ASPAI
BARCELONA • 2019

Advances in Signal Processing and Artificial Intelligence

Proceedings of the 1st International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2019)

Edited by Sergey Y. Yurish

IFSA

ASPAI
BARCELONA • 2019

Advances in Signal Processing and Artificial Intelligence:

Proceedings of the 1st International Conference on Advances in Signal Processing and Artificial Intelligence

20-21 March 2019
Barcelona, Spain

Edited by Sergey Y. Yurish

IFSA Publishing

Sergey Y. Yurish, *Editor*
Advances in Signal Processing and Artificial Intelligence
ASPAI' 2019 Conference Proceedings

E-mail (for orders and customer service enquires): ifsa.books@sensorsportal.com

Visit our Home Page on https://sensorsportal.com/ifsa_publishing.html

Neither the authors nor International Frequency Sensor Association Publishing accept any responsibility or liability for loss or damage occasioned to any person or property through using the material, instructions, methods or ideas contained herein, or acting or refraining from acting as a result of such use.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identifies as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

ASPAI Conference Website: https://aspai-conference.com/

ISSN: 2938-5350
ISBN: 978-84-09-10127-6
BN-20191803
BIC: UYQ

Contents

Foreword

On behalf of the ASPAI' 2019 Organizing Committee, I introduce with pleasure these proceedings devoted to contributions from the 1st International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2019) held in Barcelona, Spain.

Advances in artificial intelligence and signal processing are driving the growth of the artificial intelligence market as improved appropriate technologies is critical to offer enhanced drones, self-driving cars, robotics, etc. The two major factors enabling market growth are emerging artificial intelligence technologies and growth in intelligent signal processing. Today, more and more sensor manufacturers are using machine learning to sensors and signal data for analyses. The machine learning for sensors and signal data is becoming easier than ever: hardware is becoming smaller and sensors are getting cheaper, making Internet of things devices widely available for a variety of applications ranging from predictive maintenance to user behavior monitoring. Whether we are using sounds, vibrations, images, electrical signals or accelerometer or other kinds of sensor data, we can build now richer analytics by teaching a machine to detect and classify events happening in real-time, at the edge, using an inexpensive microcontroller for processing - even with noisy, high variation data.

The ASPAI' 2019 conference has been launched to provide a forum for open discussion and development of emerging artificial intelligence and appropriate signal processing technologies focused on real-word implementations by offering Hardware, Software, Services, Technology (Machine Learning, Natural Language Processing, Context-Aware Computing, Computer Vision and Signal Processing). The goal of the conference is to provide an interactive environment for establishing collaboration, exchanging ideas, and facilitating discussion between researchers, manufacturers and users.

The conference is organized by the *International Frequency Sensor Association (IFSA)* in technical cooperation with media partners – *IOS Press* (journal 'Integrated Computer-Aided Engineering') and *World Scientific* (International Journal of Neural Systems). The conference program provides an opportunity for researchers interested in signal processing and artificial intelligence to discuss their latest results and exchange ideas on the new trends.

In this year, we have got more than 60 submissions from which 41 papers (2 keynote, 33 oral and 6 posters) were selected for presentation at the Conference covering theory, design, device technology, and applications of signal processing and artificial intelligence. To accommodate this range of interests, the 1st ASPAI' 2019 Conference was organized in four dedicated regular sessions, one special session on 'Driver Monitoring using Machine Learning' and one poster session.

The proceedings contains all papers of both: oral and poster presentations. We hope that these proceedings will give readers an excellent overview of important and diversity topics discussed at the conference. Selected, extended papers will be submitted to the media partners' journals and IFSA's open access 'Sensors & Transducers' journal based on the proceeding's contributions.

We thank all authors for submitting their latest work, thus contributing to the excellent technical contents of the Conference. Especially, we would like to thank the individuals and organizations that worked together diligently to make this Conference a success, and to the members of the International Program Committee for the thorough and careful review of the papers. It is important to point out that the great majority of the efforts in organizing the technical program of the Conference came from volunteers.

Prof., Dr. Sergey Y. Yurish
ASPAI' 2019 Conference Chairman

(01)

Ifrow-multi: An Ensemble Classifier for Multiclass Imbalance Learning

E. Ramentol [1]
[1] RISE SICS Västerås, Stora Gatan 36, Västerås 722 12, Sweden
E-mail: enislay@gmail.com

Summary: In the recent years many approaches to deal with imbalanced data have emerged. Most of them have been designed for the two-class problem. When facing a multiclass imbalanced problem, the most common solution is to use a decomposition technique, such as One versus One (OVO) or One versus All (OVA) combined with a binary classifier or a resampling technique. In this paper we present a new approach for the multiclass imbalance learning. Our proposal consists on an ensemble technique that combines the IFROWANN algorithm and the decomposition techniques One versus One and One versus All. We first evaluate the classifiers behavior in the training set, then the best ones are chosen for use in the learning phase. Our experimental study is carried out using 10 datasets with different imbalanced ratios between classes. For the comparison we select 3 well-known methods from the state-of-the-art that have proved good results over multiclass datasets. Finally we include a case study applying the proposed algorithm to a real problem of fault detection.

Keywords: Multiclass imbalance learning, Decomposition strategies, Imbalanced fuzzy-rough ordered, Weighted average nearest neighbor, Ensemble techniques.

1. Introduction

The prediction of uncommon events is a challenging task for the machine learning community. This phenomenon is known as “imbalanced learning”. Faults diagnosis, medical applications, images recognitions and fraud detection are some of the most common applications of imbalanced learning [5, 6, 7, 14].

That is why in the last ten years the machine learning community has put a lot of effort into creating solutions to this problem. As a result, a significant number of approaches have been successfully proposed, more of them are designed for the two-class problem.

Solutions for two-class problem can be group in 4 categories:

Solutions at data level [4, 2, 13]: this type of solution are those that change the distribution of data by class. This resampling by creating synthetic examples of the minority class, eliminating examples of the majority class or combining the two previous techniques.

Solutions at algorithm level [3, 12, 15]: this type of solutions are those that make modifications in the classifiers favoring the less represented class.

Cost sensitive [16, 19]: this type of solution combines solutions at data level and at algorithms level, assigning different costs to the misclassification instances by classes. The cost of misclassifying a positive instance should be higher than the cost of misclassifying a negative one.

Ensembles [10, 9]: this type of solution is a combination of an ensemble learning algorithm and one of the techniques above, specifically, data level and cost-sensitive.

None of these solutions can be applied directly over multiclass imbalance data. It is necessary to decompose the original dataset into subsets of two classes. To do such a decomposition techniques must be applied as a first step.

2. Related Works

In [15] the authors introduces IFROWANN, a new classification algorithm to deal with two-class imbalance data. IFROWANN is a powerful classifier that combines the Fuzzy Rough Nearest Neighbor (FRNN) classifier and the Ordered Weighted Average vector (OWA). This algorithm showed excellent results over 102 imbalance datasets in comparison with most representative algorithms in the state-of-the-art.

The FRNN algorithm computes the classification of the test instance *x* as follows:

$$\mu P(x) = \frac{\underline{P}(x)+\overline{P}(x)}{2}, \tag{1}$$

$$\mu N(x) = \frac{\underline{N}(x)+\overline{N}(x)}{2}, \tag{2}$$

where $\underline{P}$ and $\underline{N}$ are the membership degrees to positive region of Positive and Negative class respectively, and $\overline{P}$ and $\overline{N}$ represent the membership degrees to negative region of Positive and Negative class respectively.

In order to consider the differences between the numbers of examples in classes, IFROWANN introduces the use of OWA operators to determine the positive and negative regions. Given a sequence *A* of *t* real values $A = \langle a_1, \dots, a_t \rangle$, and a weight vector $W = \langle w_1, \dots, a_t \rangle$ such that $w_i = [0,1]$ and $\sum_{i=1}^{t} w_i = 1$, the OWA aggregation of *A* by *W* is given by

$$\mathrm{OWA_W(A)} = \sum_{i=1}^{t} w_i b_i, \tag{3}$$

where $b_i = a_j$ if a_j is the i^{th} largest value in *A*.

Using the OWA aggregation the IFROWANN algorithm then determines the classification of the test instance *x* by computing:

$$\mu_P(x) = \frac{\underline{P}_{W_p^l(x)} + \overline{P}_{W_p^u(x)}}{2} = \\ = \frac{\underline{P}_{W_p^l(x)} + 1 - \underline{N}_{W_p^l(x)}}{2}, \tag{4}$$

$$\mu_N(x) = \frac{\underline{N}_{W_p^l(x)} + \overline{N}_{W_p^u(x)}}{2} = \\ = \frac{\underline{N}_{W_p^l(x)} + 1 - \underline{P}_{W_p^l(x)}}{2} \tag{5}$$

In [15] the authors proposed 6 different weighting strategies. In their experimental study they showed that some of these *W* perform better for different imbalance ratios. In this paper we will follow the author recommendation to select the best *W*.

2.1. Abording Multiclass Imbalance via Decomposition

To address the multiclass imbalance problem the research community has adopted the use of decomposition strategies. These strategies are: OVO and OVA. These strategies allow to apply two-class solutions over multiclass datasets.

One versus One (OVO): This strategy trains a binary classifier for each pair of classes. In this way the total number of binary classifiers is c (c-1) / 2 where c is the number of classes. These binary classifiers will be used in the prediction phase.

One versus All (OVA): For each class, OVA labels this class as a 'minority class' and all the other classes as 'majority classes', then it trains a binary classification model.

In [17] the author proposed an extension of IFROWANN algorithm for multiclass datasets. The authors combine OVO and IFROWANN, and also proposed an adaptive weight setting and a new dynamic aggregation method. The resulting method called FROVOCO shows very good results outperforming 3 well-known methods of the state-of-the art.

Another interesting proposal is the combination of decomposition techniques and resampling methods. In [8] the author presented a study about the use of decomposition strategies and resampling techniques showing that OVO+SMOTE+C4.5 is the best configuration. From our research we conclude that decomposition and ensembles results in the best combination approaches for multiclass imbalanced problem.

3. Ifrow-multi Algorithm

In this paper a new method based on OVA, OVO combined with IFROWANN is proposed. The Ifrow-multi is a multiclassifier that uses OVA+IFROWANN as classifier 1, OVO+IFROWANN as classifier 2 and kNN version based on similarity as classifier 3.

Below is explained how each classifier works separately and how the proposed Ensemble works.

```
Classifier 1. OVA_IFROWANN(Av-w4).
$C = \{C_1, C_2, \ldots, C_n\}$ set of classifiers, where $n$ is number of classes
training set Tra and Tst instances to be classified
$W$ weighting strategy, T-norm = Average
Double[] Pert_min: store the POS value for every class
For each instance $T_i$ to be classified
 For each class $C_i$ get_IR_OVA($C_i$)
 Create_two_tra(tra$_i$)
 if $IR \geq 3$
 w = 4
 POS = get_pos_region (tst$_i$,tra$_i$,w)
 Else
 w = 0
 POS = get_pos_region (tst$_i$,tra$_i$,w)
 Pert_min[$C_i$] = POS
C$_{final}$ = max (Pert_min),
```

where *get_IR_OVA* is a method that returns the imbalance ratio between a class and the remaining examples in the training set. *Create_two_tra_OVA* constructs a two-class dataset using one class as the minority class and the remaining examples as majorities. The method *get_pos_region* computes for a given instance the fuzzy rough membership to the positive region of the minority class and max returns the class with highest score.

The classifier 2 performs in a similar way to the classifier 1, but instead of creating the binary training sets using OVA it uses OVO.

```
Classifier 2. OVO_IFROWANN(Av-w4).
 $C = \frac{n(n-1)}{2}$, number of binary classifiers, where $n$ is number of classes
training set Tra and Tst instances to be classified
$W$ weighting strategy, T-norm = Average
Double[] Pert_min: store the POS value for every class
For each instance $T_i$ to be classified
 For each binary classifier $C_i$ get_IR_OVO($C_i$)
 Create_two_tra_OVO(C1, C2)
 if $IR \geq 3$
 w = 4
 POS = get_pos_region (tst$_i$,tra$_i$,w)
 else
 w = 0
 POS = get_pos_region (tst$_i$,tra$_i$,w)
 Pert_min[$C_i$] = POS
C$_{final}$ = max_voting (Pert_min),
```

where *get_IR_OVO* is a method that returns the imbalance ratio between two given classes. *Create_two_tra_OVO* constructs a two-class dataset given two class index. Each binary classifier grants a vote to a class, finally the max_voting function returns the class that received the most votes.

Classifier 3 is 1-NN based on a similarity function. The similarity function used for numeric features is the following:

$$\delta_k\left(x_{ik}, x_{jk}\right) = \frac{|x_{ik}-x_{jk}|}{maxA_k - minA_k}, \tag{6}$$

where maxA$_k$ and minA$_k$ are the extreme values of the domain intervals for feature k. For discrete attributes in the following way:

$$\delta_k\left(x_{ik}, x_{jk}\right) = \left\{\begin{array}{ll} 1 & if \quad k \in B \\ 0 & other\_c\,ase \end{array}\right\} \tag{7}$$

The proposed Ifrow-multi algorithm as a first step, classifiers 1 and 2 are executed on the training set. In this step, hard_instances_OVA and hard_instances_OVO are obtained, which are the number of miss classified instances in training using classifiers 1 and 2.

The final class is decided by classifier 1 if hard_instances_OVA is zero or classifier 2 if hard_instances_OVO is zero. If both are zero, then the classifier 1 decides. If none of the 2 is zero, a max voting system is applied using classifier 3.

```
Ensemble. Ifrow-multi.

Run Classifier 1 over the training set.
Compute hard_instances_OVA
Run Classifier 2 over the training set.
Compute hard_instances_OVO
For each instance $T_i$ to be classified
    possible_class1 = Classifier1
    possible_class2 = Classifier2
    possible_class3 = Classifier3
if (hard_instances_OVA = = 0)&&( hard_instances_OVO = = 0)
    then $C_{final}$ = C1
else if
if(hard_instances_OVA = = 0)&&( !hard_instances_OVO = = 0)
    then $C_{final}$ = C1
if(!hard_instances_OVA = = 0)&&(hard_instances_OVO = = 0)
    then $C_{final}$ = C2
else
    $C_{final}$ = max_voting(possible_class1,possible_class2, possible_class3)
```

4. Experimental Study

The experimental study is performed with ten datasets from the UCI repository. For the comparison we selected 3 methods from the state-of-art that have shown good results, one of them is based in IFROWANN. The remaining two methods are SMOTE-C4.5-WV (SMT) and MDO-C4.5 (MDOC). The experimental study is performed with ten datasets from the UCI repository. For the comparison we selected 3 methods from the state-of-art that have shown good results, one of them is based in IFROWANN.

The used metric is average accuracy.

Table 1. Full average accuracy results for the tree state-of-the-art classifiers and our IFROW-multi proposal.

Data	SMT	MDOC	FROVOCO	Ifrow-multi
Aut	80.644	76.478	77.156	78.544
bal	55.270	56.482	78.851	75.716
con	51.725	48.752	47.445	43.151
der	96.210	95.371	97.155	97.181
Eco	71.461	71.148	77.272	73.685
Glas	75.189	63.044	67.069	77.230
Led	63.547	64.173	64.792	71.304
new	91.389	90.444	91.111	96.444
sat	85.293	84.714	89.496	90.225
shu	96.844	91.315	91.853	93.329
% Mean	76.757	74.192	78.220	79.681

As Table 1 shows the proposed method obtained the best mean, and it wins over SMT 7 times, over MDOC 9 times and over PROVOCO 7 times.

4.1. Solving a Real Problem of Fault Detection

Multi-Ifro is an algorithm that has been designed to solve imbalanced data classification problems with multiple classes. Our goal has been to design an algorithm that achieves high precision in the smallest classes (those of greatest interest) even if it achieves worse results in the most represented classes.

Our previous experiments showed improvements in AUC. But now we would like to show, through a real data set, how to improve the classification in the less represented classes

The selected datasets is Steel plate faults [16] from the UC Irvine Machine Learning Repository. This dataset has the following distribution by classes:

Table 2. Examples per classes in Steel plate faults (in bold the most interesting classes).

Class name	#examples
Pastry	**158**
Z_Scratch	**190**
K_Scatch	391
Stains	**72**
Dirtiness	**55**
Bumps	402
Other_Faults	673

For the experiment we use a 5 fold cross validation. The following table shows the number of examples correctly classified by class. The results are for a randomly selected testing partition.

The first column is the name of the class. The second column is the number of examples in each class. The third, fourth and fifth columns are the number of examples correctly classified by Ifrow-multi, the decision tree C4.5 and OVO+SMOTE+C4.5

(SMT in the table) respectively. The last row is average of accuracy.

Table 3. Examples correctly classified by class.

Class name	#ex	Ifrow-multi	C4.5	SMT
Pastry	**32**	**20**	15	16
Z_Scratch	**38**	**36**	36	34
K_Scatch	78	77	75	75
Stains	**14**	**13**	12	13
Dirtiness	**11**	**11**	11	9
Bumps	81	56	43	48
Other_Faults	134	75	95	102
% Av_acc	-	81.988	64.071	77.955

As may be seen in Table 3, the results of C4.5 are the worst, this is due to the fact that no technique for imbalance has been applied. The results improve significantly (in 13.884 %) when a binarization technique (OVO) is applied, then over each binarized sub-set an oversampling technique (SMOTE) and finally C4.5 as a classifier is applied.

Table 3 clearly shows that our proposal achieves better precision in all classes except for the most represented.

In real problems, where some classes are significantly less represented than others, the main interest is usually in achieving good precision in those classes. Our goal is to correctly identify the less frequent faults that are the ones causing the worst damage and being the hardest to predict.

5. Conclusions and Future Work

In this paper we proposed an ensemble method based on decomposition techniques and IFROWANN algorithm. In our experimental study our method showed good results.

The future direction of this research work is to develop a more complete experimental study using more datasets and comparing with more algorithms from the state-of-art. Another direction of our work is to use the proposed method in industrial applications, such as faults and anomalies detection.

Acknowledgements

This research has been funded by the European Research Consortium for Informatics and Mathematics (ERCIM) Alain Bensoussan Fellowship Programme.

References

[1]. L. Abdi, S. Hashemi, To combat multi-class imbalanced problems by means of over-sampling techniques, *IEEE Trans. Knowl. Data Eng*, Vol. 1, 2016, pp. 238-251.

[2]. G. E. A. P. A. Batista, R. C. Prati, M.C. Monard, A study of the behaviour of several methods for balancing machine learning training data, *SIGKDD Explorations*, Vol. 6, Issue 1, 2004, pp. 20-29.

[3]. E. Bernadó-Mansilla, T. K. Ho, Domain of competence of XCS classifier system in complexity measurement space, *IEEE Transactions on Evolutionary Computation*, Vol. 9, Issue 1, 2005, pp. 82-104.

[4]. N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, *Journal of Artificial Intelligent Research*, 2002*,* Vol. 16, pp. 321-357.

[5]. Y. Chen. An empirical study of a hybrid imbalanced-class DT-RST classification procedure to elucidate therapeutic effects in uremia patients, *MedBiol. Eng. Comput.*, Vol. 54, Issue 6, 2016, pp. 983-1001.

[6]. G. Cohen, M. Hilario, H. Sax, S. Hogonnet, A. Geissbuhler, Learning from imbalanced data in surveillance of nosocomial infection, *Artificial Intelligence in Medicine*, 2006, pp. 7-18.

[7]. T. E Fawcett, F. Provost, Adaptive fraud detection, *Data Mining and Knowledge Discovery*, Vol. 3, 1997, pp. 291-316.

[8]. A Fernández, V. López, M. Galar, M. J. del Jesús, F. Herrera, Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches, *Knowledge-Based Systems*, Vol. 42, 2013, pp. 97-110.

[9]. M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: Bagging-, boosting-,and hybrid-based approaches, *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, Vol. 42, Issue 4, 2012, pp. 463-484.

[10]. M. Galar, A. Fernández, E. Barrenechea, F. Herrera, EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, *Pattern Recognition*, Vol. 46, 2013, pp. 3460-3471.

[11]. E. Hüllermeier, S. Vanderlooy, Combining predictions in pairwise classification: an optimal adaptive voting strategy and its relation to weighted voting, *Pattern Recog.*, Vol. 43, Issue 1, 2010, pp. 128-142.

[12]. T. R. Hoens, Q. Qian, N.V. Chawla, et al., Building decision trees for the multi-class imbalance problem, in Advances in Knowledge Discovery and Data Mining, *Springer*, Berlin Heidelberg, 2012, pp. 122-134.

[13]. E. Ramentol, Y. Caballero, R. Bello, F. Herrera, SMOTE-RSB*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory, *International Journal of Knowledge and Information Systems*, Vol. 33, 2012, pp. 245-265.

[14]. E. Ramentol, I. Gondres, S. Lajes, R. Bello, Y. Caballero, C. Cornelis, F. Herrera. Fuzzy-rough imbalanced learning for the diagnosis of high voltage circuit breaker maintenance: The smote-frst-2t algorithm, *Engineering Applications of Artificial Intelligence*, Vol. 48, 2016, pp. 134-139.

[15]. E. Ramentol, S. Vluymans, N. Verbiest, Y. Caballero, R. Bello, C. Cornelis, F. Herrera, Ifrowann: Imbalanced fuzzy-rough ordered weighted average nearest neighbor classification, IEEE Trans. Fuzzy Systems, Vol. 23, Issue 5, 2015, pp. 1622-1637.

[16]. Steel Plates Faults Data Set, https://archive.ics.uci.edu/ml/datasets/Steel+Plates+Faults

[17]. K. M. Ting, An instance-weighting method to induce cost-sensitive trees, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, Issue 3, 2002, pp. 659-665.

[18]. S. Vluymans, A. Fernández, Y. Saeys, C. Cornelis, F. Herrera, Dynamic a nity-based classification of multi-class imbalanced data with one-versus-one decomposition: a fuzzy rough set approach, *Knowl. Inf. Syst*., Vol. 56, 2017, pp. 55-84.

[19]. R. R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 18, 1988, pp. 183-190.

[20]. B. Zadrozny, J. Langford, N. Abe, Cost-sensitive learning by costproportionate example weighting, in *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*, 2003, pp. 435-442.

(05)

Radar Target Classification Method Against Interrupted-Sampling Repeater Jamming Based on Time-frequency Analysis

J. Chen, W. Wu, J. Zou, S. Xu and Z. Chen
National University of Defense Technology, National Key Laboratory of Science and Technology on ATR,
Changsha, China
Tel.: + 86073187003487
E-mail: chenjian591@yeah.net

Summary: The interrupted-sampling repeater jamming (ISRJ) is a coherent jamming which can flexibly form false targets in the victim radar, making it a big threat to radar. Since the time delay of the ISRJ is much shorter than a pulse repeat interval (PRI), the published electronic counter-countermeasure (ECCM) techniques did not handle this threat well. Many methods against JSRJ require a priori knowledge of the ISRJ parameters, which is not practical in real scenarios. In this paper, we present a target classification method against ISRJ based on signal component detection and feature extraction in time-frequency (TF) domain, especially for radars with linear frequency modulation (LFM) waveform. Referring to the detected target peaks in pulse compression (PC) results, the proposed method estimates the time durations of corresponding signal components in TF representation, and evaluates the signal duty ratios for further target classification. The proposed method provides a new perspective against the ISRJ problem. Experiment results demonstrate that the presented method can effectively eliminates the jamming targets and retains the real targets simultaneously.

Keywords: Radar target classification, Interrupted-sampling repeater jamming (ISRJ), Electronic countermeasure (ECM), Electronic counter-countermeasure (ECCM), Linear frequency modulation (LFM), Digital radio frequency memory (DRFM).

1. Introduction

Recent years, a new electronic countermeasure (ECM) method, named interrupted-sampling repeater jamming (ISRJ) [1], becomes a big threat to radar. ISRJ is implemented through the digital radio frequency memory (DRFM) [2] device. It samples and stores a short segment of the pulse, retransmits it several times, and repeats the process until the pulse ends. Since the jamming signal is coherent with the radar signal, a high processing gain is obtained from pulse compression (PC) or coherent integration [3, 4]. Therefore, it can flexibly forming false targets in the victim radar with relatively low power, which severely limits the ability of radar to detect, track, and recognize targets.

There are many electronic counter-countermeasure (ECCM) researches against ISRJ, including ISRJ based active echo cancellation countering method [5] and ISRJ suppression methods, such as intra-pulse frequency coded signal [6], transmitting schemes [4], and filtering methods [7-9]. In this paper, we present a target classification method based on time-frequency (TF) analysis against ISRJ, for radars with linear frequency modulation (LFM) waveform. The method provides a new perspective against the ISRJ problem. Experiment results demonstrate that the presented method can effectively eliminates the jamming targets, and simultaneously retains the real targets.

2. Method

For intuitive perception, the time, frequency, and TF reprentation of an example de-chirped, ISRJ containinated signal are plotted in Fig. 1. It is worth noting that the de-chirping operation transforms the LFM signal into a constant frequency signal, and the Fourier transform of the de-chirped signal yields the PC results. The ISRJ jammer samples 3 pieces of the radar pulse, and transmitted them twice. Therefore, there are 3 target peak groups in the PC result, which correspond to the ridges in 3 horizontal lines in the TF representation.

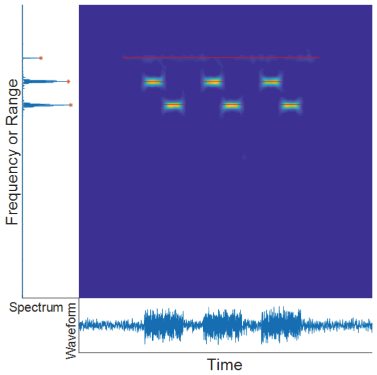


Fig. 1. Illustration of an example de-chirped, ISRJ containinated signal. Image in the main axes with blue background is the TF representation; the time waveform is under the main axes; and the spectrum, also the PC results is at the left side.

In the TF representation, the ridge of the real target signal components, which is in the top area for the example figure, is a continuous straight line. Whereas, the ridge of the jamming signals are discontinuous. We utilize the duty ratio to indicate the ridge continuity. Theoretically, the duty ratio of the jamming is no more than 50 %. The ratio is 30 % in this case. Focusing on the duty ratio feature, the steps of the proposed method are described below:

1) Detect target peaks in the PC result;
2) Calculate the TF representation of the de-chirped radar received signal;
3) Extract ridges in the TF representation referring to the targets detected in 1);
4) Estimate the total time duration of ridges at each target frequencies;
5) Calculate the duty ratio of each detected target signal component, which is defined as the ratio of the time duration to the radar pulse time width;
6) Do the judgement. If the duty ratio of a detected target is larger than 75 %, the target is true and retained; otherwise, the target is false and eliminated.

3. Experiments and Results

To verify the effectiveness of the proposed method, a series of Monte Carlo experiments are designed. The parameters for radar and ISRJ jammers are listed in Table 1. There are 4 values of ISRJ retransmition times, corresponding to 4 working modes of the ISRJ. The corresponding duty ratios of the 4 ISRJ signals are 0.5, 0.3, 0.3, and 0.1, respectively. The signal to noise ratio (SNR) ranges from -20 to 20 dB, with 5 dB interval. The jamming to signal ratio (JSR) ranges from 5 to 50 dB, with 5 dB interval. For every parameter combination of ISRJ working mode, SNR, and JSR, a 1000 times Monte Carlo experiment was carried out. The duty ratios of each detected target, was estimated, and the target was classified. The classification results of the detected targets are recorded.

Table 1. Simulation parameters for signal and ISRJ.

Parameter	Value
Pulse width	20 us
Bandwidth	40 MHz
Complex sampling frequency	50 MHz
ISRJ interception width	2 us
ISRJ retransmition times	{1,2,3,9}s
Signal to noise ratios	{5x|x = -4,-3,…,4}
Jamming to signal ratios	{5x|x = 1,2,…,10}
Range of real target	273 m

The classification result of the 4 mode of the ISRJ jammer are similar. The mean real target recall rate and false target elimination rate of the 4 mode are plotted in Fig. 2. For false targets formed by the ISRJ, the elimination rate is higher than 93 % when the jamming to noise ratio (JNR) is larger than 0 dB. The JNR equals SNR plus JSR, in dB. The recall rate of the real target is basically higher than 90 % when the SNR is larger than -10 dB and the JSR is smaller than 25 dB.

Two reasons for the low recall rate under high JSR might be: first, the detection rate of the real target is low; second, the high power jamming signals affect the TF ridge detection of the real target signal.

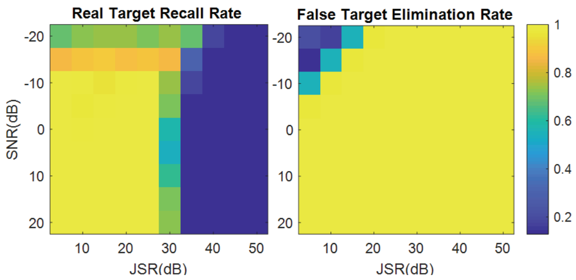


Fig. 2. Visualization of real target recall rate (left) and false target elimination rate (right) of the 4 ISRJ mode under different SNR and JSR conditions.

4. Conclusions

In this paper, we present a target classification method against ISRJ based on the parameter estimation of signal component in the TF representation. It provides a new perspective against the ISRJ problem. The experiment results demonstrate the effectiveness and implies the potential to improve the performance, if a more powerful signal component detection should be found.

References

[1]. X. S. Wang, J. C. Liu, W. M. Zhang, Q. X. Fu, Z. Liu, X. X. Xie, Mathematic principles of interrupted-sampling repeater jamming (ISRJ), *Science in China,* Vol. 50, Issue 1, 2007, pp. 113-123.

[2]. S. J. Roome, Digital Radio Frequency Memory, *Electronics & Communications Engineering Journal,* Vol. 2, Issue 4, 1990, pp. 147-153.

[3]. K. Olivier, J. E. Cilliers, M. D. Plessis, Design and performance of wideband DRFM for radar test and evaluation, *Electronics Letters,* Vol. 47, Issue 14, 2011, pp. 824-825.

[4]. C. Zhou, F. Liu, Q. Liu, An adaptive transmitting scheme for interrupted sampling repeater jamming suppression, *Sensors,* Vol. 17, Issue 11, 2017, 2480.

[5]. S. B. S. Hanbali, R. Kastantin, Technique to counter active echo cancellation of self-protection ISRJ, *Electronics Letters,* Vol. 53, Issue 10, 2017, pp. 680-681.

[6]. R. Shen, Z. Liu, J. Sui, X. Wei, Study on interrupted-sampling repeater jamming performance based on intra-pulse frequency coded signal, *Society of Photo-Optical Instrumentation Engineers*, 2017, Vol. 10420, pp. 1-5.

[7]. S. Gong, X. Wei, X. Li, ECCM scheme against interrupted sampling repeater jammer based on time-frequency analysis, *Journal of Systems Engineering and Electronics,* Vol. 25, Issue 6, 2014, pp. 996-1003.

[8]. W. Xiong, G. Zhang, W. Liu, Efficient filter design against interrupted sampling repeater jamming for wideband radar, *Eurasip Journal on Advances in Signal Processing,* 2017, 9.

[9]. H. Yuan, C. Y. Wang, X. Li, L. An, A method against interrupted-sampling repeater jamming based on energy function detection and band-pass filtering, *International Journal of Antennas and Propagation*, Vol. 2017, Issue 1, Mar. 2017, pp. 1-9.

(07)

Structural Health Monitoring using a Large Sensor Network and Bayesian Virtual Sensors

J. Kullaa
Metropolia University of Applied Sciences, Department of Automotive and Mechanical Engineering,
P.O. Box 4071, 00079 Metropolia, Finland
Tel.: +358 40 3340429
E-mail: jyrki.kullaa@metropolia.fi

Summary: Damage detection in structural health monitoring is based on unsupervised learning. Training data consist of multichannel vibration measurements of the undamaged structure under different environmental or operational conditions. The excitation (input) and the environmental or operational variables are usually unknown, but noisy response is only measured. In subsequent data analyses, the noisy measurements are replaced by Bayesian virtual sensors that are more accurate than the corresponding hardware. Environmental or operational variability between measurements can be taken into account using the minimum mean square error (MMSE) estimation. Damage detection is done in the time domain using statistical analysis including residual generation, principal component analysis, and an extreme value statistics control chart. A numerical experiment is performed for a bridge structure having a crack with different sizes. Virtual sensors outperformed the raw measurements in damage detection and damage was correctly localized to the nearest sensor.

Keywords: Virtual sensing, Structural health monitoring, Novelty detection, Bayes' rule, Vibration measurements, Sensor array, Estimation.

1. Introduction

Large structures like bridges or other important infrastructure can be monitored for an early detection of faults. Structural health monitoring (SHM) can be based on vibration measurements, which is attractive, because it is global, i.e. damage can be detected remotely from the sensor. However, in order to have an early warning, the monitoring system must be sensitive to small changes and insensitive to varying environmental or operational conditions. A small measurement error and statistical data analysis of multivariate vibration records are therefore necessary.

If a sufficiently large sensor array is installed on the structure, the system is redundant. This hardware redundancy can be used to compensate for the environmental or operational influences and reduce measurement errors using virtual sensing techniques.

The paper is organized as follows. Bayesian virtual sensing is introduced in Section 2 resulting in virtual sensors that are more accurate than the hardware. Section 3 is devoted to a damage detection algorithm capable of eliminating environmental or operational influences. Numerical simulations of a bridge under random excitation, environmental variability, and damage are performed in Section 4. Damage detection is done using the actual measurements or the virtual sensors. Finally, concluding remarks are presented in Section 5.

2. Bayesian Virtual Sensing

Virtual sensing (VS) gives an estimate of a quantity of interest using the available measurements. Empirical VS is a data-based approach, estimating a physical sensor using hardware redundancy. In structural vibrations of linear structures, the response can be written as a contribution of natural modes:

$$\mathbf{x}(t) = \mathbf{\Phi}\mathbf{q}(t) = \sum_{i=1}^{N} \varphi_i q_i(t) \approx \sum_{i=1}^{n} \varphi_i q_i(t) , \qquad (1)$$

where $\mathbf{x}(t)$ is the displacement response, N is the number of DOF in the finite element model, $n << N$ is the selected number of modes, $\mathbf{\Phi}$ is the modal matrix consisting of the mode shapes ϕ_i as columns, and $\mathbf{q}(t)$ are the modal or generalized coordinates. The last expression in (1) suggests that the response can be expressed with sufficient accuracy by including only the lowest n modes. In the case of SHM, the excitation is due to wind, waves, traffic, or micro-tremors having a narrow bandwidth. Therefore, the n lowest modes are usually able to describe the vibration response of the structure. Consequently, more than n sensors are needed to make the sensor network redundant.

Virtual sensing techniques can be used to obtain an estimate for each sensor, which is more accurate than the corresponding hardware.

Consider a sensor network measuring p simultaneously sampled variables $\mathbf{y} = \mathbf{y}(t)$ at time t. Each measurement $\mathbf{y}$ includes independent measurement error $\mathbf{w} = \mathbf{w}(t)$:

$$\mathbf{y} = \mathbf{x} + \mathbf{w} , \qquad (2)$$

where $\mathbf{x} = \mathbf{x}(t)$ are the true values of the measured degrees of freedom. The objective is to find an estimate for the true values **x** utilizing the noisy measurements **y**.

In a sensor array, a true reading for sensor *u*, x_u, can be estimated using the noisy measurements **y** by conditioning on the remaining sensors *v* and applying Bayes' rule [1]. The partitioned variables are

$$\mathbf{y} = \begin{Bmatrix} \mathbf{y}_u \\ \mathbf{y}_v \end{Bmatrix},\ \mathbf{x} = \begin{Bmatrix} \mathbf{x}_u \\ \mathbf{x}_v \end{Bmatrix},\ \mathbf{w} = \begin{Bmatrix} \mathbf{w}_u \\ \mathbf{w}_v \end{Bmatrix} \tag{3}$$

The measurement error **w** is assumed to be zero mean, independent of **x**, with a known covariance matrix

$$\mathbf{\Sigma}_w = E(\mathbf{w}\mathbf{w}^T) = \begin{bmatrix} \mathbf{\Sigma}_{w,uu} & \mathbf{\Sigma}_{w,uv} \\ \mathbf{\Sigma}_{w,vu} & \mathbf{\Sigma}_{w,vv} \end{bmatrix}, \tag{4}$$

where $E(\cdot)$ denotes the expectation operator. After partitioning, all distributions are conditioned with $\mathbf{y}_v$. Bayes' rule becomes

$$\begin{aligned} p(\mathbf{x}_u \mid \mathbf{y}) &= p(\mathbf{x}_u \mid \mathbf{y}_u, \mathbf{y}_v) \\ &= \frac{p(\mathbf{y}_u \mid \mathbf{x}_u, \mathbf{y}_v) p(\mathbf{x}_u \mid \mathbf{y}_v)}{p(\mathbf{y}_u \mid \mathbf{y}_v)} \\ &= \frac{p(\mathbf{y}_u \mid \mathbf{x}_u) p(\mathbf{x}_u \mid \mathbf{y}_v)}{p(\mathbf{y}_u \mid \mathbf{y}_v)} \end{aligned} \tag{5}$$

The distributions in (5) can be evaluated as shown in the following.

2.1. Likelihood

Assuming Gaussian noise **w**, the likelihood is obtained using Equation (2):

$$\begin{aligned} p(\mathbf{y}_u \mid \mathbf{x}_u) &= N(\mathbf{y}_u \mid \mathbf{x}_u, \mathbf{\Sigma}_{w,uu}) \\ &\propto \exp\left[-\frac{1}{2} (\mathbf{y}_u - \mathbf{x}_u)^T \mathbf{\Sigma}^{-1}_{w,uu} (\mathbf{y}_u - \mathbf{x}_u) \right] \end{aligned} \tag{6}$$

2.2. Evidence

For simplicity but without loss of generality, assume zero-mean variables **y**. The partitioned data covariance matrix $\mathbf{\Sigma}_y$ is

$$\mathbf{\Sigma}_y = E(\mathbf{y}\mathbf{y}^T) = \begin{bmatrix} \mathbf{\Sigma}_{y,uu} & \mathbf{\Sigma}_{y,uv} \\ \mathbf{\Sigma}_{y,vu} & \mathbf{\Sigma}_{y,vv} \end{bmatrix} \tag{7}$$

The precision matrix $\mathbf{\Gamma}_y$ is defined as the inverse of the covariance matrix $\mathbf{\Sigma}_y$ and is also written in a partitioned form:

$$\mathbf{\Gamma}_y = \mathbf{\Sigma}_y^{-1} = \begin{bmatrix} \mathbf{\Gamma}_{y,uu} & \mathbf{\Gamma}_{y,uv} \\ \mathbf{\Gamma}_{y,vu} & \mathbf{\Gamma}_{y,vv} \end{bmatrix} \tag{8}$$

A linear minimum mean square error (LMMSE) estimate for $\mathbf{y}_u \mid \mathbf{y}_v$ is obtained by minimizing the mean-square error (MSE) and can be computed either using the covariance or precision matrix [2, 3]. If each sensor signal is estimated in turn, the formulas based on the precision matrix result in a more efficient algorithm [2]. The expected value of the estimated variable is:

$$\hat{\mathbf{y}}_u = E(\mathbf{y}_u \mid \mathbf{y}_v) = -\mathbf{\Gamma}^{-1}_{y,uu} \mathbf{\Gamma}_{y,uv} \mathbf{y}_v = \mathbf{K}\mathbf{y}_v\ , \tag{9}$$

where $\mathbf{K} = -\mathbf{\Gamma}^{-1}_{y,uu}\mathbf{\Gamma}_{y,uv}$, and the MSE is

$$\operatorname{cov}(\mathbf{y}_u \mid \mathbf{y}_v) = \mathbf{\Gamma}^{-1}_{y,uu} \tag{10}$$

The evidence $p(\mathbf{y}_u \mid \mathbf{y}_v)$ is also Gaussian with the mean and covariance from Equations (9) and (10), respectively:

$$\begin{aligned} p(\mathbf{y}_u \mid \mathbf{y}_v) &= N(\mathbf{y}_u \mid \mathbf{K}\mathbf{y}_v, \mathbf{\Gamma}^{-1}_{y,uu}) \\ &\propto \exp\left[-\frac{1}{2} (\mathbf{y}_u - \mathbf{K}\mathbf{y}_v)^T \mathbf{\Gamma}_{y,uu} (\mathbf{y}_u - \mathbf{K}\mathbf{y}_v) \right] \end{aligned} \tag{11}$$

The evidence is merely a normalizing factor, independent of $\mathbf{x}_u$. However, this conditional probability is important in damage detection to remove the environmental or operational effects, which will be discussed in Section 3.

2.3. Prior

The prior distribution $p(\mathbf{x}_u \mid \mathbf{y}_v)$ is obtained from the measurement model (2) by partitioning and conditioning. The prior mean is

$$\begin{aligned} E(\mathbf{x}_u \mid \mathbf{y}_v) &= E(\mathbf{y}_u \mid \mathbf{y}_v) - E(\mathbf{w}_u) \\ &= E(\mathbf{y}_u \mid \mathbf{y}_v) = \mathbf{K}\mathbf{y}_v \end{aligned}\ , \tag{12}$$

and the prior covariance is

$$\begin{aligned} \mathbf{\Sigma}_{\text{prior},uu} &= \operatorname{cov}(\mathbf{x}_u \mid \mathbf{y}_v) \\ &= \operatorname{cov}(\mathbf{y}_u \mid \mathbf{y}_v) - \mathbf{\Sigma}_{w,uu} = \mathbf{\Gamma}^{-1}_{y,uu} - \mathbf{\Sigma}_{w,uu} \end{aligned} \tag{13}$$

The prior distribution is also Gaussian:

$$\begin{aligned} p(\mathbf{x}_u \mid \mathbf{y}_v) &= N(\mathbf{x}_u \mid \mathbf{K}\mathbf{y}_v, \mathbf{\Sigma}_{\text{prior},uu}) \\ &\propto \exp\left[-\frac{1}{2} (\mathbf{x}_u - \mathbf{K}\mathbf{y}_v)^T \mathbf{\Sigma}^{-1}_{\text{prior},uu} (\mathbf{x}_u - \mathbf{K}\mathbf{y}_v) \right] \end{aligned} \tag{14}$$

2.4. Posterior

The posterior distribution (5) is obtained by some manipulation, resulting in

$$p(\mathbf{x}_u \,|\, \mathbf{y}) = c \exp\left[-\frac{1}{2}(\mathbf{x}_u - \hat{\mathbf{x}}_u)^T \mathbf{\Sigma}^{-1}_{\text{post},\, uu} (\mathbf{x}_u - \hat{\mathbf{x}}_u) \right], \tag{15}$$

where c is a constant and the posterior covariance $\mathbf{\Sigma}_{\text{post},\, uu}$ is

$$\mathbf{\Sigma}_{\text{post},\, uu} = \text{cov}(\mathbf{x}_u \,|\, \mathbf{y}) = (\mathbf{\Sigma}^{-1}_{w,uu} + \mathbf{\Sigma}^{-1}_{\text{prior},\, uu})^{-1}, \tag{16}$$

and the posterior mean is

$$\begin{aligned} \hat{\mathbf{x}}_u &= E(\mathbf{x}_u \,|\, \mathbf{y}) \\ &= \mathbf{\Sigma}_{\text{post},\, uu} (\mathbf{\Sigma}^{-1}_{w,uu} \mathbf{y}_u + \mathbf{\Sigma}^{-1}_{\text{prior},uu} \mathbf{K} \mathbf{y}_v) \end{aligned} \tag{17}$$

The expected value of the posterior probability density function (PDF) is chosen for the virtual sensor reading. The statistical data analysis proceeds as usual, except that now the actual measurements are replaced with the virtual sensors.

3. Damage Detection

A single vibration measurement period is relatively short, and can be assumed to take place at constant environmental conditions. Environmental or operational variations occur between measurements. Their effects can be taken into account by building a covariance matrix using several measurements from the undamaged structure under different environmental or operational conditions.

Using the conditional probability $p(x_u|\mathbf{x}_v)$, each virtual sensor under variable environmental conditions can be estimated using the other virtual sensors in the network. For actual sensors (raw data), the corresponding conditional probability is $p(y_u|\mathbf{y}_v)$ [4]. The mean of the conditional probability (9) is often more accurate than the mean of the variable. This is illustrated in Fig. 1 showing two variables, both of which are influenced by a latent variable so that their standard deviations are large (black PDFs). The conditional PDFs $p(y_1|y_2)$ and $p(y_2|y_1)$ are shown in red after a data point (red circle) is observed. Their standard deviations are smaller. Consequently, the environmental or operational influences can be eliminated by replacing the variable means with the conditional means.

A residual vector is then generated:

$$\boldsymbol{\varepsilon}_u = \mathbf{y}_u - E(\mathbf{y}_u \,|\, \mathbf{y}_v), \tag{18}$$

where $\mathbf{y}$ can be either the actual measurement or the virtual sensor (17).

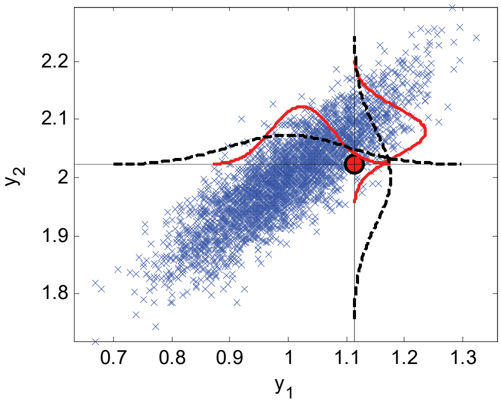


Fig. 1. An example of conditional probability distributions (red). The red circle is an observed data point.

For a reliable statistical analysis, the dimensionality is reduced using principal component analysis (PCA). The damage-sensitive features are the first principal component scores of the residuals, but they may not be Gaussian. Therefore, extreme value statistics (EVS) distribution is estimated [5] to design a control chart [6] for damage detection with appropriate control limits.

4. Numerical Experiment

A finite element model of a bridge deck was built to simulate noisy vibration measurements from 28 accelerometers under variable environmental and operational conditions (Fig. 2).

The length of the bridge was 30 m and the width was 11 m. The deck was made of concrete with a Young’s modulus of E = 40 GPa (at temperature T = 0 °C), Poisson ratio of ν = 0.15, density of ρ = 2500 kg/m^3, and thickness of 250 mm. The stiffeners were made of steel (E = 207 GPa, ν = 0.30, ρ = 7850 kg/m^3). The longitudinal stiffeners had a web with a thickness of t = 16 mm and a height of h = 1.4 m. The bottom flange had a thickness of t = 50 mm and a width of b = 700 mm. The lateral stiffeners were 1.4 m high and 30 mm thick plates.

The nodes of the bottom flanges were simply supported at both ends of the bridge. Longitudinal displacements were fixed only at one end of the bridge. The corners of the concrete deck were supported in the lateral and vertical directions.

The deck was affected by temperature variability. The ends of the bridge were at random temperatures between –20 °C and +40 °C, and the temperature distribution was linear in the longitudinal direction and constant in the lateral direction. The temperature in each cross-section had some variation having a standard deviation of 0.2 °C. The Young’s modulus of concrete had a stepwise linear relationship with temperature. Different realizations of the Young’s modulus is shown in Fig. 3. The temperature or the Young's moduli were not measured.

Two independent random loads in the vertical direction were applied at the nodes shown as green dots in Fig. 2. A steady state response was assumed.

The response was computed with a modal superposition algorithm using the first seven modes. The analysis period was 20.48 s with a sampling frequency of 50 Hz. One measurement period then included 1024 samples from each sensor.

Same amount of Gaussian noise was added to all sensors. The average signal-to-noise ratio of the sensor array was 40 dB. For the corresponding virtual sensors, the average signal-to-noise ratio increased to almost 48 dB.

Damage was a crack with an increasing size in a longitudinal steel girder. Six different crack configurations were modelled with an increasing severity by removing the contact between elements at 1-6 nodes. The nodes are shown in the detailed plot in Fig. 4 showing the order in which the nodes were separated. Each damage scenario was monitored with three measurements under random environmental and operational conditions. As a result, the last 18 measurements were from a damaged structure.

50 measurements were acquired from the undamaged structure under random environmental conditions. The training data were the first 25 measurements. They were also used to design the control charts. Test data were the last 43 measurements, from which the last 18 were from the damaged structure.

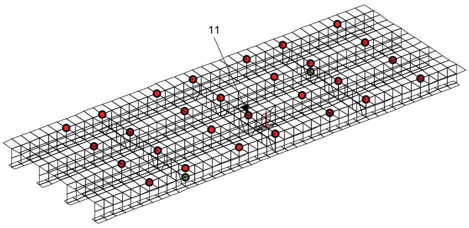


Fig. 2. Finite element model of the bridge deck. The sensor network is shown with red dots. Excitation locations are the two green dots, and the crack in a girder is plotted in red.

Damage detection was visualized with an EVS control chart, designed for the first principal component scores of the residual vector with a subgroup size of 100 samples. A generalized extreme value distribution was identified, and the PDFs of the minima and the negative maxima are plotted in Figs. 5 and 6 together with the corresponding histograms. It can be seen that the probability distributions were correctly identified. Damage detection using the actual measurements and the virtual sensors are shown in Figs. 7 and 8, respectively. With the raw measurements, only the largest crack size could be detected, whereas with the virtual sensors, the four largest crack sizes were detected. There were no false alarms when using the actual measurements (Fig. 7) and one false alarm using the virtual sensors (Fig. 8).

Damage was assumed to locate in the vicinity of the sensor *u* with the largest Mahalanobis distance [3] of the residual (Eq. (11)). The Mahalanobis distances of the residuals for each sensor are plotted in Fig. 9. Damage was localized to sensor 11 that was indeed the nearest sensor from the crack (Fig. 2).

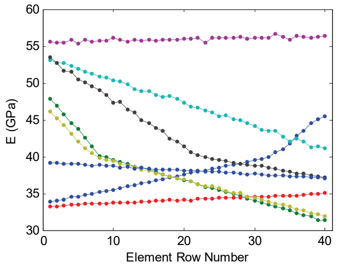


Fig. 3. Eight realizations of the longitudinal distributions of the Young's modulus of the concrete.

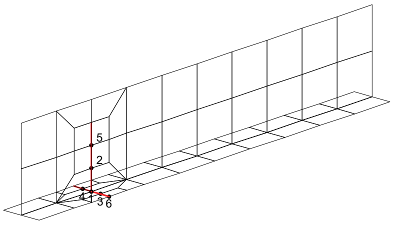


Fig. 4. A detail of the girder with damage. The crack is shown with red lines and the numbers indicate the order in which the connecting nodes separated to form an increasing crack size.

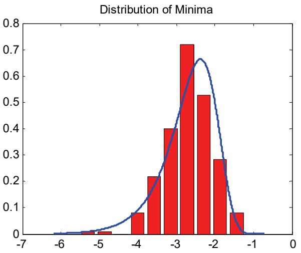


Fig. 5. Histogram and estimated PDF of the minima of the first principal component scores of the residuals in the training data.

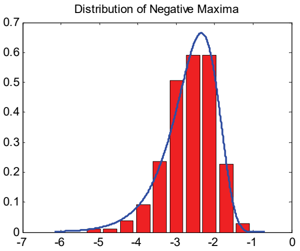


Fig. 6. Histogram and estimated PDF of the negative maxima of the first principal component scores of the residuals in the training data.

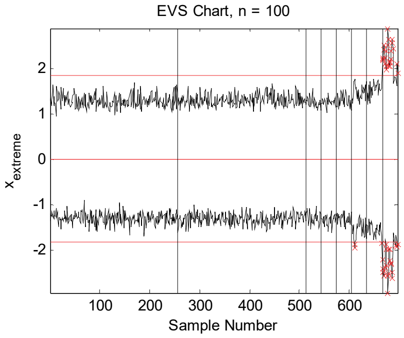


Fig. 7. EVS control chart using raw measurements. The six different damage levels are separated with the vertical lines in the right. The training data are to the left of the first vertical line.

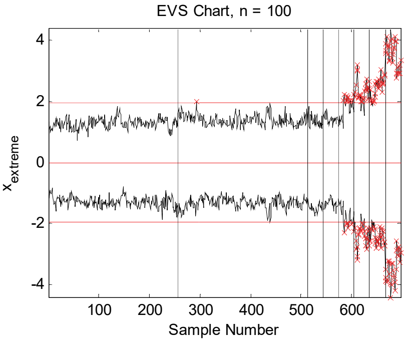


Fig. 8. EVS control chart using virtual sensors.

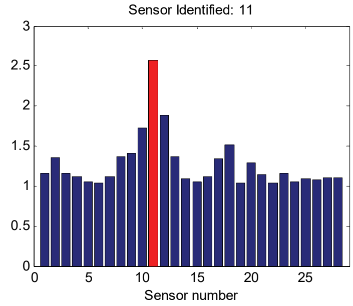


Fig. 9. Damage localization using virtual sensors.

5. Conclusions

With an increasing number of sensors, the measurement error can be reduced using Bayesian virtual sensors replacing the actual measurements and resulting in a better damage detection performance. Noise reduction was made for each measurement individually, while the environmental or operational influences were removed using all training data.

It should be noted that the proposed method was based solely on the response data, and no finite element model was used. Because damage detection was made in the time domain, no complex system identification was needed. The excitation was random and unknown. In addition, the environmental variables were not available.

Acknowledgements

This research has been supported by Metropolia University of Applied Sciences.

References

[1]. J. Kullaa, Bayesian virtual sensing in structural dynamics, *Mechanical Systems and Signal Processing*, Vol. 115, 2018, pp. 497-513.

[2]. J. Kullaa, Sensor validation using minimum mean square error estimation, *Mechanical Systems and Signal Processing*, Vol. 24, 2010, pp. 1444-1457.

[3]. C. M. Bishop, Pattern Recognition and Machine Learning, New York, *Springer*, 2006.

[4]. J. Kullaa, Eliminating environmental or operational influences in structural health monitoring using the missing data analysis, *Journal of Intelligent Material Systems and Structures*, Vol. 20, Issue 11, 2009, pp. 1381-1390.

[5]. K. Worden, D. Allen, H. Sohn, C. R. Farrar, Damage detection in mechanical structures using extreme value statistics, *SPIE Proceedings,* Vol. 4693, 2002, pp. 289-299.

[6]. D.C. Montgomery, Introduction to Statistical Quality Control, 3rd Ed., *Wiley*, New York, 1997.

(08)

Sparsity in Variational Autoencoders

Andrea Asperti [1]
[1] University of Bologna, Department of Informatics: Science and Engineering (DISI)

Summary: Working in high-dimensional latent spaces, the internal encoding of data in Variational Autoencoders becomes naturally *sparse*. We discuss this known but controversial phenomenon, sometimes referred to as overpruning, to emphasize the under-use of the model capacity. In fact, it is an important form of self-regularization, with all the typical benefits associated with sparsity: it forces the model to focus on the really important features, enhancing their disentanglement and reducing the risk of overfitting. Especially, it is a major methodological guide for the correct tuning of the model capacity, progressively augmenting it to attain sparsity, or conversely reducing the dimension of the network removing links to zeroed out neurons.

Keywords: Variational autoencoders, Sparsity, Kullback-Leibler divergence, Generative models.

1. Introduction

Variational Autoencoders (VAE) ([11, 13]) are a fascinating facet of autoencoders, supporting, among other things, random generation of new data samples. Many interesting researches have been recently devoted to this subject, aiming either to extend the paradigm, such as conditional VAE ([14, 15]), or to improve some of its aspects, as in the case of importance weighted autoencoders (IWAE) and their variants ([2, 12]). From the point of view of applications, variational autoencoders proved to be successful for generating many kinds of complex data ([8, 17]), comprising probabilistic predictions of unkown situations ([6, 16]).

Variational Autoencoders have a very nice mathematical theory (see [5] for an introduction), that we shall briefly survey in the next section. One important component of the objective function neatly resulting from this theory is the Kullback-Leibler divergence $KL(Q(z \mid X) \| P(z))$, where $Q(z \mid X)$ is the distribution of latent variables z given the data X guessed by the network, and $P(z)$ is a prior distribution of latent variables (typically, a Normal distribution). This component is acting as a regularizer, inducing a better distribution of latent variables, essential for generative sampling.

An additional effect of the Kullback-Leibler component is that, working in latent spaces of sufficiently high-dimension, the network learns representations sensibly more compact than the actual network capacity: many latent variables are zeroed-out independently from the input, and are completely neglected by the generator. In the case of VAE, this phenomenon was first observed in [2]; following a terminology introduced in [18], it is frequently referred to as overpruning, to stress the fact that the model is induced to learn a suboptimal generative model by limiting itself to exploit a small number of latent variables. From this point of view, it is usually regarded as a negative property, and different training mechanisms have been envisaged to tackle this issue: we shall survey on the recent literature in Section 5. In this article, we take a slightly different perspective, similar to the one advocated in [9, 3], looking at *sparsity* of latent variables as an important form of self-regularization, with all the typical benefits associated with it: in particular, it forces the model to focus on the really important features, typically resulting in a more robust and disentangled encoding, less prone to overfitting. Sparsity is usually achieved in Neural Network by means of weight-decay L1 regularizers (see e.g. [7]), and it is hence a pleasant surprise to discover that a similar effect is induced in VAEs by the Kullback-Leibler component of the objective function.

The most interesting consequence is that, at least for a given architecture, there seems to exist an *intrinsic* internal dimension of data. This property can be exploited both to understand if the network has sufficient internal capacity, augmenting it to attain sparsity, or conversely to reduce the dimension of the network removing links to unused neurons. Sparsity also helps to explain a loss of variability in random generative sampling from the latent space one may sometimes observe with variational autoencoders.

2. Variational Autoencoders

In latent variable models we express the probability of a data point X through marginalization over a vector of latent variables:

$$\begin{aligned} P(X) &= \int P(X \mid z, \theta) P(z) dz \approx \\ &\approx \mathbb{E}_{z \sim P(z)} P(X \mid z, \theta), \end{aligned} \tag{1}$$

where θ are parameters of the model (we shall omit them in the sequel).

Sampling in the latent space may be problematic for several reasons. The variational approach exploits sampling from an auxiliary distribution $Q(z \mid X)$. The relation between $P(X)$ and $\mathbb{E}_{z \sim Q(z|X)} P(X \mid z)$ is expressed by the following equation:

$$log(P(X)) - KL(Q(z\,|\,X)\,||\,P(z\,|\,X)) = \mathbb{E}_{z\sim Q(z|X)} log(P(X\,|\,z) - KL(Q(z\,|\,X)\,||\,P(z)) \quad (2)$$

Since the Kullback-Leibler divergence is always positive, the term on the right is a lower bound to the loglikelihood $P(X)$, known as Evidence Lower Bound (ELBO).

Supposing $Q(z\,|\,X)$ is a reasonable approximation of $P(z\,|\,X)$, the quantity $KL(Q(z)\,||\,P(z\,|\,X))$ is small; in this case the loglikelihood $P(X)$ is close to the Evidence Lower Bound, and it looks reasonable to take as learning objective its maximization.

Note that the ELBO has a form resembling an autoencoder, where the term $Q(z\,|\,X)$ maps the input X to the latent representation z, and $P(X\,|\,z)$ decodes z back to X.

The common assumption in variational autoencoders is that $Q(z\,|\,X)$ is normally distributed around an encoding function $\mu_\theta(X)$, with variance $\sigma_\theta(X)$; similarly $P(X\,|\,z)$ is normally distributed around a decoder function $d_\theta(z)$. All functions μ_θ, σ_θ and d_θ are computed by neural networks. Knowing the variance of latent variables allows sampling during training.

Provided the decoder function $d_\theta(z)$ has enough power, the shape of the prior distribution $P(z)$ for latent variables can be arbitrary, and for simplicity it is assumed to be a normal distribution $P(z) = G(0,1)$. The term $KL(Q(z\,|\,X)\,||\,P(z)$ is hence the KL-divergence between two Gaussian distributions $G(\mu_\theta(X), \sigma_\theta^2(X))$ and $G(1,0)$ which can be computed in closed form:

$$KL(G(\mu_\theta(X), \sigma_\theta(X)), G(0,1)) = \frac{1}{2}(\mu_\theta(X)^2 + \sigma_\theta^2(X) - log(\sigma_\theta^2(X)) - 1) \quad (3)$$

As for the term $\mathbb{E}_{z\sim Q(z|X)} log(P(X\,|\,z)$, under the Gaussian assumption the logarithm of $P(X\,|\,z)$ is just the quadratic distance between X and its reconstruction $d_\theta(z)$.

The problem of integrating sampling with backpropagation, is solved by the well known reparametrization trick ([11, 13]).

3. Sparsity

In a video available on line[1] we describe the trajectories in a binary latent space followed by ten random digits of the MNIST dataset (one for each class) during the first epoch of training. The animation is summarized in Fig. 1, where we use a fading effect to describe the evolution in time.

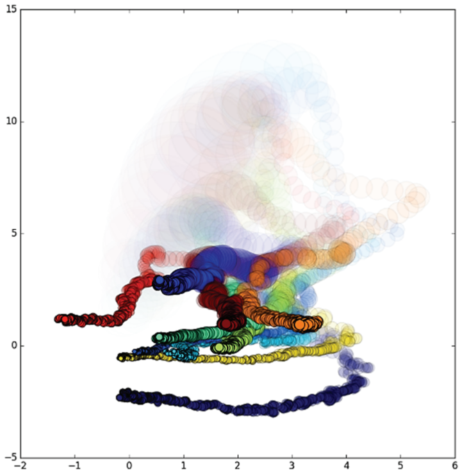


Fig. 1. Trajectories of ten MNIST digist in a binary latent space during the first epoch of training; pictures fade away with time.

Each digit is depicted by a circle with an area proportional to its variance. Intuitively, you can think of this area as the portion of the latent space producing a reconstruction similar to the original. At start time, the variance is close to 1, but it rapidly gets much smaller. This is not surprising, since we need to find a place for 60000 *different* digits. Note also that the ten digits initially have a chaotic distribution, but progressively dispose themselves around the origin in a Gaussian-like shape.

The previous behaviour is the expected one. However, augmenting the number of dimensions of the latent space, we face an interesting phenomenon: the representation becomes *sparse*.

In Fig. 2 we show the evolution during a typical training of the variance of latent variables in a space of dimension 16.

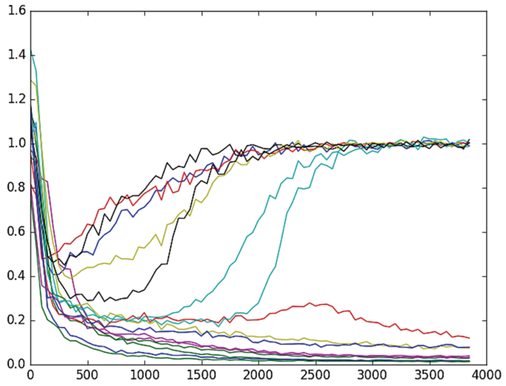


Fig. 2. Evolution of the variance along training (16 variables, MNIST case). On the x-axis we have numbers of minibatches, each one of size 128.

[1] http://www.cs.unibo.it/~asperti/variational.html

Table 1 provides relevant statistics for each latent variable at the end of training, computed over the full dataset: the mean of its variance (that we expect to be around 1, since it should be normally distributed), and the mean of the computed variance $\sigma_\theta^2(X)$ (that we expect to be a small value, close to 0). The mean value is around 0 as expected, and we do not report it.

Table 1. Inactive variables in the VAE for generating MNIST digits (dense case).

no.	variance	mean($\sigma_\theta^2(X)$)
0	258.847272e-05	25 0.9802212
1	250.00011756	250.99551463
2	256.665453e-05	250.98517334
3	0.97417927	0.008741336
4	0.99131817	0.006186147
5	1.0012343	0.010142518
6	0.94563377	0.057169348
7	250.00015841	250.98205334
8	0.94694275	0.033207607
9	250.00014789	250.98505586
10	1.0040375	0.018151345
11	0.98543876	0.023995731
12	250.000107441	250.9829797
13	254.5068125e-05	250.998983
14	250.00010853	250.9604088
15	0.9886378	0.044405878

All variables highlighted in red have an anomalous behavior: their variance is very low (in practice, they *always* have value 0), while the variance $\sigma_\theta^2(X)$ computed by the network is around 1 for each X. In other words, the representation is getting *sparse*! Only 8 latent variables out of 16 are in use: the other ones are completely ignored by the generator. For instance, in Fig. 3 we show a few digits randomly generated from Gaussian sampling in the latent space (upper line) and the result of generation when inactive latent variables have been zeroed-out (lower line): they are indistinguishable.

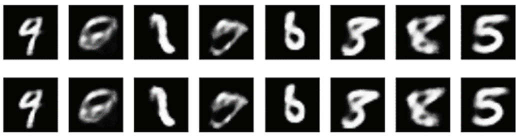


Fig. 3. Upper line: digits generated from a vector of 16 normally sampled latent variables. Lower line: digits generated after "red" variables have been zeroed-out: these latent variables are completely neglected by the generator.

4. KL-divergence and Sparsity

Let us consider again the loglikelihood for data X.

$$\mathbb{E}_{z\sim Q(z|X)} log(P(X\,|\,z) - KL(Q(z\,|\,X)\,||\,P(z))$$

If we remove the Kullback-Leibler component from the previous objective function, or just keep the quadratic penalty on latent variables, the sparsity phenomenon disappears. So, sparsity must be related to that component, and in particular to the part of the term trying to keep the variance close to 1, that is

$$-\sigma_\theta^2(X) + log(\sigma_\theta^2(X)) + 1\,, \qquad (4)$$

whose effect typically degrades the distinctive characteristics of the features. It is also evident that if the generator ignores a latent variable, $P(X\,|\,z)$ will not depend on it and the loglikelihood is maximal when the distribution of $Q(z\,|\,X)$ is equal to the prior distribution $P(z)$, that is just a normal distribution with 0 mean and standard deviation 1. In other words, the generator is induced to learn a value $\mu_\theta(X) = 0$, ans a value $\sigma_\theta(X) = 1$; sampling has no effect, since the sampled value for z will just be ignored.

During training, if a latent variable is of moderate interest for reconstructing the input (in comparison with other variables), the network will learn to give less importance to it; at the end, the Kullback-Leibler divergence may prevail, pushing the mean towards 0 and the standard deviation towards 1. This will make the latent variable even more noisy, in a vicious loop that will eventually induce the network to completely ignore the latent variable.

We can get some empirical evidence of the previous phenomenon by artificially deteriorating the quality of a specific latent variable. In Fig. 4 we show the evolution during training of one of the active variables of the variational autoencoder in Table 1 subject to a progressive addition of Gaussian noise. During the experiment, we force the variables that were already inactive to remain so, otherwise the network would compensate the deterioration of a new variable by revitalizing one of the dead ones.

In order to evaluate the contribution of the variable to the loss function we compute the difference between the reconstruction error when the latent variable is zeroed out with respect to the case when it is normally taken into account; we call this information *reconstruction gain*.

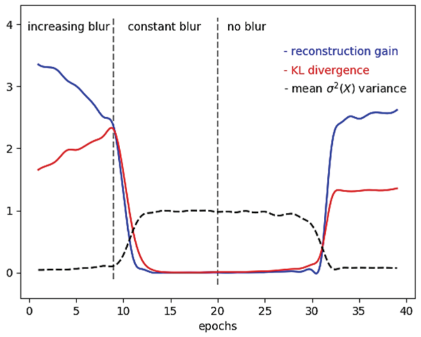


Fig. 4. Evolution of reconstruction gain and KL-divergence of a latent variable during training, acting on its quality by addition of Gaussian blur. We also show in the same picture the evolution of the variance, to compare their progress.

After each increment of the Gaussian noise we repeat one epoch of training, to allow the network to suitably reconfigure itself. In this particular case, the network reacts to the Gaussian noise by enlarging the mean values $\mu_x(X)$, in an attempt to escape from the noisy region, but also jointly increasing the KL-divergence. At some point, the reconstruction gain of the variable is becoming less than the KL-divergence; at this point we stop incrementing the Gaussian blur. Here, we assist to the sparsity phenomenon: the KL-term is suddenly pushing variance towards 1 (due to equation (4)), with the result of decreasing the KL-divergence, but also causing a sudden and catastrophic collapse of the reconstruction gain of the latent variable.

Contrarily to what is frequently believed, sparsity seems to be reversible, at some extent. If we remove noise from the variable, as soon as the network is able to perceive a potentiality in it (that may take several epochs, as evident if Fig. 4), it will eventually make a suitable use of it. Of course, we should not expect to recover the original information gain, since the network may have meanwhile learned a different repartition of roles among latent variables.

5. A Controversial Topic

Sparsity has been mostly intrerpreted as a negative property of Variational Autoencoders, responsible for an *overpruning* of the network capacity: an issue to be solved.

The most typical approach to tackle overpruning is that of using a parameter to trade off the contribution of the reconstruction error with respect to the Kullback-Leibler regularizer [1]:

$$-log(P(X)) \approx \\ -\mathbb{E}_{z\sim Q(z|X)} log(P(X\mid z) + \lambda KL(Q(z\mid X) \,||\, P(z))$$

The theoretical role of this λ-parameter is not so evident; let us briefly discuss it. In the closed form of the traditional logloss there are two parameters that seems to come out of the blue, and that may help to understand the λ. The first one is the variance of the prior distribution, that seems to be arbitrarily set to 1. However, as e.g. observed in [5], a different variance for the prior may be easily compensated by the learned means $\mu(X)$ and variances $\sigma^2(X)$ for the posterior distribution $Q(z\mid X)$: in other words, the variance of the prior has essentially the role of fixing a unit of measure for the latent space. The second choice that looks arbitrary is the assumption that the distribution $P(X\mid z)$ has a normal shape around the decoder function $d_\theta(z)$: in fact, in this case, the variance of this distribution may strongly affect the resulting loss function, and can justify the introduction of a balancing λ parameter.

Tuning down λ reduces the number of inactive latent variable, but this may not result in an improved quality of generated samples: the network use the additional capacity to improve the quality of reconstruction, at the price of having a less regular distribution in the latent space, that becomes harder to exploit by a random generator.

More complex variants of the previous technique comprise an annealed optimization schedule for λ [1] or enforcing minimum KL contribution from subsets of latent units [10]. It is generally true that inactive latent variables are annihilated at a quite early stage of training, and that relevant, quick learned features may kill less significant ones (suggesting to use a specific $lambda$ for each feature), but it is not evident that these kind of therapeutic obstinancy are really worth the effort. As observed in [18], all these schemes not only require hand-tuning, but especially risk to “take away the principled regularization scheme that is built into VAE.”

A more recent turn along the previous line is that offered by β-VAE [9, 3]. Formally, the shape of the objective function is the same of equation 5 (where the parameter λ is renamed β), but in this case the emphasis is in pushing β to be *high*. This is reinforcing the sparsity effect of the Kullback-Leibler regularizer, inducing the model to learn more disentangled features. To prove this fact, the authors also introduce an interesting metrics to measure the degree of disentanglement learned by the model, and provide experimental results confirming the beneficial effects of a strong regularization.

Coming back to over-pruning, a different way to tackle the issue is that model-based, consisting in devising architectural modifications that may alleviate the problem. For instance, in [19] the authors propose a probabilistic generative model composed by a number of sparse variational autoencoders called *epitoms* that partially share their encoder-decoder architectures. The intuitive idea is that each data X can be embedded into a small subspace K_X of the latent space, that is however specific to the given data.

Similarly, in [4] *skip-connections* are advocated as a possible technique to address over-pruning.

While there is no doubt that particular network architectures show less sparsity than others (for instance, convolutional networks usually are usually less sparse than dense ones for a similar dimension of the latent space), in order to claim that the aforementioned approaches are general technique for *tackling overpruning* it should be proved that they systematically lead to improved generative models across multiple architectures and many different data sets, that is a result still in want of confirmation.

References

[1]. S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, S. Bengio, Generating sentences from a continuous space, *CoRR*, arXiv:1511.06349, 2015.

[2]. Y. Burda, R. B. Grosse, R. Salakhutdinov, Importance weighted autoencoders, *CoRR*, arXiv:1509.00519, 2015.

[3]. C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding disentangling in β-VAE, in *Proceedings of the* 2017 *NIPS Workshop on Learning Disentangled Representations,* 2017.

[4]. A. B. Dieng, Y. Kim, A. M. Rush, D. M. Blei, Avoiding latent variable collapse with generative skip models, *CoRR*, arXiv:1807.04863, 2018.

[5]. C. Doersch. Tutorial on variational autoencoders, *CoRR*, arXiv:1606.05908, 2016.

[6]. S. M. A. Eslami, D. J. Rezende, et al., Neural scene representation and rendering, *Science*, Vol. 360, Issue 6394, 2018, pp. 1204-1210.

[7]. I. Goodfellow, Y. Bengio, A. Courville, Deep Learning*, MIT Press*, 2016.

[8]. K. Gregor, I. Danihelka, A. Graves, D. Wierstra, DRAW: A recurrent neural network for image generation, *CoRR*, arXiv:1502.04623, 2015.

[9]. I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, beta-VAE: Learning basic visual concepts with a constrained variational framework, in Proceedings of the *ICLR 2017 Conference,* 2017.

[10]. D. P. Kingma, T. Salimans, M. Welling, Improving variational inference with inverse autoregressive flow, *CoRR*, *arXiv:1606.04934*, 2016.

[11]. D. P. Kingma, M. Welling, Auto-encoding variational bayes, *CoRR*, *arXiv:1312.6114*, 2013.

[12]. T. Rainforth, A. R. Kosiorek, et al., Tighter variational bounds are not necessarily better, in *Proceedings of the 35th International Conference on Machine Learning (ICML'18),* Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, pp. 4274-4282.

[13]. D. J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in *Proceedings of the 31th International Conference on Machine Learning (ICML'14),* Beijing, China, 21-26 June 2014, pp. 1278-1286.

[14]. K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, in Advances in Neural Information Processing Systems (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett, Eds.), Vol. 28, *Curran Associates Inc.*, 2015, pp. 3483-3491.

[15]. J. Walker, C. Doersch, A. Gupta, M. Hebert, An uncertain future: Forecasting from static images using variational autoencoders, *CoRR*, *arXiv:1606.07873*, 2016.

[16]. J. Walker, C. Doersch, A. Gupta, M. Hebert, An uncertain future: Forecasting from static images using variational autoencoders, in *Proceedings of the 14th European Conference on Computer Vision (ECCV'16),* Amsterdam, The Netherlands, October 11-14, 2016, pp. 835-851.

[17]. R. A. Yeh, Z. Liu, D. B. Goldman, A. Agarwala, Semantic facial expression editing using autoencoded flow, *CoRR*, *arXiv:1611.09961*, 2016.

[18]. S. Yeung, A. Kannan, Y. Dauphin, L. Fei-Fei, Tackling over-pruning in variational autoencoders, *CoRR*, *arXiv:1706.03643*, 2017.

[19]. S. Yeung, A. Kannan, Y. Dauphin, Epitomic variational autoencoder, in *Proceedings of the ICLR 2017 Conference*, 2017.

(09)

Decisional Bayesian Network for Risk Assessment in Industry

N. Aissani [1], I. H. M. Guetarni [1] and Z. Djemai [1]
[1] Laboratoire de l'ingénierie de la sécurité et Du developpement Durable, Institut de Maintenance et de sécurité Industrielle, Université Mohamed Ben Ahmed Oran2, Algeria
Tel.: + 213663666205
E-mail: aissani.nassima@univ-oran2.dz

Summary: In this paper, Bayesian approach will be experimented on safety data of a cement factory (CFWA) to develop a decisional support tool for risk assessment. In this company, accident data are saved from 7 years ago, however some records are uncomplete or description is not clear so a filtration method is developed to deal with incompleteness and uncertainty of data. Then, decisional support system for risk assessment (DSRS) is developed as conceptual and generic model with a protocol for assessing the risk of injuries on. The model is experimented by generating many scenarios to understand and create knowledge about causes or conditions that favored those dreaded events: person injuries.

Keywords Industrial Safety, Bayesian approach, Decision making, Data treatment, Cement factories.

1. Introduction

The Bayesian method makes it possible to deduce the probability of an event from those of other events already evaluated. It's based on conditional probability. From a long time Bayesian approach was used in risk analysis, especially in economics area [1] like in The Credit Granting Decision [2] and also in health prevention and diagnosis in 80 ties it was called an art [3].

In the last few years there has been a growing interest in using BN (Bayesian Network) for risk modeling in industrial context and especial in high risk context like on offshore installation [4] and also in other types of industry like construction industry [5]. In this paper BN is experimented to assess cement factory risks.

2. Decisional Process

The decisional engine (DE) is based on Bayesian network developed on the cases analysis (Fig. 1).

1. New & validation data: Data base which contains the case of study data. It can be the first stage data that is used in the BN creation or the validation data or update data that are used to adjust the BN results every time new data are available.

2. BN Inference: BN nodes are the data base items; the decisional item is the central node which depends on the other descriptive items. Each node has a list of values determined by the data treatment module after the first integration of data.

3. Data treatment: In the data base, items with a list of value will not be treated because possible values are identified for this item. However, for textual items possible values are not identified and to define this list an algorithm is developed.

4. Interrogation interface: It is a simplified interface in HTML5 format to be downloaded on any device. User can describe its situation according a list of question and a list of possible responses which meet the BN nodes values. This query defines a scenario that can be applied on the BN and then give a risk evaluation or possible causes as response.

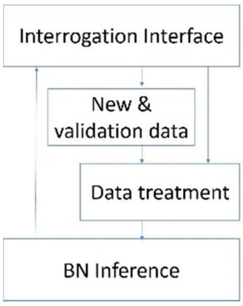


Fig. 1. Decisional process architecture.

3. Case of Study and Problem

The cement factory, our case of study, is a new construction; it is composed of two production lines: grey one and black one. It produces about 8000 t/day. The line is composed of 4 sections: crusher, sand crusher, raw flour mill, kiln and white cement crusher. The production is non-stop with 3 teams each one works for 8 hours. This intensive activity and the process complexity make the factory not safe, from 2011 until now, about 200 incidents, involving people with different levels of severity, were registered. The safety department is responsible for reporting and identifying any type of accident. However, sometimes description is not clear and data are incomplete. That's why; any exploitation of these data needs use of evolutionary methods to deal with incomplete and uncertainty. Its why data treatment module is integrated in decisional process.

3.1. Data Treatment

Accident data are saved in database were each description is saved under item name. Each item could have different value, these values should be standardized in order to achieve learning process. An algorithm is developed to select similar values and replace them by the most used.

Another algorithm is developed to standardize a textual item description (values) it select key word and return a key short synthesis that could be intersection of many stored values and replace them.

At this stage, data are ready to be used in Bayesian learning to feed nodes probabilities.

3.2. Bayesian Network Node Identification

At this step, data are filtered and relevant items are identified, then node can be modeled. The node is each item and the list of its possible values are the labels of that node. The dreaded event, which is the “accident”, is the node to predict in this network, so it is at the center of the network as shown in Fig. 2. Some items are consequence of the dreaded event like: “potential gravity" describing the degree of damage and "injured body part" describing the level of body injury. These items are designed as cause nodes for the dreaded event. For each node a list of possible values is identified it is called node states.

3.3. Data Exploitation and Bayesian Inference

Now data are able to be used in the network inference and the probability tables can be filled. For each item the probability is elementary. But for the dreaded event, which is the centre of the Bayesian network, its probability depends on the other nodes that are the causes and according to the Bayesian theorem it is given by the formula:

$$p\left(de = value / c_1 = value_1 \,\&\ldots\&\, c_n = value_n\right),$$
$$= \frac{p\left(c_1 = value_1 \,\&\ldots\&\, c_n = value_n \;/\; de = value\right)}{p\left(de = value\right)},$$

where *de* is the dreaded event, *ci* are the cause events and *n* is the number of causes. A Bayesian inference allows all events’ probabilities update. The dreaded event is “Categorie de l’incident” or “Category of the incident” it is also a decisional node as well as “causes” and “parties du corps blesses” or “body injured parts”. The other nodes are descriptive nodes.

3.4. Scenario’s Analysis

A scenario means the assumption of a truth and the observation of its consequences on the dreaded events as well as the other consequent events. In this paper, 6 scenarios are developed (Fig. 3). According to this analysis, the serious and sever accidents mainly involve not very vigilant subcontractors. As a result, the following recommendations could be proposed:

- Enlighten and inform workers, especially subcontractors because they have been the most affected, about risks and measures of prevention and safety.
- Respect limitation of work time, to maintain the concentration and vigilance of workers.
- Convince the workers about the obligation to respect the internal rules of the company (work permit, PPE...).

3.5. Comparison with External Data

The decisional module could be feed with new data at any time to update its confidence probabilities. In “globalcement” (http://www.globalcement.com) data base about 86 fatal accident and 115 in all enclosing Fatal, Critical and Accident with stop, this database enclose only critical and fatal accident. Even if these new data addressed fatal accident it meets our BN conclusion, where all scenarios are confirmed (Fig. 3) except the fourth one: when company worker had accident with high severity it has 5.55 % chance that he had a hand or finger burn or cut.

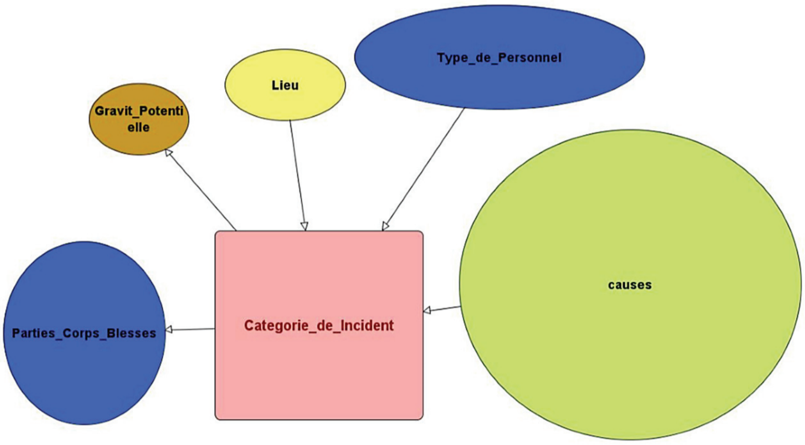


Fig. 2. Developed Bayesian Network.

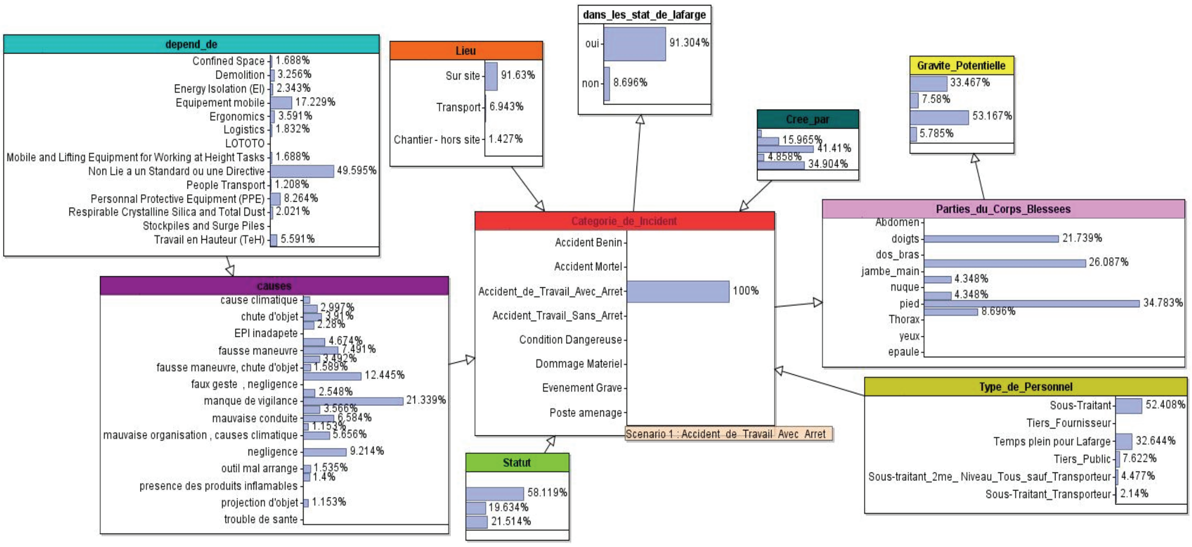


Fig. 2. Bayesian Network for injury risk prediction.

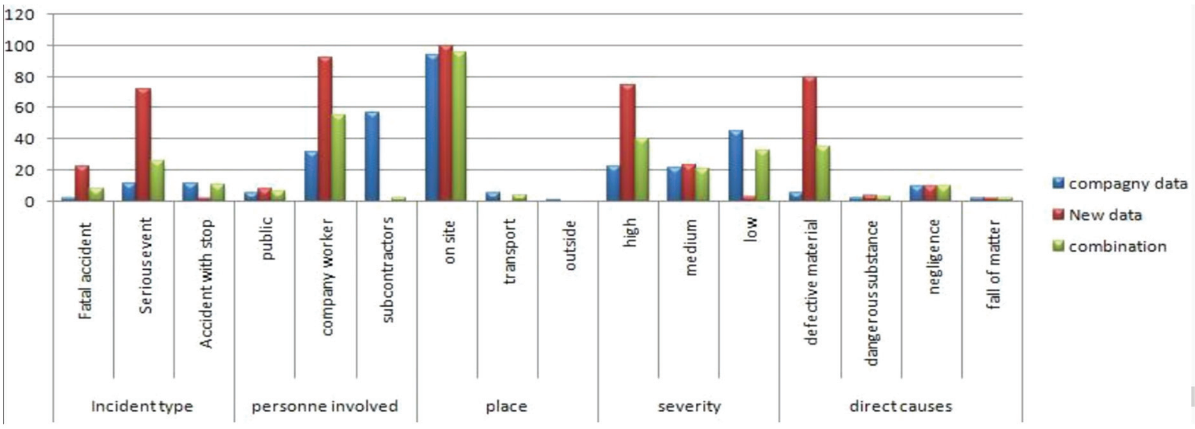


Fig. 3. Bayesian Network for injury risk prediction.

4. Conclusions

In this paper an application of Bayesian approach is presented to develop a decisional support system for risk assessment. It should be updated over the time with new recorded data and also validate it with other factories accident data to confirm some knowledge especially those with a little confidence like fatal accidents this was done in the last section.

References

[1]. A. Zellner, Bayesian analysis in econometrics, *Journal of Econometrics*, Vol. 37, Issue 1, 1988, pp. 27-50.

[2]. H. Bierman, W. H. Hausman. The credit granting decision. manage, *Sci.*, Vol. 16, Issue 8, 1970, pp. B-519-B-532.

[3]. D. M. Eddy, C. H. Clanton, The art of diagnosis, *New England Journal of Medicine*, Vol. 306, Issue 21, 1982, pp. 1263-1268

[4]. J. Ren, I. Jenkinson, J. Wang, D. L. Xu, J. B. Yang, An offshore risk analysis method using fuzzy Bayesian network, *Journal of Offshore Mechanics and Arctic Engineering*, Vol. 131, 2009, 041101.

[5]. S. S. Alizadeh, S. B. Mortazavi, M. M. Sepehri, Assessment of accident severity in the construction industry using the Bayesian theorem, *International Journal of Occupational Safety and Ergonomics*, Vol. 21, 2015, pp. 551-557.

(10)

Inductive Matrix Completion for Network Fusion

Pengwei Hu [1, 2], Shaochun Li [2], Shuhang Gu [3], Lun Hu [4] and Keith C.C. Chan [1]
[1] Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[2] IBM Research, Beijing, China
[3] Computer Vision Lab, ETH Zurich, Switzerland
[4] School of Computer Science and Technology, Wuhan University of Technology
E-mails: csphu@comp.polyu.edu.hk, lishaoc@cn.ibm.com, shuhanggu@gmail.com, hulun@whut.edu.cn,
cskcchan@comp.polyu.edu.hk

Summary: Network is a set of nodes linked together. Given two or more networks with unique topology and content that are related to some extent, good fusion is a key step to increase understanding of networks. Most fusion techniques focus solely on patterns that seem to be common to most networks. Using an adjacency matrix to represent a network, network fusion problem becomes matrix issue. In this work, we propose complete network fusion (CNF) by first introduce matrix completion in the fusion step. Lost complementary information has largely been recovered. Experiments indicate that our model outperforms the state-of-the-art approaches.

Keywords: Low-rank approximation, Fusion, Clustering, Matrix completion, Social media intelligence.

1. Introduction

While the information networks have become ubiquitous, the challenge of collecting comprehensive network from multiple networks still persists [3, 5]. On top of that, multi-sensor network, multi-target network, social network and disease network, all applications need to be integrated from multiple sources. Therefore, many network fusion methods have been proposed and aroused great research interest, among which the core goal is to generate the optimal representation of co-directed entities based on a variety of information. However, many times the fused network is incomplete with nodes and edges missing. FCAN [3] is proposed to concatenate two networks directly for clustering complex network. CESNA [5] handle multiple networks which utilize generative models. MISAGA [2] is proposed to solve multi-networks as a constrained optimization problem. Commonly, only a part of the network information can be observed and we would like to infer the unobserved part of the network. We address this issue by using a novel model called complete network fusion (CNF). CNF adopts an advanced iterative method to update the existing multiple networks, aiming to make the network data more similar. Finally, CNF revaluates the instantaneous network at each step. Let's take the network as a matrix, some values of matrix may be lost in the induction process. We use weighted nuclear norm minimization (WNNM) based matrix completion to estimate the missing part of the network. Experiments on two real-world networks show that our approach can effectively fuse the networks and recover some useful network values which miss in the process of fusion. Finally, we introduce the latest network clustering algorithm for comparative experiments, and the experimental results show that the communities detected by CNF are more matching with the verified communities. In general, CNF is a comprehensive network model capable of learning to express multiple networks. Compared with the existing methods, the matrix completion proposed by CNF completes the missing node relations in the fusion network, and fuses the iteratively completed network into a more reliable network to facilitate the discovery of the communities in the network.

2. Proposed Method

Suppose each sample has m feature types, and there are n samples in total, which means that we can construct m representation networks for these n samples. For a network on multiple networks, let $G = \{V, E\}$ represent a network structure, then $V = \{u_1, u_2, \dots, u_n\}$ in G represent that n vertices in the network, E represent the edge between the vertices in the network. In a topological network, the presence or absence of edges indicates whether the two points are connected. In a content network, the higher the value of edges means the higher the similarity between the two points. Here we use an n-by-n matrix M to represent the edge weights. CNF introduces the complete kernel concept on the vertex set V [4], and uses the normalized weight matrix $K = D^{-1}M$ to obtain the fused network representation from the multiple network representation. Let's say that D is a diagonal matrix, and we have $D(i,i) = \sum_j M(i,j)$, and $\sum_j K(i,j) = 1$. Here, we place the degree of self-similarity on the diagonal entries of the M and keep $\sum_j K(i,j) = 1$, so as to obtain a better normalization result:

$$K(i,j) = \frac{M(i,j)}{2\sum_{k \neq i} M(i,k)}, \qquad (1)$$

subject to the constraints: $i \neq j$, otherwise $K(i,j) = \frac{1}{2}$. Here, we introduce KNN algorithm to measure local affinity, and then assume N_i as a set of u_i's neighbors including u_i in G:

$$Q(i,j) = \frac{M(i,j)}{\sum_{k\in N_i} M(i,k)}, \tag{2}$$

subject to the constraints: $j \in N_{i,}$ otherwise $Q(i,j) = 0$. In this study, there are content network and topological network, so let $m = 2$. The initial v state network should be $K_{t=0}^{(v)}$, the kernel matrix should be $Q_{t=0}^{(v)}$ and the CNF fusion step is to iteratively update each network:

$$K^{(v)} = Q^{(v)} \times \left(\frac{\sum_{k\neq v} K^{(k)}}{m-1}\right) \times \left(Q^{(v)}\right)^T, \quad v = 1,2,3,\dots,m \tag{3}$$

In order to solve the problem of information loss in above step, CNF takes benefit from the network node correlation, and find a low-rank approximation of $K^{(v)}$ to recover the missing values. Some early studies use factorization-based methods to find a low-rank approximation of the input network, while recent nuclear norm based methods get the low-rank approximation by regularizing its singular values. The recently proposed WNNM method is able to adeptly regularize the singular values of matrix, has shown its advantages in different applications. Here, CNF use the WNNM matrix completion method to complete the missing values in the fused network $K^{(v)}$ in each step of iteration, which lead to the following optimization problem $\min_X \|X\|_{w,*}\ s.t. \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(K^{(v)})$ and we reformulate it by introducing a variable *E:*

$$\min_X \|X\|_{w,*}\ s.t. X + E = K^{(v)}, \mathcal{P}_\Omega(E) = 0, \tag{4}$$

where w is the weight vector and indicator matrix *Ω* generated by the nodes that value less than a threshold in the fused network. We solve (4) by the ADMM algorithm, which alternatively updates *X*, *E* and the augmented Lagrange variable L like previous study [1]. We use the stop criterion as the stopping condition: $\|Y - X_{k+1} - E_{k+1}\|_\text{F}/\|Y\|_\text{F} > \theta$. Then we update the network by *X*. CNF executes (1) on $K^{(v)}$ after each update. Run v parallel interchanging diffusion process through the above update state network, and finally get $K^{(v)} = \frac{\sum_{v=1}^{m} K^v}{m}$. Spectral clustering will be used to obtain network clusters

3. Experiments

Data Sets:

In this study, we employ experiments on two data sets: Caltech and Ego-Facebook gathered from [2]. Each data collection consists of several known ground truth clusters. In these two data collections, both the number of users and the number of communities are very different scale. They also include attribute values associated with the vertex that is used to generate content-based network and user following relations are used to generate link-based network.

1) Caltech: This is a small set of social network data collected from student friendships at Caltech. This data used the attribution of student dormitory to group the users of social network, and a total of 35 clusters were confirmed manually first. More specifically, the network data has 769 vertices, and 16656 edges representing associations between students.

2) Ego-Facebook: t is a much larger data set collected from Facebook with 193 manually identified communities. The data set contains 4039 Facebook users, including 88234 user connections.

Evaluation:

In this paper, we evaluate and compare our approach with three states of the art methods: MISAGA proposed in [2], CESNA proposed in [5] and original spectral clustering. A common and effective clustering task metric is NMI, which is a method of measurement based on information theory. The measurement concept of NMI is to measure the quality of clustering by matching the matching degree between discovered clusters and verified clusters. We adopted NMI as the evaluation metric in our experiment. The experimental results on all methods are shown in Table 1.

Table 1. Comparison of NMI scores.

Method	CNF	MISAGA	CESNA	SC
Facebook	**0.593**	0.543	0.58	0.52
Caltech	**0.361**	0.310	0.221	0.338

4. Conclusions

This paper studied multiple network fusion with matrix completion. To the best of our knowledge, our paper is the first study which uses matrix completion technique to recover missing information in the process of network fusion. Experimental results show that proposed model performs better than several existing methods. In the future, we plan to introduce fuzzy algorithms into CNF, allowing each node to belong to multiple communities. This scheme is more in line with the actual situation. Based on this scheme, it would be interesting to use deep representation to re-express the fused network features.

References

[1]. S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, L. Zhang, Weighted nuclear norm minimization and its

applications to low level vision, *International Journal of Computer Vision*, Vol. 121, Issue 2, 2017, pp. 183-208.

[2]. T. He, K. C. Chan, MISAGA: An algorithm for mining interesting subgraphs in attributed graphs, *IEEE Transactions on Cybernetics*, Vol. 48, Issue 5, 2017, pp. 1369-1382.

[3]. L. Hu, K. C. Chan, Fuzzy clustering in a complex network based on content relevance and link structures, *IEEE Transactions on Fuzzy Systems*, Vol. 24, Issue 2, 2016, pp. 456-470.

[4]. B. Wang, A. M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, B. Haibe Kains, A. Goldenberg. Similarity network fusion for aggregating data types on a genomic scale, *Nature Methods*, Vol. 11, Issue 3, 2014, pp. 333-337.

[5]. J. Yang, J. McAuley, J. Leskovec, Community detection in networks with node attributes, in *Proceedings of the* IEEE *International Conference on Data Mining (ICDM'13)*, 2013, pp. 1151-1156.

[6]. P. Hu, K. C. Chan, T. He, H. Leung, Deep fusion of multiple networks for learning latent social communities, in *Proceedings of the IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI'17)*, 2017, pp. 765-771.

(11)

Deep Reinforcement Learning for Route Optimization in Baggage Handling Systems

R. A. Sørensen [1,2], M. Nielsen [2] and H. Karstoft [1]
[1] Aarhus University, Department of Engineering, Finlandsgade 22, 8200 Aarhus N, Denmark
[2] BEUMER Group, P. O. Pedersens vej 10, 8200, Aarhus N, Denmark
Tel.: + 4527615984
E-mail: ras@eng.au.dk

Summary: In Baggage Handling Systems, traffic congestion and deadlocks are common problems, often caused by using static shortest path routes. Developing site specific routing rules to replace some of the shortest path routes can overcome these problems in certain situations. However, this is a time-consuming and costly approach, making it an interesting point of improvement. This paper presents a proof of concept in which a Deep RL method, called Double Deep Q-Network, is used to optimize routing in a simplified model of a Baggage Handling System. The Double Deep Q-Network method is compared with Dijkstra's shortest path algorithm computed statically and dynamically. When comparing the number of steps used to route a set of items, the Double Deep Q-Network method proved superior by, on average, using fewer steps in the tested environment.

Keywords: Deep reinforcement learning, Double DQN, Baggage handling system, Route optimization.

1. Introduction

This paper addresses the problem of route planning for Baggage Handling Systems (BHSs) in airports. From check-in to departure, baggage can travel several kilometres in specialized rail systems under the floor of airport terminals. A BHS is composed of many different elements, such as straights, curves, diverters, mergers, toploaders and discharges. When transporting baggage in a BHS, there are different alternatives. Destination Coded Vehicles, specialized baggage totes, conveyor belts, or a combination is used. This paper focuses on the use of specialized baggage totes. To correctly route these totes, current methods rely on using routing tables calculated with a mixture of Dijkstra’s shortest path (SP) algorithm [1] and next-hop routing. However, a BHS is a complex and dynamic environment with the possibility of events causing congestions or, in worst case, deadlocks in the system. Congestions lower the total throughput, decreasing the capacity of the BHS. Deadlocks must be solved by manually removing the blocking totes.

To overcome such difficulties, site-specific routing rules can be applied. However, developing such rules is a time-consuming and often very difficult task due to the complexity of BHSs. In this paper, a simplified model of a BHS is used to compare two different shortest path approaches with a Deep Reinforcement Learning (Deep RL) approach.

The expectation is that by using Deep RL, it will be possible to avoid scenarios resulting in congestions and deadlocks, thus reducing the difficulties of BHSs and thereby limiting the need for additional site-specific routing rules.

1.1. Related Work

In graph related routing problems, some of the common routing schemes make use of Dijkstra’s SP algorithm [1] or its variants. The SP algorithm finds the shortest path from one node to all other nodes. Given a continuous unchanged flow in the system and the absence of negative edges and infinite loops, the SP algorithm provably returns the shortest routing possible. However, a BHS is not a system with continuous unchanged flow, and it often contains loops. To solve this challenge, different alternatives have been proposed as an alternative to SP.

In Hallenborg et al [2], multi-agent control software is used to make routing decisions in a BHS. Local routing strategies are made for the elements controlling the totes in the BHS, making them act as collaborative agents. This work mentions strategies both for routing decisions, overtaking totes, empty tote handling, and saturation management. It shows that by using this method, it is possible to improve the performance of BHSs. However, it comes at the cost of a large communication overhead and introduces a task of manually designing the routing strategies.

In Tarau et al. [3], a comparison is made between centralized, decentralized, and distributed Model Predictive Control (MPC) methods. They find that the best performance is gained from using a computationally heavy centralized MPC approach. Using the decentralized MPC approach lowers the computational challenges. However, it also lowers the performance. The distributed MPC approach gives the best trade-off between the performance and the computational efforts of the three compared MPC methods. To further reduce the computational requirements, they propose two heuristic variations.

Applying these heuristic approaches results in a drop of performance, but also drastically lowers the computation time.

The performance of [2, 3] is high, but both have high computational costs and thus are difficult to use as real time control methods without loss of performance. This paper proposes an alternative method, namely the Deep RL method. By using Deep RL methods, compared to [2], it is possible to avoid manually designing routing strategies, thus allowing the model to optimize without human influence. From the work of [3], it is seen that using a centralized method performs better than using a decentralized and distributed method, disregarding the computational speedup achieved. This paper shows that Deep RL can use information from the entire BHS, as is the case of the centralized method from [3]. The computational efforts for inference are not analysed, but they are expected to be lower than the centralized method from [3] since only one forward pass through a Deep Neural Network (DNN) is needed.

RL methods for dynamic routing have been implemented with success in the internet package routing domain [4, 5]. One of the main differences between this domain and the BHS is that in BHSs the decision of routing is final, the moment it is committed, i.e. when a tote is sent in a certain direction, you cannot call it back and try another direction. Furthermore, congestions can extend to affect previous decision points so that they are unable to send totes in alternative directions.

In recent years, Deep RL has become a popular method due to its ability to learn how to control complex environments with large state spaces. Deep RL is a type of RL where a Deep Neural Network is used as function approximator. The Deep RL method DQN has beaten expert human players in the complex domain of Atari 2600 games, using only the image frame and the score as an input [6]. The method is based on using an agent to explore an environment and a DNN to predict the total future reward for the agent's possible actions given a state of the environment. By using experience replay they manage to get the DNN to converge towards the superhuman strategy, and without changing the method, they can do so for a majority of the Atari 2600 games. DQN is further developed into Double DQN [7]. They show that DQN is prone to overestimating the reward, resulting in worse performance. Their suggestion is to decouple the action from the reward estimation. They do so by letting actions be chosen with an *online* network with parameters θ, while the expected reward is estimated by a *target* network with parameters $\bar{\theta}$, which, in practice, is an older version of the online network. They define the target for the DNN as in (1).

$$Y_t = r_{t+1} + \gamma Q(s_{t+1}, \underset{a}{argmax}\, Q(s_{t+1},a;\theta);\bar{\theta}) \qquad (1)$$

This paper shows, as a proof of concept, that it is possible to use Double DQN to optimize routing of totes in an environment, modelling a simplified BHS. The method is compared with the performance of Dijkstra's SP, computed statically (sSP) and dynamically (dSP).

2. Method

2.1. Environment

To train a model using RL, an environment is needed. This environment must be able to receive actions, take steps and return rewards and state representations. The environment used in this work was a simplification of a BHS. It was implemented as a graph with nodes and edges using a Petri net. Petri nets can conduct stepwise processing, which was used to model the movement of totes in the simulated BHS. They have a special naming convention for nodes which are named places and edges which are named transitions. The Petri net was implemented using the python library snakes [8]. Transitions were implemented as FIFO queues to describe the length, i.e. the number of steps needed for totes to traverse them and the capacity, i.e. the maximum number of allowed totes on them, as seen in Fig. 1. To ensure that queueing would appear, the capacity of the transitions was set lower than the number of steps needed to traverse them, i.e. capacity < length. If capacity ≥ length, the only way a queue could appear was if the exit of the transition was blocked by other totes.

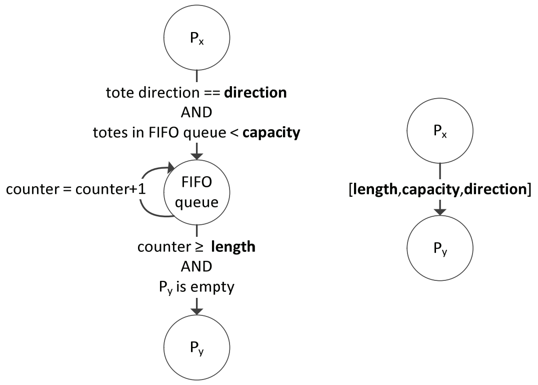

P_x
tote direction == **direction**
AND
totes in FIFO queue < **capacity**
counter = counter+1
FIFO queue
counter ≥ **length**
AND
P_y is empty
P_y
P_x
[length,capacity,direction]
P_y

Fig. 1. *Left*: The Petri net notation is shown for how totes were moved from place to place via a FIFO queue consisting of a place with an ingoing, looping and outgoing transition. The *tote direction* was set by the RL model while *direction*, *length* and *capacity* were staticky defined by the environment. *Right*: Throughout the rest of the paper, this notation will be used to describe the FIFO queue shown to the left.

Places, such as P_x and P_y in Fig. 1, are referred to as decision points. At these places, totes could be directed in different directions, simulating the behaviour of diverters. The places can also act as mergers to collect totes from different transitions. Some places are simulating toploaders where baggage enters the system. Lastly the places can simulate discharges, which enables totes, with that discharge as destination, to be removed from the system. In real systems the tote would stay in the system and only the baggage would be dropped. However, this is not considered in this paper.

2.1.1. Step

When the environment received an action, it took a step which consisted of the following operations:

1) One tote from each place was moved into the transaction which matched the tote direction defined by the action. If the transaction had reached its capacity, the tote was blocked until next step.
2) The counter for all totes in transitions was incremented by one.
3) If the step counter of the first tote in the FIFO queue had exceeded the size attribute of the transition, the tote was moved to the next place.

2.1.2. Action

The actions were controlling the tote directions at all decision points. The action was therefore a vector $\boldsymbol{a}$ of length P which was the number of decision points in the BHS. The values of $\boldsymbol{a}$ were integer values from 0 to d_p - 1, where d_p was the number of possible decisions at the p'th decision point.

2.1.3. State

When a step had been made, the environment returned a state to the RL agent. The environment was described as a $[E×F]$ matrix, where E was the number of elements in the environment and F was the number of features related to the totes. The features were:

1) *Destination* – the place the tote had to be directed to release a reward.
2) *Occupied* – a binary indication on whether a tote was present. This information was redundant since the remaining rows only contained non-zero values when no tote was present. It was, however, included to give a constant value to the network to rely on.
3) *Distance to place* – the distance to the next decision point. This was included to ease the work of the network, since it introduced a measure of how close a tote was to a point where an action could affect it directly.
4) *Time in transition* – number of steps a tote had been in its current transition. This was included to describe if totes had been waiting longer than usual at a specific transaction.
5) T*ime in system* – number of steps a tote had been in the system, which was used as an accumulated addition to negative reward.

Each feature was found empirically and while it would have been interesting to see the individual contribution of the features, it has not been done as a part of this paper.

2.1.4. Reward

Besides the state, the environment also returned a reward to the RL agent. The reward signal was used to describe the goals of the agent. In this simplified BHS, the goal was to get a high throughput and a low delay of totes, meaning that the totes had to traverse the BHS using as few steps as possible. In real systems, each piece of baggage must be delivered within a certain time slot. This is, however, not considered in this paper.

The stepwise reward r_t was calculated as a sum of scores scaled by three parameters (2):

1) *Destination score* – every time a tote reached its destination, a positive score of dst_{score} was given. This was done to get a positive edge towards getting totes to their destinations.
2) *Time penalty* – for each step a tote had been in the system, a counter describing the total time in system was increased by one. The sum of the counters of all active totes was multiplied with a negative score of $t_{penalty}$ and added to the reward. This was done to get a positive edge towards getting totes quickly to their destinations.
3) *Step penalty* – at each step, a negative score of $s_{penalty}$ was given. This was done to make sure that the system optimized towards using few steps.

$$r_t = n_t*dst_{score} - \sum \boldsymbol{c}_t*t_{penalty} - s_{penalty}, \qquad (2)$$

where n is the number of totes that arrived at their destinations in step t and $\boldsymbol{c}$ is an array containing the time counters of all active totes in step t.

2.2. Neural Network Architecture

The DNN seen in Fig. 2 was designed specifically to the BHS task and mainly consisted of convolutional layers. First, the input was converted from having dimensions $[E×F]$ to $[E×1×F]$, allowing the convolutional layer to have *same* padding, without excessive use of zero padding. Then a convolutional layer was used, followed by three parallel convolutions with different filter sizes to capture different sizes of features. After this, three serialized convolutional layers were used to allow for higher level features. The output consisted of a $[P×d_{max}]$ matrix, where P was the number of decision points and d_{max} = max(d_p).

The DNN was trained to estimate the Q-function, $Q(s,\boldsymbol{a})$, returning a vector of estimated reward values dependent on which action vector $\boldsymbol{a}_t$ was chosen. The expected discounted total future reward R' was then found by summation (3).

$$R'_{t+1} = \sum Q(s_t,\boldsymbol{a}_t), \qquad (3)$$

The loss was found using the Huber loss function of the difference between the expected reward and the actual reward of the next step (4). The Adam update strategy is used to update the weights of the network.

$$L=L_\delta(\sum Q(s_t,\boldsymbol{a}_t) - \gamma \sum Q'(s_{t+1},\boldsymbol{a}) - r_{t+1},\delta=1), \qquad (4)$$

where

$$Q'(s_{t+1},\boldsymbol{a})=Q(s_{t+1}, \underset{\boldsymbol{a}}{argmax}\, Q(s_{t+1},\boldsymbol{a};\theta);\bar{\theta}), \qquad (5)$$

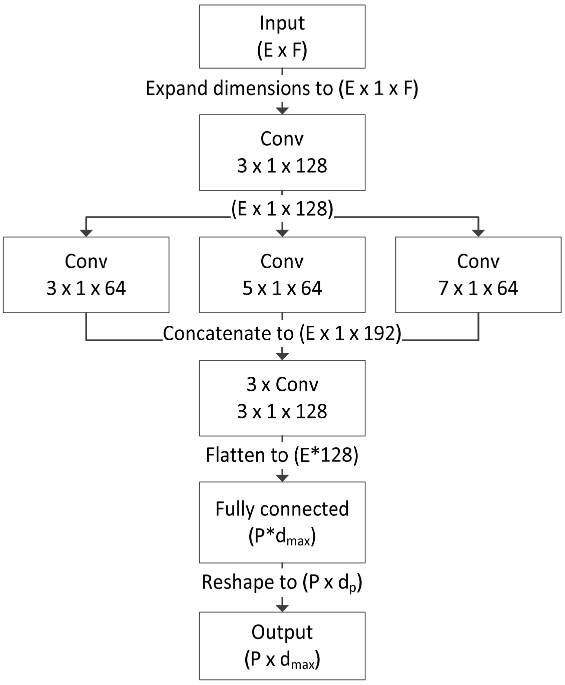


Fig. 2. The DNN structure, where *E* was the number of elements, *F* was the number of features per element, *P* was the number of decision points and d_{max} was the maximum value of d_p. Only the first d_p values at each decision point were considered after the output.

3. Experiments

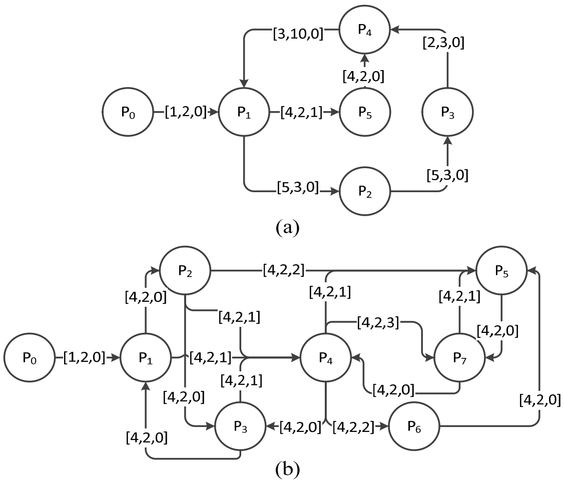
The performance of the Double DQN was compared to the performance of both the sSP and the dSP method on the two different environments seen in Fig. 3. In both environments, totes were sourced from P_0 and had destinations among all other places.

The two SP methods used the length of the transitions as cost. The only difference between sSP and dSP was that: when using dSP, if the number of totes in a FIFO queue reached its capacity, a large cost was added to the transition and the route was recalculated based on this. When the congestion had disappeared, the cost was removed, and the route recalculated again.

To test different setups, three Double DQN models were trained on each environment with a different number of totes to control. One was trained with 10 totes, one with 10-50 totes randomly chosen per episode, and one with 50 totes. They were trained until convergence in three steps with decreasing learning rate. Table 1 shows the hyperparameters for training. To test the performance, they were all tested for 100 episodes with 1 to 100 totes each, a total of 10,000 episodes.

4. Results

The results from testing on env. A shown in Fig. 4, revealed that, in general, the more totes that were introduced, the better the Double DQN performed compared to the sSP. Curiously, the model trained with 10-50 totes (RL 10-50) always performed worse than the one trained solely on 50 totes (RL 50), even when a low number of totes were used. The model trained on 10 totes (RL 10) got about the same performance as dSP.



(a)

(b)

Fig. 3. The two test environments used in the experiments. They are referred to as *env. A* (a) and *env. B* (b).

Table 1. Hyperparameters for training. The models were trained until convergence with each learning rate.

Batch size	32
Replay buffer size	100,000
Update target model	Every 5,000 steps
Exploration	0.5 to 0.02 in 1,000,000 steps
Learning rate	[1e-3, 1e-4, 1e-5]
γ	0.9
dst_{score}	5.0
$t_{penalty}$	0.01
$s_{penalty}$	1.0

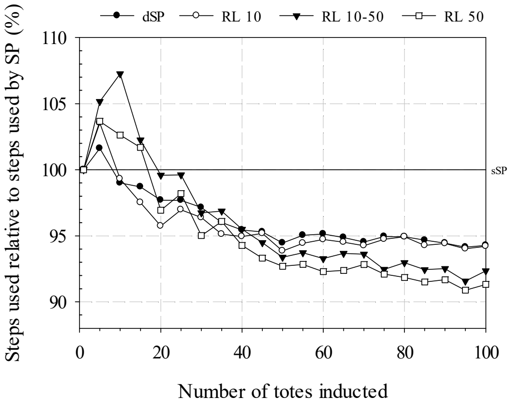


Fig. 4. The results from testing on env. A. The plot showed the relative difference from the average number of steps used to the number of steps used by sSP (100 %) when tested with different number of totes. Only every fifth measurement is shown for clarity.

The results from testing on env. B shown in Fig. 5, stated that, again, the general trend was that the more totes that were introduced, the better the Double DQN models performed compared to the sSP. RL 10 used more steps than dSP when exceeding 30 totes, while RL 10-50 and RL 50 used fewer steps than dSP and sSP from 1 to 100 totes and used a similar number of steps in general.

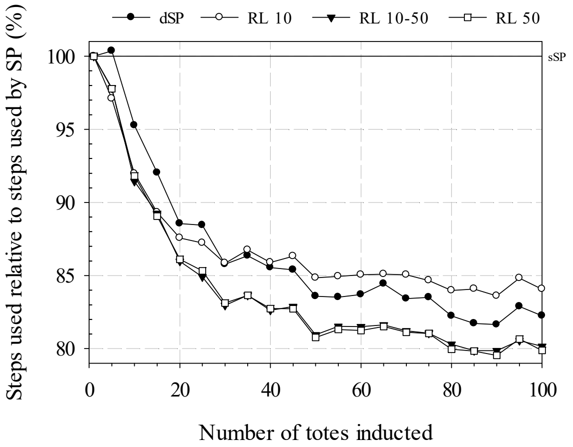


Fig. 5. The results from testing on env. B shown in the same setting as Fig. 4.

5. Discussion and Conclusions

Using Deep RL for routing seems to be a viable solution in the tested environment. Using only 10 totes to train the model was not sufficient to effectively handle a higher number of totes. There was, however, not much difference in the performance of a model trained with 10-50 or 50 totes. The gain from training with a higher number of totes could be that the model experienced a higher number of congestions and thereby learned routing policies to avoid those situations. Increasing the number of totes even further would result in a longer training time, and since the environments used were relatively small, a larger number of totes would probably not have changed a lot for the performance.

Comparing the performance on the two environments, it could be seen that there was a better opportunity for optimization in env. B. This was not surprising since the environment was larger and had more alternative routes than env. A. It did however show that Double DQN could find patterns to utilize optimization opportunities in simplified BHSs. This makes Deep RL for routing in BHSs a promising avenue for further research, which could include larger and more complex environments or handling of empty totes after delivery of baggage.

Acknowledgements

We thank BEUMER Group A/S for guidance and materials. This work is partly funded by the Innovation Fund Denmark (IFD) under File No. 8053-00040B.

References

[1]. E. W. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematik*, Vol. 1, Issue 1, 1959, pp. 269-271.

[2]. K. Hallenborg, Y. Demazeau, DECIDE: Applying multi-agent design and decision logic to a baggage handling system, in *Proceedings of the International Workshop Engineering Environment-Mediated Multi-Agent Systems (EEMMAS'07)*, 2008, pp. 148-165.

[3]. A. N. Tarau, B. De Schutter, J. Hellendoorn, Model-based control for route choice in automated baggage handling systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 40, Issue 3, 2010, pp. 341-351.

[4]. J. A. Boyan, M. L. Littman, Packet routing in dynamically changing networks: A reinforcement learning approach, in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS'94)*, Vol. 6, 1994, pp. 671-678.

[5]. L. Peshkin, V. Savova, Reinforcement learning for adaptive routing, in *Proceedings of the International Joint Conference on Neural Networks (IJCNN'02)*, Honolulu, Hawaii USA, 12-17 May 2002, Vol. 2, pp. 1825-1830.

[6]. V. Mnih, et. al., Human-level control through deep reinforcement learning, *Nature*, Vol 518, Issue 7540, 2015, pp. 529-533.

[7]. H. van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-learning, in *Proceedings of the Thirtieth Conference on Artificial Intelligence (AAAI'16)*, Phoenix, Arizona USA, 12-17 February 2016, pp. 2094-2100.

[8]. F. Pommereau, SNAKES: A flexible high-level Petri nets library, in *Proceedings of the Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS'15)*, 21-26 June 2015, pp. 254-265.

(13)

A Neural Model for the Prediction of Pathogenic Genomic Variants in Mendelian Diseases

Alessio Cuzzocrea [1], Luca Cappelletti [1] and Giorgio Valentini [1]
[1] AnacletoLab, Dipartimento di Informatica, Università degli Studi di Milano, Italy
E-mail: valentini@di.unimi.it

Summary: The detection of pathogenic genomic variants associated with genetic or cancer diseases represents an open problem in the context of the Genomic Medicine. In particular the detection of mutations in the non-coding regions of human genome represents a particularly challenging machine learning problem, since the number of neutral variants largely outnumber the pathogenic ones, thus resulting in highly imbalanced classification problems. We applied neural networks to the detection of pathogenic regulatory genomic variants in Mendelian diseases and we showed that leveraging imbalance-aware techniques and deep learning algorithms, we can obtain state-of-the-art results, using a less complex model than those proposed in literature for this challenging prediction task.

Keywords: Neural networks, Imbalance-aware neural networks, Deep learning, Prediction of pathogenic genomic variants, Mendelian diseases.

1. Introduction

An open problem in the context of Precision Medicine is the detection of the pathogenic variants associated with genetic Mendelian diseases. Indeed for most of the about 8000 different Mendelian diseases no known causative gene is known and hence no therapy is available for affected patients [14]. Recently several studies showed that most of the pathogenic variants associated with Mendelian disorders lie in the non-coding regulatory regions of the human genome [2].

For this reason several computational methods have been proposed to disentangle the regulatory mechanisms underlying Mendelian diseases and other disorders ranging from complex genetic diseases to cancer, using mainly supervised Machine Learning-based techniques to predict the pathogenicity of genomic variants in regulatory regions of the human genome [7, 15, 5].

Unfortunately only a very small amount of positive (pathogenic) variants are available to train learning machines and in this very imbalanced context, where neutral variants (negative examples) largely outnumber positive ones, machine learning methods are severely biased toward the majority class and are not able to detect pathogenic variants with a sufficient reliability. Very recently novel imbalance-aware machine learning methods have been proposed in this context, showing that applying together ensemble and sampling techniques we can significantly improve prediction results [8, 10].

Motivated by these results and by the very recent successful application of deep neural learning methods to Genomic Medicine [13], in this work we investigate whether a neural model, by adopting imbalance-aware techniques and deep learning techniques can obtain state-of-the-art results in this challenging prediction task.

In the next sections we propose two imbalance-aware neural models able to deal with highly imbalanced genomic data, and we experimentally show that they largely outperform “vanilla” neural models, achieving state-of-the-art results in the prediction of pathogenic regulatory variants in Mendelian diseases.

2. Methods

We introduce two imbalance-aware neural methods, able to deal with highly imbalanced genomic data. The first one *MiMiS-Net* (Mini-batch Minority class Sized Neural Networks) simply enlarges the mini-batch size applied during the training of the neural network. The second one *MiBa-Net* (Mini-batch Balanced Neural Networks), inspired by [9], uses sampling techniques to balance positive and negative examples in the mini-batch.

2.1. Mini-batch Minority Class Sized Neural Networks (*MiMiS-Net*)

The main idea behind this approach consists in improving the likelihood that at least one positive example will be included in each mini-batch during the training phase. We show that this can be accomplished by simply appropriately enlarging the size of the mini-batch itself. Indeed when the data are highly imbalanced, the update of the weights is likely performed with most of the mini-batches including only examples of the majority (negative) class: in this situation the neural network tends to be biased toward the negative class, since it learns only from negative examples, and hence cannot recognize positive examples.

More precisely, let N be the overall number of available examples of the training set T, n the size of the mini-batch, and p the probability that a positive example will be randomly extracted from the overall training set. If N_p is the total number of positive examples in the training set, we can estimate $p \simeq \frac{N_p}{N}$.

Let X_n be a random variable that counts how many positives have been randomly drawn from T into a mini-batch of size n. Then X_n is distributed according to a binomial distribution $B(p,n,k)$ where k is the number of successes (positive examples) across n Bernoulli experiments each one with probability of success p. Then the probability $P(X_n \geq 1)$ that we have at least on positive example in a mini-batch of size n is:

$$P(X_n \geq 1) = \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} \tag{1}$$

We can observe that

$$P(X_n \geq 1) = 1 - P(X_n = 0) = \\ = 1 - \binom{n}{0} p^0 (1-p)^n = 1-(1-p)^n \tag{2}$$

Hence Eq. (1) can be rewritten as:

$$P(X_n \geq 1) = 1-(1-p)^n \tag{3}$$

If we would like to estimate the size n of the mini-batch needed for having at least one positive in the mini-batch itself with probability $P(X_n \geq 1)$, we can apply a $\log$ transform to Eq. (3):

$$n = \frac{\log(1-P(X_n \geq 1))}{\log(1-p)} \tag{4}$$

Eq. (4) shows the mini-batch size n needed for having with probability $P(X_n \geq 1)$ at least one positive example in each mini-batch. It is easy to see that n is large for large values of $P(X_n \geq 1)$ and for small values of p, i.e. when we would like to be confident that at least one example is included in the mini-batch and when the data in the training set are imbalanced (Fig. 1). For a reasonable probability (say $P = 0.8$) of having at least one positive example in the mini-batch, when data are imbalanced (say $p = \frac{1}{5000}$) we need a mini-batch size of at least $n = 8046$, a size significantly larger than those usually applied for mini-batch learning.

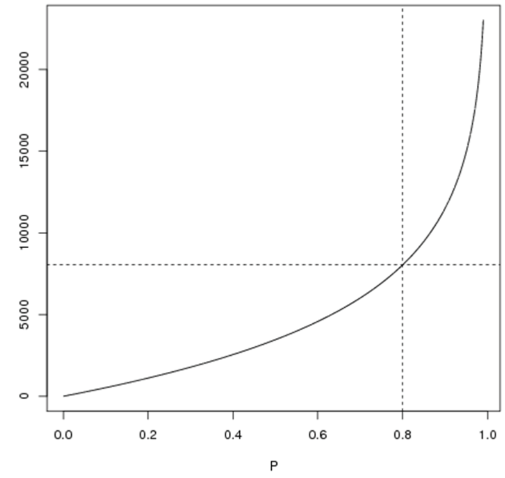


Fig. 1. Plot of the size n of the mini-batch (vertical axis) for drawing with probability P (horizontal axis) at least one positive example included in it when the frequency of the positives in the training set is about $p = \frac{1}{5000}$.

2.2. Mini-batch Balanced Neural Networks (*MiBa-Net*)

Sampling procedures to deal with the imbalance of the data have just been proposed in machine learning and neural network literature [4] and have been proven successful in the context of the analysis of genomic data with ensemble methods [8, 9]. Here we propose to balance the mini-batch during the training of the neural network, in order to provide a number of positive examples (the minority examples) comparable with those of the majority (negative class). In this way at each mini-batch the weights of the network are updated taking into account in a balanced way both positive and negative examples.

The mini-batch generator samples with replacement, according to a uniform distribution, the positive examples by drawing a sample ratio $r_p \in (0, r_p^{max}]$ of the available positive examples: if $r_p < 1$ we subsample the positives, if $r_p = 1$ we have a bootstrap sample, for $r_p > 1$ we perform oversampling. Negative examples are sub-sampled without replacement according to the ratio $r_n \in (0, r_n^{max}]$ between the negatives and the positives in the mini-batch: if $r_n < 1$ we will have less negatives than positives in the mini-batch, if $r_n = 1$ positives and negatives are equally sized, and for $r_n > 1$ negatives outnumber positives in the mini-batch. As an example, if we have $N_n = 10^6$ negative examples and $N_p = 10^2$ positive examples, we have an imbalance $N_p / N_n = 1/10^4$. If we set $r_p = 1$ and $r_n = 1$ we can obtain a perfectly balanced mini-batch with 100 positives and 100 negatives. An epoch, with this generator, is considered to be finished when all the negative samples are used. Notice that the positive samples may appear repeatedly among different batches in the same epoch, while each negative will appear only once in one specific mini-batch at each epoch.

3. Results

We evaluated the proposed methods *MiMiS-Net* and *MiBa-Net* on Mendelian data, by comparing them with a baseline "vanilla" Neural Network and with *hyperSMURF* [9], an imbalance-aware hyper-ensemble method that significantly outperformed other state-of-the-art methods such as *CADD* [7], *DeepSEA* [15], *Eigen* [5] and *GWAVA* [8] on this specific task [9].

3.1. Experimental Set-up

For the experiments we used the data set of Mendelian Single Nucleotide Variants (SNV) in non-coding regions of the human genome originally collected in [11]. From this data set we used all the available manually curated 406 positive examples, and from the available 14 millions of neutral variants (negative examples) we randomly drew one million of examples, thus resulting in an imbalance $p \simeq \frac{1}{2500}$. To each SNV example are associated 26 features representing different characteristics of the genomic variants, ranging from G/C content, population-based features, to conservation scores and transcription and regulation annotations (see [11] for more details).

We trained the neural networks on all the genomic variants except those belonging to chromosome 19 (19018 examples) that have been left out for evaluating the generalization performance. In other words we performed a "chromosome aware" hold-out procedure and we did not use the examples of the test set (chromosome 19) to train the model. The main hyper-parameters of the model, i.e. different number of hidden layers (ranging from 1 to 4), the number of hidden neuron per layer (ranging from 2 to 100) have been selected by 5-fold cross-validation on the training set. We used the ReLU activation function for the hidden layers and a sigmoid for the output layer. We chose as loss function to be optimized the hinge loss with the logit function applied to the sigmoid output, and we applied both the Stochastic Gradient Descent (*SGD*) with fixed learning rate (0.01) and the *Adam* method [6] as optimization algorithms. The weight matrix of each layer have been initialized using the Glorot normal initializer [3]. Before training each feature has been standardized by subtracting its mean and dividing by its standard deviation across examples.

For evaluating the performance of the different methods we used the Area Under the Precision recall Curve (AUPRC), since it is well-known that in very imbalanced learning problems this metric is more informative than the Area Under the Receiving Operating Characteristic curve (AUROC) [1]. All the experiments and the new neural models have been implemented by deriving new Python classes from the Keras library [16] using Tensorflow as backend.

3.2. *MiMiS-Net* Results

At first we trained and test the state-of-the-art method *hyperSMURF* on the Mendelian data set, obtaining an AUPRC = 0.911 and an $\text{AUROC} = 0.999$. The best "vanilla" neural model, i.e. a neural network that does not adopt any imbalance-aware learning strategy, achieved an $\text{AUPRC} = 0.078$ and an $\text{AUROC} = 0.968$. This is not so surprising since a previous work clearly showed that imbalance-unaware strategies are not able to obtain good results on this challenging learning task [9].

The proposed *MiMiS-Net* imbalance-aware method, by setting the batch size $n = 5000$, corresponding to a probability $P(X_n \geq 1) \simeq 0.85$ of drawing at least one positive example in the mini-batch in the training set (eq. 4) led to significantly better results than the vanilla Neural Network (Fig. 2). On the test set we obtained an AUPRC = 0.794 and an AUROC = 0.973, significantly lower than that obtained by *hyperSMURF* but an order of magnitude larger than that obtained by the vanilla neural model. Fig. 2 shows that *Adam* optimization achieves significantly better results than *SGD* and as expected feature standardization is necessary to improve performances. Nevertheless, looking at Fig. 3(a), we can observe a certain overfitting of *MiMiS-Net* and for this reason we applied dropout techniques [12] to try to avoid this effect. Results show that *MiMiS-Net* with dropout reduces overfitting (Fig. 3(b)) and achieves significantly better results on the test set ($\text{AUPRC} = 0.879$).

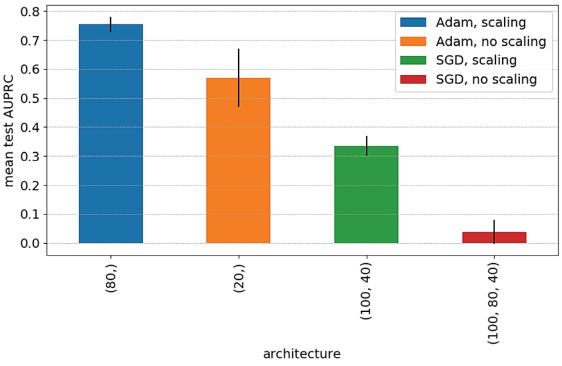
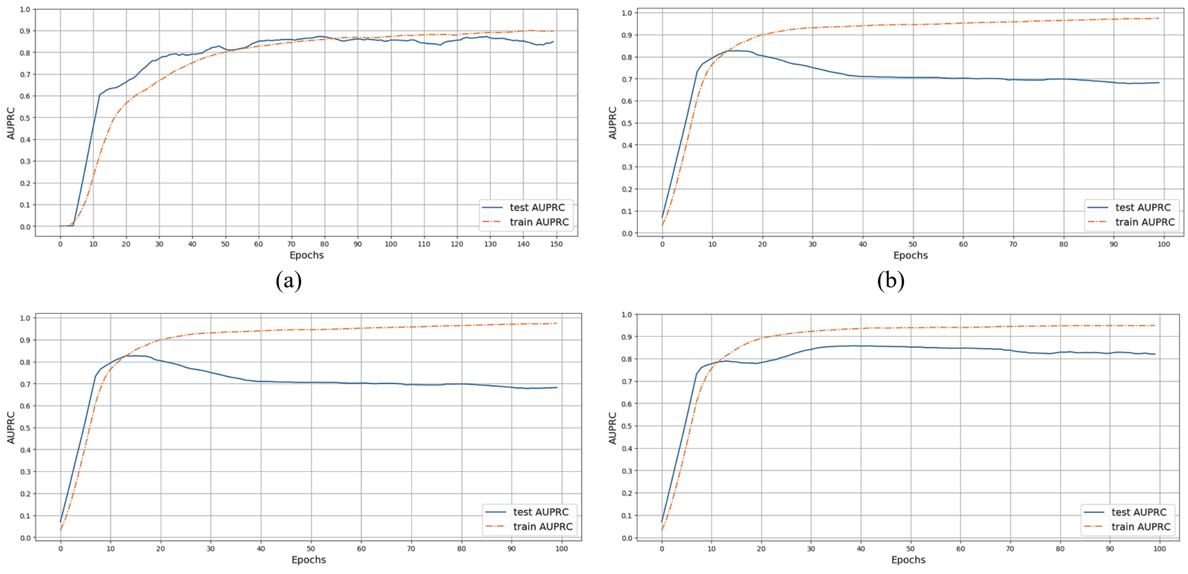
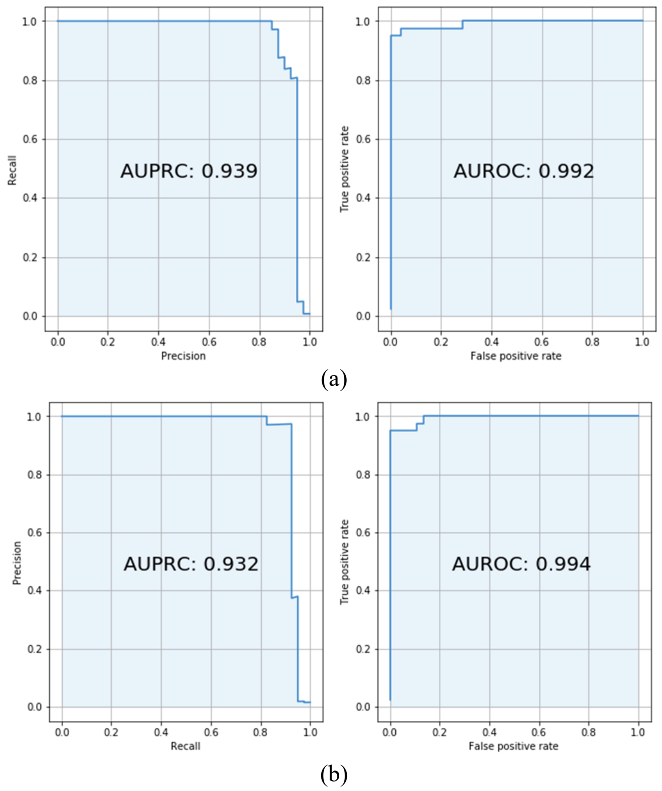


Fig. 2. *MiMiS-Net* cross-validation results on the training set, using Adam and SGD optimization algorithms with and without feature normalization. In abscissa the number of hidden neurons for each layer of the selected best models is reported. The vertical lines represent the standard deviation across folds.



(a) (b) (c) (d)

Fig. 3. MiMiS-Net and MiBa-Net training and test AUPRC across epochs. Horizontal axis: epochs; vertical axis: AUPRC. Orange and blue lines represent respectively test and train AUPRC results. (a) MiMiS-Net (b) MiMiS-Net with dropout (c) MiBa-Net with dropout (d) MiBa-Net with dropout and Max norm regularization.



(a)

(b)

Fig. 4. Precision Recall and ROC curves on the test set obtained with the best *MiMiS-Net* and *MiBa-Net* models using feature decorrelation, dropout and regularization techniques. (a) *MiMiS-Net* (b) *MiBa-Net*.

3.3. *MiBa-Net* Results

Results with *MiBa-Net* show that also this neural imbalance-aware technique can boost pathogenic Mendelian variants detection. Indeed *MiBa-Net* with dropout obtains on the test set an AUPRC = 0.674, but with a serious overfitting towards the training set (Fig. 3(c)). Recalling that regularization through maximization of the norm has been shown to work nicely when paired with dropout [12], we applied jointly dropout and Maxnorm regularization techniques, thus reducing overfitting (Fig. 3(d)) and achieving a test set AUPRC = 0.835.

Even if we achieved results close to that obtained by the state-of-the-art method *hyperSMURF*, we tried to further improve performances by analyzing the correlation between the 26 features associated with the genomic variants. By systematically applying the Pearson correlation between each pair of features we individuated sets of highly correlated features, and removed accordingly 5 of them and then we retrained both *MiMiS-Net* and *MiBa-Net* with the reduced set of 21 features using dropout and regularization. Results show a further significant enhancement of the performances (Fig. 4), with AUPRC values even better than those achieved by the state-of-the-art *hyperSMURF* method.

4. Conclusion

Several machine learning methods have been recently proposed in literature for the detection of pathogenic genomic variants, associated with several diseases ranging from genetic disorders to cancer. We showed that in the case of the detection of rare SNV mutations in non-coding genome causative of Mendelian diseases, imbalance-aware neural models based on mini-batch sampling techniques (*MiBa-Net*) and on the enlargement of the mini-batch (*MiMiS-Net*), we can significantly improve results obtained with imbalance-unaware "vanilla" neural models. In particular by using deep learning techniques together with imbalance-aware methods we can achieve results at least comparable with state-of-the-art results. Finally we observe that in the context of Mendelian diseases the best results have been obtained with relatively simple neural models with one or two hidden layers and some tens on hidden neurons, while state-of-the-art models used ensembles or hyper-ensemble of learning machines, characterized by a significantly larger complexity and training time.

References

[1]. J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in *Proceedings of the 23rd International Conference on Machine Learning (ICML'06)*, New York, NY, USA, 2006, pp. 233-240.

[2]. S. L. Edwards, J. Beesley, J. D. French, A. M. Dunning, Beyond gwas: Illuminating the dark road from association to function, *American Journal of Human Genetics*, Vol. 93, 2013, pp. 779-797.

[3]. X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS'11)*, April 2011, pp. 315-323.

[4]. H. He, E. Garcia, Learning from imbalanced data, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, Issue 9, 2009, pp. 1263-1284.

[5]. I. Ionita-Laza, et al., A spectral approach integrating functional genomic annotations for coding and noncoding variants, *Nature Genetics*, Vol. 48, Issue 2, 2016, pp. 214-220.

[6]. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15)*, 2015.

[7]. M. Kircher, et al., A general framework for estimating the relative pathogenicity of human genetic variants, *Nature Genetics*, Vol. 46, Issue 3, Mar. 2014, pp. 310-315.

[8]. G. Ritchie, I. Dunham, E. Zeggini, P. Flicek. Functional annotation of noncoding sequence variants, *Nature Methods*, Vol. 11, Issue 3, Mar. 2014, pp. 294-296.

[9]. M. Schubach, M. Re, P. N. Robinson, G. Valentini, Imbalance-aware machine learning for predicting rare and common disease-associated non-coding variants, *Scientific Reports*, Vol. 7, 2017, 2959.

[10]. M. Schubach, M. Re, P. N. Robinson, G. Valentini, Variant relevance prediction in extremely imbalanced training sets, in *Proceedings of the 25th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB) and 16th European Conference on Computational Biology (ECCB),* 2017.

[11]. D. Smedley, et al., A whole-genome analysis framework for effective identification of pathogenic regulatory variants in Mendelian disease, *American Journal of Human Genetics*, Vol. 99, Issue 3, 2016, pp. 595-606.

[12]. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, Vol. 15, 2014, pp. 1929-1958.

[13]. A. Telenti, C. Lippert, P. Chang, M. DePristo, Deep learning of genomic variation and regulatory network data, *Human Molecular Genetics*, Vol. 27, Issue R1, 2018, pp. R63-R71.

[14]. Y. Yang, et al., Clinical whole-exome sequencing for the diagnosis of mendelian disorders, *New England Journal of Medicine*, Vol. 369, 2013, pp. 1502-1511.

[15]. J. Zhou, O. G. Troyanskaya, Predicting effects of noncoding variants with deep learning-based sequence model, *Nature Methods*, Vol. 12, Issue 10, August 2015, pp. 931-934.

[16]. Keras: The Python Deep Learning Library, https://keras.io/

(15)

Phantomisation in State-based HTN Planning

Ilche Georgievski [1] and Marco Aiello [1]
[1] University of Stuttgart, Institute of Architecture of Application Systems, Service Computing Department,
Universitaetsstrasse 38, 70569 Stuttgart, Germany
E-mail: firstname.lastname@iaas.uni-stuttgart.de

Summary: Hierarchical Task Network (HTN) planning is a particularly useful planning technique due to its rich domain knowledge. This knowledge needs to be conceived by a domain author in such a way that planners can find (effective) plans. This is especially true for HTN planners that search for plans in the space of states. In fact, these planners require various ‘tricks’ in the knowledge, making the domains elaborate and large. One trick is for the handling of phantomisation – knowledge on how to recognise that some task is already accomplished by another task. Such conceived knowledge purely depends on the domain author’s ability and experiences to identify and encode phantomisation. We propose an approach for automatic phantomisation during planning that does not need domain tricks. With this approach, one can solve more planning problems than using planners without it. Other potential benefits may include less encoding effort and simpler and smaller domains.

Keywords: AI planning, Hierarchical task networks, Knowledge representation, Phantomisation.

1. Introduction

Hierarchical Task Network (HTN) planning is a planning technique well suited for domains in which some hierarchical representation is desirable or known in advance, domains that involve composite constructs (e.g., business process models) or structured strategies (e.g., cloud provisioning processes), and domains that are intrinsically epistemic [1]. HTN planning requires an initial state, a task network as an objective to be accomplished, and domain knowledge consisting of networks of primitive and compound tasks. A *task network* represents a hierarchy of tasks each of which can be executed, if the task is *primitive*, or decomposed via *methods* into subtasks, if the task is *compound*. Planning starts by decomposing the initial task network and continues until all compound tasks are decomposed. The solution is a *plan* which equates to a set of primitive tasks applicable to the initial state.

Considering the kind of space the search for solutions is performed in, one can distinguish between plan-based HTN planning and state-based HTN planning. Domains provided to planners of the latter kind contain well-conceived knowledge. For example, state-based HTN planners require a larger and more elaborated domain model for solving block-worlds planning problems than their counterpars do [1]. One reason for this is the need for handling phantomisation explicitly in the encodings of compound tasks.

Phantomisation is a stage after the process of recognising that the purpose of some task is already accomplished by other task(s) at some place in a task network. It involves a substitution of an element of a plan with a “phantom” element to indicate that the step is a no-op or not needed in the given situation [2]. The phantomisation is mainly used to avoid redundant, unnecessary steps in plans. In some cases, the phantomisation can be key to satisfactory performance of HTN planners [3]. Thus, the advantage of using phantomisation is the planner’s ability to take into account which and when tasks are really necessary, and therefore, to produce more efficient plans. While in most plan-based HTN planners the phantomisation process is accomplished automatically during the planning process, in state-based HTN planners phantomisation information must be provided in the domain knowledge. The weak points of this type of phantomisation are its identification and encoding, and along with that, an increased domain size.

We address the possibility of diminishing the strenuous and tedious process of writing effective domain knowledge by introducing enhanced reasoning in state-based HTN planners so as to recognise and handle phantomisation automatically and without explicit domain encodings.

The paper is organised as follows. Section 2 reviews related work. Section 3 gives a preliminary on state-based HTN planning. Section 4 introduces our proposed formalism for phantomisation in state-based HTN planning. Section 5 provides an overview of the algorithm behind the planning process. Section 6 gives an example that demonstrates our approach and some of its advantages. Section 7 provides concluding remarks.

2. Related Work

A number of studies deal with the problem of reducing unnecessary plan steps. Several hierarchical planners use domain expressivity to reason about such redundancy and no planner has incorporated that ability into the planner itself [6, 2, 7].

Tate [2] has introduced the concept of phantom task, which says that such a task can be accomplished by doing nothing if the task is placed in the task network at a point where its effect is still holding. Task reduction schemas are given to a planner a priori as part of the domain knowledge. Such phantomisation is also used in the framework for plan modification and reuse [8].

Charniak and McDermott [9] proposed the idea to merge tasks with similar actions. To merge two tasks means that their actions must "match" in the sense that they would differ only in the slots where one or both have anonymous constants. This idea has neither a formalism nor an implementation.

Foulser et al. [10] proposed a formalism and heuristics-based algorithms for finding minimum-cost merged plans. In their formalism, a set of operators is mergeable with a (merged) operator if and only if the operator can achieve all the "useful" effects of the operators in the set, the operator's preconditions are subsumed in the preconditions of the set, and the cost of operator is less than the cost of the set of operators. Although our idea essentially seems similar to theirs, we use a less constrained task-to-task interaction. Also, we consider two types of tasks, primitive and compound tasks.

In a multi-agent environment, Cox and Durfee [11] proposed an algorithm that uses a merging approach to help agents remove redundant plan steps while at the same time preserve their autonomy. Basically, their approach removes redundant steps from plans instead of not adding them at all.

3. State-Based HTN Planning

HTN planning is popular and well suitable for various domains due to its rich domain knowledge [1]. The domain consists of tasks that can be accomplished by operators or methods. An operator represents a transition from one state to another, while a method predefines how to decompose some task into greater details. Given an HTN planning problem, which consists of an initial state, an initial task network and sets of operators and methods, planning is performed by repeatedly decomposing tasks from the initial task network until operators that are executable in the current state are reached. Formal definitions follow.

A state is a set of ground predicates in which the closed-world assumption is adopted. A *predicate* consists of a predicate symbol, and a list of terms $\tau_1, \cdots, \tau_k$. A *term* is either a constant, or a variable. Each predicate can be true or false, and a predicate is *ground* if its terms contain no variables.

Characteristic for HTN planning are the notions of primitive and compound tasks. A *primitive task* is an expression of the form $t_p(\tau)$, where t_p is a primitive–task symbol, and $\tau = \tau_1, \cdots, \tau_k$ are terms. A compound task is defined similarly. The set of primitive and compound tasks is a finite set of tasks T.

Each primitive task is represented by a single operator defined similarly to a STRIPS operator [5].

An *operator* o is a triple $\langle t_p(o), pre(o), eff\ (o)\rangle$, where $t_p(o)$ is a primitive task, $pre(o)$ and $eff(o)$ are preconditions and effects, respectively.

A transition from a state to another one is accomplished by an instance of an operator whose preconditions are a logical consequence of the current state. That is, an operator o is *applicable* in state s, iff $pre(o) \subseteq s$. The application of o to s results into a new state $s[o] = s \cup eff(o) = s'$.

Each compound task is associated with one or more methods as ways of its accomplishment. A *method* m is a tuple $\langle t_c(m), pre(m), tn(m)\rangle$, where $t_c(m)$ is a compound task, $pre(m)$ is a precondition, and $tn(m)$ is a task network. A method m is *applicable* in a state s iff $pre(m) \subseteq s$. Given a task $t_c(m)$ and a method m, applying m to s results into a task network $tn(m) = (s[m], t_c)$. A task network tn is a pair $\langle T', \prec\rangle$, where T'⊆ T, and $\prec$ defines the ordering in T'.

Definition 1 (HTN planning problem). An HTN planning problem P is a tuple $\langle s_0, tn_0, O, T\rangle$, where s_0 is an initial state, and tn_0 is an initial task network, O and T are sets of operators and tasks, respectively.

Definition 2 (Solution). Given an HTN planning problem P, a sequence of operators $o_1, \cdots, o_n$ or a plan is a solution to P, if and only if there exists a task $t \in T_0$, where $tn_0 = \langle T_0, \prec_0\rangle$, such that $(t, t') \in \prec_0$ for all $t' \in T_0$ and

- t (or o_1) is primitive and applicable in s_0 such that $o_2, \cdots, o_n$ is a solution to $P = \langle s_0[o_1], tn_0 \backslash \{o_1\}, O, T\rangle$; or
- t is compound and there exists an applicable method m such that $tn(m) = (s0[m], t)$, $tn' = tn_0 \backslash \{t\} \cup tn(m)$, and $o_1, \cdots, o_n$ is a solution to $P = \langle s_0, tn', O, T\rangle$.

4. Phantomisation

Our approach to automatic phanotmisation includes a notion of agenda. Intuitively, given an HTN planning problem, the agenda contains all atoms that are valid up to the *i*-th state. An atom is a predicate. An atom may not be in the *i*-th state, but it may still be valid. That is, an atom is *phantom* in the *i*-th state if and only if its value evaluates to true in some *(i-k)*-th state and whose truthfulness holds between the *(i-k)*-th and *i*-th states but not in the *i*-th state.

Consider a modern home with rooms r_1, r_2, and r_3, a domestic robot Tars, which is in room r_1, and a cup c also in r_1. Let r_1 be adjacent to r_3 and r_3 to r_2. The objective is to move Tars from r_1 to r_2 and transfer c from r_1 to r_3. If we move Tars from r_1 to r_3, the fact *(at Tars* r_1*)* is no longer true in the state, but it is a valid fact for the task of transferring the cup from r_1 to r_3. Recording this type of fact validity enables the task to be further decomposed by unloading the cup at r_3 and moving Tars to r_2.

Definition (Agenda). Let P be an HTN planning problem and s_i the current state. An agenda is a set $A \subseteq s_0 \cup \cdots \cup s_i$ such that all logical atoms in A are phantoms in s_i.

We can now check when some operator is already accomplished during the planning process. We consider such an operator to be *matchable* to an already applied operator if and only if both represent the same primitive task.

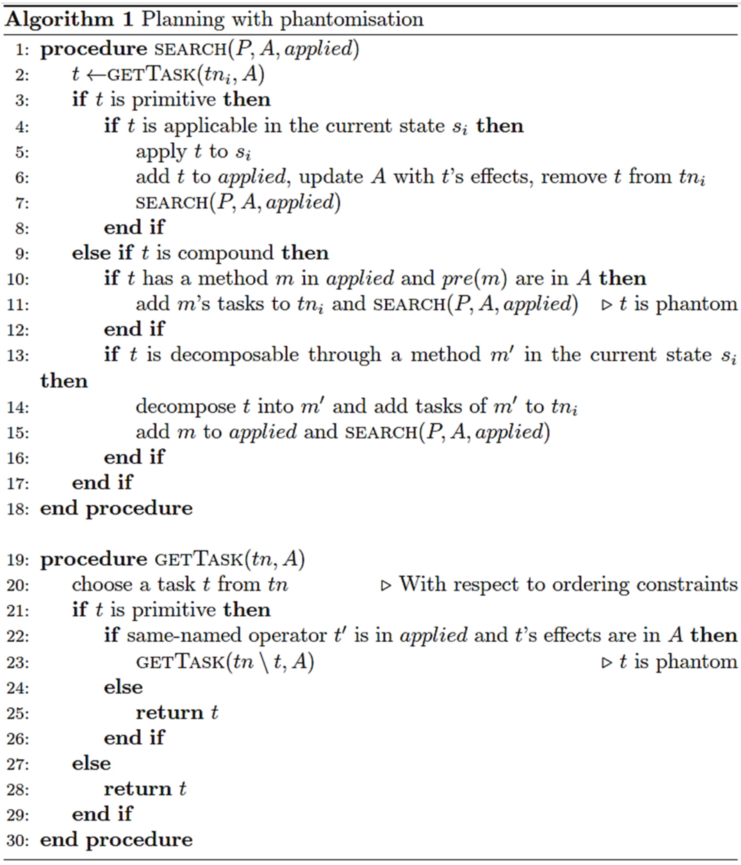
Definition (Phantom primitive task). Let P be an HTN planning problem and A an agenda. A primitive task $t_p(\tau)$ is phantom if and only if there exist another primitive task $t'_p(\tau')$ in the current plan π such that $t_p = t'_p, \tau = \tau'$, and $eff(\tau') \in A$.

In addition, we need to check the methods of a compound task for their applicability. Without phantomisation, the methods' precondition may not be applicable as certain variable bindings do not exist explicitly in the current state. In our case, relevant methods are those that have been already successfully instantiated (for example, their primitive tasks, if any, are part of the potential plan).

Definition (Phantom compound task). Let P be an HTN planning problem and A an agenda. Let t_c be the current compound task, m its method, and s_i the current state. Task t_c is phantom if t_c is decomposable by m and m is applied to state s_k such that $k < i$ and $pre(m) \in A$.

5. Algorithm

Algorithm 1 shows the algorithm we developed and takes as input an HTN planning problem $P = \langle s_0, tn_0, O, T\rangle$, an agenda A initialised to s_0, and an empty set *applied* that will contain applied tasks. The main procedure starts with an interleaving step represented by the GetTask procedure. In this procedure, each primitive task, which is already applied (i.e., part of the potential plan) and its effects are elements of the agenda, is pruned.



Algorithm 1 Planning with phantomisation

```
 1: procedure SEARCH($P, A, applied$)
 2:     $t \leftarrow$GETTASK($tn_i, A$)
 3:     if $t$ is primitive then
 4:         if $t$ is applicable in the current state $s_i$ then
 5:             apply $t$ to $s_i$
 6:             add $t$ to $applied$, update $A$ with $t$'s effects, remove $t$ from $tn_i$
 7:             SEARCH($P, A, applied$)
 8:         end if
 9:     else if $t$ is compound then
10:         if $t$ has a method $m$ in $applied$ and $pre(m)$ are in $A$ then
11:             add $m$'s tasks to $tn_i$ and SEARCH($P, A, applied$)   ▷ $t$ is phantom
12:         end if
13:         if $t$ is decomposable through a method $m'$ in the current state $s_i$
    then
14:             decompose $t$ into $m'$ and add tasks of $m'$ to $tn_i$
15:             add $m$ to $applied$ and SEARCH($P, A, applied$)
16:         end if
17:     end if
18: end procedure

19: procedure GETTASK($tn, A$)
20:     choose a task $t$ from $tn$            ▷ With respect to ordering constraints
21:     if $t$ is primitive then
22:         if same-named operator $t'$ is in $applied$ and $t$'s effects are in $A$ then
23:             GETTASK($tn \setminus t, A$)                        ▷ $t$ is phantom
24:         else
25:             return $t$
26:         end if
27:     else
28:         return $t$
29:     end if
30: end procedure
```

Planning continues depending on whether the chosen task is primitive or compound. If it is primitive and applicable in the current state s_i, its effects are added to the agenda. When the chosen task is compound, a typical state-based HTN planner skips task's methods for which bindings do not exist. However, with phantomisation, we may also consider some of those methods those already applied. Therefore, in lines 10-12, we add logic to the algorithm that handles such methods and calls the procedure recursively with methods' tasks added to the current task network.

Previously identified phantomisation yields additional interleaving steps between the tasks in the task network. But, since the task has already been accomplished, many of the interleaving steps are not necessary – they produce redundant searching. For instance, if a planner finds a plan at some point after successful phantomisation and backtracks to try other combinations without controlling the search, it will find plans which are equivalent to the first found one. Hence, we add a control ability that prunes these steps. The algorithm continues to search for other possible methods of the compound task for which appropriate bindings exist in the current state. Thus, when a certain task is decomposable in the current state, its applicable method is added to the set of applied tasks.

6. Example

We provide an example of phantomisation from a scenario of planning for ubiquitous computing [4]. The scenario is articulated around the adaptation of an office equipped with an air conditioning system, lamp, radio, and computer. We encapsulate this information in two compound tasks as shown in Fig. 1. One task is for adapting the office to certain ambience (i.e., *adjust-office*), and the other one is for adjusting the working desk (*adjust-desk*). The *adjust-office* task can be further decomposed into a network of three tasks, namely *set-ac* for setting the air conditioning system to a temperature level *t*, *turn-on-lamp* for switching on the lamp *l*, and *turn-on-music* for controlling the radio *m*. The *adjust-desk* task contains a decomposition of two tasks, namely *turn-on-lamp* and *start-computer* for invoking the computer *c*. *set-ac*, *turn-on-lamp*, *turn-on-music*, and *start-computer* are all primitive tasks.

We may also have some constraints on the order of tasks in the task networks of the compound tasks, for example, that *set-AC* must occur before *turn-on-lamp* and the latter one must occur before *turn-on-music*. Thus, the tasks of *adjust-office* should be accomplished in the sequence given in Fig. 1.

Say that we want to perform both tasks, i.e., the initial task network consists of preparing the working desk in the office *r* and adjusting *r* after some time being empty. We assume that one of the most effective solutions would be when the air conditions are comfortable, the lamp is turned on, the music is playing, and the computer is started and ready for work. That is, the plan π equlas to *set-AC, turn-on-lamp, turn-on-music, start-computer* in that order.

Let us examine the situation when our algorithm is on the right way of finding such a good solution. For the purpose of demonstration, we consider a straightforward application of the first two operators of the *adjust-office* task. At this point, their effects are added to the agenda. The planning process continues by interleaving the task *adjust-desk(r,l,c)* and decomposing it to its task network. Task's first subtask is *turn-on-lamp(l,r)* which is already a part of the potential plan. The algorithm reasons that this task is already achieved and that its effect (the lamp is on) is still valid. Therefore, the algorithm is allowed to prune the task from interleaving and continues by processing the rest of available tasks. In few steps, the algorithm finds the correct sequence of operators.

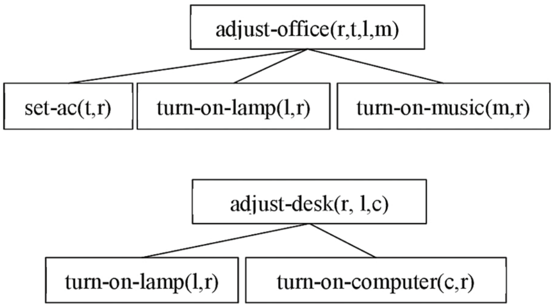
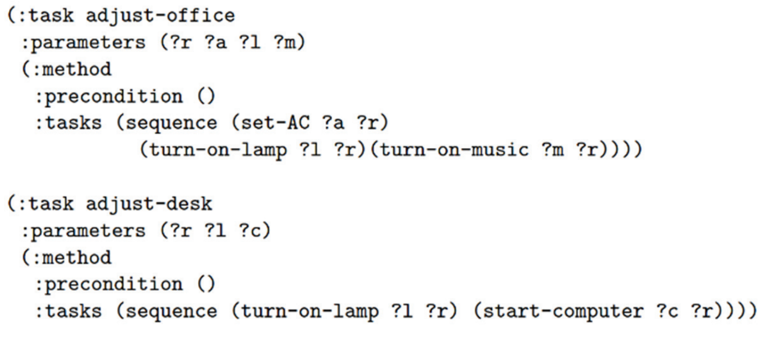


Fig. 1. Examples of two compound tasks.

In contrast to our approach, a typical state-based HTN planner will not find a solution given the domain in Fig. 2. For such planner to come up with a plan, more effective domain descriptions are required. We enclose such enhanced encodings in Fig. 3. Thus, we need to include an additional task *light-helper* that has a method representing phantomisation – doing nothing when the lamp is already turned on. Comparing Fig. 2 and Fig. 3, one may notice that the first encoding is a simpler and more compact domain representation.

7. Concluding Remarks

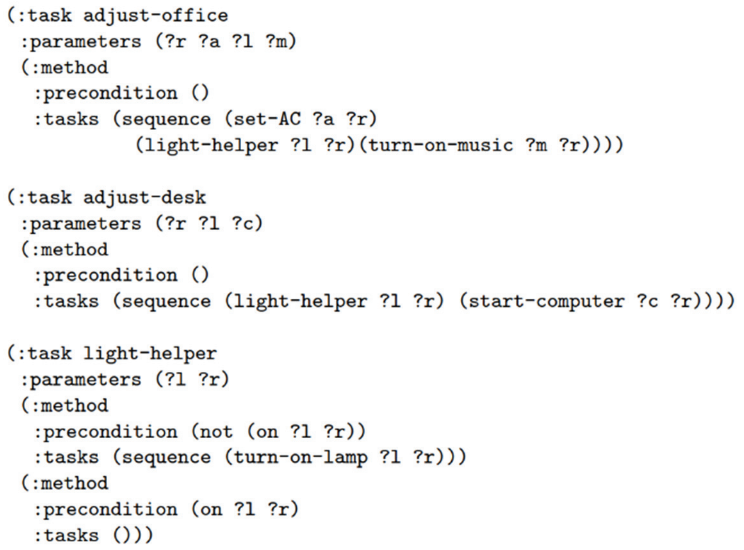
We enhanced state-based HTN planning with a feature for automatic phantomisation. We showed how can this be implemented and demonstrated it on an example. We postulate that the main benefit of this approach is the simplification and reduction in size of the domain knowledge. It is possible to improve some plans even more in a sense of reducing the number of steps by extending this task-to-task matching into a task-to-tasks matching.



```
(:task adjust-office
 :parameters (?r ?a ?l ?m)
 (:method
  :precondition ()
  :tasks (sequence (set-AC ?a ?r)
         (turn-on-lamp ?l ?r)(turn-on-music ?m ?r))))

(:task adjust-desk
 :parameters (?r ?l ?c)
 (:method
  :precondition ()
  :tasks (sequence (turn-on-lamp ?l ?r) (start-computer ?c ?r))))
```

Fig. 2. Simple task encodings specified using the Hierarchical Planning Domain Language (HPDL).



```
(:task adjust-office
 :parameters (?r ?a ?l ?m)
 (:method
  :precondition ()
  :tasks (sequence (set-AC ?a ?r)
         (light-helper ?l ?r)(turn-on-music ?m ?r))))

(:task adjust-desk
 :parameters (?r ?l ?c)
 (:method
  :precondition ()
  :tasks (sequence (light-helper ?l ?r) (start-computer ?c ?r))))

(:task light-helper
 :parameters (?l ?r)
 (:method
  :precondition (not (on ?l ?r))
  :tasks (sequence (turn-on-lamp ?l ?r)))
 (:method
  :precondition (on ?l ?r)
  :tasks ()))
```

Fig. 3. Enhanced task encodings specified in HPDL.

References

[1]. I. Georgievski, M. Aiello, HTN planning: Overview, comparison, and beyond, *Artificial Intelligence*, Vol. 222, Issue 5, 2015, pp. 124-156.

[2]. A. Tate. Generating project networks, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'77)*, 1977, pp. 888-893.

[3]. R. M. Young, M. E. Pollack, J. D. Moore. Decomposition and causality in partial-order planning, in *Proceedings of the International Conference on Artificial Intelligence Planning Systems (AIPS'94)*, 1994, pp. 188-194.

[4]. I. Georgievski, M. Aiello, Automated planning for ubiquitous computing, *ACM Computing Surveys*, Vol. 49, Issue 4, 2017, pp. 63:1-63:46.

[5]. R. E. Fikes, N. J. Nilsson, STRIPS: A new approach to the application of theorem proving to problem solving, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'71)*, 1971, pp. 608-620.

[6]. E. D. Sacerdoti, A structure for plans and behavior, PhD Thesis, *Stanford,* CA, USA, 1975, aAI7605794.

[7]. D. E. Wilkins, Practical Planning: Extending the Classical AI Planning Paradigm, *Morgan Kaufmann Publishers Inc.*, San Francisco, CA, USA, 1988.

[8]. S. Kambhampati, J. A. Hendler, A validation-structure-based theory of plan modification and reuse, *Artif. Intell.*, Vol. 55, Issue 2, 1992, pp. 193-258.

[9]. E. Charniak, D. McDermott, Introduction to Artificial Intelligence, *Addison-Wesley*, 1986.

[10]. D. E. Foulser, M. Li, Q. Yang, Theory and algorithms for plan merging, *Artif. Intell.*, Vol. 57, Issue 2-3, 1992, pp. 143-181.

[11]. J. S. Cox, E. H. Durfee, Discovering and exploiting synergy between hierarchical planning agents, in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'03)*, 2003, pp. 281-288.(16)

(16)

Supervised Learning to Identify Roundabouts Using IMU

M. G. Altarabichi [1], M. Uddin Ahmed [1] and S. Begum [1]
[1] School of Innovation Design and Engineering (IDT), Mälardalen University, Västerås, Sweden
E-mail: {mohammed.ghaith.altarabichi, mobyen.ahmed, shahina.begum}@mdh.se

Summary: Events and interactions that take place in road junctions (i.e., roundabouts and intersections) are interesting to Naturalistic Driving Studies (NDS) analysing safety and behavioural aspects of road users. This work investigates several machine learning techniques including Artificial Neural Network (ANN), Random Forest (RF), k-Nearest Neighbours (k-NN), Logistic Regression (LR) and Support Vector Machines (SVM) to develop a model to identify roundabouts using Inertial Measurement Unit (IMU) sensory data. The study is based on NDS data of 16 volunteers' part of SimuSafe project collected between May and August 2018. GPS signal is used to label the ground truth of roundabouts manually. The ANN model outperforms other classifiers achieving 81 % F_1 score for roundabout identification, The ANN also generalize and identify cross-driver patterns better than any other tested model. Feature selection shows the importance of *lateral acceleration*, *yaw* and *pitch* signals in identifying roundabouts.

Keywords: Road safety, Machine learning, Naturalistic driving, Road junctions, Roundabout, GPS, IMU.

1. Introduction

In 2010, The European Commission issued a policy [1] aiming to halve the number of road deaths by 2020 with the vision of setting 2050 as a deadline to achieve zero road fatalities in EU [2]. Meeting these objectives promotes initiatives to conduct researches to reduce the rate of road deaths and improve road safety.

Naturalistic Driving Studies (NDS) is a research field that analyses and monitors drivers in their natural settings and has shown usefulness in the analysis of driver behaviour [3]. However, identification of events of interest (e.g. near-collisions, crashes, and violations) that occur during NDS cannot be predictably retrieved. Due to this unpredictably, data has to be collected along a large time frame which make the task of identifying events manually by a human time-consuming, tedious and impractical. These challenges inspired the need to develop automated tools capable of reliably identifying potential events of interest.

A road junction is by definition a certain structure in which two or more roads cross (i.e., roundabouts and intersections). This path crossing of vehicles and road users (pedestrians, cyclist and motorcyclists) makes road junctions a prime candidate for NDS analysing safety and behavioural aspects of road users [4-8]. The typical approach of road junction identification is based either on aerial images [9-12] or cameras fixed to the vehicle [13].

2. Related Work

A vision-based algorithm of identifying road junctions is presented in [13]. The algorithm uses videos coming from a camera mounted on the roof of a moving vehicle, the authors report detection of position of the road junction in 165-195 milliseconds. However, the reported experiment is performed in a controlled settings within Bristol University Campos and at a constant given speed.

The use of aerial images to identify junctions are observed in several studies. Authors of [9] employed Artificial Neural Network (ANN) to capture junctions from aerial images. In [10] junctions are identified from satellite images using Extended Kalman Filter (EKF) and Particle Filter (PF). High resolution Synthetic Aperture Radar (SAR) images are used to detect junctions in [11], while in [12] authors use camera images taken by an aircraft.

The use of IMU sensory data for roundabout identification is discussed in [14], the authors created a yaw rate profile for different exit scenarios of roundabout data recorded during a field operation test done in Gothenburg, as part of the euroFOT project. The documented results demonstrated an area under the curve (AUC) above 0.8 for right roundabout manoeuvres while the AUC of going straight and left in a roundabout exceeded 0.9. A method of path curve identification is implemented [15] based on variation of heading angle acquired through IMU sensor. The start-end points of the curve (Point of Curve PC, Point of Tangent PT) is identified using this method in addition to the length of the curve. The ANOVA test showed that a kinematic approach showed the best results in comparison to geometry-based and lateral-acceleration methods when compared against the ground truth measured by field tests. Authors of [16] proposed a path planning method for automated vehicles that divide the driving process in a roundabout to three stages: entrance maneuverer, driving within the roundabout and exit manoeuvre. A similar three divisions of roundabout manoeuvres are discussed in [17] for verification of roundabout safety design that demonstrated correlation between roundabout design and entering vehicle speed.

This research contributes a novel roundabout detection model designed to extract driving manoeuvres from various IMU signals. The use of vehicular data provides rich insights on important features that represent forces and angular rates observed during driving manoeuvres that can't be retrieved with an image-based approach. Machine Learning is used to demonstrate the importance of using other IMU signals in addition to *yaw rate* used in the literature [14-15] to identify roundabouts. Feature importance performed using Random Forest highlights the importance of *lateral acceleration* and *pitch rate* signals for the identification of roundabouts. The results of validating the created machine learning models suggest that ANN succeed to generalize and identify cross-driver patterns better than any other tested model. The ANN outperforms other classifiers by more than 6 % in F_1 performance.

3. Materials and Method

The roundabout detection model is built to recognize sequences of driving maneuvers using Yaw Rate of IMU sensor as a base signal. The paper will demonstrate the roundabout identification method in accordance to the following stages of model development.

3.1. Data Collection

The analysis of this work is performed using timestamped sensor data that corresponds to NDS trips. The data analysed is part of SimuSafe project of 16 volunteers and is collected between May and August 2018. The roundabout detection model is developed to use IMU sensor data for the purpose of classification, eight signals are utilized from IMU sensory data as observed from Table 1.

Table 1. IMU signals utilized to construct features of the roundabout classification model.

No.	Signal	Description
1.	RollRateExtSns	Vehicle roll rate
2.	RollExtSns	Roll angle of vehicle chassis
3.	PitchRateExtSns	Vehicle pitch rate
4.	PitchExtSns	Pitch angle of vehicle chassis
5.	YawRateExtSns	Vehicle yaw angle
6.	LongAccelExtSns	Longitudinal acceleration
7.	VertAccelExtSns	Vertical acceleration
8.	LatAccelExtSns	Lateral acceleration

3.2. Feature Extraction

Signals of Table 1 are resampled from 14 Hz to 1 Hz in order to reduce signal noise, and a Gaussian rolling window is applied in order to smoothen the resampled signals. The *YawRateExtSns* is selected as a base signal in order to establish the start and end time of driving manoeuvres. The integral of *YawRateExtSns* readings between start and end time is calculated to measure the corresponding *Yaw angle* of the performed manoeuvre. The time series of sensor readings that represent driving trip is transformed accordingly into a sequence of manoeuvres performed by the vehicle. Driving manoeuvres are presented by eight time-domain features that correspond to the mean value of each of the eight IMU signals over the time period of the manoeuvre. A ninth feature were added that correspond to the calculated *Yaw angle*, and a feature that correspond to the *duration of the manoeuvre in seconds* is added to total 10 features per manoeuvre.

3.3. Feature Engineering

The literature highlighted dividing the driving process in a roundabout into three stages: entrance, driving within the roundabout and exit manoeuvres [16-17]. In order to account for this manoeuvres sequence, each dataset instance is built using a rolling window with three manoeuvres as a window size. By following this approach each instance is presented by 30 features that summarize IMU sensory data of the last three consecutive manoeuvres performed by the vehicle.

Fig. 1 demonstrates an example of the feature creation method applied to *YawRateExtSns* signal of a trip recorded on 9th May 2018 between 05:50:00 and 05:50:35. Positive values of *YawRateExtSns* represent left turns while negative values indicates right turns. By using the calculated integral of Area-1 in Fig. 1 it was observed that the vehicle completed first a 46° right turn maneuver to enter the roundabout between 05:50:06 and 05:50:13, Positive values of Area-2 between 05:50:13 and 05:50:26 indicates a 148° left turn within the roundabout which represent the second maneuver. Lastly, the vehicle make another 88° right turn on 05:50:26 after exiting the roundabout.

This approach generated a training data set that consist of 22,080 instances. GPS signals were used to label 249 roundabouts occurrences manually according to map images of the driving trips. The generated dataset was used to train several classifiers including Artificial Neural Network (ANN), Random Forest (RF), k-Nearest Neighbours (k-NN), Logistic Regression (LR) and Support Vector Machines (SVM). The Python Deep Learning library Keras[1] is used for the implementation of ANN. While the machine learning scikit-learn library[2] in Python is used for all other models. Hyperparameters optimization is performed using a grid search to tune different classifier parameters. For the k-NN classifier (k = 1), while the RF (n_estimators = 41, min_samples_ leaf = 2, bootstrap = False, max_features = sqrt), the SVM (kernel = rbf, class_weight = balanced) and finally LR (solver = liblinear). The problem of class imbalance in the dataset is addressed by performing validation using different values of class-weight [1, 2, 4, 8, 16, 99]. F_1 score of the roundabout class is selected as the main metric to compare different classifier performance.

[1] https://keras.io/

[2] https://scikit-learn.org/stable/

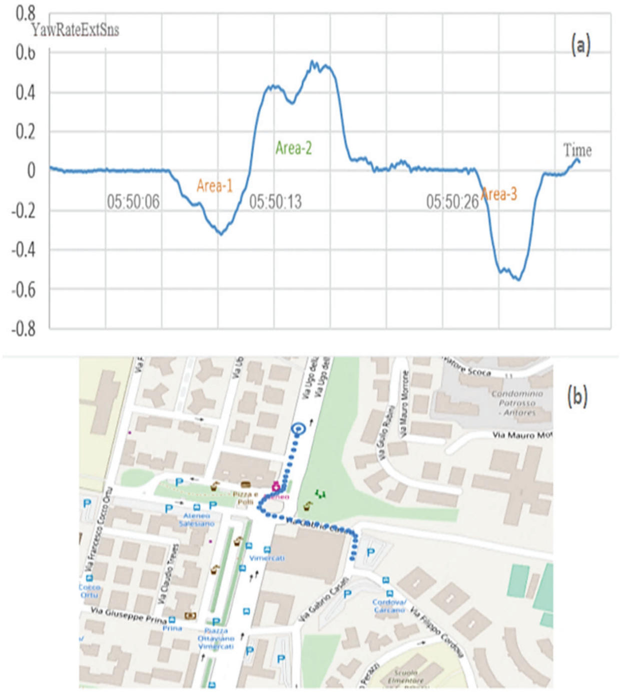


Fig. 1. (a) The plot of corresponding YawRate signal for the trip recorded on 9th May 2018.
(b) A GPS drawing of the path of the trip recorded on 9th May 2018.

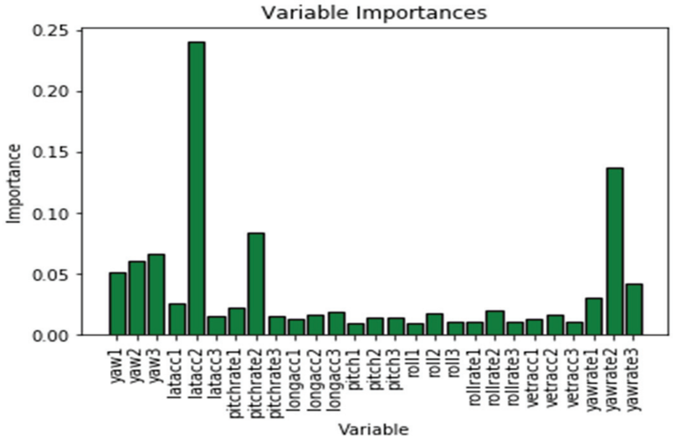
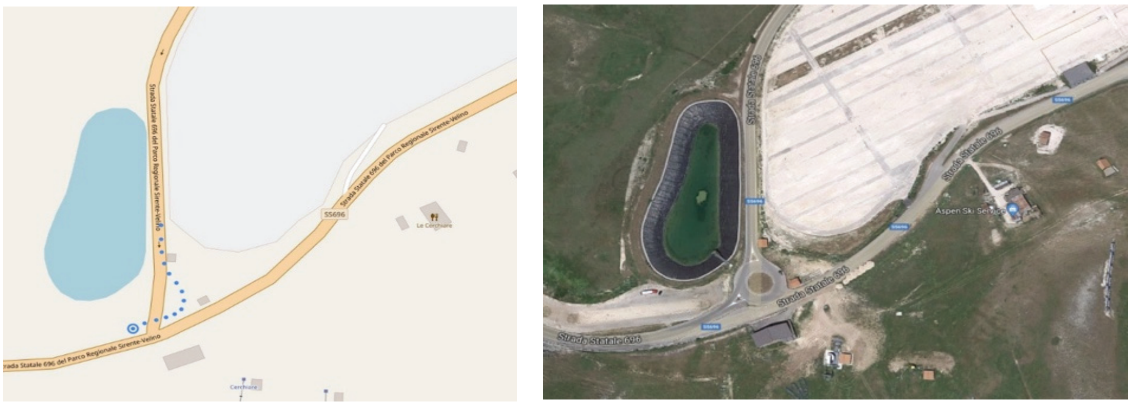


Fig. 2. Features importance using random forest.



(a) (b)

Fig. 3. A roundabout case identified by the mode. (a) The blue dots in the left map image represent the sequence of manoeuvres classified as a roundabout by the ANN model; (b) The satellite image to the right shows a roundabout under construction in Strada Statale 696, 67047 Rocca di cambio AQ, Italy (42.211361, 13.454691).

4. Results and Discussion

To identify important features for the detection of roundabouts Random Forest is used to perform feature ranking. As observed from Fig. 2 *lateral acceleration*, *yaw rate* and *pitch rate* are the top ranked features that contributed notably to the roundabout classification decisions. It could also be noted that the long acceleration of the roundabout entrance manoeuvre *longacc1* is not important for the classification. This result could indicate that drivers enter roundabouts in different styles, they don't necessarily deaccelerate to enter roundabouts.

The created models were validated using 10-folds cross validation (CV). As observed from Table 2, the ANN model (using two fully connected layers with 128 neurons in each layer and a dropout = 0.05) achieved the best F_1 score with a balanced precision 81 % and recall 82 % performance for the roundabout class. Table 3 demonstrates the confusion matrix of the best performing classifier. The validation process demonstrated the model capability to identify roundabouts that were challenging to spot even by the manual process of inspecting GPS map images, as demonstrated in the classification examples of Fig. 3. Examples of correctly classified roundabouts of all shapes and scenarios could be viewed in Fig. 4.

Table 2. F_1 Score, precision and recall of different classifiers for the roundabout class 10-fold CV.

No	Classifier	F_1	Precision	Recall
1.	ANN	**81.36**	81.20	**81.53**
2.	k-NN	80.65	81.82	79.52
3.	SVM	77.14	90.76	67.07
4.	RF	73.22	**94.30**	59.84
5.	LR	63.53	71.72	57.03

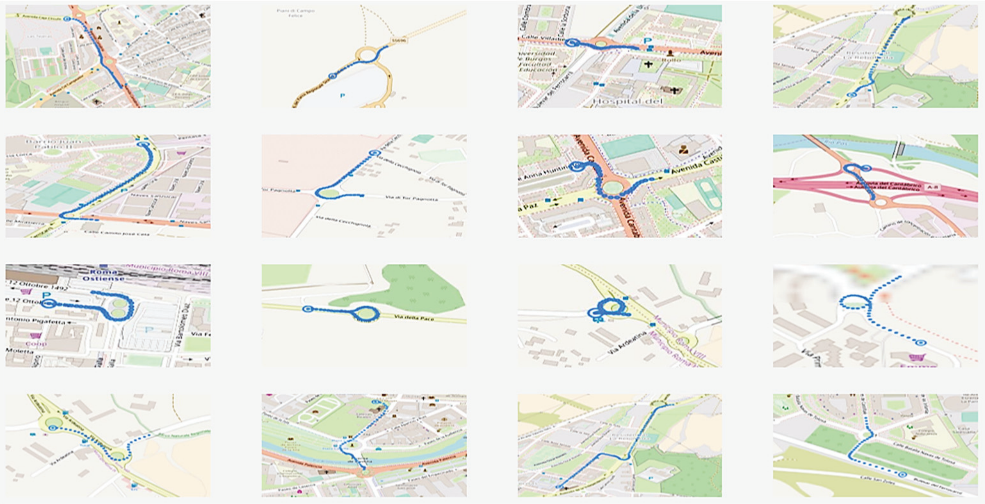


Fig. 4. Examples of correctly classified roundabouts by the ANN model of different shapes and scenarios.

Table 3. ANN model confusion matrix for 10-fold CV.

		Predicted Class	
		Other	**Roundabout**
True Class	Other	21,784	47
	Roundabout	46	203

The developed models demonstrates capability of identifying roundabouts by recognizing sequences of driving manoeuvres. The models were validated further using a leave-one-out cross validation (LOOCV) approach. This validation is performed to assess the model generalization capability in detecting driving manoeuvres of an unseen driver. In each fold (16 folds that correspond to 16 drivers) of the validation, driving manoeuvres of a certain driver is excluded from training and is used to form a test dataset. Results of LOOCV in Table 5 shows that ANN outperforms other classifiers by more than 6 % in F_1 performance. This LOOCV results suggests that the ANN model succeed to generalize and identify cross-driver patterns, unlike k-NN for instance that suffers more than 13 % drop in F_1 performance in comparison to 10-folds CV.

Table 4. F_1 Score, precision and recall for the roundabout class LOOCV.

No.	Classifier	F_1	Precision	Recall
1.	ANN	**76.92**	84.95	**70.28**
2.	SVM	70.53	88.48	58.63
3.	RF	67.87	**94.29**	53.01
4.	k-NN	67.33	67.20	67.47
5.	LR	63.98	72.22	57.43

Class imbalance in the dataset is addressed by performing validation using different values of class weight [1, 2, 4, 8, 16, 99]. The best performing model ANN is selected for this optimization process of selecting weight. Table 5 list results of ANN classifiers trained with different values of class weight. We may infer from the results that assigning higher weight value for the roundabout class is leading the loss function to optimize for a higher roundabout recall. However, precision performance drops with a higher weight value. This allows to dynamically select the best weight value based on requirement of ND study. Given the priority of recalling as much roundabout event as possible for instance, a network with weight set to 16 could recall 90 % of roundabout occurrences of an unseen driver, at the expense of encountering higher false positives rate.

Table 5. F_1 Score, precision and recall for 10-folds CV and LOOCV using different values of class weight.

10-Folds CV			
Class Weight	**F_1**	**Precision**	**Recall**
1	**81.36**	**81.20**	81.53
2	79.55	74.05	85.94
4	77.09	70.43	85.14
8	74.15	62.30	91.57
16	69.60	54.86	95.18
32	64.66	49.06	94.87
99	57.18	40.43	**97.59**
LOOCV			
Class Weight	**F_1**	**Precision**	**Recall**
1	**76.92**	**84.95**	70.28
2	76.07	77.50	74.40
4	75.36	68.65	83.53
8	68.15	56.28	86.35
16	67.07	53.61	89.56
32	60.18	44.73	91.97
99	52.34	36.21	**94.38**

In order to validate the potentials of improving the performance further using a larger dataset, we plot the performance of different k-NN models (since it is fast to train and evaluate, using k = 1) trained with an increasing number of roundabouts. As observed from Fig. 5. The model achieves 73 % F_1 score when trained with over 50 roundabouts, the accuracy rises to 77 % with 150 and exceeds 80 % with more than 200 roundabout occurrences in training data.. A future work is planned to investigate further if adding more manoeuvres and drivers to the training dataset could contribute to a better generalization capability.

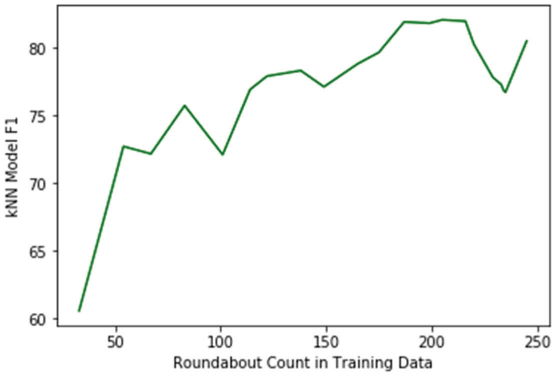


Fig. 5. k-NN Model F_1 Performance with more roundabout occurrences in training data.

A planned future work is to extend the tests of the model by including other types of road junctions (i.e., intersections) and driving manoeuvres (i.e., overtaking). Performing this future work could provide valuable information and insights about different driving styles in other road junctions and manoeuvres.

Sensor fusion of other vehicular signals like GPS could potentially improve some of the current observed limitations of the model. The model classifies some of the S-shaped road curves as roundabouts as observed in Fig. 6. Feature selection is limited in this work to the use of Random Forest, other methods of feature selection could enrich our understanding of important features and signals that could potentially improve the automatic detection capability of road structures and driving manoeuvres.

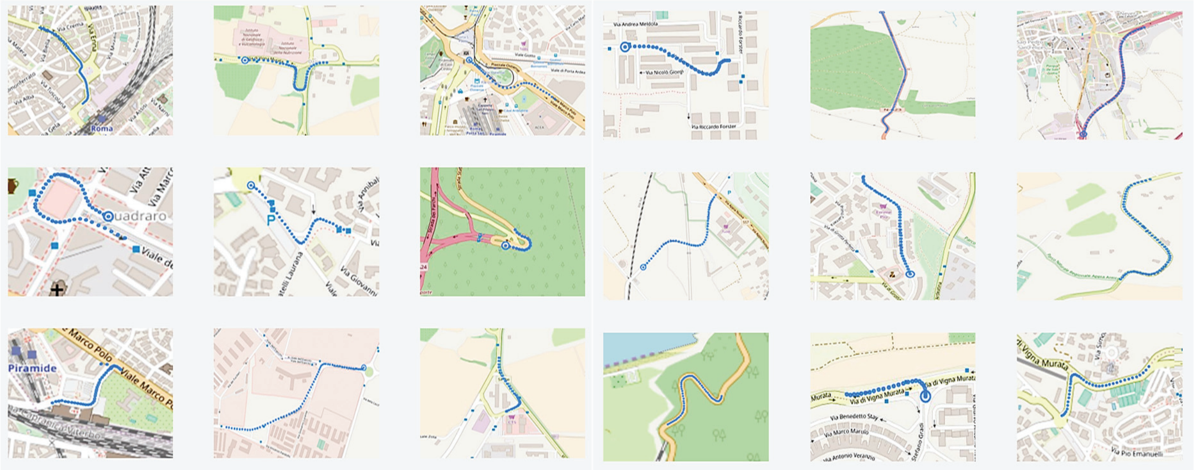


Fig. 6. Examples of debatable and wrong roundabout classifications by the ANN model.

Acknowledgements

The authors would like to acknowledge the project SimuSafe, the project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723386.

References

[1]. KPMG (2012), Self-Driving Cars: The Next Revolution, KPMG and the Center for Automotive Research, www.kpmg.com/Ca/en/IssuesAndInsights/ArticlesPublications/Documents/self-driving-cars-nextrevolution.pdf

[2]. D. J. Fagnant, K. M. Kockelman, Preparing a Nation for Autonomous Vehicles: Opportunities, Barriers and Policy Recommendations, Eno Foundation (www.enotrans.org), www.enotrans.org/wp-content/uploads/wpsc/downloadables/AV-paper.pdf

[3]. J. S. Hickman, R. J. Hanowski, An assessment of commercial motor vehicle driver distraction using naturalistic driving data, *Traffic Injury Prevention*, Vol. 13, Issue 6, 2012, pp. 612-619.

[4]. R. Kulmala, Safety at Rural Three-and Four-arm Junctions: Development and Application of Accident Prediction Models, Vol. 233, *VTT Technical Research Centre of Finland*, 1995.

[5]. R. Kulmala, Measuring the safety effect of road measures at junctions, *Accident Analysis Prevention*, Vol. 26, Issue 6, 1994, pp. 781-794.

[6]. C. Hydn, A. Vrhelyi, The effects on safety, time consumption and environment of large-scale use of roundabouts in an urban area: A case study, *Accident Analysis Prevention*, Vol. 32, Issue 1, 2000, pp. 11-23.

[7]. D. D. Clarke, R. Forsyth, R. Wright, Behavioural factors in accidents at road junctions: The use of a genetic algorithm to extract descriptive rules from police case files, *Accident Analysis Prevention*, Vol. 30, Issue 2, 1998, pp. 223-234.

[8]. Y. Chen, H. Meng, Z. Wang, Safety improvement practice for vulnerable road users in Beijing junctions in *Proceedings of the Transportation Research Board 88th Annual Meeting*, 2009.

[9]. A. Barsi, C. Heipke, May. Detecting road junctions by artificial neural networks, in *Proceedings of the 2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas (URBAN'03)*, 2003, pp. 129-132.

[10]. S. Movaghati, A. Moghaddamjoo, A. Tavakoli, Road extraction from satellite images using particle filtering and extended Kalman filtering, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 48, Issue 7, 2010, pp. 2807-2817.

[11]. M. Negri, P. Gamba, G. Lisini, F. Tupin, Junction-aware extraction and regularization of urban road networks in high-resolution SAR images, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 44, Issue 10, 2006, pp. 2962-2971.

[12]. L. Wu, Y. Hu, 2009, Vision-aided navigation for aircrafts based on road junction detection, in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS'09)*, Vol. 4, 2009, pp. 164-169.

[13]. M. Ekinci, B. T. Thomas, Road junction recognition and turn-offs for autonomous road vehicle navigation, in *Proceedings of the 13th International Conference on Pattern Recognition*, 1996, Vol. 3, pp. 318-322.

[14]. F. Jorge, Detecting roundabout manoeuvres using OpenStreetMap and vehicle state, *Chalmers University of Technology*, Göteborg, Sweden, 2012.

[15]. W. Luo, L. Li, K. C. Wang, Automated pavement horizontal curve measurement methods based on inertial measurement unit and 3D profiling data, *Journal of Traffic and Transportation Engineering*, Vol. 3, Issue 2, 2016, pp. 137-145.

[16]. D. Gonzalez, J. Prez, V. Milans, Parametric-based path generation for automated vehicles at roundabouts, *Expert Systems with Applications*, Vol. 71, 2017, pp. 332-341.

[17]. H. Pilko, D. Bri, N. Ubi, Study of vehicle speed in the design of roundabouts, *Graevinar: Asopis Hrvatskog Savezagraevinskih Inenjera*, Vol. 66, Issue 5, 2014, pp. 407-416.

(17)

Enhancing Computerized-aided Instruction by Providing Dynamically Adaptive Learning Material through Artificial Intelligent Techniques

Konstantina Chrysafiadi, Christos Troussas and Maria Virvou
Department of Informatics, University of Piraeus, Piraeus, Greece
Tel.: + 302104142131
E-mail: {kchrysafiadi, ctrouss, mvirvou}@unipi.gr

Summary: Nowadays, there is a fertile ground for the research area of e-learning. E-learning systems can promote education by providing the learning experience to students regardless of the place they are and the time of the learning delivery. Therefore, the learners of an e-learning system have a different knowledge level and background, abilities and needs. So, there is a growing need for tailored education. This is achieved by the incorporation of "intelligence" into the teaching and learning processes of e-learning systems, since it allows the provision of personalized instruction to students. In this paper, an intelligent mechanism (ICALM) that allows an adaptive learning system to create and adapt dynamically the learning material on the fly, taking into consideration each time the needs and abilities of each individual learner, is presented. ICALM adapts the learning material at three levels: i) knowledge level, ii) content, iii) display mode. It uses artificial neural networks, multi-criteria decision analysis and machine learning techniques. The presented mechanism has been incorporated in two adaptive e-learning systems and was evaluated. The evaluation results are very encouraging.

Keywords: Adaptive e-learning, Adaptive learning material, Artificial neural network, Personalized tutoring, Learning style, Machine learning.

1. Introduction

In recent years, the rapid development of technology has affected several aspects of people's everyday life. Indeed, people tend to use technology in order to build new approaches in many fields, including education. Moreover, the socio-economic changes worldwide necessitate the employment of new technological approaches in education; hence, they give birth to e-learning, which is a way of distance education that overcomes the obstacles posed by place and time, offering a more adaptable instruction in comparison to traditional classrooms [10]. However, e-learning systems are used from heterogeneous groups of learners meaning that such groups are distinguished by different needs, preferences and interests [2]. A solution to this challenge is the employment of artificial intelligence (AI) in learning systems so that they adapt the learning material or the tutoring and learning processes to each individual student's needs and abilities, offering him/her a personalized learning experience. More specifically, such techniques allow the perception and determination of learners' needs [9, 12]. Also, they help the system create a routine for each learner concerning the dynamic adaptation of the learning and teaching processes, including the delivery of the domain being taught and the tailored assessment strategies, etc. [20]. As such, intelligent and adaptive educational systems usually provide an individualized learning path to each student.

Adaptive educational systems and applications use a variety of AI techniques [7]. Examples of such techniques are the artificial neural networks (ANN) and the machine learning (ML) techniques. An ANN mimics certain aspects of the information processing and physical structure of the human brain with a web of neural connections. They are used in the field of e-leaning towards providing personalization to students' needs by finding the similarity of the domain concept data representation pattern between the students' and the learning object's profiles [16, 18]. In [2, 10], the authors focused on the adaptation of the e-learning system to the students' specific needs and preferences using ANNs. Other authors used ANN to form a recommendation system to support students [20] by delivering them adaptive instruction [12, 19], or proposing them a learning path that meets in a better way their needs and abilities [19]. Finally, ANN has been also used for emotion recognition in e-learning systems [9, 16]. ML is related to algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on models and inference instead [5]. It involves algorithms that predict possible outcomes based on students' data. The system identifies certain patterns and trends, and then learns from the data in order to provide greater personalization. ML techniques have been employed towards offering individualized learning pathways. For example, in [5, 11, 17], the authors employed machine learning techniques in order to support collaborative learning and find patterns of interaction between learners showing that prior knowledge and communication skills of learners are likely to influence effectiveness of collaborative learning. Other efforts have been focused on the amelioration of learners' knowledge sources as well as on adaptive pedagogy. For example, in [4, 8, 13], machine learning techniques have been used in order to increase student engagement and improve learning outcomes. A lot of research effort has been also placed on the diagnosis of mistakes using machine learning. For example, in [1, 15, 22], the authors reported that e-learning can be enhanced by the identification of learners' errors and the adaptation of the teaching strategy to the weaknesses of students.

Furthermore, decision analysis is also employed in the field of e-learning. It is used to build systems, designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as if–then rules [11]. Decision analysis is mainly used in circumstances that a decision has to be taken. As an example, in e-learning, decision analysis is used to specify the proper domain knowledge to be delivered to students or the appropriate assessment to be given to students. Decision analysis processes have been used in the related scientific literature in order to provide tailored assessments to students [6, 26]. For example, in [6], the authors create automated adaptive tests using multiple-criteria decision analysis taking into consideration multiple students' criteria along with the types of exercises and the desirable learning objective. Also, in [14] a decision-making model and method are proposed to evaluate suitability, acceptance and use of personalised learning units. Furthermore, present multi-criteria decision analysis approaches for selecting and evaluating digital learning objects [3].

Taking into consideration the above, this paper presents an intelligent mechanism that allows an adaptive learning system to create and adapt dynamically the learning material on the fly, taking into consideration each time the needs and abilities of each individual learner. The presented mechanism is implemented by an Intelligent Creator of Adaptive Learning Material (ICALM). ICALM adapts the learning material that is addressed to a particular learner at three levels: i) knowledge level, ii) content of the learning material (which concepts will be presented and focused), and iii) display mode of the learning content (i.e. video, text, images, audio etc.). The presented intelligent mechanism uses artificial neural networks, a method of multi-criteria decision analysis and machine learning techniques. The knowledge level of the students is used to identify the domain knowledge concept. The artificial neural network takes as input the learning style of the students which is based on the VARK model (Visual, Auditory, Reading and Kinesthetic Learners) and gives as output the display mode of the learning material. The calculation of the activation function of the presented ANN is based on the Weighted Sum Model (WSM), which is a method for multi-criteria decision analysis. More specifically, each sensory modality of the VARK model uses different weights. This means that if a leaner is a visual learner wants to be shown the learning content in the mode of text, video narration, images, diagrams and examples but in a different percentage. The other modalities take the weights following the same rationale. Then, the activation function, which is based on WSM, determines the output of neural network by mapping the resulting values. Regarding the partitioning of students' learning style, it is specified using the k-means clustering algorithm. Finally, the types of learners' errors, which are diagnosed using machine learning algorithms, and specifically String Matching algorithm and String Meaning Similarity technique, serve for the content of the domain concept.

As a testbed for our research, two fully operating and evaluated e-learning systems have been used: i) a web-based educational application for programming languages tutoring and ii) a web-based e-learning system for English and French language tutoring. Examples of operation of these two systems, using ICALM, attests that ICALM is proved to be effective for optimizing e-learning, while providing great assistance to learners in a computer-aided instruction environment. Finally, a wide-range evaluation is presented. Its results are very encouraging, showing that the modules of ICALM incorporating either machine learning or an artificial neural network and the weighted sum model can enhance computer-aided instruction.

The remainder of this paper is organized as follows. Section 2 describes ICALM. Evaluation process and results are presented in Section 3. Finally, in Section 4, conclusions are drawn and future plans are discussed.

2. The Description of ICALM

ICALM is the name of the presented Intelligent Creator of Adaptive Learning Material. Its aim is to adapt the learning material that is addressed to a particular learner at three levels: i) knowledge level, ii) content of the learning material (which concepts will be presented and focused), and iii) display mode of the learning content (i.e. video, text, images, audio etc.). For the adaptation of the learning material to the knowledge level of the learner, the system checks the results of the learner's assessment and decides which the concepts are that coincide with her/ his knowledge level. The selection of the content of the domain concept is based on the types of errors that the learner usually does. The diagnosis of errors' types is achieved through the String Matching algorithm and String Meaning Similarity technique. In addition, the decision for the display mode of the learning material is based on the learning style of each particular learner and is achieved through an artificial neural network in conjunction with the weighted sum model. The logical architecture of ICALM is depicted in Fig. 1. The mechanism of adaptation of the learning material is described in more details below for each adaptation level.

2.1. Adaptation to Knowledge Level

The knowledge level of a particular student is used to identify the domain concepts that have to be delivered to her/him meeting his/her educational needs and abilities. The learner's knowledge level is determined by the results of her/his assessments. According to these results, the system selects the domain concepts of the learning material that coincides with the learner's knowledge level and delivers them to her/him for studying.

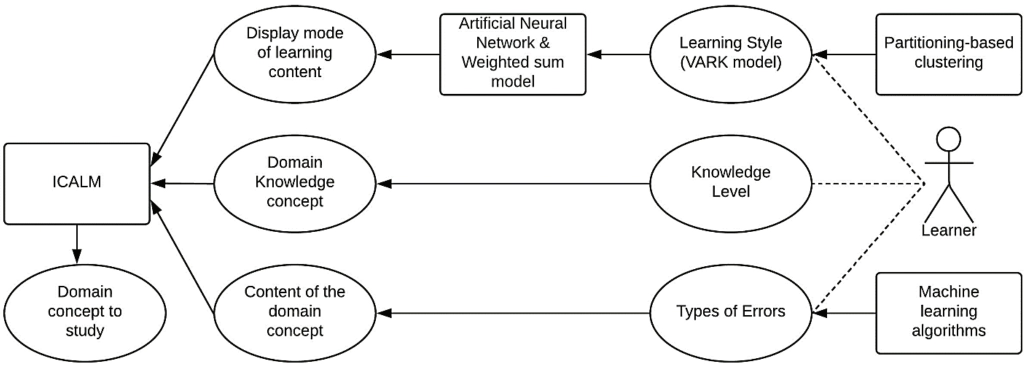

Display mode of learning content
Artificial Neural Network & Weighted sum model
Learning Style (VARK model)
Partitioning-based clustering
ICALM
Domain Knowledge concept
Knowledge Level
Learner
Domain concept to study
Content of the domain concept
Types of Errors
Machine learning algorithms

Fig. 1. The logical architecture of ICALM.

2.2. Adaptation of the Content

For deciding about the issues of each delivered domain concept that have to be presented and focused for a particular learner, the types of errors (i.e. grammatical, syntactic, logical or knowledge transfer mistakes in language learning) that s/he, usually does, have to be detected. The diagnosis of errors' types is based on machine learning algorithms and specifically String Matching algorithm and String Meaning Similarity technique. String Matching algorithm tries to find strings that match a pattern approximately in order to identify the kind of mistakes between grammatical, syntactic or logical. String Meaning Similarity is responsible for identifying knowledge transfer mistakes in language learning and works by finding and translating patters, determining the source language of knowledge. Since it is not the scope of this paper to present how the aforementioned algorithms function, an in-depth analysis of them is presented in the authors' previous research works [23, 25].

2.3. Adaptation of the Display Mode

For the adaptation of the display mode of the learning material an artificial neural network is used, operating in conjunction with the multi-criteria decision analysis (MCDA). The artificial neural network takes as input the learning style of the students which is based on the VARK model (Visual, Auditory, Reading and Kinesthetic Learners). More specifically, each sensory modality of the VARK model uses different weights based on the Weighted Sum Model which is the best known and simplest MCDA method for evaluating a number of alternatives in terms of a number of decision criteria [21]. This means that if a learner is a visual learner, then s/he wants the learning content to be shown in the mode of text, video narration, images, diagrams and examples but in a different percentage. The other modalities take the weights following the same rationale. Then, the activation function, which is calculated using WSM, determines the output of neural network by mapping the resulting values, as shown in Fig. 2. The weights in the ANN have been defined by 14 professors and teachers (3 teachers of primary school, 5 teachers of secondary school and 6 university professors) of different knowledge domains. Each of these 14 tutors has at least 14 years experience in educational process and instruction. However, the defined weights of the presented ANN can be changed according to the preferences of each particular tutor. Let's see an example of operation of the presented ANN. Mike is a student that his learning style is 70 % Reading and 30 % Visual, according to the system's classification. So, XV = 0.3, XA = 0, XR = 0.7 and XK = 0. The values of the corresponding weights of our ANN are the following: W1R = 0.8, W4R = 0.1, W5R = 0.1, W1V = 0.1, W2V = 0.2, W3V = 0.3, W2V = 0.3 and W5V = 0.1. According to the activation function that is based on WSM, we have the following results:

$$y_1 = 0.3 * 0.1 + 0 + 0.7 * 0.8 = 0.59,$$

$$y_2 = 0.3 * 0.2 + 0 + 0 = 0.06,$$

$$y_3 = 0.3 * 0.3 = 0.09,$$

$$y_4 = 0.3 * 0.3 + 0.7 * 0.1 = \\ = 0.03 + 0.56 = 0.16,$$

$$y_5 = 0.3 * 0.1 + 0 + 0.7 * 0.1 = 0.1$$

From the above results, the system concludes that the content of the domain concept, which is going to be delivered to Mike, has to be displayed, manly, with text, which will include some diagrams.

Regarding the partitioning of students' learning style, it is specified using the k-means clustering algorithm. The goal of this algorithm is to find groups in the data. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity. Each centroid of a cluster is a collection of feature values which define the resulting groups. More analysis on k-means algorithm in the context of e-learning is presented in the authors' previous research work [24].

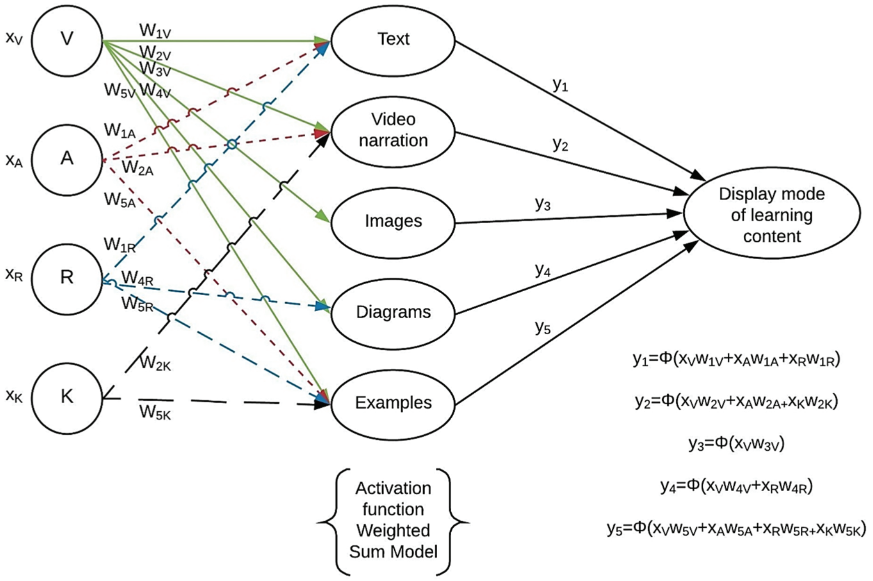

x_V
V
x_A
A
x_R
R
x_K
K
W_{1V}
W_{2V}
W_{3V}
W_{5V}
W_{4V}
W_{1A}
W_{2A}
W_{5A}
W_{1R}
W_{4R}
W_{5R}
W_{2K}
W_{5K}
Text
Video narration
Images
Diagrams
Examples
y_1
y_2
y_3
y_4
y_5
Display mode of learning content
$y_1=\Phi(x_Vw_{1V}+x_Aw_{1A}+x_Rw_{1R})$
$y_2=\Phi(x_Vw_{2V}+x_Aw_{2A+}x_Kw_{2K})$
$y_3=\Phi(x_Vw_{3V})$
$y_4=\Phi(x_Vw_{4V}+x_Rw_{4R})$
$y_5=\Phi(x_Vw_{5V}+x_Aw_{5A}+x_Rw_{5R+}x_Kw_{5K})$
Activation function Weighted Sum Model

Fig. 2. Construction of the ANN with MCDA.

3. Evaluation

The evaluation is a core phase in the systems development life cycle since it measures the effectiveness of the system and seeks potential enhancements. Then, the evaluation provides metric for the success of the software. Even though there are several frameworks and questionnaires that can be used in the evaluation study. There is not any standard agreed approach for evaluating adaptive learning material delivery. Hence, for the evaluation of ICALM, we have created a questionnaire that includes questions that examines the user experience, learning results and system efficiency. The questions were close-ended based on Likert scale with the responses ranging from "Unsatisfactory" (1) to "Excellent" (5). The questionnaire is depicted in Table 1.

Table 1. Questionnaire.

User experience	Q1	Is the user interface friendly?
	Q2	Did you like the domain knowledge units?
	Q3	Is your opinion about the tutoring process positive?
	Q4	Rate your experience during the tutoring process.
Learnin g results	Q5	Do you feel that you upgraded your knowledge during the tutoring process?
	Q6	How accurate are the learning objectives?
	Q7	How accurate is the identification of you knowledge level by the system?
System efficiency	Q8	Were you self-assured during the tutoring process?
	Q9	Do you believe that the domain knowledge units corresponded to your knowledge level?
	Q10	Do you feel that your lack of knowledge was properly handled by the system?
	Q11	Do you feel that the system was adapted to your knowledge level?
	Q12	Do you feel that the system was adapted your learning needs?

For the evaluation study, two fully operating e-learning systems have been used: i) a web-based educational application for programming languages tutoring and ii) a web-based e-learning system for English and French language tutoring. Examples of operation of these two systems, using ICALM, attests that ICALM is proved to be effective for optimizing e-learning, while providing great assistance to learners in a computer-aided instruction environment. In particular, 25 students used the first system and 30 students used the second system. They used the systems for a period of 6 weeks. The students were assisted during the whole evaluation process by 3 instructors and the evaluators. The contribution of the instructors in the evaluation was very important.

Fig. 3 illustrates the results of the three aforementioned categories of questions. These results include the average of answers of 55 students. The results of each category depict an average score of the questions belonging to each category. In all categories, ICALM seems to achieve great scores since it constructs a personalized and adaptive learning environment where students are engaged in learning and upgrade their knowledge by making use of all its modularities.

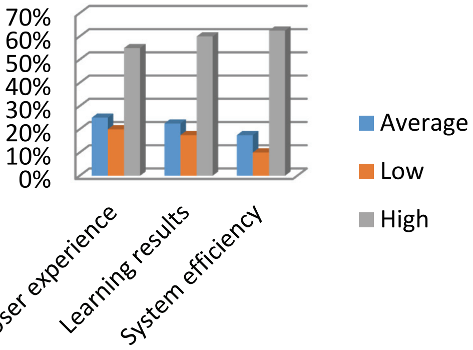


Fig. 3. The evaluation results.

4. Conclusions and Future Work

In this paper, an intelligent mechanism that allows an adaptive learning system to create and adapt dynamically the learning material on the fly, taking into consideration each time the needs and abilities of each individual learner has been presented. The presented mechanism is implemented by module, which is called ICALM (Intelligent Creator of Adaptive Learning Material). ICALM uses techniques of Artificial Intelligence in order to manage to adapt the learning material that is addressed to a particular learner to her/his knowledge level, abilities and needs. These techniques are an artificial neural network and the machine learning techniques of the k-means clustering algorithm, the String Matching algorithm and the String Meaning Similarity technique. Furthermore, our system uses the Weighted Sum Model, which is a method for multi-criteria decision analysis. The adaptation of the learning material is achieved in three levels: i) knowledge level, ii) content of the learning material (which concepts will be presented and focused), and iii) display mode of the learning content (i.e. video, text, images, audio etc.). The aim of the presented system is to use proficiency and determine what a student really knows and to accurately and logically move students through a sequential learning path to prescribed learning outcomes and skill mastery.

It is in our future plans to add more functionalities to our novel mechanism, such as the affect recognition. The affect recognition functionality can provide the opportunity to adapt the learning material to students based on their affect state. Furthermore, we will add to the presented artificial neural network the ability to be self-trained, in order to be able to adjust itself the values of the weights. This can be done in the future due to the fact that we will have in our disposal a large amount of data from the system's application in e-learning systems.

References

[1]. U. Z. Ahmed, P. Kumar, A. Karkare, P. Kar, S. Gulwani, Compilation Error Repair: For the Student Programs, From the Student Programs, in *Proceedings of the IEEE/ACM 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET'18)*, Gothenburg, Sweden, 2018, pp. 78-87.

[2]. J. Almotiri, K. Elleithy, A. Elleithy, Comparison of autoencoder and principal component analysis followed by neural network for e-learning using handwritten recognition, in *Proceedings of the IEEE Long Island Systems, Applications and Technology Conference (LISAT'17)*, Farmingdale, NY, 2017, pp. 1-5.

[3]. S. Başaran, Multi-criteria decision analysis approaches for selecting and evaluating digital learning objects, in *Proceedings of the 12th International Conference on Application of Fuzzy Systems and Soft Computing (ICAFS'16)*, Vienna, Austria, 2016, pp. 251-258.

[4]. A. Calma, D. Kottke, B. Sick, S. Tomforde, Learning to learn: Dynamic runtime exploitation of various knowledge sources and machine learning paradigms, in *Proceedings of the IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W'17)*, Tucson, AZ, 2017, pp. 109-116.

[5]. P. Chopade, S. M. Khan, D. Edwards, A. von Davier, Machine learning for efficient assessment and prediction of human performance in collaborative learning environments, in *Proceedings of the IEEE International Symposium on Technologies for Homeland Security (HST'18)*, Woburn, MA, 2018, pp. 1-6.

[6]. K. Chrysafiadi, C. Troussas, M. Virvou, A framework for creating automated online adaptive tests using multiple-criteria decision analysis, in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC'18)*, Miyazaki, Japan, 2018, pp. 226-231.

[7]. K. Chrysafiadi, M. Virvou, Student modeling approaches: A literature review for the last decade, *Expert Systems with Applications*, Vol. 40, Issue 11, 2013, pp. 4715–4729.

[8]. M. Dlamini, W. S. Leung, Evaluating machine learning techniques for improved adaptive pedagogy, in *Proceedings of the IEEE IST-Africa Week Conference (IST-Africa)*, Gaborone, 2018, pp. 1-10.

[9]. O. El Hammoumi, F. Benmarrakchi, N. Ouherrou, J. El Kafi, A. El Hore, Emotion recognition in

e-learning systems, in *Proceedings of the* 6^{th} *International Conference on Multimedia Computing and Systems (ICMCS'18)*, Rabat, 2018, pp. 1-6.

[10]. M. Holmes, A. Latham, K. Crockett, J. D. O'Shea, Near real-time comprehension classification with artificial neural networks: Decoding e-learner non-verbal behavior, *IEEE Transactions on Learning Technologies*, Vol. 11, Issue 1, 2018, pp. 5-12.

[11]. N. Joseph, N. Pradeesh, S. Chatterjee, K. Bijlani, A novel approach for group formation in collaborative learning using learner preferences, in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI'17)*, Udupi, 2017, pp. 1564-1568.

[12]. P. A. Khodke, M. G. Tingane, A. P. Bhagat, S. P. Chaudhari, M. S. Ali, Neuro fuzzy intelligent e-learning systems, in *Proceedings of the IEEE Online International Conference on Green Engineering and Technologies (IC-GET'16)*, Coimbatore, 2016, pp. 1-7.

[13]. R. Kokku, S. Sundararajan, P. Dey, R. Sindhgatta, S. Nitta, B. Sengupta, Augmenting classrooms with AI for personalized education, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'18)*, Calgary, AB, 2018, pp. 6976-6980.

[14]. E. Kurilovas, Advanced machine learning approaches to personalise learning: learning analytics and decision making, *Behavior & Information Technology*, 2018.

[15]. S. Larabi Marie-Sainte, N. Alalyani, S. Alotaibi, S. Ghouzali, I. Abunadi, Arabic natural language processing and machine learning-based systems, *IEEE Access*, Vol. 7, 2019, pp. 7011-7020.

[16]. G. Li, Y. Wang, Research on leamer's emotion recognition for intelligent education system, in *Proceedings of the IEEE* 3^{rd} *Advanced Information Technology, Electronic and Automation Control Conference (IAEAC'18)*, Chongqing, 2018, pp. 754-758.

[17]. I. Matazi, A. Bennane, R. Messoussi, R. Touahni, I. Oumaira, R. Korchiyne, Multi-agent system based on fuzzy logic for e-learning collaborative system, in *Proceedings of the International Symposium on Advanced Electrical and Communication Technologies (ISAECT'18)*, Rabat, Morocco, 2018, pp. 1-7.

[18]. S. G. Rabiha, A. Kurniawan, J. Moniaga, D. I. Wahyudi, E. Wilson, Sasmoko, Face detection and recognition based e-learning for students authentication: Study literature review, in *Proceedings of the International Conference on Information Management and Technology (ICIMTech'18)*, Jakarta, 2018, pp. 472-476.

[19]. T. Saito, Y. Watanobe, Learning path recommender system based on recurrent neural network, in *Proceedings of the* 9^{th} *International Conference on Awareness Science and Technology (iCAST'18)*, Fukuoka, 2018, pp. 324-329.

[20]. X. Shen, B. Yi, Z. Zhang, J. Shu, H. Liu, Automatic recommendation technology for learning resources with convolutional neural network, in *Proceedings of the International Symposium on Educational Technology (ISET'16)*, Beijing, 2016, pp. 30-34.

[21]. E. Triantaphyllou, Multi-Criteria Decision Making Methods: A Com- parative Study, *Kluwer*, Norwell, MA, 2000.

[22]. C. Troussas, K. Chrysafiadi, M. Virvou, Machine learning and fuzzy logic techniques for personalized tutoring of foreign languages, in *Proceedings of the Artificial Intelligence in Education (AIED'18)*, London, UK, 2018, pp. 358-362.

[23]. C. Troussas, M. Virvou, E. Alepis, Collaborative learning: group interaction in an intelligent mobile-assisted multiple language learning system, *Informatics in Education*, Vol. 13, Issue 2, 2014, pp. 279-292.

[24]. C. Troussas, M. Virvou, E. Alepis, Comulang: Towards a collaborative elearning system that supports student group modeling, *SpringerPlus*, Vol. 2, 387.

[25]. C. Troussas, M. Virvou, A. Vougiouklidou, K. J. Espinosa, Automatic misconception diagnosis in multiple language learning over social networks, in *Proceedings of the International Conference on Information, Intelligence, Systems and Applications (IISA'13)*, Piraeus, Greece, 2013, pp. 1-6.

[26]. R. Weis, E. L. Dean, K. J. Osborne, Accommodation decision making for postsecondary students with learning disabilities: Individually tailored or one size fits all?, *Journal of Learning Disabilitie*s, Vol. 49, Issue 5, 2016, pp. 484-498.

(19)

Trainable Monotone Combination of Classifiers

Sergey Grosman
Siemens PPAL, Konstanz, Germany
E-mail: srg.grosman@gmail.com

Summary: This paper suggests a new method for combining multiple classifiers that deliver a continuous valued output. The core of this combination is a trainable mapping with continuous valued output, which yields the monotonic behaviour with respect to each of the underlying base classifiers. Given a training sample the intermediate feature space is decomposed into regions of decreasing quality. It is shown that there exists a unique partitioning for which the associated combiner has the maximal possible AUC (area under the curve) within the family of monotone combiners. Furthermore, an algorithm is provided which finds the optimal partitioning and hence the associated monotone combiner in polynomial time. Although the proposed approach has historically emerged as a combiner of classifiers it can also be seen as a monotonic classifier acting on an abstract feature space.

Keywords: Machine learning, Classifier combination, Monotonic classifier, Receiver operating characteristic, Area under the curve.

1. Motivation: From naive Thresholding to a Monotone Combination

Suppose we have a binary classification problem. For the sake of clarity and ease of comprehension we call the classes positives and negatives. Let us assume that an ensemble of k base classifiers is given defined by means of their discriminant functions. The base classifiers can be any classifiers yielding continuous-valued output, e.g. neural networks or decision trees.

1.1. Simple Thresholding

As we start trying to use multiple base classifiers together, the first idea which comes to mind is to prescribe an individual threshold value to each of the base classifiers. The class label is then determined by checking all the base classifier outputs in the intermediate feature space; see Fig. 1.

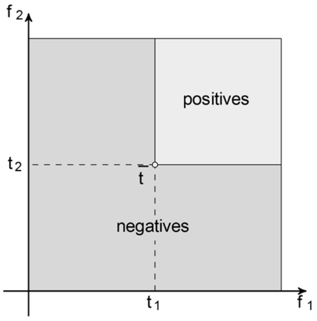


Fig. 1. Threshold combination.

1.2. Threshold Refinement

In practical application one very quickly comes to the point where simple thresholding is not flexible enough. Therefore, it has to be extended to multiple combinations of minimal acceptable scores and prescribe a positive class label as soon as a combination of the base classifier scores is not less than at least one of them: see Fig. 2.

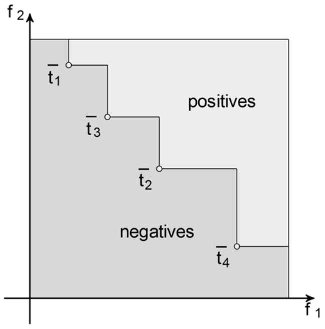


Fig. 2. Complex threshold refinement.

1.3. Quality Levels as a Discriminant Function

Depending on the application and its performance measures, one may be interested in defining quality areas for the class label determination. If there is only one boundary, then the case deteriorates to the one described in the previous subsection. In case of multiple areas, their boundaries split the space into stripes corresponding to the different quality levels, or in other words degrees of certainty; see Fig. 3. Each quality level can be mapped to a real number (a score) in [0; 1] in such a way that the higher the degree of certainty the larger the prescribed value is. The described mapping represents nothing but a discriminant function. Due to its construction the discriminant function yields a special property, which is worth to emphasize: it is monotonically increasing with respect to the base classifier scores. Thus, the monotonicity of a discriminant function is a kind of a very natural condition when the base classifiers are combined.

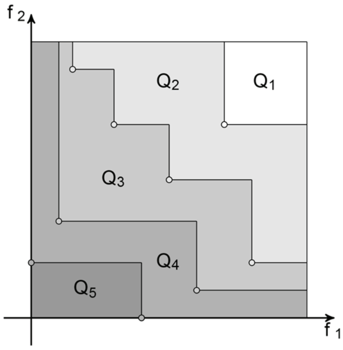


Fig. 3. Quality levels can be interpreted as values of the corresponding discriminant function.

2. Objective Function, Uniqueness of a Trained Combination and a Training Algorithm

A new approach, the trainable monotone combiner, is derived considering the monotonicity as an explicit constraint. There are different ways to measure a combiner performance. In order to design the combiner and its training procedure an appropriate objective function is required. The AUC (area under the curve) is chosen as an objective function which is to be maximized; see [1] for an extensive introduction into receiver operating characteristic (ROC) curves and the AUC concept. It is worth mentioning that the AUC is equal to the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one [2].

In the current work the decompositions of the intermediate space into quality levels of the form shown in Fig. 3 are considered. It is proved that the optimal partitioning based on a training sample is unique. Moreover, a training algorithm is provided that constructs this optimal partitioning and hence the combiner in polynomial time. For the complete theory and discussions see the full paper [3].

4. Toy Example

A prototype implementation of the proposed approach has been done. For the experiments I used a MATLAB pattern recognition toolbox PRTools [4]. A toy training set is displayed in Fig. 4 (left) and the resulting TMC layers (geodesic lines) can be seen in Fig. 4 (right). The prototype implementation and further examples are available for download on [3].

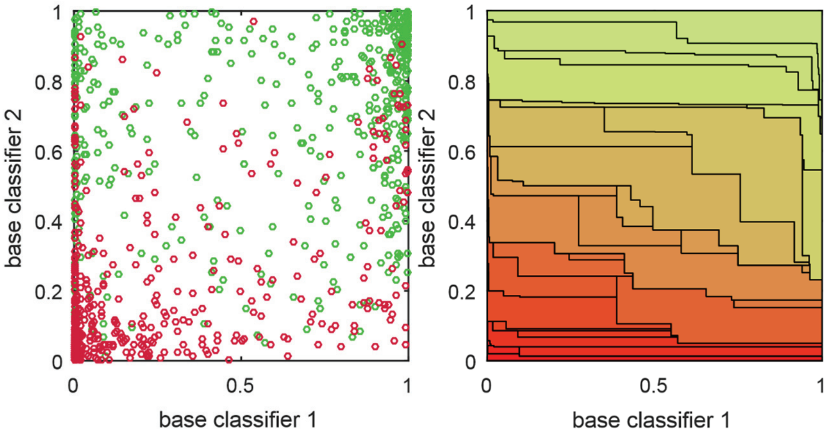


Fig. 4. TMC combiner: toy example: training set (left) and TMC layers (right).

References

[1]. T. Fawcett, An introduction to ROC analysis, *Pattern Recognition Letters*, Vol. 27, Issue 8, 2006., pp. 861-874

[2]. J. A. Hanley, B. J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, *Radiology*, Vol. 143, Issue 1, 1982, pp. 29-36.

[3]. Trainable Monotone Combiner, http:// http://www.tmcombi.org

[4]. R. P. W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. M. J. Tax, S. Verzakov, Prtools, A MATLAB Toolbox for Pattern Recognition, http://www.prtools.org

(20)

AirCSI – Remotely Criminal Investigator

P. Araújo Jr., M. Mendonça and L. Oliveira
Intelligent Vision Research Lab, Federal University of Bahia, Brazil
Tel.: +55 71 3283-9472
E-mails: pompilio.araujo@ufba.br, lrebouca@ufba.br

Summary: Once a location associated with a committed crime must be preserved, even before criminal experts start collecting and analyzing evidences, the crime scene should be recorded with minimal human interference. In this work, we introduce an autonomous system for investigation of crime scene using a drone. Our proposed intelligent system recognizes objects considered as important evidence of the crime scene, and defines the trajectories through which the drone performs a detailed search to record evidences of the scene. We used our own method, called Air-SSLAM, to estimate drone's pose, as well as proportional–integral–derivative (PID) controllers for aircraft stabilization, while flying through the paths defined by the environment recognition step. We evaluated the performance of our system in a simulator, also preparing a real-drone system to work in a real environment.

Keywords: Criminal scene investigation, Object detection, Intelligent drones, SLAM, Autonomous drone, Self localization, Object detect.

1. Introduction

When a crime is committed, no matter how careful the criminal is, evidences are spread throughout the crime scene, and must be recorded and collected by a team of experts. Evidences are not perennial, decreasing in quantity and quality as they spatially and temporally come far from the crime scene. The goal of collecting and recording evidences is to preserve the maximum amount of information so that experts, prosecutors and judges can analyze the dynamics of the facts, deciding in the courts on the culpability of those ones involved [1]. The gathering and recording of evidences are tasks of the criminal expert, who is designated by the state to analyze the site, as well as any material evidence that may clarifies the crime. The expert must use all technological resources available to store the evidence, since the fragile elements can be lost after releasing from the crime scene.

Very few works propose methods of autonomous drones to search for crime evidence. In [2], the authors make a comparison between the ways of acquiring data from the scene with laser and with images. The goal is to compare traditional methods using theodolite to geo-refer the evidence. The data collected through these 3D techniques are free of time-consuming problems and can be used at any time, while sharing between different operators. The authors propose two case studies, discussing practical aspects of data acquisition from a crime scene. However, authors perform data acquisition, manually. An environmental data acquisition system is presented in a police context [3], specifically in external environments focused on environmental crimes. Some aspects of data captured with aerial images are used, but there is no calculation of trajectory and autonomous flight. A detection of objects with images captured in a drone with an NVIDIA Jetson TX2 module for GPU processing on board the aircraft is addressed in [4]. In [5], you only look once (YOLO) method is used to detect objects, achieving the best performance at that time.

Here we introduce AirCSI, an image collection and recording system of crime scene evidence, which autonomously operates on board in an unmanned aerial vehicle (drone). The AirCSI uses a stereo camera installed on the drone, which has the goal of capturing the images for the aircraft positioning system in real time with our AirSSLAM, a simultaneous localization and mapping method [6]. A downward-facing monocular camera, also equipping the drone, is used to detect and help estimate the object coordinates at the crime scene. Although AirCSI can use any object detector as a baseline to find the evidences, in our experiments, we used YOLO-v3 [7], which was specially trained for our proposed goals. The YOLO was chosen because it uses a single CNN network to classify and locate the object, which provides faster detection.

AirCSI brings three important contributions, as follows: (i) The introduction of a method to calculate trajectories for finding objects, considering the objects already found in a crime scene, (ii) a new methodology to increase the accuracy of object detection based on multiple perspectives, and, finally, (iii) an evaluation in a realistic environment using the AirSim simulator [8].

2. Outline of the AirCSI

Air-CSI creates a coordinate system that originates from the drone starting point. Fig. 1 summarizes our proposed method described in five steps, as follows:

(1) The drone initiates the movement in the vertical direction and stabilizes at the height $h < hmax$;
(2) Using the monocular camera at the bottom of the drone, each detected object is classified as a type

of evidence, which has a relevance coefficient ρ defined by the user;

(3) Trajectory calculation is performed according to the coefficient ρ of the detected evidences; a coverage radius is created for each detected evidence and the drone should pass through the coverage area of all evidences;

(4) The control module performs stabilization and displacement of the aircraft in the trajectory defined by the system; there are eight proportional-integral-derivative (PID) controllers: two cascades in each direction of the quadrotor drone movements, one for velocity, and another one for the position;

(5) From the data collected, AirCSI creates reports with sketches, highlighted evidence and images gathered during the scanning.

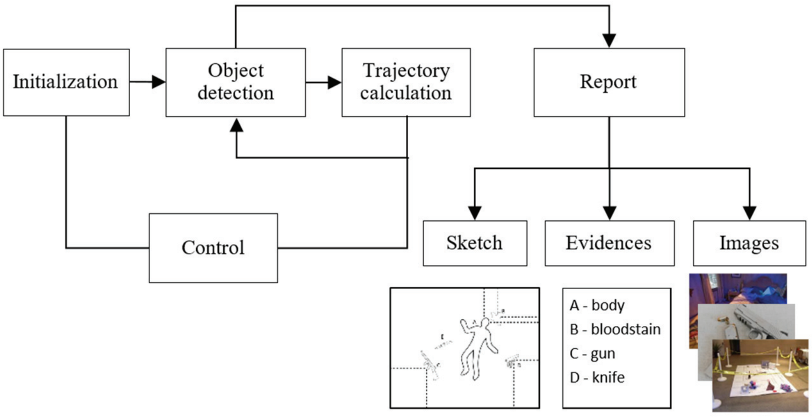


Fig. 1. Initialization – the drone takes off from a position near the crime scene, being positioned at a height **h**; **object detection** – the camera at bottom captures images and detects suspicious objects; **trajectory calculation** – a trajectory is calculated from the positioning of the detected objects, and the drone performs the trajectory for refined object detection; **report** – the result of the scan is presented in a report.

2.1. Evidence Detection

YOLO was used as a baseline detector. Although this detection method is not the most accurate, it is one of the fastest. YOLO was also the best choice, since precision has less relevance than the detection rate. This is so because the object will be detected from more than one perspective, and only objects that has their detections confirmed in all perspectives will be considered. In other words, after the first detection, object position is recorded, demanding the drone to detect the object again, in a different pose. This situation makes the object detection module to have higher mAP as the drone approaches to the object.

YOLO applies a single neural network to every whole image. The network divides the image into regions and provides bounding boxes and probabilities for each region. The bounding boxes are weighted by the predicted probabilities. YOLO was trained for the detection of the following objects: human body, revolver, pistol, machine gun and knife. The following parameters were used to train the object detector: batch = 64, momentum = 0.9 and decay = 0.0005. Images were preprocessed by changing their resolution to 608×608 from the original images acquired. To train the YOLO detector, the MS-COCO dataset with 3000 additional weapon images were used.

After detecting the objects in the scene, the object bounding box is projected onto the ground plane, providing two dimensional information of the object location (see Fig. 2). The stereo camera is used to estimate the distance between the drone and the evidence, as well as the drone and the ground. These two distances provides also an estimation of the object height.

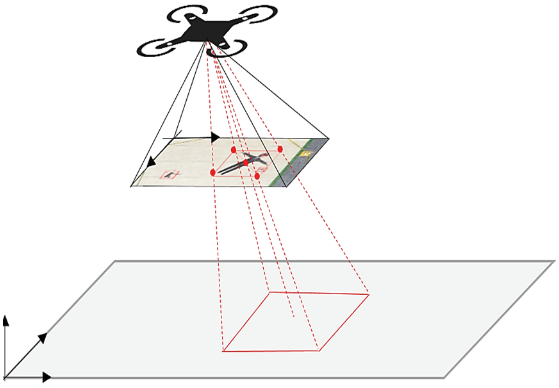


Fig. 2. The five points of the bounding box of the monocular camera image are translated to the world coordinate system by multiplying the target vector by the inverse of the pose matrix.

Our proposed system considers six degrees of freedom that determines the pose of the drone $[x\ y\ z\ \varphi\ \chi\ \psi]^T$, where x, y and z are the coordinates of the drone position, and ψ is the yaw rotation. The angles φ and χ (roll and pitch) are considered null values when the drone is in equilibrium, while these values are very low during the drone movement. These constraints are completely suitable, because the drone moves at very low velocity.

To internally represent a detected evidence, five points of the detected bounding boxes (the four vertices and the geometric center of the rectangle) are used. Each point (p_i) in the bounding box is defined with the coordinates $X_{Ci} = [x_i \ y_i \ z_i]^T$ with respect to the camera coordinate system. Using this camera pose P, points are translated to the world coordinate system X_{Wi}.

$$X_{Ci} = P^{-1} \cdot X_{Wi} \quad (1)$$

Each time an evidence is found, its position is stored and a counter is incremented. At the end of the scan, each evidence will have recorded the number of times it was detected. The accuracy of the deviation is proportional to the value recorded by the counter, because drone overlaps the evidence many times in different perspectives.

3. Self Localization

To perform the control of the drone pose in a real situation, a GPS-independent, self-locating method is exploited by using only images from the stereo camera (refer to [2] for more details on AirSSLAM). AirSSLAM relies on good features to-track (GFTT) [9] to extract keypoints, which are lately described by rotated-binary robust-independent elementary features (rBRIEF) [10, 11]. To estimate drone pose, the keypoints of the two views are matched in order to calculate the transformation matrix between two consecutive timed frames. The keypoints provide an initial map that is used as a reference to be tracked posteriorly. Air-SSLAM performs a periodic map maintenance around image patches, which are also used as quality indicators to improve keypoint tracking. An optimization procedure algorithm is applied to include new keypoints in the continuously updated map. This procedure is necessary to minimize the error between the current keypoint and the map (already inserted) keypoints. After that, the pose of the drone is periodically recalculated by a bundle adjustment optimization method. All map keypoints and the drone pose calculated are used to perform this refinement. Air-SSLAM presents a novel method of point matching, starting the search at a probable point location, then gradually increasing the search area. As each keypoint is found, the probable location for the subsequent points are updated and corrected. Air-SSLAM applies a Kalman filter to stabilize the calculated camera pose. This method allows real-time location of the drone in environment.

In the simulator, AirSSLAM is not used. Another computer runs the AirSim program, which transmits the pose to the drone onboard computer (NVIDIA Jetson TX2 [12]). AirSim is a simulator created on the Unreal Engine that offers physically and visually realistic simulations designed to operate on high frequency real-time looping hardware simulations. The AirSim was experimentally tested with a quadrotor as a stand-alone vehicle, comparing the software components with real-world flights. A change was made in the original code of the *reportState* method [12], which was carried out to transmit the drone pose, via network, with the UDP protocol in order to have faster transmission boudrates. This communication channel is also used to stream the controller commands to the simulator. Fig. 3 illustrates the configuration to perform the simulation and how it was designed to work in real situation.

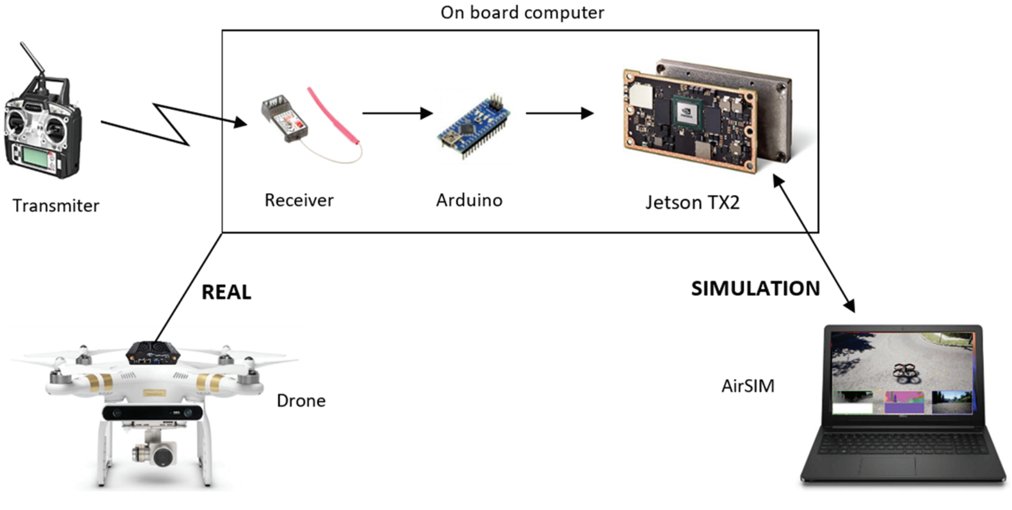


Fig. 3. Simulation tests – AirCSI runs within an NVIDIA Jetson TX2 module. Another computer runs the AirSim program, which transmits the pose to drone on board computer. Real projected situation - AirSSLAM and AirCSI run within an NVIDIA Jetson TX2 module. A transmitter and receiver as well as an Arduino are used for manual control.

4. Controlling the Drone

A double control is applied in each direction of the drone coordinates, [*x y z ψ*]. For each variable, we used two controllers, one for velocity and another for position. Disturbances influencing the position are corrected by the first controller, which does not allow a great interference on the velocity (see Fig. 4) [13]. In addition, the phase delay in the secondary part of the process is measurably reduced by the secondary loop. This improves the response of the velocity in the primary mesh [14].

The input of the controller C_V is the velocity error $e_{\dot{x}_c}$, given by

$$e_{\dot{x}_c} = \left(\frac{x_{c(n)} - x_{c(n-1)}}{T}\right) - P^{-1}\dot{x}_{ws}\,, \tag{2}$$

where $x_{c(n)}$ and $x_{c(n-1)}$ is the position of the drone in the camera coordinate system in the current and previous samplings, respectively; T is the sampling period, P is the drone pose matrix and $\dot{x}_{ws}$ is the reference velocity in the global coordinate system that is received from the position controller output. The input of the controller C_P is the position error e_{x_W}, which is defined by

$$e_{X_w} = X_w - X_{ws}\,, \tag{3}$$

where X_w is the position of the current drone and X_{ws} is the desired position.

The PID controllers are used by the transfer function:

$$u\,(t) = K_p e\,(t) + K_i \int e\,(t)\,dt + K_d \frac{de}{dt}\,, \tag{4}$$

where K_p, K_i, K_d are the proportional, derivative integral constants, respectively. The method 2p2z was implemented with sampling period of 160 ms. The output is given by:

$$y\,[n] = e\,[n]\,b_0 + e\,[n-1]\,b_1 + e\,[n-2]\,b_2 + y\,[n-1]\,, \tag{5}$$

where y[n] is the control signal at the output of the controller, and e[n] is the error in the controlled variable (position or velocity). The constants b_0, b_1 and b_2 are:

$$\begin{aligned} b_0 &= K_p + \frac{K_i \cdot T}{2} + \frac{K_d}{T} \\ b_1 &= K_p + \frac{K_i \cdot T}{2} - \frac{2 \cdot K_d}{T} \\ b_2 &= \frac{K_d}{T} \end{aligned} \tag{6}$$

The controllers were tuned by the Ziegler-Nichols closed-loop method. The values of the constants of the velocity and position controllers are listed in Table 1.

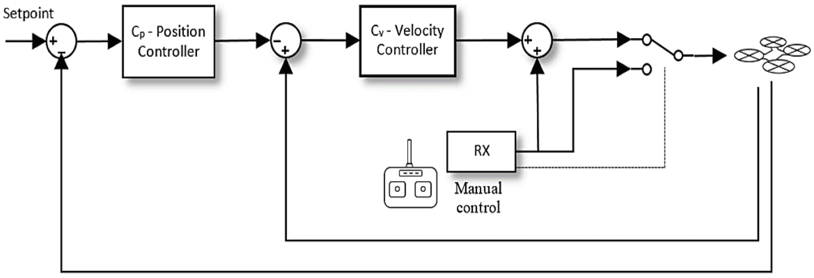


Fig. 4. Control system: the control in each direction is made by two controllers: C_V (velocity) and C_P (position). The output of the velocity controller goes into a switch that alternate the control from manual to automatic.

Table 1. Coefficients of the controllers used in the tests.

	x		*y*		*z*		*ψ*	
	Cp	*Cv*	*Cp*	*Cv*	*Cp*	*Cv*	*Cp*	*Cv*
Kp	*5*	*30*	*5*	*30*	*10*	*40*	*10*	*20*
Ki	*1*	*2*	*1*	*2*	*2*	*2*	*5*	*5*
Kd	*4*	*3*	*4*	*3*	*3*	*3*	*0.08*	*0.01*

In our experiment, a remote control is used to manually control the drone during the tuning phase, in order to avoid accidents. A switch key was implemented to switch from manual to automatic control. A routine was implemented to keep the drone in the current position whenever the key is triggered. This allows the user to set an initial pose for the drone manually.

Fig. 6 illustrates the result of the tests with the controllers running in the simulator. The controllers were evaluated by adjusting the set point with a variation of eight meters in each direction *x*, *y* and *z*, and a variation of 90º in the angle yaw (*ψ*). An average accommodation time of 20 ms in all directions was obtained. With respect to the angle yaw (*ψ*), the accommodation time was 10 s. This time can be considered as satisfactory, because a small value of velocity is needed to give time to perform the detection of objects. A large overshoot was observed only in the *x* direction.

The main reason may not have to do with the system itself, but to some delay in transmitting the position over the Ethernet network. This hypothesis was confirmed when the tests were repeated, and the problem did not occur all the time.

To stabilize the calculated camera pose, a Kalman filter was applied only on the position controllers, using the following parameters: number of states = 2, measure states = 1, and time of measurement = 0.500 s.

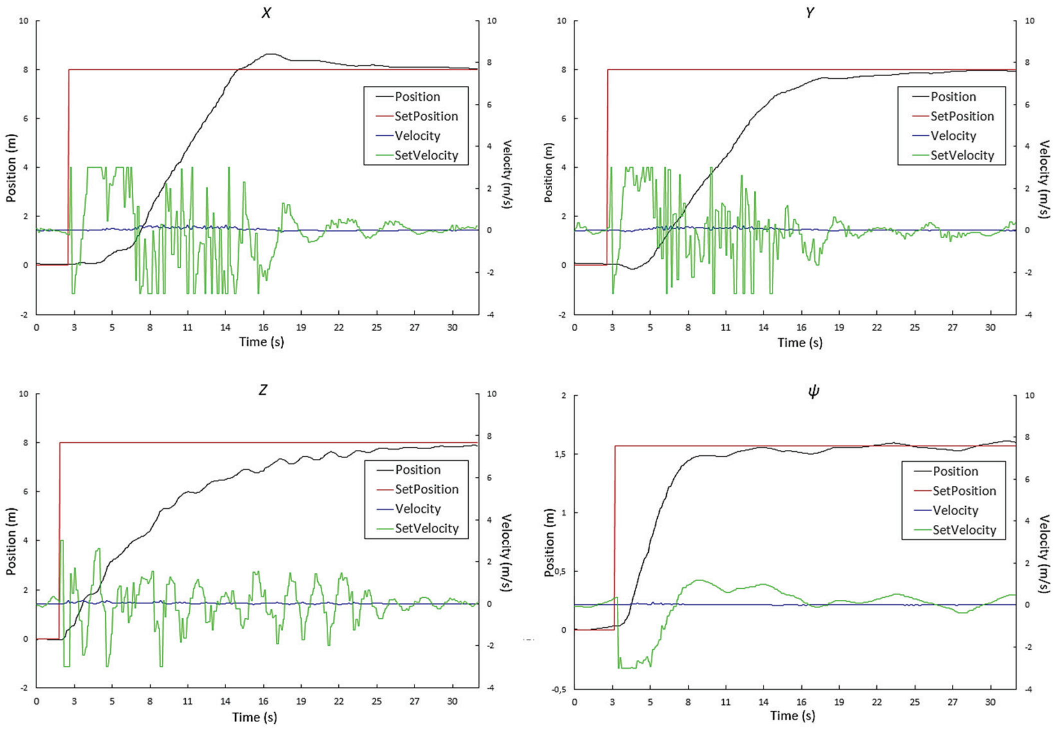


Fig. 5. The plots show the position and velocity variation over a period of 30 seconds. The average accommodation time of 20 seconds was observed in the directions *x*, *y*, *x,* and 10 seconds in the angle *ψ*.

4. Trajectory Calculation

After detecting the first evidence from a height ***h***, AirCSI flies down to perform a detailed search for more evidences. Each type of evidence has a span radius value ρ_i that defines a scan area. In the tests performed, we used the following radius for some evidence samples: human body (ρ_1 = 3 m), revolver (ρ_2 = 2 m), pistol (ρ_3 = 2 m), machine gun (ρ_4 = 2 m) and knife (ρ_5 = 1 m).

The scan is performed in-line following a path that fills the rectangle *R* that circumscribes the circle of radius ρ_i (see Fig. 6). That rectangle is calculated using the method described in [15]. This method builds a rectangle of minimum area enclosing an n-vertex convex polygon. To define the in-line scan, the drone moves following a zig-zag path, according to a sequence of points, which is defined as follows: Let V_1, V_2, V_3 and V_4 be the vertices of the rectangle circumscribing the circles, *h* the height of the drone during scanning and *θ* the horizontal aperture angle of the camera, the trajectory follows the points T_i as follows

$$T_i = V_t + \left(\frac{1}{2} + j\right) \cdot \left(\frac{V_{t+2} - V_t}{\| V_{t+2} - V_t \|}\right) \cdot h \cdot \tan tan\, \frac{\theta}{2}, \quad (7)$$

where *i* is the index of each point of the trajectory *j* is:

$$\begin{gathered} j = (i-1)\ div\ 2 \\ t = 1 + [(i-1)\ div\ 2]\ mod\ 2 \end{gathered} \quad (8)$$

With this trajectory the drone manages to sweep all areas close to the evidence, guaranteeing that there would be no point without going through the vision of the camera. The example shown in Fig. 6 shows a trajectory of a crime scene, where four evidences were found: a human body, a revolver, a knife and a machine gun.

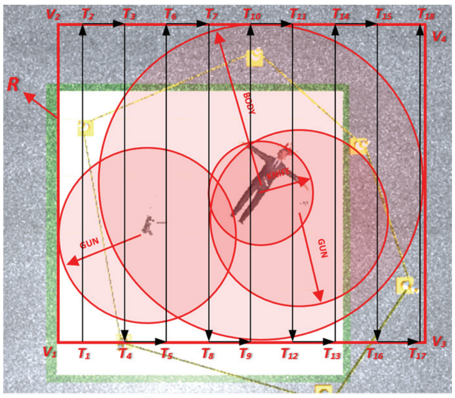


Fig. 6. Drone trajectory calculated as a function of the weight coefficient ρ_i for each evidence. The human body is more relevant than other evidences, so it has a larger scanning area.

5. Conclusion

AirCSI uses a drone to sweep an area that contains evidence of a crime. The proposed system uses YOLO as a real-time object detector to find evidences in a scene. In search for crime evidences, a low false negative value is wanted, since a human analysis will always be done by an expert after the automatic search. The system control is designed for the drone to move steadily. The cascaded controllers allowed an explicit velocity adjustment, which did not allow very fast scrolling to impair the capture of images. Thus, the found accommodation time of 20 s can be considered suitable for scanning the tested distance of 8 m. As a future work, we are working on a real implementation of AirCSI in a drone. Also, the goal of our research is to training a detector to search for other classes of evidence such as ammunition cases, projectiles, bloodstains and other types of objects used in crime.

Acknowledgements

The authors would like to thank FUNDAÇÃO DE AUXÍLIO À PESQUISA DO ESTADO DA BAHIA (FAPESB) to support with the project under grant APP0015/2016, and the Federal Police of Brazil, for the time granted to the project.

References

[1]. J. T. Fish, L. S. Miller, M. C. Braswell, E. W. Wallace Jr, Crime Scene Investigation, *Routledge*, 2003.

[2]. S. R. Bucheli, Z. Pan, C. L. Glennie, A. M. Lynne, D. P. Haarman, J. M. Hill, Terrestrial laser scanning to model sunlight irradiance on cadavers under conditions of natural decomposition, *International Journal of Legal Medicine*, Vol. 128, Issue 4, 2014, pp. 725-732.

[3]. M. Lega, C. Ferrara, G. Persechino, P. Bishop, Remote sensing in environmental police investigations: aerial platforms and an innovative application of thermography to detect several illegal activities, *Environmental Monitoring and Assessment*, Vol. 186, Issue 12, 2014, pp. 8291-8301.

[4]. N. Tijtgat, W. Van Ranst, B. Volckaert, T. Goedemé, F. De Turck, Embedded real-time object detection for a UAV warning system, in *Proceedings of the International Conference on Computer Vision (ICCV'17)*, 2017, pp. 2110-2118.

[5]. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 2016, pp. 779-788.

[6]. P. Araújo, R. Miranda, D. Carmo, R. Alves, L. Oliveira, Air-SSLAM: A visual stereo indoor slam for aerial quadrotors, *IEEE Geoscience and Remote Sensing Letters*, Vol. 14, 2017, pp. 1643-1647.

[7]. J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, Tech Report, *arXiv:1804.02767*, 2018.

[8]. S. Shah, D. Dey, C. Lovett, A. Kapoor, Airsim: High-230 fidelity visual and physical simulation for autonomous vehicles, *arXiv:1705.05066*, 2017.

[9]. J. Shi, C. Tomasi, Good features to track, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994, pp. 593-600.

[10]. E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in *Proceedings of the Intl. Conference on Comput. Vision (ICCV'11)*, 2011, pp. 2564-2571.

[11]. M. Calonder, V. Lepetit, C. Strecha, P. Fua, BRIEF: Binary robust independent elementary features, in *Proceedings of the Europ. Conference on Comput. Vision (ECCV'10)*, 2010, pp. 778-792.

[12]. NVIDIA Jetson TX2, http://www.nvidia.com/object/embedded-systems-dev-kits-modules.html

[13]. P. K. Saha, Process Control and Instrumentation, Web course, *National Programme on Technology Enhanced Learning*, 2014.

[14]. T. E. Marlin, Process Control: Designing Processes and Control Systems for Dynamic Performance, Vol. 2, *McGraw-Hill*, New York, 1995.

[15]. D. S. Arnon, J. P. Gieselmann, A Linear Time Algorithm for the Minimum Area Rectangle Enclosing a Convex Polygon, Report Number: 83-463, *Purdue University*, 1983.

(21)

Curve Registration and Mean Estimation Using Warplets

Nenad Mijatovic [1], Anthony O. Smithy [2] and Adrian M. Peterz [3]
Florida Institute of Technology, 150 W University Blvd, Melbourne, FL 32901, USA
E-mails: [1] nmijatov2005@my.fit.edu, [2] yanthonysmith@fit.edu, [3] zapeter@fit.edu

Summary: In this work, we propose a novel approach for simultaneously aligning one-dimensional time series and estimating the mean curve of the set. Our method employs compositional warplets to decompose the underlying domain, allowing for localized time warping of the curves and giving better registration accuracies. On synthetic and real data sets, our experimental results are competitive or better than other state-of-art approaches.

Keywords: Warplets, Curves registration, Mean curve.

1. Introduction

In many time series applications we often need the ability to align two curves or estimate the mean given two or more curves. Commonly, aligning different signal features is referred to curve registration or time warping. The current work presents a novel approach to curve registration and mean curve estimation that leverages the localized, compositional aspects of warplets [1]. Whereas general time warp approaches often result in overly restrictive global smoothness properties, our employment of warplets to model curve deformations provides parameterized control over local warping characteristics. This captures a richer class of deformations and allows one to more accurately register unaligned curves. Our proposed framework also extends the use of warplets to estimate the mean deformation warp for a set of unaligned curves, which consequently results in a mean curve for the group.

1.1. Related Work

Many authors have chosen different flavors of Dynamic Time Warping (DTW) for curve registration, making this method the most preferable one in speech analysis, engineering, biology and medicine, see [2] for more details. Berndt and Clifford, in their seminal work [3], solved the problem of detecting common patterns between two sequences with the use of dynamic programming formulation. The authors in [4], suggested a regularized cost function accompanied with DTW, in order to find an optimal time shift between different noisy curves.

Petitjean *et al.* in [5] suggested a DTW framework that used a global averaging strategy to determine a mean curve. The authors termed their method DTW Barycenter Averaging (DBA). In order to calculate a mean among a set of sequences, DBA was defined as a two step process: (1) DTW was performed between the mean and each individual sequence to determine associations between corresponding elements across all curves and temporary average curve and (2) each element of the mean curve was updated based on the associations assigned during step (1). The authors in [6] improved DBA using Nearest Centroid to allow meaningful averaging of warped sequences.

Functional Data Analysis (FDA) [7] enables a convenient treatment of the curves as functions, giving rise to many important features, such as derivatives, maxima, minima, slope of the curves, etc. James in [8] utilized one feature, referred as 'moments', in the FDA paradigm in order to capture local shifts between the curves. Ramsay and Li [9], approached the problem of time transformation by estimating the strictly increasing and invertable time-warping function. Their non-parametric method utilized the minimization of a distance between the curves. The authors in [10] developed a computationally effective method that estimated locally monotone transformations. Sangalli *et al.* in [11] proposed *k*-mean to reduce amplitude and phase variability. As a result, the method efficiently aligned curves from the same groups.

In [1], the authors introduced local warping component functions as warplets, denoted as τ_w, whose composition forms a global warping function, τ_c. They are similar in spirit to the wavelet and B-spline decomposition of warps. Since each warplet is defined within a specific interval, it is responsible for local function deformation in time or space, enabling to a multi-resolution warplet approach. In order to estimate warplet parameters, the authors resorted to a Bayesian inference strategy, formally implemented as a Markov Chain Monte Carlo (MCMC) sampling algorithm. The authors conveniently provided an R package (with some C functions) for multi-resolution time warping applied to functional data, refer to [12].

In other related work, Sleats *et al.* [13] postulated that each curve (time series) consists of a common mean function with added "phase" noise that is presented as a linear combination of asymmetric quadratic kernels. The authors utilized a linear regressor in order to calculate warplet parameters by using MCMC. A more geometrical approach used for functional alignment was proposed in [14]. Instead of using Euclidean or point-wise distance in the original space, this method used the square-root velocity transformations. Using this approach, the Fisher-Rao Riemannian metric in the quotient space becomes the standard `1 norm. In [15], the authors also transformed functions in the quotient space, but they went further to introduce the ambient space, in which the functions were normalized to have zero mean and unit variance. The authors applied a Bayesian approach, in which inference was carried out by the MCMC simulation and priors were included as the amount of warping. To obtain the mean estimators, the authors used Dynamic Programming (DP) in order to search for the optimal warping function. Some recent work by Panaretos and Zemel [16] proved, through experimentation, that the point warping process can be solved with the geometry of the Monge problem of optimal transportation.

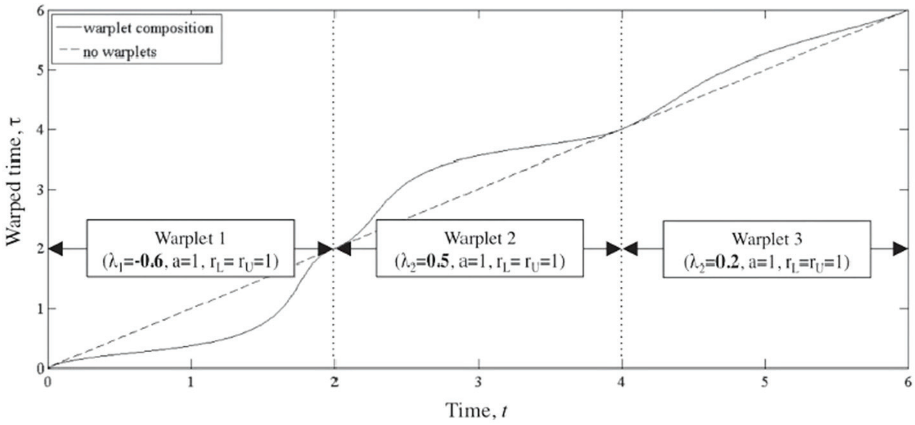


Fig. 1. Composition of three warplets with different intensity parameters $\lambda = \{-0.6, 0.5, -0.2\}$ specified in three non-overlapping intervals (with the same radius $r = 1$ and centers a at 2,3, and 5). As a result of warping, the original time t (dashed straight line) is deformed into the warped time (solid wiggly line).

The methods previously detailed utilize complex signal transformations and computational statistical models in order to calculate the amount of time and space deformation as well as the underlying mean curve. Our proposed work with compositional warplets adopts a more principled optimization framework to align curves, and extends them to the estimation of mean curve functions. Specifically, in what follows, we present a Coordinate Gradient Descent method to learn the warplet parameters. We demonstrate that out method is more efficient, in performance and implementation, than the previous frameworks that rely on expensive MCMC approaches.

2. Warplets for Curve Alignment

The warping component function [1] is defined as

$$\tau(t) = \begin{cases} a + rg\left(\lambda; \frac{t-a}{r}\right), & t \in [a-r, a+r] \\ t, & \text{otherwise} \end{cases}, \qquad (1)$$

where $\lambda \in [-1,1]$ presents the warplet intensity parameter, $a > 0$ is the center of the warplet, $r > 0$ is the warplet radius, and $g(\lambda;y) = z + \lambda K(z)$ for which z is the solution to $z - \lambda K(z) = y$. The warplet kernel $K(\cdot)$ is a symmetrical continuous function on $[-1,1]$, where the first derivative satisfies $\sup_z|K(z)| = 1$.

Based on (1), the warping component function performs dilation and compression on the closed interval $t \in [a - r, a + r]$. The larger the absolute λ value, the more compression / dilation will be performed by the warplet. The sign of the λ parameter indicates the order of warping: for positive values, the warping component function first performs dilation and then compression of the function; for negative values, the compression is done before the dilation. If $\lambda = 0$, there is no warping. We choose to use the quartic kernel defined as

$$K(z) = \begin{cases} \frac{3\sqrt{3}}{8}\left(1 - z^2\right)^2, & z \in [-1, 1] \\ 0, & \text{otherwise} \end{cases}, \qquad (2)$$

which meets two desired features: (1) the first derivative is a smooth function and (2) its support is limited.

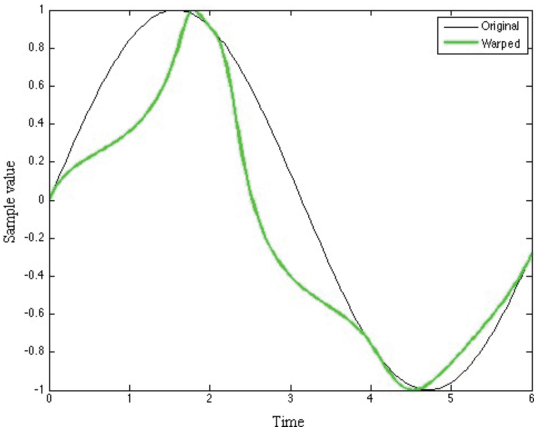


Fig. 2. Warped signal (in green) after applying the composition of three warplets from Fig. 1 on the original function (in black).

For a given function $f: [0, T] \rightarrow \mathrm{R}$, based on (1), we can introduce local dilation and compression by acting on t (real physical time or space). The warped version of $f(t)$, using the wth warping component function τ_w, becomes $f_w(t) = f(\tau_w(t)) = f(a_w, r_w \lambda_w; t)$.

Furthermore, we can expand this idea and define a composition τ_c of the warping components as

$$\tau_c(t) = \tau_W \circ \cdots \circ \tau_2 \circ \tau_1 (t), \tag{3}$$

where W is the total number of warplets. An example of warped time τ_c, using 3 different quartic warplets that act on the original time t, is illustrated in Fig. 1. As a result, the original function is deformed into its warped version, see Fig. 2.

2.1. Learn Warplets Intensity Parameters

In order to determine the warplet intensity parameters λ_w in the composition (1), assuming that the warped function $f_w(t)$ is known, we define the following optimization problem

$$\arg \min_{\{\lambda_w\}_{w=1}^{W}} \frac{1}{2} \left(\mathbf{f}_w - \hat{\mathbf{f}}_w\right)^T \left(\mathbf{f}_w - \hat{\mathbf{f}}_w\right) (\tau(\lambda_w)), \tag{4}$$

where $\mathrm{f}_w = [f_w(t_1) \ldots f_w(t_P)]^T \in \mathbb{R}^{P \times 1}$ and $\hat{\mathrm{f}}_w = [\hat{f}_w(\tau_1) \ldots \hat{f}_w(\tau_P)]^T \in \mathbb{R}^{P \times 1}$ are vector representations of the warped and estimated warped functions respectively (each vector element represents a time sample at time t_p and τ_p, for $p = 1,\ldots,P$), and P is the total number of signal points (samples). We apply the Gradient Descent (GD) method in order to find an optimal set of warplet intensity parameters $\{\lambda_w^*\}_{w=1}^{W}$. The gradient of (4), with respect to λ_w, can be calculated as

$$-\left(\mathbf{f}_w - \hat{\mathbf{f}}_w\right)^T \frac{\partial \hat{\mathbf{f}}_w}{\partial \lambda_w},$$

and rewritten as

$$-\sum_{p \in [a-r, a+r]} \left[f_w(t_p) - \hat{f}_w(\tau_p)\right] \frac{\partial \hat{f}_w(\tau(\lambda; t_p))}{\partial \lambda_w} \tag{5}$$

In order to recover the original time / space continuum, we need to find an inverse function of the composition of warplets. This can be performed by applying the opposite sign to each individual warplet, since $\tau_w(\tau_w^{-1}) = \tau_w(\tau_w(-\lambda_w)) = t$. Formally, once the optimal set of the warplet parameters is found, we can recover the original function as $\hat{f}(t) = \hat{f}_w\left(\tau_c^{-1}\right)$. Here τ_c^{-1} presents the inverse warping component function, calculated as the following composition of the warplets

$$\tau_c^{-1}(t) = \tau_W(-\lambda_W^*) \circ \cdots \circ \tau_2(-\lambda_2^*) \circ \tau_1(-\lambda_1^*)(t) \tag{6}$$

It is worth mentioning that the opposite signs of the estimated parameters are used in the composition (6) in order to find an overall inverse warp transformation. Finally, to obtain the original, unwrapped version, the inverse warp mapping is performed on the warped signal.

2.2. Mean Warplet and Curve Using CGD

Here, we extend our initial inverse warping idea from a single time series curve to a set of curves of size N that we want to register. Furthermore, our intent is to find a mean curve from the set of warped functions. We can think of the unknown warplet composition of the mean function as some function that has an underlying mean warplet, $\tau_{\$}$. We set up the following objective function

$$E = \frac{1}{2} \sum_{n=1}^{N} (\mathbf{f}_n - \boldsymbol{\varpi})^T (\mathbf{f}_n - \boldsymbol{\varpi}), \tag{7}$$

where the warped signals $\{\mathbf{f}_n\}_{n=1}^{N}$, as well as the mean curve $\boldsymbol{\varpi}$, depend on unknown warplet compositions defined by their warplet parameters, denoted as $\lambda_{n,w}$ and $\lambda_{\varpi,w}$ respectively. More specifically, an unknown warplet composition of the mean curve is defined as

$$\tau_{\$}(t) = \tau_{\$,W} \circ \cdots \circ \tau_{\$,1}(t) \tag{8}$$

Thus, the partial derivative of (7) w.r.t. the wth warplet, acting on the nth curve, with an intensity parameter $\lambda_{n,w}$, can be calculated as

$$\frac{\partial E}{\partial \lambda_{n,w}} = (\mathbf{f}_n - \boldsymbol{\varpi})^T \frac{\partial \mathbf{f}_n}{\partial \lambda_{n,w}} \tag{9}$$

Similarly, we can find the partial derivative of (7) w.r.t. the wth warplet for the mean curve, $\boldsymbol{\varpi}$, as

$$\frac{\partial E}{\partial \lambda_{\varpi,w}} = -(\mathbf{f}_n - \boldsymbol{\varpi})^T \frac{\partial \boldsymbol{\varpi}}{\partial \lambda_{\varpi,w}} \tag{10}$$

Formally, we can establish the Coordinate Gradient Descent (CGD) iterative method, as presented in Algorithm 1. First, we initialize W warplet parameters $\{\lambda_{n,w}\}_{w=1}^{W}$ for each warped signal, f_n, as well as the mean curve $\boldsymbol{\varpi}$ and its parameters $\{\lambda_{\varpi,w}\}_{w=1}^{W}$.

An initial value of the mean curve was set either by random selection of one of the curves from the input set or as a point-wise average of all input signals. Note that we can cycle through all warplet parameters from the warplet compositions for the warped functions, as well as the mean curve. Since the warplet supports are not overlapping, we can exclusively update only one λ parameter at each iteration. Once all parameters are updated using CGD with predefined step size δ, we can compute the composition of the warplets and apply its inverse to the signal, as suggested in (6). Once the norm $|\cdot|$ of the objective function (7), calculated using the updated versions curves, becomes less than ϵ, the algorithm provides a final set of aligned input signals as well as their corresponding mean curve.

Algorithm 1: Coordinate Gradient Descent for curves registration and mean curve calculation.

```
input : a set of warped curves, $\{\mathbf{f}_n\}_{n=1}^{N}$, CGD
        step size $\delta$, and error threshold $\epsilon$.
output: estimated warplet parameters and
        estimated mean curve $\hat{\varpi}$
Initialize warplet parameters;
Initialize mean curve $\varpi$;
repeat
    for $n = 1$ to $N$ do
        for $w = 1$ to $W$ do
            Compute $\frac{\partial E}{\partial \lambda_{n,w}}$ using (9);
            Update $\lambda_{n,w} \leftarrow \lambda_{n,w} - \delta \frac{\partial E}{\partial \lambda_{n,w}}$;
        end
        Compute inverse warplet composition
          $\tau_{n,c}^{-1}$ using (6);
        Warp the $n$th function $\mathbf{f}_n \leftarrow \mathbf{f}_n\left(\tau_{n,c}^{-1}\right)$;
    end
    for $w = 1$ to $W$ do
        Compute $\frac{\partial E}{\partial \lambda_{\varpi,w}}$ using (10);
        Update $\lambda_{\varpi,w} \leftarrow \lambda_{\varpi,w} - \delta \frac{\partial E}{\partial \lambda_{\varpi,w}}$;
    end
    Compute inverse mean curve composition
      $\tau_{\varpi,c}^{-1}$ using (6);
    Warp the mean curve $\varpi \leftarrow \varpi\left(\tau_{\varpi,c}^{-1}\right)$;
    Update $E$ using (7);
until $|E| < \epsilon$
```

3. Experimental Results

To demonstrate the performance of our method, we used both synthetic and real datasets.

3.1. Synthetic Dataset

The synthetic dataset contained five sinusoidal curves – each deformed in time from a standard sine wave by varying warplet intensity level and sign parameters. This caused the original sinusoidal curves to become locally deformed; they were first compressed and then dilated (or vice versa), see Fig. 3a. The na¨ıve mean curve was calculated as the point-wise average between all warped curves. Similarly, we calculated the standard deviation between all deformed curves. Refer to Fig. 3c for an illustration of the na¨ıve mean (in blue), as well as the confidence interval, constructed using the mean +/– standard deviation curves (in red). Note that this simplified method expressed large variations of the signals. In order to register the original signals, we ran our method and obtained the almost perfectly aligned signals that corresponded to underlying sinusoidals with different amplitudes, see Fig. 3b. In Fig. 3d, the mean curve (in blue) and the confidence intervals (in red), estimated using our method, are illustrated. One can readily observe that the mean curve resembles a standard sine wave. Also, the respective standard deviation intervals obtained using our method are narrower compared to the na¨ıve approach.

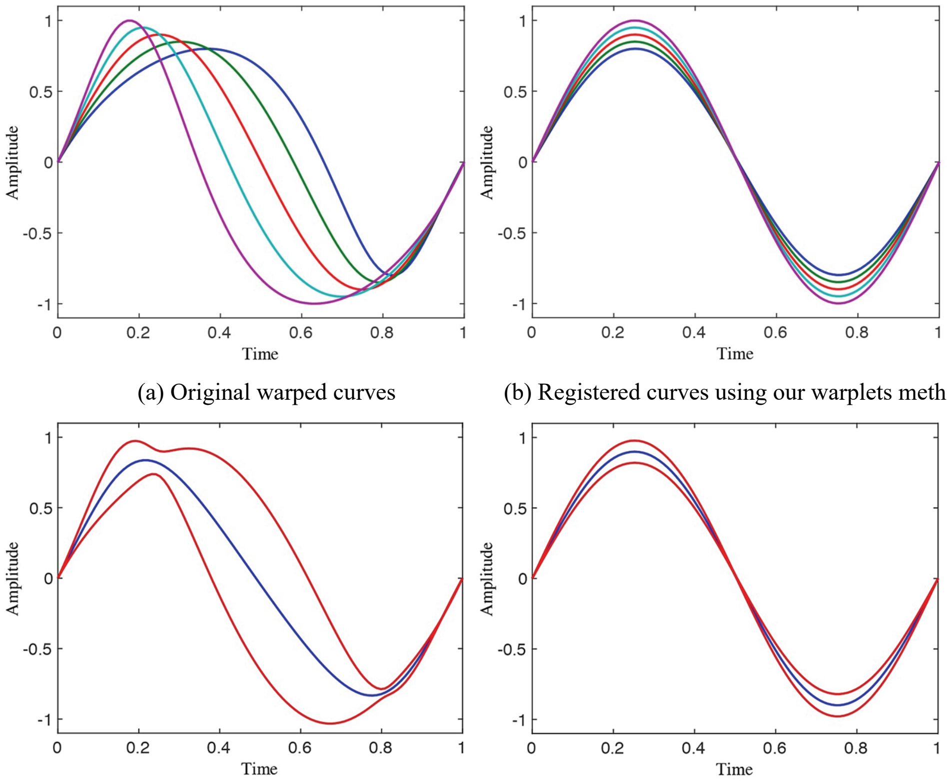
3.2. Real Dataset

For an example of our method performed on the real data, we used a Fish dataset from the UCR Time Series Data Mining Archive repository, publicly available at [17]. The UCR repository presents a growing archive of time-series datasets split into training and testing groups with a different number of curves assigned to different classes. Predominately, the datasets have been used for assessing performances between different supervised classification algorithms. The Fish dataset was generated as sequence of two-dimensional image contours matching different species. As such, the contours representing the same fish species might not be aligned due to the different sizes, translations, and rotations of different images. We used the dataset to obtain a mean curve using our proposed approach and the competing Fisher-Rao method [14], see Fig. 4. The figure presents a close overlap between the na¨ıve (in black) and the Fisher-Rao (in green) means – these are unwanted results, since the competing method was unable to improve the overall registration of shifted fish contours. On the other hand, our method using the warplets illustrates (in red) a better set of aligned curves.

4. Conclusion

In this paper, we proposed a novel approach for aligning the curves, deformed in both time and space domains, using multi-resolution warplets. The suggested method presents a simple and computationally inexpensive method that uses the Coordinate Gradient Descent (CGD) framework to first estimate the warplet parameters, then to register curves, and finally to calculate the mean curve. Our CGD iterative algorithm learns the warplet intensity parameters used to deform the curves. In each iteration, our method adjusts the warplet parameters of the mean curve and each individual curve's warplet parameters. The proposed approach benefits from the localized, non-overlapping characteristics of warplets. This results in a richer class of transformations when compared to other contemporary time warp methods, ultimately yielding more faithful registrations and mean estimations.

In our future work, we plan to propose a High Performance Computing (HPC) method that will update the respective warplet intensity parameters using multiple Graphics Processing Unit (GPU) cores at the same time. This will allow a large collection of curves to be registered simultaneously. Also, we will enhance our method to estimate other warplet parameters, such as the total number of warplets in the composition, as well as their center and radius values.



(a) Original warped curves

(b) Registered curves using our warplets method

(c) Mean +/- std. dev. point-wise registered curves

(d) Mean +/- std. dev. warplet registered curves

Fig. 3. Curves registration with mean curve and confidence interval calculations. (a) The original set of deformed sine curves. (c) The na¨ıve point-wise average curve (in blue) with mean +/- standard deviation ranges (in red). (b) The registered curves using the warplets with CGD and (d) the mean (in blue) and confidence interval curves (in red).

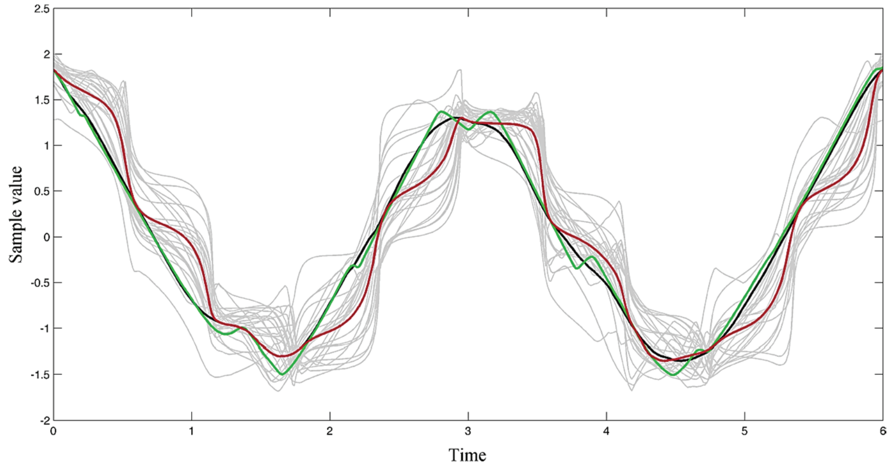


Fig. 4. The mean curve calculated using our method (in red) using real curves (in grey). The mean curves using naïve or point-wise (in black) and Fisher-Rao [14] (in green) methods are presented for comparison.

Acknowledgements

The authors acknowledge partial support from NSF grant No. 1263011 and No. 1560345. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. Also, the authors would like to thank Antonina “Nina” Mijatovic for her kind review of this work.

References

[1]. G. Claeskens, B. W. Silverman, L. Slaets, A multiresolution approach to time warping achieved by a Bayesian prior- posterior transfer fitting strategy, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 72, September 2010, pp. 673-694.

[2]. M. Muller, Dynamic time warping, *Information Retrieval for Music and Motion*, 2007, pp. 69-84.

[3]. D. J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in *Proceedings of the Knowledge Discovery and Data Workshop (KDD'94)*, Vol. 10, Seattle, WA, 199, pp. 359-3704.

[4]. K. Wang, T. Gasser, *et al.*, Alignment of curves by dynamic time warping, *The Annals of Statistics*, Vol. 25, Issue 3, 1997, pp. 1251-1276.

[5]. F. Petitjean, A. Ketterlin, P. Gancarski, A global averaging method for dynamic time warping, with applications to clustering, *Pattern Recognition*, Vol. 44, Issue 3, 2011, pp. 678-693.

[6]. F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, E. Keogh, Dynamic time warping averaging of time series allows faster and more accurate classification, in *Proceedings of the IEEE International Conference on Data Mining (ICDM'14)*, 2014, pp. 470-479.

[7]. J. Ramsay, B. Silverman, Functional Data Analysis, 2nd Ed., *Springer*, 2005.

[8]. G. M. James, *et al.*, Curve alignment by moments, *The Annals of Applied Statistics*, Vol. 1, Issue 2, 2007, pp. 480-501.

[9]. J. O. Ramsay, X. Li, Curve registration, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 60, Issue 2, 1998, pp. 351-363.

[10]. A. Kneip, X. Li, K. MacGibbon, J. Ramsay, Curve registration by local regression, *Canadian Journal of Statistics*, Vol. 28, Issue 1, 2000, pp. 19-29.

[11]. L. M. Sangalli, P. Secchi, S. Vantini, V. Vitelli, K-mean alignment for curve clustering, *Computational Statistics & Data Analysis*, Vol. 54, Issue 5, 2010, pp. 1219-1233.

[12]. L. Slaets, G. Claeskens, B. Silverman, Warping functional data in R and C via a Bayesian multiresolution approach, *SSRN Electronic Journal*, 2010.

[13]. L. Slaets, G. Claeskens, M. Hubert, Phase and amplitudebased clustering for functional data, *Computational Statistics & Data Analysis*, Vol. 56, January 2012, pp. 2360-2374.

[14]. A. Srivastava, W. Wu, S. Kurtek, E. Klassen, J. Marron, Registration of functional data using Fisher-Rao metric, *arXiv:1103.3817*, March 2011.

[15]. W. Cheng, I. L. Dryden, X. Huang, *et al.*, Bayesian registration of functions and curves, *Bayesian Analysis*, Vol. 11, November 2016, pp. 447-475.

[16]. V. M. Panaretos, Y. Zemel, *et al.*, Amplitude and phase variation of point processes, *The Annals of Statistics*, Vol. 44, Issue 2, 2016, pp. 771-812.

[17]. H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, The UCR Time Series Classification Archive, https://www.cs.ucr.edu/%7Eeamonn/time_series_data_2018/

(23)

Exploiting Higher-order Patterns for Community Detection in Attributed Graphs

L. Hu [1], **P. Hu** [1] **and T. He** [2]
[1] Wuhan University of Technology, 122 Luoshi Road, Wuhan, China
[2] The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China
E-mail: hulun@whut.edu.cn

Summary: As a fundamental task in cluster analysis, community detection is crucial for the understanding of complex network systems in many disciplines such as biology and sociology. Recently, due to the increase in the richness and variety of attribute information associated with individual nodes, detecting communities in attributed graphs becomes a more challenging problem. Most existing works focus on the similarity between pairwise nodes in terms of both structure and attribute information while ignoring the higher-order patterns involving more than two nodes. In this paper, we explore the possibility of making use of higher-order information in attributed graphs to detect communities. To do so, we first compose tensors to specifically model the higher-order patterns of interest from the aspects of network structures and node attributes, and then propose a novel algorithm to capture these patterns for the task of community detection. Experiments on two real-world datasets demonstrated the promising performance of our algorithm.

Keywords: Attributed graph, Community detection, Higher-order patterns.

1. Introduction

A graph consists of nodes that represent individual objects and edges that connect nodes to describe the relationship between them. Different types of graphs are widely used to represent the complex network systems in many practical applications. Due to the rapid development of information technology, in parallel with the ever increasing network sizes has been a concomitant increase in the richness and variety of attribute information associated with the nodes, such as social networks with user-generated content and protein-protein interaction networks with functional attributes of proteins. A formal representation describing such networks is attributed graph.

Two sources of information are available in an attributed graph, one is the structure information of graph topology and the other is the attribute information of nodes. Hence, new challenges have been raised for the problem of community detection in attributed graphs (CDAG), as there is a necessity for us to take into account these two sources of information simultaneously.

However, existing algorithms are constrained for only considering the clustering consistency between pairwise nodes and cannot extend the highly desirable consistency in both structure and attribute information to the other nodes in the same community, especially for many real-world applications, whose attributed graphs often suffer from the shortcomings of sparse structures and dispersed distributions of attribute values [1]. Obviously, there is a necessity for us to consider the higher-order patterns involving more than two nodes from the aspects of network structures and node attributes for improved performance on the CDAG problem.

In this paper, we extend previous works to attributed graphs and propose a new algorithm, namely TODA, for efficiently detecting communities based on higher-order patterns available in the structure and attribute information.

2. Methodology

Here, an attributed graph is represented with a 3-element tuple $G=\{V,E,\Lambda\}$ where $V=\{v_i\}$ is a set of all n_V nodes, $E=\{e_{ij}\}$ denotes a total of n_E links, and $\Lambda=\{\Lambda_m\}$ consists of n_Λ attributes that are available to be associated with each of nodes in V . If there is a link $e_{ij}\in E$, it means that the two nodes v_i and v_j are connected in the network.

Given an arbitrary attribute $\Lambda_m\in\Lambda$, we define its domain, i.e., $dom(\Lambda_m)$, as a set of possible values that can be taken by Λ_m and $|dom(\Lambda_m)|$ is the size of $dom(\Lambda_m)$. A $n_V\times|dom(\Lambda_m)|$ matrix $\mathbf{X}_m$ is used to describe whether an attribute value of Λ_m is taken by a node. For instance the entry $x_{ip}^m=1$ denotes that the p -th value in $dom(\Lambda_m)$ is taken by v_i .

Regarding the higher-order patterns about G , we compose them based on the structure and attribute information in G. Regarding the higher-order structural patterns, triangle motifs are preferred, as they are the fundamental units in complex networks. To represent the corresponding third-order structural patterns for triangle motifs, we use a three-mode tensor $\underline{\mathbf{T}}_S$ defined as:

$$\underline{\mathbf{T}}_{\mathrm{S}} = \left(\underline{t}_{\mathrm{S}}(i,j,k)\right), \tag{1}$$

where $1 \le i, j, k \le n_V$ and

$$\underline{t}_{\mathrm{S}}(i,j,k) = \begin{cases} 1, \ if \ e_{ij}, e_{ik}, e_{jk} \in E \\ 0, otherwise \end{cases} \tag{2}$$

Unlike most of existing algorithms that require a conversion from multi-valued attributes to binary attributes in advance, the use of tensor allows us to preserve the similarity information for each of attributes in Λ . To indicate the similarity among nodes v_i , v_j and v_k in terms of Λ_m , a five-mode tensor $\underline{\mathbf{T}}_{\mathrm{A}} = \left(\underline{t}_{\mathrm{A}}(m,i,j,k,p)\right)$ is adopted. The entry at index (m,i,j,k,p) is defined as:

$$\underline{t}_{\mathrm{A}}(m,i,j,k,p) = \begin{cases} 1, \ if \ x_{ip}^m, x_{jp}^m, x_{kp}^m = 1 \\ 0, otherwise \end{cases} \tag{3}$$

Since all entries in $\underline{\mathbf{T}}_{\mathrm{S}}$ and $\underline{\mathbf{T}}_{\mathrm{A}}$ involve three nodes, we incorporate these higher-order patterns into a second-order Markov chain. The transition probability tensor of this chain is denoted as $\underline{\mathbf{P}} \in \Re^{n_V \times n_V \times n_V}$, which is a three-mode tensor. In $\underline{\mathbf{P}}$, each entry $p(i,j,k)$ is the probability of moving to the node v_i depends on the current node v_j and the previous node v_k .

To approximate the second-order Markov chain, an equivalent first-order Markov chain can be derived from the stationary distribution of the spacey random walk [2]. Assuming that $\mathbf{M}$ and $\mathbf{x}$ are the transition matrix of the first-order Markov chain and the corresponding stationary distribution respectively, the equations of $\mathbf{M}$ and $\mathbf{x}$ can be obtained with the use of an iterative fixed-point algorithm. Hence, the stochastic process of the first-order Markov chain can be determined and we then solve the CDAG problem under the framework of spectral clustering.

3. Experiments

To evaluate the performance of TODA, we have performed a series of extensive experiments on two real-world attributed graphs, i.e., Cora and Twitter datasets. The Cora dataset is a citation network where the attributes are the keywords of publications while the Twitter dataset is a social network with profile information. For the purpose of benchmarking, TODA has been compared with several state-of-the-art algorithms specifically developed for solving the CDAG problem and they were CODICIL [3], GBAGC [4] and niMM [5]. Among them, CODICIL is a distance-based algorithm while the other three algorithms are model-based. The experiment results are presented in Table 1. The promising performance of TODA is an indication to the rationality behind the use of higher-order patterns in the structure and attribute information of attributed graph.

Table 1. Performance EVALUATION ON REAL-WORD ATTRIBUTED GRAPHS.

	Cora Dataset		**Twitter Dataset**	
	NMI	***Accuracy***	***NMI***	***Accuracy***
TODA	**0.41**	0.54	**0.31**	**0.69**
CODICIL	0.37	**0.55**	0.25	0.57
GBAGC	0	0.3	0.16	0.63
niMM	0.01	0.25	0.22	0.52

4. Conclusion

In this work, we developed a tensor-based community detection algorithm (TODA) for attributed graphs with the use of the higher-order patterns. The introduction of higher-order patterns ensures the high consistence in terms of both structure and attribute information for more than two nodes (i.e., three nodes in our work), thus improving the performance of community detection as indicated by the experiment results of TODA.

References

[1]. A. Clauset, M. E. Newman, C. Moore, Finding community structure in very large networks, *Physical review E*, Vol. 70, Issue 6, 2004, 066111.

[2]. T. Wu, A. R. Benson, D. F. Gleich, General tensor spectral coclustering for higher-order data, in *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS'16)*, Barcelona, Spain, 5-10 December 2016, pp. 2559-2567.

[3]. Y. Ruan, D. Fuhry, S. Parthasarathy, Efficient community detection in large networks using content and links, in *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*, Rio, Brazil, 13-17 May 2013, pp. 1089-1098.

[4]. Z. Xu, Y. Ke, Y. Wang, H. Cheng, J. Cheng, GBAGC: A general Bayesian framework for attributed graph clustering, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Vol. 9, Issue 1, 2014, 5.

[5]. X. Fan, R. Y. Da Xu, L. Cao, Y. Song. Learning nonparametric relational models by conjugately incorporating node information in a network, *IEEE Transactions on Cybernetics*, Vol. 47, Issue 3, 2017, pp. 589-599.

(24)

A Novel Automatic Video Magnification System

En-Jian Cai [1], Dong-Sheng Li [2] and Jie-Zhong Huang [1]
[1] State Key Laboratory of Coastal & Offshore Engineering, Dalian Univ. of Technology, Linggong RD.2,
Dalian 116023, China
[2] Guangdong Engineering Center for Structure Safety and Health Monitoring, Department of Civil
and Environmental Engineering, Shantou University, Guangdong, 515063, China
Tel.: +86 (411) 84708402, fax: +86 (411) 84708501
E-mail: dsli@dlut.edu.cn

Summary: Video magnification techniques may assist to pick up important and informative subtle movements invisible to the naked eyes. However, existing amplification methods show weakness in the isolation of possible severe corruptions, and video temporal variation information are also not sufficient so false manual adjustments may be introduced. This paper presents an advanced video amplification technique to counter the above issues. Firstly the original video is modeled as a sparsely corrupted observation tensor, the alternating direction method of multipliers (ADMM) and tensor singular value decomposition (tensor-SVD) are combined to solve the convex optimization problem. Then followed by spatial decomposition like the condition in traditional amplification methods, what's more, the optimal Wiener filter and state-space model are applied to remove residual dense spatial corruptions and update the purely temporal variations. Finally the magnified signal is added back to the original video and the spatial pyramid is collapsed to obtain the final video. Our experimental results demonstrate the superior performance of the proposed magnification method for obtaining significant video magnification even for very small variations as well as reducing artefacts compared to other existing methods.

Keywords: Tensor, Spatio-temporal analysis, State-space model, Wiener filter, Video magnification.

1. Introduction

The human visual system has limited sensitivity to many spatiotemporal events, but these variations may contain useful information which can be used in many applications. For example, human wrist vessels exhibit small periodic motion with blood circulation, although this motion is invisible to the naked eyes, a person's heart rate can be judged directly by this information [1]. The cardiac activity can also be measured by analyzing the subtle head motion resulting from the cardiac cycle of blood from the heart to the head via the carotid arteries [2]. What's more, some damage information of the building structure can be detected by measuring some small vibrations in video [3]. As a kind of effective techniques that can assist people to identify important and informative subtle movements invisible to the naked eyes, video magnification analyzes these signals existing in video and amplify them to reveal imperceptible changes, therefore arousing wide concern in the development of video amplification researches and gradually promoting its wider applications both in biomedical and engineering.

Existing video motion magnification methods can be mainly classified into two categories: Lagrangian perspective and Eulerian perspective. In Lagrangian processing [2], the feature points of the video signal are firstly clustered and tracked, and then the amplitude of motion is enlarged. However, this method has high computational complexity (10 h at that time) and the related parameters are difficult to control, which easily introduces artefacts and restricts its applications. Compared to Lagrangian perspective, Eulerian approaches do not estimate changes explicitly. Instead, they amplify video changes by processing and enhancing the temporal variations of signal that evolve in frame over time, which is more robust and motivates the development of video magnification to reveal unperceivable video information. Wu et al [3] used a spatial decomposition motivated by the first-order Taylor expansion to analyze all the pixels' value over time, followed by separating the frequency band of interest and then enhancing it, thereby eliminating the need for costly optical flow computation and achieving superior result of amplifying the motion information, but it only supports small magnification factors at high spatial frequencies and the noise level in original video will be linearly increased with increasing magnification factor. Wadhwa et al [4] used the complex steerable pyramid to extract a phase-based representation and then amplify the temporal phase variations, thus effectively suppressing noise and supporting a larger magnification. To overcome the shortcoming of computational complexity in complex steerable pyramid, Wadhwa et al [5] used the Riesz pyramid to reduce over-completeness and execution time, but the Riesz pyramid fails to maintain the power of the video signal, which will cause some tiny artefacts. Our amplification technique belongs to the Eulerian perspective, but our decomposition can separate the low multi-rank subtle signal from corrupted video, which avoids fewer edge artefacts and generates better noise characteristics.

One key component of the previous video magnification techniques is the temporal filtering over the representations, which assists to isolate changes of interest and to prevent noise from being amplified. Wu et al [4] and Wadhwa et al [5] mainly established standard frequency bandpass filters in amplification processing. Their techniques achieve high-quality results, but easily suffer from degraded quality when false manual adjustments are introduced. To solve the problem of the determination of frequency band parameter, Sushma et al [6] proposed a semi-automated magnification method automatically employing the information obtained from the first two frames of video, but only two frames' information is not enough. To simultaneously handle the large movements in video, Elgharib et al [7] used affine transformation to model large motions, while Zhang et al [8] used a different temporal processing equivalent to a second-order derivative. What's more, Xu et al [9] employed depth cameras and bilateral filters such that the magnification can be applied on all pixels located at the same depth to find the pixels whose changes should be magnified, but the need for depth information is introduced. With the development of deep learning, TH et al [10] used deep convolutional neural networks (CNN) to learn all the filters directly from examples and then amplify all the learned motions. It did not need temporal filter at all, but almost 200,000 marked images are requisite to train the CNN and the running time is too long. On the other hand, our method achieves significant amplification quality by combining the Wiener filter and state-space model, which helps to remove some residual dense spatial corruptions and iteratively estimate the purely temporal variations, and simultaneously avoid inaccurate manual adjustments.

The contributions of this paper are as follows: 1) A new magnification technique is proposed that can optimally identify both dense and sparse corruptions, and then efficiently remove it, thus magnifying subtle changes at a deeper level compared to the previous methods. 2) We show that the temporal variations can be recursively predicted, which overcomes the requirement of manual adjustments in traditional processing and achieves significant video magnification. 3) The proposed method outperforms relevant video magnification techniques both in observed output quality and in the quantitative evaluation.

2. Video Automatic Magnification

In this section, the novel video magnification processing is proposed and the main algorithms are also discussed. The specific framework can be briefly summarized as follows: Firstly the original video is modeled as a sparsely corrupted observation tensor, the alternating direction method of multipliers (ADMM) [11] and tensor singular value decomposition (tensor-SVD) are combined to recursively solve the convex optimization problem of low multi-rank component recovery. Then followed by decomposing the processed video into different spatial frequency bands, which makes temporal changes with low amplitude, and lower spatial frequencies can be modeled based on fluid flow analysis. What's more, the optimal Wiener filter is employed to remove subtle residual dense spatial corruptions, and the temporal variations are directly predicted by state-space model to avoid inaccurate manual adjustments and generate a significant amplification result. Finally, the subtle signals of interest are extracted via multiplying by a magnification factor, the amplified signal is added back to the original video and the spatial pyramid is collapsed to obtain the final video.

2.1. Low Tensor-Multi Rank Recovery

The low rank component identification is the primary task of our video processing due to the high temporal correlation contained in, and this task can be achieved by modeling the original video signal as a sparsely corrupted observation tensor. Thus generating the expression of third-order tensor T:

$$\mathrm{T} = \mathrm{L} + \mathrm{S}, \tag{1}$$

where L represents low tensor-multi-rank and S denotes the sparse tensor which is assumed as tube-wise sparse in this paper. To isolate the low rank from the sparse components given the observation T, the following optimization problem is obtained:

$$\begin{gathered} min\|L\|_{TNN} + \lambda\|\mathrm{S}\|_{1,1,2}, \\ \mathrm{s.t.T} = \mathrm{L} + \mathrm{S}, \end{gathered} \tag{2}$$

where TNN denotes the tensor-nuclear-norm, and the specific proof that TNN is a valid norm which can be found in [12]. The parameter $\lambda > 0$ and $\|\mathrm{S}\|_{1,1,2}$ for 3-D tensors can be defined as $\sum_{i,j}\|\mathrm{S}(i,j,:)\|_F$ which is a Matlab notation.

The task in equation (2) is to automatically locate pixels and recover the video. Although this may be done by processing each frame but if the video features and noise artefacts are aligned, it is needed to detect efficiently the noise and then estimate the corrupted video component.

The ADMM algorithm is employed in order to solve the convex optimization problem of equation (2), thus generating the following recursion:

$$\begin{gathered} L^{k+1} = \arg{}^{min}_{\;L}\, \|L\|_{TNN} + \\ + \tfrac{\rho}{2}\|L + S^k - T + W^k\|_F^2, \end{gathered} \tag{3}$$

$$\begin{gathered} S^{k+1} = \arg{}^{min}_{\;S}\, \|S\|_{1,1,2} + \tfrac{\rho}{2}\|L^{k+1} + S - \\ -T + W^k\|_F^2, \end{gathered} \tag{4}$$

$$W^{k+1} = W^k + L^{k+1} + S^{k+1} - T \tag{5}$$

In the above equations, $\mathrm{W} = \rho\mathrm{Y}$ and $\mathrm{Y} = P_\Omega(T)$, where P is the orthogonal projector onto the span of tensors vanishing outside of Ω. The TNN [12] denoted by $\|L\|_{TNN}$ can be defined as the sum of the singular values of all the frontal slices of parameter $\hat{L}$ is a norm and can also be considered as the tightest convex relaxation to l_1 norm of the tensor multi-rank.

Algorithm 1. Tensor-SVD.

```
Input: Sparsely corrupted observation tensor $T \in R^{n_1 \times n_2 \ldots \times n_3}$, extended parameter $\rho = n_3 n_4 \ldots n_N$.
For i = 3,…,N do
        $F \leftarrow \mathrm{fft}(T, [], i)$;
End
For i = 1,…, ρ do
        $[U, S, V] = SVD(F(:,:,i))$;
        $\hat{U}(:,:,i) = U; \hat{S}(:,:,i) = S$;
        $\hat{V}(:,:,i) = V$;
End
For i = 3,…, ρ do
        $U \leftarrow \mathrm{ifft}(\hat{U}, [], i); S \leftarrow \mathrm{ifft}(\hat{S}, [], I)$;
        $V \leftarrow \mathrm{ifft}(\hat{V}, [], i)$
Output: Decomposition matrices $\{U, S, V\}$.
```

To provide a solution to equation (3), we should firstly transform it into the Fourier domain along the third dimension, thus generating the following expression:

$$\hat{L}^{k+1} = \arg\overset{min}{\hat{L}} \left\|\mathrm{blkdia}g(\hat{L})\right\|_* + \\ +\frac{\rho}{2}\left\|\hat{L} - (\hat{T} - \hat{S}^k - \hat{W}^k)\right\|_F^2, \tag{6}$$

where $blkdiag(\hat{L})$ represents the l_1 norm of the tensor multi-rank, for which the tightest convex relaxation is the the nuclear norm of $blkdiag(\hat{L})$ which is TNN of L, thus $blkdiag(\hat{L})$ which is a block diagonal matrix can be defined as:

$$blkdiag(\hat{L}) = \begin{bmatrix} \hat{L}^{(1)} \\ \hat{L}^{(2)} \\ . \\ . \\ \hat{L}^{(n_3)} \end{bmatrix}, \tag{7}$$

where $\hat{L}^{(i)}$ denotes the i_{th} frontal slice of $\hat{L}$, and $i = 1,,2, \ldots, n_3$. According to the particular format in equation (6), it can be broken up into n_3 independent minimization problems. Let $\hat{L}^{k+1,(i)}$ represent the i_{th} frontal slice of $\hat{L}^{k+1}$. Similarly define $\hat{T}^{(i)}$, $\hat{S}^{k,(i)}$ and $\hat{W}^{k,(i)}$, thus separating equation (6) as:

$$\hat{L}^{k+1,(i)} = \arg\overset{min}{Z} \|Z\|_* + \\ +\frac{\rho}{2}\left\|Z - (\hat{T}^{(i)} - \hat{S}^{k,(i)} - \hat{W}^{k,(i)})\right\|_F^2 \tag{8}$$

For $I = 1,,2, \ldots, n_3$, we can calculate each i_{th} frontal slice of $\hat{L}^{k+1}$ through equation (8). Now note that, if $USV^T = (\hat{T}^{(i)} - \hat{S}^{k,(i)} - \hat{W}^{k,(i)})$ is the SVD of $(\hat{T}^{(i)} - \hat{S}^{k,(i)} - \hat{W}^{k,(i)})$, then the solution to (8) is $UD_\tau(S)V^T$, where $D_\tau(S) = diag(S_{i,i} - \tau)_+$ for some positive constant $\tau > 0$ and "+" denotes the positive part. It is equivalent to multiplying the parameter $\left(1 - \frac{\tau}{S_{i,i}}\right)_+$ to the i_{th} singular value of S, thus we can calculate each frontal slice of $\hat{L}^{k+1}$ by employing this shrinkage [13] on each frontal slice of $(\hat{T}^{(i)} - \hat{S}^{k,(i)} - \hat{W}^{k,(i)})$. Now let $U * S * V^T = (T - S^k - W^k)$ be the tensor- SVD [14] (see Algorithm 1) of $(T - S^k - W^k)$ and $\hat{S}$ denotes the Fourier transform of S along the third mode. Thus each component of the singular tubes of $\hat{L}^{k+1}$ can be considered as the result of multiplying every entry $\hat{S}(i,i,j)$ with $\left(1 - \frac{\tau}{\hat{S}(i,i,j)}\right)_+$ for $\tau > 0$. Since this process is implemented in the Fourier domain, it is equivalent to convolving each tube parameter S(I, I, ☺ of S with a real valued tubal vector $\vec{\tau}_i$ which denotes the inverse Fourier transform of the vector $[\left(1 - \frac{\tau_i(1)}{\hat{S}(i,i,1)}\right)_+, \left(1 - \frac{\tau_i(2)}{\hat{S}(i,i,2)}\right)_+, \ldots, \left(1 - \frac{\tau_i(n_3)}{\hat{S}(i,i,n_3)}\right)_+]$. What's more, the above operation can also be captured by $S * J$, where J denotes the f-diagonal tensor with i_{th} diagonal tube to be $\vec{\tau}_i$. Thus generating the expression:

$$L^{k+1} = U * (S * J) * V^T \tag{9}$$

What's more, the closed solution to equation (4) can be generated by:

$$S^{k+1}(i,j,:) = \\ = \left(1 - \frac{\lambda}{\rho\|S^k(i,j,:)\|_F}\right)_+ S^k(i,j,:) \tag{10}$$

In summary, the above ADMM algorithm and tensor-SVD can be combined to separate the sparse noise from the original video, and our corruptions analysis will be showed in Section 3.

2.2. Spatio-temporal Automatic Processing

As analyzed above, the initial part of our processing in Section 2.2.1 mainly utilizes both the ADMM algorithm and tensor-SVD to efficiently isolate the stubborn sparse random corruptions existing in original video, but there are always some residual dense spatial corruptions slightly affecting the quality of amplification. What's more, the lack of temporal selection ability will also cause the loss of requisite signal information.

2.2.1. Processing Problem Statement

In this section, the relationship between spatio-temporal processing and magnification will be firstly explained, and the following theoretical derivation is mainly formalized by a special case of one-dimensional video signal for easy readability.

For input image signal $I(x,t)$ at position x and time t, the case of linear video magnification [3] is firstly considered due to the ability of exaggerating both motions and color changes at fixed positions. We assum a displacement function $\delta(t)$, an initial function $L(x) = \mathrm{I}(x, 0)$, thus the function can be expressed as:

$$\mathrm{I}(x, t) = L\big(x + \delta(t)\big), \tag{11}$$

where $L(x)$ denotes the recovered low multi-rank component in original video, the goal of video magnification is to enlarge the temporal changes α times, so the amplified signal can be represented as:

$$\tilde{\mathrm{I}}(x, t) = L\big(x + (1 + \alpha)\delta(t)\big) \tag{12}$$

Then followed by decomposing the signal at time t by a first-order Taylor series expansion around x and adding the temporal bandpass filter $B_\omega(x, t) = \omega\delta(t)$, thus obtaining the final magnification expression:

$$\tilde{\mathrm{I}}(x, t) = L(x) + (1 + \alpha)B_\omega\big(\delta(x, t)\big) \tag{13}$$

But all above equations do not consider the influence of the residual dense spatial noise that can also change with time t, so the modified equation should be expressed as:

$$\tilde{\mathrm{I}}(x, t) = L(x) + (1 + \alpha)B_\omega\big(\delta_o(x, t) + \\ + \delta_n(x, t)\big), \tag{14}$$

where $\delta_o(x, t)$ is the requisite signals' displacement function at position x and time t, and $\delta_n(x, t)$ is the residual dense spatial corruptions' displacement function at position x and time t.

It is obvious that the following amplification quality is mainly depended on two parameters: B_ω and $\delta_n(x, t)$. Thus our amplification technique aims at removing the residual dense spatial corruptions and iteratively estimating the purely temporal variations via exploiting both the Wiener filter and state-space model.

2.2.2. Dense Corruptions Optimal Removal

In this section, we aim at removing the subtle residual dense spatial corruptions in video, and different from the previous model, the corrupted video is modeled as a Gaussian distribution model. What's more, the optimal Wiener filter is employed to achieve this task for its optimal property in minimizing the mean squared error between the original and the recovered signal.

The Wiener filter is widely used in signal processing, and its purpose is to optimally remove the amount of corruptions in a signal, which is achieved by comparing the original signal with an estimation of a desired noiseless signal. Wiener filter assumes its input signal to be wide-sense stationary and takes a statistical approach to solve its goal of removing the corruptions from video signal. Wiener filter works mainly by linear time-invariant filtering of an observed noisy process, simultaneously assuming signal and the additive corruptions as wide sense stationary with known spectral characteristics. What's more, the Wiener filtering scheme runs on minimizing the mean square error between the estimated signal and the desired signal, and the main purpose of the Wiener filter is statistically and optimally estimate an unknown signal, it employs a related signal as an input and filters that known signal to generate the estimated signal as an output. The wiener filter in Fourier domain can be obtained as follows:

$$W(f_1, f_2) = \frac{\mathrm{H}(f_1, f_2)\mathrm{S}_{xx}(f_1, f_2)}{|\mathrm{H}(f_1, f_2)|^2\mathrm{S}_{xx}(f_1, f_2) + \mathrm{S}_{nn}(f_1, f_2)}, \tag{15}$$

where $\mathrm{S}_{xx}(f_1, f_2)$ and $\mathrm{S}_{nn}(f_1, f_2)$ are respectively the power spectra of the original image signal and additive corruptions, $\mathrm{H}(f_1, f_2)$ denotes the blurring filter [15, 16].

The Wiener filter has two main parts: an inverse filtering part to deblur the signal and a corruption smoothing part to remove the noises. To implement the Wiener filter in practice, the signal X(k) is considered which has been corrupted by a white Gaussian noise N(k), σ_n^2 denotes the noise variance. The output signal can be denoted by Y(k), and if X(k) is an independent and identically distributed Gaussian noise with mean μ_k and variance σ_x^2, then the estimation filter H(f) can be estimated directly by:

$$H(\mathrm{f}) = \frac{\mathrm{S_{xy}(f)}}{\mathrm{S_y(f)}} = \frac{\mathrm{S_x(f)}}{\mathrm{S_y(f)} + \mathrm{S_n(f)}} \tag{16}$$

Thus the optimal estimate of $X(k)$ can be computed by the observations $Y(k)$:

$$X(k) = \frac{\sigma_x^2}{(\sigma_x^2 + \sigma_n^2)}\big(Y(k) - \mu_y\big) + \mu_x \tag{17}$$

Since the parameter N(k) is zero mean, $\mu_x = \mu_\mathrm{y}$.

In addition, this problem can be solved by dividing the image signal in some small blocks and then construct a different filter for each block, and we consider a 5 × 5 neighborhood around a pixel in a given frame in this section.

2.2.3. Temporal Variations Prediction

As mentioned above, the state-space model is established to automatically estimate the purely temporal variations, and the concrete implementation process mainly employs the Kalman filter to calculate the estimated frame value and the measured value in each iteration.

The temporal video signal value in each iteration can be considered as a linear combination of its previous value and the process noise. What's more, the measured temporal value that we obtain is a linear combination of the frame value and the measurement noise. The purpose of Kalman filter in this algorithm is to find the most optimum averaging factor for each consequent state. The estimates are updated employing a weighted average, and more weight can be obtained to estimates with higher certainty, thus the Kalman filtering can converge the recovered frame value to the original value over the time.

The Kalman Filter can be expressed by the following equation:

$$X_i = K_i Z_i + (1 - K_i) X_{i-1} \tag{18}$$

where X_i denotes the current estimation of the ith frame signal and X_{i-1} represents the previous estimation of frame signal, K_i is the Kalman gain and Z_i is the measurement value.

In addition, the equation (18) can further be divided into two different equations:

$$X_i = X_{i-1} + w_{i-1} \tag{19}$$

$$Y_i = X_i + v_i, \tag{20}$$

where w_{i-1} denotes the process noise and v_i represents measurement noise in current frame, Kalman filtering algorithm obtains the best estimate by the forward recursion method, and the specific algorithm steps applied in our video amplification are concluded in Algorithm 2.

```
Algorithm 2. Temporal Variations Prediction.
Input: Low mutil-rank matrices: $L_1, L_2, \ldots, L_{n_3}$, filter gain: $G = 0.5 \; or \; 0.8$, noise variance: $0 < V < 0.1$.
Let prediction seed: $I_2 = L_1$, error seed: $E_2 = V$.
For $i = 2, \ldots, n_3 - 1$ do
    Compute the Kalman gain:
    $K_i = E_i/(E_i + V)$.
    Update the prediction with $(M_i \leftarrow L_{i+1})$ measurement:
    $I_i = G * I_i + (1.0 - G)M_i + K_i(M_i - I_i)$.
    Update the variance estimate:
    $E_i = E_i(1.0 - K_i)$.
    Predict the next image:
    $I_{i+1} = I_i$.
    Predict the variance:
    $E_{i+1} = E_i$.
End
Output: New matrices $\{I_1, I_2, \ldots, I_{n_3}\}$.
```

3. Experimental Results

The presented technique is demonstrated on real videos as well as synthetic ones with dense or sparse corruptions added by user, we mainly carry out two comparisons including both motion amplification and color amplification. The motion part is compared with Wu et al and Wadhwa et al due to the superior performance of noise resistance generated by Laplacian or Fourier pyramids, while the recently proposed techniques [7-10] mainly handle large motions existing in video, and the color part is mainly compared with Wu et al because of the poor performance of complex steerable pyramid in color processing.

3.1. Motion Amplification Analysis

In this section, we apply the video magnification to the throat (Fig. 1) to demonstrate the superior performance of our technique. In addition, some sparse and dense corruptions are added to the original video to give an intuitive comparison. Our interested signal commonly suffers from a degree of noise when the magnification is implemented, especially with a large magnification factor.

As the results in Fig. 1 showed, both sparse and dense corruptions existing in original video appear different trends when various amplification techniques are applied. According to the amplification result generated by Wu et al in (a), the traditional nonautomatic linearly magnification can not efficiently isolate both stubborn dense and sparse corruptions, thus causing the significant artefacts and the loss of some requisite motion information. While the technique proposed by Wadhwa et al generates a better amplification result than Wu et al in the isolation of corruptions, but the shortcoming is that it causes more loss of the requisite motion components in original signal due to the false filtering by complex steerable pyramid. Different from the above methods, the amplification result in (c) avoids fewer edge artefacts and generates better noise characteristics, which shows that our technique can efficiently isolate both dense and sparse corruptions, what's more, the automatic selection steps can avoid the loss of some requisite motion information.

3.2. Color Amplification Analysis

In this section, the blackman video (Fig. 2) was amplified in color case by Wu et al and our automatic processing. As shown in Fig. 2, the video amplified by Wu et al generates plenty of edge artefacts which submerges the original signal like the motion amplification case, and the video processed by our technique generates significant noise characteristics.

4. Conclusions

This paper proposes a novel amplification technique to counter two issues existing in current amplification methods: Low performance of noise handling and the lack of automatically identifying temporal variations. The original video is modeled as a sparsely corrupted observation tensor, the ADMM algorithm and tensor-SVD are combined to solve the convex optimization problem. Different from the traditional amplification methods, the optimal Wiener filter and state-space model are applied to remove residual dense spatial corruptions and update the purely temporal variations. Our experimental results show that the proposed method can obtain significant motion magnification even for very small motions as well as reducing artefacts compared to other existing methods.

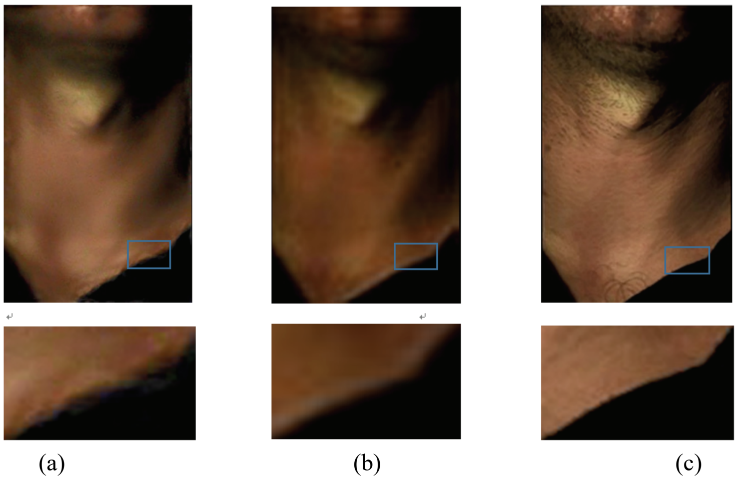


Fig. 1. Comparison of the motion amplification results on throat sequences. (a) The amplification result generated by Wu, et al, which mainly employs spatial decomposition; (b) The amplification result generated by Wadhwa et al which mainly employs complex steerable pyramid decomposition; (c) The magnification result obtained by our technique.

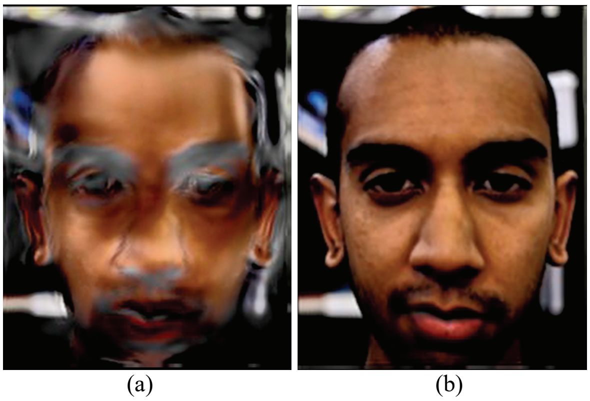


Fig. 2. Comparison of the color amplification results on blackman sequences. (a) The amplification result generated by Wu et al which mainly employs spatial decomposition; (b) The magnification result obtained by our technique.

Acknowledgements

This research work is jointly supported by the National Natural Science Foundation of China (Grant No. 51578107, 51778103, 5177081947, and 51121005), and 973 Project (Grand No. 2015CB057704), and the Fundamental Research Funds for the Central Universities (Grand No. DUT18 LAB07) and STU Scientific Research Foundation for Talents (Grant No. NTF18012).

References

[1]. H. Y. Wu, M. Rubinstein, E. Shih, et al., Eulerian video magnification for revealing subtle changes in the world, *ACM Transactions on Graphics*, Vol. 31, Issue 4, 2012, pp. 1-8.

[2]. C. Liu, A. Torralba, W. T. Freeman, et al., Motion magnification, in *Proceedings of the Conference ACM SIGGRAPH*, 2005, pp. 519-526.

[3]. Y. J. Cha, J. G. Chen, O. Büyüköztürk, Output-only computer vision based damage detection using phase based optical flow and unscented Kalman filters, *Engineering Structures*, Vol. 132, 2017, pp. 300-313.

[4]. N. Wadhwa, M. Rubinstein, F. Durand, et al., Phase-based video motion processing, *ACM Transactions on Graphics (TOG)*, Vol. 32, Issue 4, 2013, 80.

[5]. N. Wadhwa, M. Rubinstein, F. Durand, et al., Riesz pyramids for fast phase-based video magnification, in *Proceedings of the IEEE International Conference on Computational Photography (ICCP'14)*, 2014, pp. 1-10.

[6]. M. Sushma, A. Gupta, J. Sivaswamy, Semi-automated magnification of small motions in videos, in *Proceedings of the International Conference on*

Pattern Recognition and Machine Intelligence (PReMI'13), 2013, pp. 417-422.

[7]. M. A. Elgharib, M. Hefeeda, F. Durand, et al., Video magnification in presence of large motions, in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'15)*, 2015, pp. 4119-4127.

[8]. Y. Zhang, S. L. Pintea, J. C. V. Gemert, Video acceleration magnification, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, 2017, pp. 502-510.

[9]. L. Xu, J. Jia, Depth-aware motion deblurring, in *Proceedings of the IEEE International Conference on Computational Photography (ICCP'12)*, 2012, pp. 1-8.

[10]. T. H. Oh, R. Jaroensri, C. Kim, et al., Learning-based Video motion magnification, in *Proceedings of the 15th European Conference on Computer Vision (ECCV'18),* 2018.

[11]. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends R in Machine Learning*, Vol. 3, Issue 1, 2011, pp. 1-122.

[12]. O. Semerci, N. Hao, M. E. Kilmer, E. L. Miller, An iterative, in *Proceedings of the Second International Conference on Image Formation in X-ray Computed Tomography*, 2012.

[13]. J. F. Cai, E. J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, *SIAM Journal on Optimization*, Vol. 20, Issue 4, 2010, pp. 1956-1982.

[14]. M. Kilmer, K. Braman, N. Hao, R. Hoover, Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging, *SIAM Journal on Matrix Analysis and Applications*, Vol. 34, Issue 1, 2013, pp. 148-172.

[15]. J. D. Gibson, B. Koo, S. D. Gray, Filtering of colored noise for speech enhancement and coding, *IEEE Transaction on Signal Processing*, Vol. 39, Issue 8, August 1991, pp. 1732-1742.

[16]. E. Ercelebi, S. Koc, Lifting-based wavelet domain adaptive Wiener filter for image enhancement, *IEE Proceedings – Vision, Image and Signal Processing*, Vol. 153, Issue 1, 2006, pp. 31-36.

[17]. T. Kailath, Some new algorithms for recursive estimation in constant linear systems, *IEEE Transactions on Information Theory*, Vol. 19, Issue 6, 1973, pp. 750-760.

(25)

Identification of Sleep Apnea Based on Short Length Segments of HRV Data using Simple Perceptron

A. Hossen
Sultan Qaboos University, College of Engineering, P.O. Box 33 Al-Khoud, 123 Muscat, Oman
Tel.: + 96824141303, Fax: + 96824414336
E-mail: abhossen@squ.edu.om

Summary: Sleep Apnea is a well-known disorder that is identified using a complex and an expensive overnight sleep test (Polysomnography). Many researchers used frequency-domain analysis and/or time-domain analysis to identify patients with sleep apnea based on the heart rate variability of the whole night recorded ECG signal. The whole night recording could vary from 6 to 8 hours. In this paper, a simple neural network (a perceptron with two inputs and a hard limit output neuron) has been used to identify patients of sleep apnea based on short-length segments of the whole record. The data used in this work was obtained from the MIT data bases and consists of 30 records (20 with sleep apnea and 10 for normal subjects) for training and similar size record for test. The input features to the perceptron are the three classical frequency band power ratios LF/HF, VLF/LF, and VLF/HF. The power of the frequency bands has been obtained using the wavelet-packet decomposition of the HRV signal. The best result is obtained using the whole record that was 93 %, and has been reduced to 88 % and 86 % and 84 % and 82 % when the signal length is divided by 5, 10, 20 and 30, respectively. Such acceptable results with short length segments is very essential to simplify the fully automatic screening test.

Keywords: Sleep apnea, Identification, MIT-data, HRV, Neural networks, Wavelet packets, Simple perceptron, Short segments.

1. Introduction

Sleep apnea is defined as a complete or partial stoppage of breath during sleep. Patients with sleep apnea suffer from snoring during night, fatigue, morning headings, day time sleepiness, hypertension [1]. There is a clear correlation between severe sleep apnea and development of cardiovascular diseases [2]. The difficulty in breath for sleep apnea patients will cause the brain to activate both sympathetic and parasympathetic components of the autonomic nervous system. Sleep apnea patients have increase in the sympathetic tone and decrease in the parasympathetic tone compared to normal subjects. Such changes in the sympathetic and parasympathetic tones will increase the heart rate [2].

Obstructive sleep apnea (OSA) is the common form of apnea that occurs when the upper airway is completely or partially obstructed due to the relaxation of dilating muscles. Hypoapnea occurs when the airway is partially collapsed and causes 50 % reduction of air accompanied by oxygen desaturation of 4 % or greater [1]. The severity of apnea is measured by a commonly used standard such as the apnea/hypoapnea index (AHI), which is the number of apnea and hypoapnea events per hour [1]. Most clinicians consider an apnea index below 5 as normal, and an apnea index of 10 or more as pathologic.

In the general population, the greatest challenge for primary care providers lies in determining which patients with some symptoms such as snoring warrant further evaluation, as most patients with OSA snore, but most snorers do not have OSA [2].

In developed countries the cost of investigating these symptoms has increased considerably during the last decade as the only reliable method for the diagnosis of OSA until now is overnight sleep studies (Polysomnography) [2], which is a cumbersome, time consuming and expensive procedure requiring specially trained polysomnographers and needs recording of EEG, EOG, EMG, ECG, nasal air respiratory effort and oxygen saturation [1].

Since the golden test method (the polysomnographic recording) is an expensive and complex test for both patients and specialists, deep research has been done to identify patients from only ECG records [3-4]. Cost efficient technologies for sleep disorders diagnosis and follow-up monitoring of treatment are important [3].

Therefore, it is of a great importance and interest to have automatic screening algorithms on a single-lead ECG signals. This will reduce the pressure on sleep laboratories and make it possible to diagnose sleep apnea inexpensively from ECG recordings acquired in the patient's home [3-4].

Heart rate variability (HRV) is referred to as the beat-to-beat variation in heart rate. Instantaneous heart rate is measured as the time in seconds between peaks of two consecutive R waves of the ECG signal. This time is referred to as the RRI. The variation of heart rate accompanies the variation of several physiological activities such as breathing, thermoregulation and blood pressure changes [2].

The frequency spectrum of the heart rate variability has been divided into three main bands: the very low frequency band (VLF) and low frequency band (LF) and the high frequency band (HF). The VLF band

(0.0033-0.04 Hz) is associated with thermoregulation, while the LF band (0.04-0.15 Hz) reflects the sympathetic tone and the HF band (0.15-0.4 Hz) reflects the parasympathetic tone [1]. The LF/HF ratio has been used by some researchers to represent the sympathetic modulation of the heart rate.

The organization of the paper is as follows: in Section 2, both data sets (the trial data and test data) and pre-processing steps are described. Section 3 contains the main idea of the wavelet-packet based spectral analysis and the simple perceptron technique. The results of implementation of the proposed method on full-length and short-segments of HRV records are given in Section 4. Section 5 contains conclusions of the presented work.

2. Data

The single-channel 60 ECGs extracted from polysomnographic recording with a sampling rate of 100 Hz and with an average duration of 492 minutes are downloaded from the physionet website [5]. The HRV data are computed and filtered and resampled at 1 Hz [4]. The 60 data are already divided into two groups training group set and test group set. Each group consists of 30 subjects (20 apnea and 10 normal).

The RRI data usually undergoes several pre-processing stages before it can be used in a classification algorithm.

The accuracy of the identification algorithm depends mainly on the accuracy of the (QRS) detection (R peak detector) that is used to obtain the RRI signal from the raw ECG data. The QRS detection and beat classification (normal or not) is accomplished by the single-lead threshold based ECG wave boundaries detector "Ecgpuwave", which is available on the physionet website [5]. The RRI data used in this work are the Normal-to-Normal (NN) intervals obtained directly from the QRS detector without any smoothing and filtering steps; therefore it could contain false intervals, missed and/or ectopic beats. The simple approach to exclude false RR interval is to bounds the RR intervals with lower and upper limit, typically 400-2000 msec. Therefore, all RR intervals beyond these limits are excluded. Removing of outliers is achieved by using of 41-points Moving Average Filter (MAF). A local mean is computed for every 41 consecutive RR intervals excluding the central interval. The output of the MAF is used as a reference measure to reject or accept the central RR intervals.

Re-sampling at 1 Hz and substituting of missed peaks are then achieved by simple linear interpolation that has been implemented by the MATLAB. Re-sampling and estimation of missed value are intended to generate an equally spaced RRI data and preserve the temporal sequences that are necessary for the frequency domain analysis. At this point we assume that we have a clean RRI data sampled at 1 Hz and contains no missed or outlier values.

3. Methods

The frequency domain features are obtained using FFT and wavelet packet decomposition with 32 bands using 5 decomposition stages. Fig. 1 shows an example of a three stage wavelet-packet decomposition stages. The power of the three frequency bands is obtained either directly from FFT spectrum or using the wavelet packet decomposition for better spectrum resolution. In case of the wavelet packets, the power of the VLF band is approximated by summing the power of the second and the third bands. The power of the LF band is obtained by summing the power of bands 4 to 10, while the power of the HF band is computed by adding the powers of bands 11 to 26 [4]. Three ratios are computed: r1 (LF/HF) and r2 (VLF/LF) and r3 (VLF/HF).

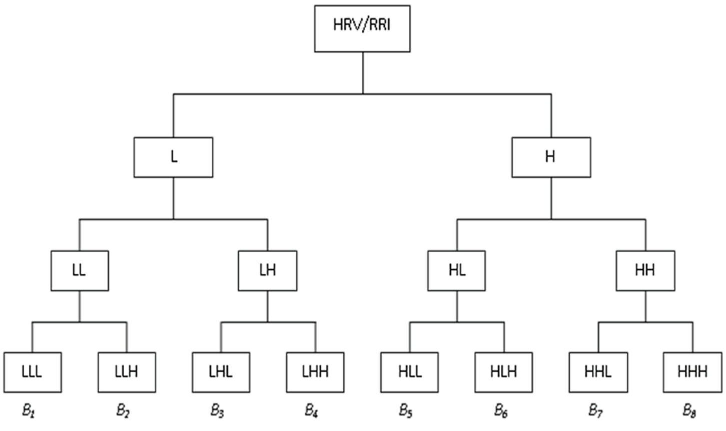


Fig. 1. Three stages wavelet-packet decomposition.

A simple neural network (perceptron) with two inputs (any two of the three ratios: r1, r2, and r3) and one output obtained from the hard limit output neuron of the perceptron has been used in the work (see Fig. 2). The perceptron was trained with the training set and tested on the test set.

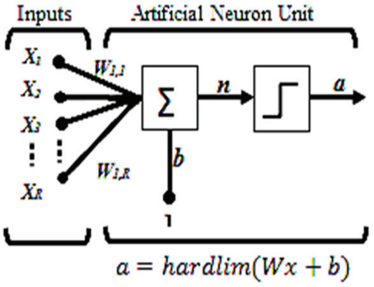


Fig. 2. A Simple perceptron.

4. Results

Firstly, the network is trained and tested using the complete overnight records. Table 1 shows the results obtained using FFT and three different wavelet filters (db1, coif1, and fk8). The best result (accuracy of 28/30: 93.335) is obtained using coif1 or fk8 wavelet filters with r2 and r3 as inputs to the perceptron. The idea of this work is to identify patients on the basis of short length HRV records. For this purpose, the data is divided into segments, for example 5 to obtain 150 records for training and 150 records for test. The perceptron is trained on the short length records and tested also on the short records. Table 2 shows the results using FFT and the same 3 wavelet filters used in Table 1 with the same combinations of power ratios of the three different bands. Four different number of segments have been used 5, 10, 20, and 30. The results of identification show that best identification (number of correctly classified records) reduces from 132/150 (88 %) to 259/300 (86 %) to 504/600 (84 %) and to 736/900 (82 %) when the segments are increased from 5 to 10 to 20 and to 30, respectively.

Table 1. Number of identified full-length records.

Method	r1&r2	r2&r3	r1&r3
FFT	26	25	25
db1	27	27	26
coif1	27	28	27
fk8	27	28	26

Table 2. Number of identified short length segments.

Method	5 segments: 150 records	10 segments: 300 records	20 segments: 600 records	30 segments: 900 records
	r1&r2 r2&r3 r1&r3	r1&r2 r2&r3 r1&r3	r1&r2 r2&r3 r1&r3	r1&r2 r2&r3 r1&r3
FFT	119 122 115	235 240 228	453 481 469	673 680 646
db1	128 126 125	248 245 249	500 499 488	716 721 729
coif1	130 131 126	259 251 248	504 499 490	725 722 727
fk8	130 132 129	258 253 246	498 501 491	727 736 725

5. Conclusions

A simple perceptron is used in this paper for identification of patients with sleep apnea from normal subjects based on two features computed from short length segments of the wavelet-based spectral analysis of the HRV signal. The best accuracy obtained based on full-length data is 93.3 %. while the best accuracy obtained based on 30 segments data is about 82 %. This shows that a simple method based on about 10-15 minutes of HRV signal can be sufficient for identification of patients with OSA. The result can be improved more by using other more sophisticated complex neural networks.

Acknowledgements

Many thanks for Sultan Qaboos University for the support in a form of internal research grant provided (and for the author for doing the research) and for the support attending the conference.

References

[1]. AASM task force report, Sleep-related breathing disorders in adults, recommendations for syndrome definition and measurement techniques in clinical research, *Sleep*, Vol. 22, 1999, pp. 667-689.

[2]. T. Young, P. Peppard, M. Palta, M. Hla, K. M. Finn, L. Morgan, J. Skatrud, Population-based study of sleep-disordered breathing as a risk factor for hypertension, *Arch. Intern. Medicine*, Vol. 157, 1997, pp. 1746-1752.

[3]. T. Penzel, J. McNames, P. de Ghazal, B. Raymond, A. Murry, G. Moody, Systematic comparison of different algorithms for apnea detection based on electrocardiogram recordings, *Med. Biol. Eng. Comp.*, Vol. 40, 2002, pp. 402-407.

[4]. A. Hossen, B. Al Ghunaimi, M. O. Hassan, Subband decomposition soft-decision algorithm for heart rate variability analysis in patients with obstructive sleep apnea and normal controls, *Signal Processing*, Vol. 85, 2005, pp. 95-106.

[5]. CinC Challenge 2000 data sets, Data for development and evaluation of ECG-based apnea detectors, http://www.physionet.org/physiobank/database/apnea-ecg/

(27)

Adversarial Multi-task Learning of Speaker-invariant Deep Neural Network Acoustic Models for Speech Recognition

L. Tóth [1] and G. Gosztolya [2]
[1] University of Szeged, Institute of Informatics, Árpád tér 2, H-6720 Szeged, Hungary
[2] MTA-SZTE Research Group on Artificial Intelligence, Tisza Lajos krt. 103, H-6720 Szeged, Hungary
E-mails: {tothl, ggabor}@u-szeged.hu

Summary: The performance of speech recognition systems has greatly improved with the introduction of Deep Neural Network (DNN) acoustic models. However, making these systems robust against all possible kinds of environmental conditions is still an important research topic. The adversarial multi-task DNN training method was proposed recently, and it has already been successfully applied to increase the domain and noise robustness of DNN acoustic models. Here, we evaluate the efficiency of this training method in increasing the speaker-invariance of a speech recognition system that is based on a convolutional neural network (CNN). Moreover, we propose a solution to handle those cases where speaker labels are not available for the training dataset. In the supervised case we report relative error rate reductions of 3-4 %. With the unsupervised method the improvements are somewhat smaller, but consistent across all tested parameter values.

Keywords: Speech recognition, Deep neural networks, Adversarial training, Multi-task training, Speaker-invariant.

1. Introduction

The introduction of deep learning in speech recognition has significantly reduced the error rate of speech recognition systems [1]. However, improving the robustness of these recognizers to various environmental factors is still in the focus of research [2], as the performance of current systems may drop drastically in different background noise, in reverberant environments, or simply with different speaker accents, just to name but a few possible adversarial conditions.

The sensitivity to these environmental factors can partly be explained by the fact that neural networks are inclined to overfit the actual training data, which reduces their generalization ability. Regularization methods are frequently applied to tackle this overfitting phenomenon. For example, it is frequently observed that presenting multiple tasks to the network at the same time – known as multi-task training [3] – also has a regularization effect. That is, having to solve two (or more) similar, but slightly different tasks at the same time forces the network to find a more general and more robust inner representation. Multi-task training has been successfully applied in several speech recognition studies [4, 5].

While multi-task training seeks to minimize the error of both tasks, there is a newer variant of the method known as *adversarial* multi-task training. Here, the error of the secondary task is *maximized*. With this modification, we expect the network to find an inner representation that is invariant with respect to the secondary task [6]. In speech technology, adversarial multi-task training has mostly been applied to increase the noise-robustness of DNN-based acoustic models [7, 8], as sensitivity to the background noise of the actual application domain is perhaps the most common adversarial factor. But we can also find examples where it is used to make the system less sensitive to other factors like accented speech [9].

The performance of speech recognizers may also vary significantly among speakers. In this paper, we seek to apply the adversarial multi-task training method to alleviate this issue. Our starting point will be the recent study of Meng et al. [10]. In contrast with their study, here we work with convolutional neural nets instead of fully connected DNNs. As the convolutional structure already makes the network less sensitive to speaker variance, it is not clear whether adversarial training can reduce this sensitivity any further. A second difference is that here we use the TIMIT database, which contains shorter samples from significantly more speakers than the corpus used in [10], so the task is presumably more difficult.

The approach of Meng et al. assumes that speaker-level annotation is available for the training data, permitting supervised training. However, most of the datasets available for training speech recognizers do not contain any information about the speakers. Thus, we also describe an experiment where we apply a clustering method which assigns the files to automatically designed speaker clusters, and the CNN is trained using these cluster labels. We will refer to this method as the unsupervised version of the approach.

2. Adversarial Multi-task Training

In multi-task training we train the neural network to solve multiple (in this case two) tasks in parallel, based on the same set of input features [6]. The two tasks should be related, but slightly different. Multi-task training requires a special network architecture where the network has separate output layers dedicated to the two tasks, and the uppermost hidden layers are also task-specific. However, there is only one, shared input layer, and the lowermost hidden layers are also shared between the two tasks. The multi-task DNN architecture is illustrated in Fig. 1.

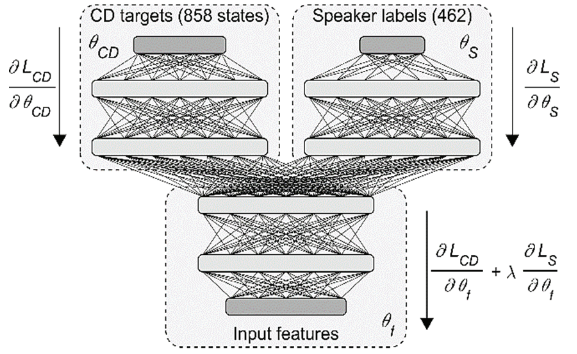

CD targets (858 states)
Speaker labels (462)
θ_{CD}
θ_S
$\frac{\partial L_{CD}}{\partial \theta_{CD}}$
$\frac{\partial L_S}{\partial \theta_S}$
$\frac{\partial L_{CD}}{\partial \theta_I} + \lambda \frac{\partial L_S}{\partial \theta_I}$
Input features
θ_I

Fig. 1. Architecture of the (adversarial) multi-task deep neural network.

In our case, one of the output layers is trained to recognize the context-dependent (CD) states of the Hidden Markov Model (HMM) speech recognizer. The other, secondary output layer is trained to identify the speaker label of the actual training sentence. During multi-task training, we minimize the error functions (L_{CD} and L_S) of the two output layers simultaneously. Thus, during the backpropagation training process we have to sum the error values of the two branches when they reach the shared layers. This means that λ = 1 in the formula of Fig. 1. The fact that the lower layers are shared between the two tasks forces the network to find a hidden representation that is useful for minimizing both error functions.

In the case of *adversarial* multi-task training, the goal is to minimize the error of the main task, and *maximize* the error of the secondary task at the same time. Ganin et al. proposed the following solution for this [6]: we will keep minimizing the error of the task-specific layers. However, when the error backpropagation reaches the shared layers, the sign of the error for the secondary task is flipped, which is technically realized by using a negative λ value. This modification practically turns minimization into maximization with respect to the shared layers. This way, the network will seek a shared hidden representation that is optimal for solving the first task, but contains no useful information for solving the secondary task. As in our case the secondary task is speaker identification, the optimal hidden representation would be totally speaker-invariant, and the classification error rate of the secondary branch would be 100 %.

In his original paper, Shinohara suggested introducing the adversarial secondary task only gradually by slowly increasing the (absolute) value of λ [7]. That is, in the *k*th training iteration the value of λ would be set to

$$\lambda_k = \min(\frac{k}{c},1)\cdot\lambda , \qquad (1)$$

which means in practice that λ attains its final value after *c* training epochs. He proposed setting *c* to 10, but we also experimented with the value of 7, as we observed that during the backpropagation process the halving of the learning rate typically starts after 6-7 training epochs.

Another meta-parameter of the model is the number of layers in the network, and their division between the shared and the task-specific parts. As in our previous studies (see e.g. [11]) we obtained the best results with 4-5 hidden layers, here we worked with a network depth of 4 hidden layers. As regards the depth where the two branches should join, it seems reasonable that having more shared layers is better when the two tasks are quite similar, while quite different tasks would require more task-specific layers. However, the optimal structure can only be found experimentally. For example, in an earlier paper where the tasks were relatively different, we found an early division to be optimal [12]. For the actual speech plus speaker recognition task, Meng et al. applied a network with 2 shared and 2-5 task-specific layers [10]. In the pilot tests we obtained the best results with 3 joint and 1-1 task-specific layers, so we present detailed results only with this network architecture.

3. Experimental Set-up

The neural network model we applied here contained convolutional neurons in its lowest layer, which performs convolution along the frequency axis (for more details, see our earlier study [11]). The shared part consisted of three hidden layers, while the task-specific parts contained one hidden layer for both tasks. All fully connected hidden layers contained 2000 rectified linear (ReLU) neurons. The main output layer consisted of 858 softmax neurons, corresponding to the context-dependent states of the hidden Markov model speech recognizer. The output layer for the secondary task contained 462 softmax neurons, and it was trained to discriminate the speakers of the training dataset. The training was performed using the standard backpropagation algorithm with a mini-batch size of 1000 training vectors. The learning rate was fixed at a value of 0.001 until the training error kept decreasing on the validation set. Afterwards, the learning rate was halved in each training epoch. The cost function applied was the standard cross-entropy error rate for both output layers, measured at the level of the training vectors (acoustic data frames). As the task of the CNN is framewise classification, in some cases we will directly report the frame error rates obtained. The speech recognition system applies the standard HMM/DNN hybrid scheme to convert the frame-level probability estimates into a sequence of phones [11]. To evaluate the accuracy of the whole speech recognizer, we will report the phone error rates attained.

As the training database we used the TIMIT English speech corpus, which contains speech samples from 462 speakers in the training subset. The core test set we used here contains samples from 24 speakers who are separate from the training set. As the development set, we randomly held out the samples of 44 speakers from the train set, which roughly corresponds to 10 % of the training material.

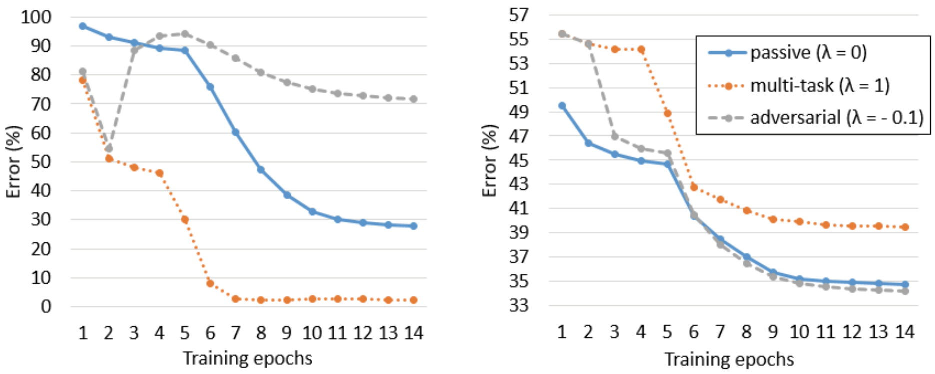


Fig. 2. The error curve for the secondary task on the train set (left), and the error curve for the main task on the development set (right) as a function of the number of training iterations.

4. Results and Discussion

Fig. 2 shows a typical example of how the (frame-level) classification error rate of the CNN changes during the training epochs. On the left we plotted the error rate of the secondary (speaker classification) task on the train set. We note that the secondary output will not be used by recognizer, so we plotted it only to gain an insight into what happens during adversarial multi-task training. The figure on the right shows the error for the main task – in this case on the development set, as this is the value that we seek to minimize. When λ = 0, the secondary branch of the network is allowed to learn, based on the hidden representation formed in the uppermost shared hidden layer, but it cannot modify this representation. Hence, we called this case the ‘passive learning’ scenario, and the result obtained in this configuration for the main task will serve as our baseline. We observe that in this case the speaker classifier branch can achieve an error rate below 30 % on the train set, while the error curve for the main output stops just below 35 %. By setting λ = 1, we get a classic multi-task model. In this case the training error rate of the secondary branch goes down to 3 %, but the price is that the error rate of the main task increases to 39 %. In the last experiment, we let the system run in multi-task mode for 2 iterations (to aid visualization), but then we turned on adversarial training by setting λ to -0.1. As the result, the corresponding error curve on the left quickly raises to the 70-90 % range, and remains there until the end of training. However, adversarial training has a beneficial effect on the main task, as the corresponding error curve goes slightly below that of the baseline (passive) model on the right.

Having found that adversarial training can outperform the standard multi-task training approach, we looked for the optimal parameter values. Table 1 shows how the parameters λ and *c* influence the error rates of the speech recognizer. As regards the main task, we report phone recognition error rates (PhER) from the full system on the development and test sets, while for the secondary task we report only frame-level error rates (FrER) of the neural network on the train set, as this output is not used by the recognizer. To reduce the effect of random initialization, we report the average scores of repeating the training 3 times. The results indicate that the error rate of the secondary task consistently increased as λ increased, just as expected. Moreover, compared to the baseline, the recognition error rate also consistently improved for all λ values both on the development and test sets, reaching the optimum at λ = -0.15 and *c* = 10. The relative error rate reduction lay between 3 % to 4 % on average, and it was 4.2 % in the best case.

Table 1. The frame error rates of the CNN (secondary task, train set) and the phone error rates of the speech recognizer (main task, development and test sets).

Parameters		**FrER (train set, secondary task)**	**PhER (dev. set, main task)**	**PhER (test set, main task)**
λ	**c**			
0 (baseline)		36 %	16.6 %	18.8 %
-0.03	7	57 %	16.3 %	18.3 %
-0.06	7	73 %	16.2 %	18.4 %
-0.10	7	82 %	16.1 %	18.3 %
-0.10	10	79 %	**16.0 %**	**18.0 %**
-0.15	10	85 %	16.2 %	18.1 %
-0.20	10	90 %	16.3 %	18.2 %

Table 2. The error rates of the speech recognizer as a function of the number of clusters.

No. of Clusters	Parameters		Phone Error Rate (%)	
	c	λ	dev. set	test set
–	baseline		16.6 %	18.8 %
50	7	-0.10	16.3 %	18.6 %
100	10	-0.15	16.0 %	18.0 %
150	10	-0.10	16.0 %	18.3 %
200	10	-0.10	16.2 %	18.4 %
250	10	-0.10	16.0 %	18.3 %

5. Unsupervised Case

The TIMIT corpus is an old-fashioned database in the sense that its content was carefully planned, and it also contains a detailed description of the speakers. Nowadays, we use much larger databases and we prefer to record these under realistic application conditions. Unfortunately, speaker annotation is not available for most of the newer databases. In these cases, we cannot directly apply the adversarial training method, as the missing speaker labels must be replaced by some other training targets. A possible solution is to use automatically determined labels. Here, we utilized a refined version of the unsupervised speaker clustering algorithm called bottom-up Hierarchical Agglomerative Clustering with a Generalized Likelihood Ratio (GLR) [13-15] for this purpose. This algorithm arranges the files into clusters, based on the similarity of the speakers' voices. The only assumption of the algorithm is that each file contains the voice of only one speaker, which is satisfied in the case of TIMIT, where each file contains a single sentence. The only meta-parameter of the algorithm is the number of clusters. We set this parameter to 50, 150, 150, 200 and 250, and then we repeated the DNN training experiment, but this time using the speaker clusters as training targets. Table 2 shows the results we obtained, for each cluster size reporting the score of just the best performing meta-parameters. As we can see, the improvements and the best result on the development set were comparable to those for the supervised method. On the test set, the decrease in the error rate was the same in the best case, though slightly less on average. However, the improvement was consistently present for all cluster sizes used.

6. Conclusions

Here, we evaluated the adversarial multi-task training method proposed by Meng et al. within the framework of CNNs. Moreover, we found a way to make the method work when speaker annotation is not available. We found that the adversarial training method is beneficial for CNNs as well, and that our unsupervised training approach can attain error rate reductions that are comparable to those of the original method.

Acknowledgements

We acknowledge the support of the Ministry of Human Capacities, Hungary, grant 20391-3/2018/FEKUSTRAT. László Tóth was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and the UNKP-18-4 New Excellence Program of the Hungarian Ministry of Human Capacities. The Titan X graphics card used in this research was donated by the Nvidia Corporation.

References

[1]. D. Yu, L. Deng, Automatic Speech Recognition – A Deep Learning Approach, *Springer*, 2015.

[2]. J. Li, L. Deng, Y. Gong, R. Haeb-Umbach, An overview of noise-robust automatic speech recognition, *IEEE/ACM Trans. on Audio, Speech and Language Processing*, Vol. 22, Issue 4, 2014, pp. 745-777.

[3]. R. Caruana, Multitask learning, *Journal of Machine Learning Research*, Vol. 28, 1997, pp. 41-75.

[4]. M. Seltzer, J. Droppo, Multi-task learning in deep neural networks for improved phone recognition, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'13)*, 2013, pp. 6965-6969.

[5]. P. Bell, S. Renals, Regularization of deep neural networks with context-independent multi-task training, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'15)*, 2015, pp. 4290-4294.

[6]. Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, *Journal of Machine Learning Research*, Vol. 17, 2016, pp. 1-35.

[7]. Y. Shinohara, Adversarial multi-task learning of deep neural networks for robust speech recognition, in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'16)*, 2016, pp. 2369-2372.

[8]. P. Denisov, N. Vu, F. Font, Unsupervised domain adaptation by adversarial learning for robust speech recognition, in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'16)*, 2016, pp. 2369-2372.

[9]. S. Sun, C. Yeh, M. Huang, M. Ostendorf, L. Xie, Domain-adversarial training for accented speech recognition, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18)*, 2018, pp. 4854-4858.

[10]. Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gong, B. Juang, Speaker-invariant training via adversarial learning, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18),* 2018, pp. 5969-5973.

[11]. L. Tóth, Phone recognition with hierarchical convolutional deep maxout networks, *EURASIP*

Journal on Audio, Speech and Music Processing, 2015, 25.

[12]. L. Tóth, T. Grósz, A. Markó, T. Csapó, Multi-task learning of speech recognition and speech synthesis parameters for ultrasound-based silent speech interfaces, in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'18)*, 2018, pp. 3172-3176.

[13]. K. J. Han, S. Kim, S. S. Narayanan, Strategies to improve robustness of Agglomerative Hierarchical Clustering under data source variation for speaker diarization, *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 16, Issue 8, 2008, pp. 1590-1601.

[14]. W. Wang, P. Lu, Y. Yan, An improved speaker clustering, *Acta Acustica*, Vol. 33, Issue 1, 2008, pp. 9-14.

[15]. H. Kaya, A. Karpov, A. Salah, Fisher vectors with cascaded normalization for paralinguistic analysis, in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'15)*, 2015, pp. 909-913.

(28)

Deep Learning Authentication Based on Arm and Wrist Movements

Nilay Tufek[1], **Ozan Ata** [1], **Zeynep Oner** [1] **and Zehra Cataltepe** [1]
Istanbul Technical University, Maslak, 34467 Sariyer/Istanbul, Turkey
+ 90 554706 25 17
E-mail: tufek@itu.edu.tr

Summary: Machine learning practices are becoming more important with the advances in digitalization. Recognizing human activities and using resulting activity information for authorization purposes are still challenging topics. In this study, we designed and implemented a solution with an accurate success rate for log-in purposes by using a wearable device. In this concept, each user can log in the system with a specific arm movement or wrist gesture as a signature by wearing a smart watch. Also, in the scope of the study, each user has written a specific word on a paper and consequently the system detects who writes the word by using wrist movements. The inputs of the system are an accelerometer and a gyroscope sensor data on the smart watch. The system model is designed and implemented by using state of the art deep learning methods and their combinations including data augmentation, Long-short Term Memory (LSTM), 1D-2D Convolutional Neural Networks (CNN), LSTM+CNN, Dynamic Time Wrapping (DTW) and k Nearest Neighbour (kNN), with several combinations being implemented for each generated data set.

Keywords: LSTM, CNN, Deep learning, Activity recognition, Authentication, Sensor-based.

1. Introduction

Efficient analysis of sensor data is becoming more essential in machine learning [1]. With increasing usage and capabilities of the sensors and with new generation IoT systems; interpretation of time series sensor data has a higher significance [2].

Nearly all systems tend to be digitalized for making them more user friendly. Login is an important part of such a system and consequently transforming. By using accelerometer and gyroscope sensors, determining user activities as a signature is the main idea of this study. In other words, the detection of the owner's arm or wrist movements is the research topic of this study. Therefore, an end-to-end authentication solution is designed and implemented on the cutting edge of technology for a bank login system.

Within the scope of this study, a unique data set was created and contributed to this academic study. In literature, there is no data set published for arm and wrist movements aiming to authenticate passwords via activity recognition. To address this need, a mobile and a smart watch applications were designed and implemented to collect data from users. The main contribution is to propose a log-in system which benefits from state-of-the-art machine learning algorithms applied on the data sets with several approaches. Also, the resulting accuracy, space and time consumption comparisons of these methods were made.

2. Literature Review

In literature, there are general purpose activity recognition studies. Most of them are about full body activity recognitions. One of the most popular data sets is called UCI HAR data set [7] which is occurred by "walking", "sitting", "standing", "lying", "climbing up-stairs" and "climbing down-stairs" actions. This data set is generated via accelerometer and gyroscope sensors on mobile phone. Another popular dataset on this area is "Opportunity Dataset" [8].

While these popular data sets are based on whole body physical actions, there is one similar study on wrist movements [3] as well. However, the data set is not publicly published. While this study used classical machine learning methods, state of the art deep learning methods were evaluated in our study. Moreover, we have used a smart watch do achieve authorization.

3. Theoretical Background

3.1. DTW + kNN

Dynamic Time Warping (DTW) currently exists well-known pattern-matching method to measure similarity between two time dependent sequences. How DTW carries out as a pattern matching technique in recognition systems is extracting one template from whole data stream and training it as a DTW template. Thereafter, computation of the similarity cost between two data streams is obtained by substituting vector lengths as shown in equation (1).

$$DTW = \sqrt{} (Dx^2+Dy^2+Dz^2), \quad (1)$$

DTW is the dynamic time wrapping distance in Euclidean mode. In this equation it is shown for three dimensional vectors.

K Nearest Neighbours (kNN) is a classification algorithm that identifies the k most related gestures among given N training vectors based on feature similarity. The algorithm appeals real world data while determining model structure instead of obeying theoretical assumptions like in linear regression models. In Online Accelerometer Gesture Recognition using Dynamic Time Warping and K-Nearest Neighbours Clustering with Flawed Templates article, while DTW is implemented to define similarity between two template data streams using distances, kNN as a distance-based classifier extracts k nearest patterns among all measurements. [1] In this study, similarity cost between two different time series data collected from accelerometer and gyroscope sensors is measured by DTW formula and then classification is performed by kNN algorithm.

3.2. LSTM

Long Short-Term Memory (LSTM) has been discovered by Hochreiter and Schmidhuber as an effective and scalable solution for various machine learning problems related with sequential data [2]. Experiment results of [2] indicate that Vanilla LSTM cells, full back-propagation through time training and logistic sigmoid function as activation function, performs well on the most of the problems and modifications on hyper-parameters such as number of hidden nodes, learning rate and hidden layer size have great effect on the performance of the system. Below figure represents LSTM architecture and its back-propagation mechanism.

3.3. CNN

Convolutional Neural Networks (CNN) are one of the most common feature extraction and classification methods. It is mostly used in computer vision on images. The major advantage of CNN is extracting features basing on the data-sets without any pre-determined functions. In our study, we used CNN for time series data by converting them into image format [9]. There are used two different kind of CNN architectures including 1D CNN and 2D CNN. All are generated and trained as from scratch networks.

3.3.1. 1D CNN

Here, an image format is generated from sensor data of an action and the convolution is used on only one dimension. When the data is 131 filtered sample and each sample is occurred by 7 different data (3 acc + 3 gyro + 1 time interval), then the features are extracted separately by using the changes on time and on sensory data. The architecture of 1D CNNs is created and trained from scratch as described in the Section 5.2.

3.3.2. 2D CNN

2D CNN uses two dimensions filters on images. It is most common CNN usage for machine learning models.

3.4. CNN + LSTM

CNN + LSTM is used as a state-of-the-art benchmark on time series data including activity recognition and video level computer vision solutions. For 2D image input which is our data converted version, 1D CNN is used for future extraction and LSTM is added at the end of the CNN for understanding the behaviour in time.

4. System Design

In a login concept, each smart watch account can be used as a user name. Recognition result of arm or wrist movement can be used as a password. System overview is shown in Fig. 1.

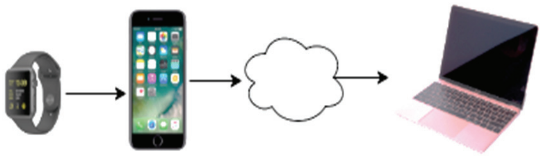


Fig. 1. End-to-end system overview including sensor, mobile phone, internet connection and the server for analysing and login.

Various challenges are achieved in this study including comprehensive system design, data collection and organization, implementations of mobile applications, data augmentations and from scratch implementations of deep learning methods.

4.1. Data Collection

There are three different kinds of data sets which are collected from two unique aspects: the first one is the “no specified action” case. Each user should specify his/her own arm movement which will be interpreted as a password. They are allowed to create their own arm movement like drawing big circle or zigzag on the air. This is entirely user specific action. Then, the proposed system would determine if this is an expected customer or not via the user’s movements. This is called a “*paraph*” data set. A sample from accelerometer and gyroscope point of view is seen in Figs. 2 and 3.

Secondly, wrist movements are also used for authorization. Each user writes the same word “*elma*” on the paper whilst the smart watch is collecting data and consequently it determines the user. These data sets are constructed both with *left* and *right* handed data. A sample for accelerometer and gyroscope for an action is shown in Figs. 4 and 5. As it is seen, because of small writ movement of writing with a pen, the change in range is very small compare to arm movements.

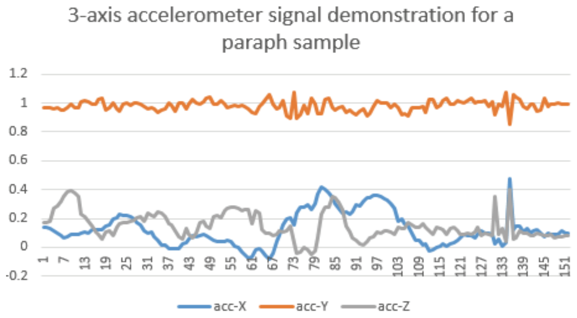


Fig. 2. Accelerometer measurement of paraph action in time. While x-axis represents number of samples, y-axis represents gravity.

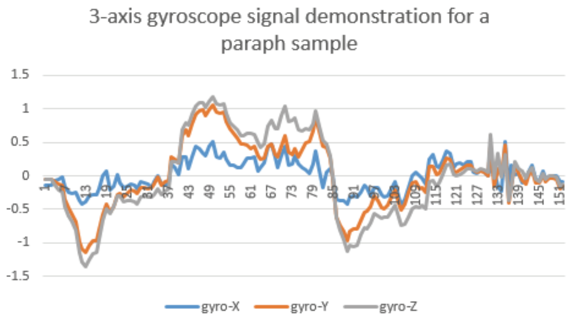


Fig. 3. Shows gyroscope measurements in time. Y-axis represents rad/sec^2.

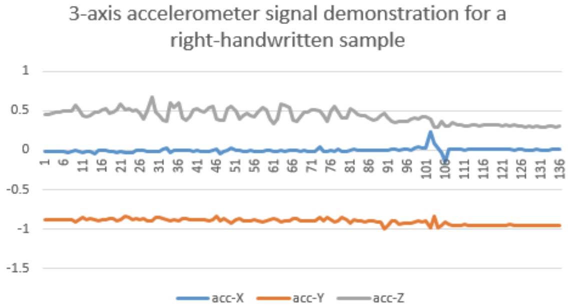


Fig. 4. Accelerometer measurement of right hand-written action in time. While x-axis represents number of samples, y-axis represents gravity.

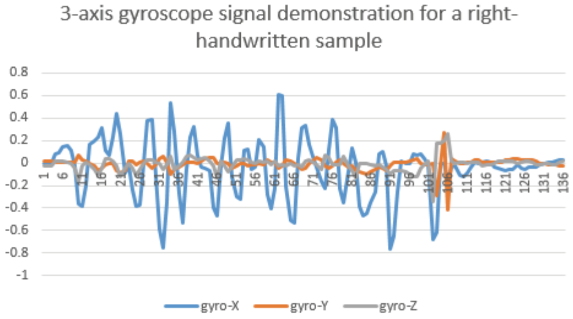


Fig. 5. Shows gyroscope measurements in time. Y-axis represents rad/sec^2.

In this data set 30 different subjects, healthy and between the ages of 18-40 years old were involved. Each gave data for *paraph*, *left-handwritten* and *right-handwritten* data sets 5 times. The sampling sensor rate was 50 Hz and each action was captured for 4 seconds.

4.2. Pre-processing

Before the implementation, there are pre-process steps including filtering, zero padding and separating into training and test data sets randomly by protecting continuity in ¼ proportion. Also, data augmentation is implemented to extend training set twice. The strategy of data augmentation is executed via random shifting and random noise adding.

4.3. Learning Processes

From the artificial intelligence point of view, there is a classification problem with 30 classes. The phases of learning system are occurred by two different parts called training phase and testing phase. Training phase is about generating intelligent model which is classification model in this study. Training data, initial values and model structure are needed to implement this phase. After several iterations, the model is prepared. The schema of this phase is shown in Fig. 6.

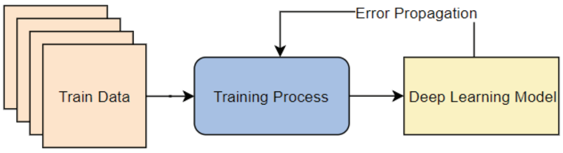


Fig. 6. Deep learning training process schema.

Another phase of the system is testing. After training phase is finished with a determined success which is about the error approximation, test could be run. For this step, we can get the classification results of the test data by using trained machine learning model. It can be seen in Fig. 7. Then evaluation could be applied this results.

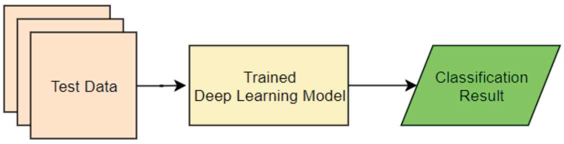


Fig. 7. Deep learning testing process schema.

5. Algorithm Design

For this problem, state of the art deep learning techniques [4] are used. All deep learning networks are designed and trained from scratch. **1D-CNN** is used on both time dimension and sensor dimension separately. **2D-CNN** [5] is used after preparing the sensor data as image data. **CNN+LSTM** method is created by 1D convolution with 1- Layer LSTM. **2 Layer** and **3 Layer LSTMs** are implemented [6] as well. One of the most popular time series classical machine learning method [3], **DTW+kNN** is implemented as well, while k is 4. Moreover, 4-Cross Fold Validation is used.

In the implementation of Machine Learning and Deep Learning algorithms, Keras and Tensorflow frameworks are used in Python 3.7. The data is prepared before implementing including zero padding and random splitting to train and test datasets in proportion of 4/1. Then following machine learning and deep learning algorithms are implemented for three different data sets by using 4 Fold Cross Correlation.

5.1. DTW + kNN

DTW and kNN were used together. For DTW, Euclidean distance is used. In order to determine k value, it is tried from 1 to label size and it is determined that k = 5 gives the best accuracies.

5.2. CNN

In this work, CNN architecture is used from scratch. Also because of the comparison purposes, 1D CNN (time), 1D CNN (sensors) and 2D CNN are used. In order to use CNN, each single action in dataset migrated into image format as 160×7 means 160 time stamps and 7 different sensors (pre-features).

5.2.1. 1D CNN – Time Convolution

The convolution is used on only one dimension. Here, the image is generated from sensor data of an action. When the data is 160 sample and each sample is occurred by 7 different data (3 acc + 3 gyro + 1 time interval), then the features are extracted only by using the changes on time series for each channel. The architecture is created and trained from-scratch as below: batch size: 24; Epoch 130; loss function: categorical cross-entropy; optimizer: SGD; learning rate 0.01; result metric: accuracy. Input shape is 131×7; Layer 1: 1D Convolution Layer: kernel size: 3, filters: 32, activation function: "relu", stride:2 Layer 2: 1D Max-pooling: pool-size: 2, stride 1 Dropout: 0.1 Layer 3: 1D Convolution Layer: kernel size: 2, filters: 32, activation function: "relu" Layer 4: Dense: node: 100, activation "relu" Layer 5: Dense: 24, activation: "softmax".

5.2.2. 1D CNN - Sensor Convolution

Here, only sensors have convolution, and time series had no convolution with the same architecture. It is specified as below: The same network as 1D CNN-Time network and parameters used but only changes on the input parameters occurred because of the convolution direction transposition. Input size: 7×131.

5.2.3. 2D CNN

The image-formatted data is used as an image here in generic 2D CNN. From scratch networks is described with the parameters in Keras as below: batch size: 24; Epoch 300; loss function: ’categorical cross-entropy’; optimizer: SGD; learning rate 0.01; result metric: accuracy. Input shape 131×7; Layer 1: 2D Convolution Layer: kernel size: 3×3, filters: 32, activation function: "relu", stride:1×1 Layer 2: 2D Max-pooling: pool-size: 2×2, stride 1×1 Dropout: 0.4 Layer 3: 2D Convolution Layer: kernel size: 3×3, filters: 32, activation function: "relu" Dropout: 0.4 Layer 4: Dense: node: 1000, activation "relu" Layer 5: Dense: 24, activation: "softmax".

5.3. CNN + LSTM

1D CNN and LSTM were used as a hybrid model for this experiment. The aim is feature extraction by using CNN on both sensors and LSTM are used for the time series behaviour on time line. Batch size: 24; Epoch 130; loss function: 'categorical cross-entropy'; optimizer: 'Adadelta'; learning rate 0.01; result metric: accuracy. Input shape (131,7); The architecture is as below: Layer 1: 1D Convolution Layer: kernel size: 3, filters: 20, activation function: "elu" Layer 2: 1D Max-pooling: pool-size: 2, stride 1 Layer 3: 1D Convolution Layer: kernel size: 2, filters: 180, activation function: "elu" Layer 4: 1D Max-pooling: pool-size: 1, stride 1 Layer 5: Dense: node: 512, actiation "elu" Layer 6: Dense: node: 512, activation "elu" Layer 7: LSTM: 256 node Dropout 0.2 Layer 8: Dense: 24, activation function: "sigmoid".

5.4. LSTM

In this study LSTM is used as 2 layer and 3 layer LSTM architectures with fallowing parameters batch size: 24; Epoch 10; loss function: 'binary cross-entropy'; optimizer: 'rmsprop'; learning rate 0.01; result metric: accuracy. Input shape is 131×7.

5.4.1. 2 Layer LSTM

The parameters of the algorithm for each layer is shown as following. 1. Layer: LSTM: 32 node, return sequence true, stateful true 2. Layer: LSTM: 32 node 3. Layer: Dense: 24, activation function: softmax.

5.4.2. 3 Layer LSTM

The parameters of the algorithm for each layer is shown as following. 1. Layer: LSTM: 32 node, return sequence true, stateful true 2. Layer: LSTM: 32 node, return sequence true 3. Layer: LSTM: 32 node 4. Layer: Dense: 24, activation function: softmax.

6. Results

Empirical evaluations demonstrate that the results are successful from both artificial intelligence and security perspectives. In Table 1, test accuracies, training durations and updating parameters are shown.

The accuracy results for the case with data augmentation are mostly better than the case without data augmentation as seen on the Table 2.

Table 1. Results of test accuracy, time and space usage for each method and each dataset.

Models	Paraph	RightHand Written	LeftHand Written	Duration	Updated Parameters
DTW+kNN	89 %	94 %	97 %	<1 sec	0
1D-CNN time conv.	90 %	91 %	96 %	60 sec	206,908
1D-CNN sensor conv	87 %	91 %	95 %	20 sec	24,412
2D CNN	83 %	88 %	91 %	15 min	2,050,592
2-Layer LSTM	90 %	93 %	95 %	4 sec	14,232
3-Layer LSTM	**93 %**	**96 %**	**98 %**	9 sec	22,552
1D CNN + LSTM	88 %	91 %	94 %	3 min	1,248,032

Table 2. Results of test accuracy, time and space usage for each method and each augmented data set.

Models	Paraph	RightHand Written	LeftHand Written	Duration	Updated Parameters
DTW+kNN	89 %	95 %	97 %	3 sec	0
1D-CNN time conv.	91 %	93 %	97 %	100 sec	206,908
1D-CNN sensor conv	88 %	91 %	93 %	40 sec	24,412
2D CNN	85 %	88 %	94 %	20 min	2,050,592
2-Layer LSTM	92 %	93 %	95 %	8 sec	14,232
3-Layer LSTM	95 %	97 %	**98 %**	18 sec	22,552
1D CNN +LSTM	89 %	92 %	94 %	16 min	1,248,032

7. Conclusions

As a result, deep learning based solutions are successfully proposed and realized to address login problem. As it is visible from the empirical evaluations, the most suitable deep learning model for sensor based activity recognition is **3-Layer LSTM** due to its high accuracy which is **98 %** with acceptable space and time consumption compared to the others overall. Moreover, it is shown that data augmentation is important to improve training processes.

Acknowledgements

The sponsor of this project is Yapı Kredi Teknoloji. First of all, thanks to Ozlem Yurdakurban and Mustafa Isbilen for their support. Also, thanks to all the people for joining and contributing to data collection in advance.

References

[1]. J. Wang, et al., Deep learning for sensor-based activity recognition: A survey, *Pattern Recognition Letters*, 2018.

[2]. S. Aljawarneh, et al., A similarity measure for temporal pattern discovery in time series data generated by IoT, in *Proceedings of the International Conference on Engineering MIS (ICEMIS'16)*, 2016, pp. 1-4.

[3]. B. Nassi, et al., Handwritten signature verification using hand-worn devices, *arXiv:1612.06305*, 2016.

[4]. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, Vol. 521, 2015, pp. 436-444.

[5]. Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, in The Handbook of Brain Theory and Neural Networks, *MIT Press*, 1998, pp. 255-258.

[6]. A. F. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with LSTM, in *Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN'99)*, 1999, pp. 850-855.

[7]. D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones, in *Proceedings of the 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'13)*, Bruges, Belgium, 24-26 April 2013.

[8]. D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G.Tröster, J. Doppler, Collecting complex activity datasets in highly rich networked sensor environments, in *Proceedings of the IEEE Seventh International Conference on Networked Sensing Systems (INSS'10)*, 2010, pp. 233-240.

[9]. J. Yang, M. N. Nguyen, P. San, X. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, *IJCAI*, Vol. 15, 2015, pp. 3995-4001.

(29)

An Adaptive Threshold for Change Detection Methods using an Entropy Criterion – Application to Fault-tolerant Fusion in Mobile Robotics

B. Daass [1], **D. Pomorski** [1] **and K. Haddadi** [2]
[1] University of Lille, CRIStAL Laboratory, 59650 Villeneuve d'Ascq, France
[2] University of Lille, IEMN Laboratory, 59650 Villeneuve d'Ascq, France
Tel.: + 33 (0)7 69 60 56 19
E-mail: bilal.daass@univ-lille.fr

Summary: This paper proposes a reformulation of the change detection strategy using an entropy-based criterion. The proposed criterion allows to calculate an adaptive threshold, in contrast with the Bayes criterion. Our approach can be used by any change detection method based on the (generalized) likelihood ratio. As validation, we propose to consider the entropy criterion by applying it to a commonly used change detection technique (Cusum). The approach is illustrated on a well-known example of the literature. Finally, this entropy-based change detection is experimentally validated from an extended Kalman filter in mobile robotics, using a fault-tolerant fusion methodology. We apply and compare the classical and the proposed (with adaptive threshold) techniques to detect sensor faults. We show that our strategy is much more robust with respect to false alarms and missed detections. Our fault detection approach with adaptive thresholding leads to a more accurate pose of the mobile robot.

Keywords: Entropy criterion, Bayes criterion, Change detection, Adaptive thresholding, Fault-tolerant fusion, mobile robotics, Extended Kalman filter.

1. Introduction

Nowadays, process supervision occupies an important place in quality control, in cooperative localization in mobile robotics, etc. Indeed, any failure of such processes can reduce performance and have serious consequences. The development of statistical methods capable of detecting and locating anomalies in these dynamic systems as quickly as possible is of real interest.

Continuous efforts have been made in the development of robust and effective change detection strategies since Shewhart control charts [1]. Subsequently, various tools, techniques, approaches and their related applications in various fields were introduced. In this respect, many statistical process monitoring and control techniques use a (generalized) likelihood ratio. In particular, this is the case of the cumulative sum (Cusum) control charts, initiated by E.S. Page [2], and studied by many authors [3], [4] thereafter.

These methods use a fixed predetermined threshold [5]. Practically, the choice of the threshold value determines the system performances. By considering a threshold too low, we are exposed to a large number of false alarms; while a too high threshold leads in the best of case to a time delay in detection, and in the worst case, to a set of missed detections. In order to overcome this constraint, we propose in this paper to reformulate the detection strategy using an entropy-based criterion whose properties are compared with those obtained by the Bayesian criterion [6], [7].

Our approach is flexible because it allows to learn automatically, and in real time, the most suitable threshold for a given problem. In addition, our approach can be used by any change detection method based on the likelihood ratio calculation. The proposed strategy has a certain robustness with respect to rare events, allowing especially its use in in any change detection algorithm. We show that our strategy is much more robust with respect to false alarms and missed detections.

This paper is organized as follows. Section 2 presents a detection methodology based on an entropy criterion, to determine an adaptive threshold that can be applied to any likelihood ratio method. The following section details the Cusum algorithm, which is a commonly used change detection technique. Our entropy-based approach, associated with this detection technique, is then applied to a well-known example of the literature. Section 4 presents an experimental validation on real data from a mobile robotics platform. We propose a fault-tolerant fusion methodology based on an extended Kalman filter.

2. Detection Based on an Entropy Criterion

In this section, a reformulation of the detection strategy using an entropy-based criterion is proposed. The properties of the entropy-based criterion are then compared with those obtained by the Bayesian criterion [6], [7].

2.1. Entropy-based Criterion

Based on observations from N sensors, the detection theory addresses the problem of discriminating a finite number of hypotheses. In the context of a binary detection:

- H is the true hypothesis. It takes two possible values H_0 and H_1.
- Based on the knowledge of the vector of measurements $y = (y_1, y_2, \dots, y_N)$ issued from a vector of sensors $Y = (Y_1, Y_2, \dots, Y_N)$, a final decision u_0 is taken. It has two possible values 0 or 1, depending on whether H is considered to be H_0 or H_1 (Fig. 1).

The prior probabilities P_j and conditional probability densities $p(y_1, y_2, \dots, y_N/H_j)$ $(j = 0{,}1)$ are assumed to be known for each hypothesis.

Furthermore:

- $P_D = P(u_0 = 1/H_1)$ is the probability of detection.
- $P_F = P(u_0 = 1/H_0)$ is the probability of false alarm.

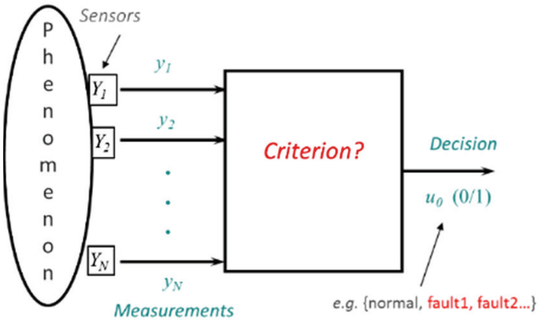


Fig. 1. Detection system.

The optimization criterion chosen in this work is the minimization of the part of H which is not explained by the final decision u_0. This quantity is defined as the Shannon's conditional entropy $h(H/u_0)$ [8]–[10]:

$$h(H/u_0) = E\left\{\log\left(\frac{1}{P(\mathrm{H} / u_0)}\right)\right\}$$
$$= -\sum_{i\in\{0,1\}}[\alpha_i \log\frac{\alpha_i}{\alpha_i+\beta_i} + \beta_i \log\frac{\beta_i}{\alpha_i+\beta_i}] \quad (1)$$

with:

$$\begin{cases} \alpha_1 = P_0 P_F & \beta_1 = (1-P_0)P_D \\ \alpha_0 = P_0(1-P_F) & \beta_0 = (1-P_0)(1-P_D) \end{cases} \quad (2)$$

Minimizing $h(H/u_0)$ then consists in minimizing a mean risk whose costs are not constant (unlike the Bayesian criterion), but depend on the posterior probabilities.

Taking into account the following notation:

$$\begin{cases} C_{00} = \log\frac{\alpha_0+\beta_0}{\alpha_0} & C_{10} = \log\frac{\alpha_1+\beta_1}{\alpha_1} \\ C_{01} = \log\frac{\alpha_0+\beta_0}{\beta_0} & C_{11} = \log\frac{\alpha_1+\beta_1}{\beta_1} \end{cases} \quad (3)$$

And under the reasonable assumption that $C_{10} > C_{00}$ and $C_{10} > C_{11}$, the decision rule can be expressed as a likelihood ratio:

$$\frac{P(u_0/H_1)}{P(u_0/H_0)} \underset{u_0\,=\,0}{\overset{u_0\,=\,1}{\gtrless}} \frac{P_0}{1-P_0} \times \frac{C_{10}-C_{00}}{C_{01}-C_{11}} = \lambda \quad (4)$$

This structure is similar to that obtained by the Bayes or Neyman-Pearson criterion. In this entropy-based approach, the costs are not constant, but depend on the posterior probabilities.

An exhaustive search could be proposed to find the optimal value λ^* of the threshold. It consists in testing all the possible values of the threshold λ and keeping only the value minimizing $h(H/u_0)$. In most cases, a gradient-based technique is preferable [11].

2.2. Entropy-based Criterion Properties

In this section, a simple example is chosen. It makes it possible to highlight interesting properties of the entropy optimization.

This example consists in optimizing a detector associated with a single sensor for which conditional probability densities are Gaussian $p(y/H_i) = \mathcal{N}_i(i, 1)$ $i = 0{,}1$. This results are compared to those obtained using the Bayes criterion for which the costs are commonly set at $C_{00} = C_{11} = 0$ and $C_{01} = C_{10} = 1$.

Fig. 2 (a) represents the ROC (receiver operating characteristic) curve defined by the detection probability P_D as a function of the false alarm probability P_F. One point of this curve corresponds to a unique value of the parameter P_0.

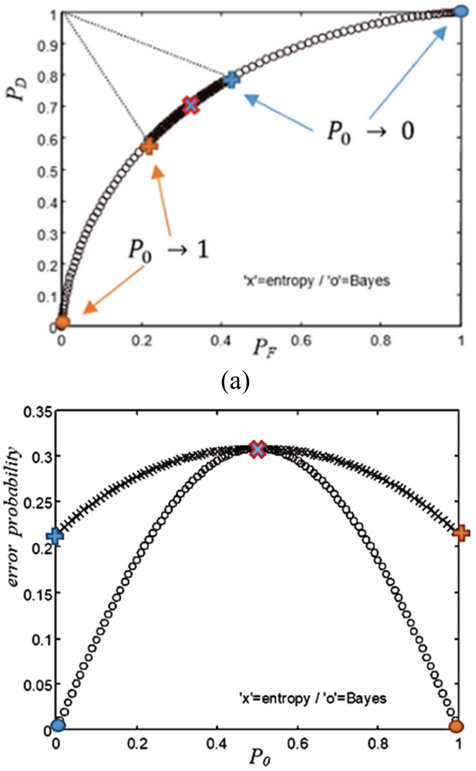


Fig. 2. (a) ROC curve; (b) Error probability – according to $\boldsymbol{P_0}$.

Note that the entropy-based criterion allows the reduction of the set of potential values of P_F and P_D (smaller arc): it is therefore a robust criterion with respect to P_0. In particular, when P_0 is close to 0 or 1, the ROC curve remains close to the ideal point ($P_F = 0$, $P_D = 1$). This is not the case with the Bayesian criterion: when an event is rare (e.g. faults, P_0 tends to 1), the Bayes criterion neglects this event. The entropy-based criterion takes this event into account significantly, to the detriment of the error probability, maximized, by definition, by the Bayes criterion (Fig. 2 b).

3. Change Detection: Cumulative Sum

The Cusum charts are typically used for monitoring change detection. They are effective in detecting small changes in the mean of a process.

When the process is under control, the ith measurement x_i is issued from a normal distribution with a mean μ_0 and a standard deviation σ. The Cusum algorithm works by accumulating deviations from μ_0 that are above (*resp.* below) the target μ_0 with a statistic C_i^+ (*resp.* C_i^-). These statistics are called the Upper and Lower Cusum respectively:

$$\begin{cases} C_i^+ = \max[0, x_i - (\mu_0 + K) + C_{i-1}^+] \\ C_i^- = \max[0, -x_i + (\mu_0 - K) + C_{i-1}^-] \end{cases} \qquad (5)$$

with: $C_0^+ = C_0^- = 0$.

K is the reference value (or tolerance threshold). It is often chosen halfway between the target μ_0 and the out-of-control value of the mean μ_1 (the mean for which we consider that the process is out of control).

If either C_i^+ or C_i^- exceeds the threshold H, the process is considered out of control. The value of this threshold is often determined without any real justification (apart from an experimental justification). However, the chosen value will condition the performance of the Cusum. A typical value of H is usually defined as five times the standard deviation σ of the process.

We propose to apply the entropy-based criterion to the Cusum algorithm using the well-known example of Douglas C. Montgomery [12] page 401, consisting of 30 observations. The first 20 observations were randomly drawn from a normal distribution $\mathcal{N}(10; 1)$; while the last 10 observations were drawn from a normal distribution $\mathcal{N}(11; 1)$.

Fig. 3 shows the Upper Cusum C^+(black), the fixed threshold used by [12] (red), and the adaptive threshold calculated iteratively (blue) using the entropy-based criterion presented in the previous section.

Note that we get a few false alarms in the first five observations. Indeed, the threshold can not be learned with a minimum of observations. From 7 observations, the threshold becomes stable, then it decreases slowly (the conditional entropy then increases). To avoid such false alarms, a solution is to initialize (at time t=0) the threshold to a predefined value (e.g. 5).

Using a fixed threshold, the model change is detected only from the 29th observation using Cusum. While, in our approach, we detect the model change from the 23rd observation, i.e. two steps after the model change. Therefore, our approach detects the model change more efficiently (faster) and without prior knowledge of the detection threshold.

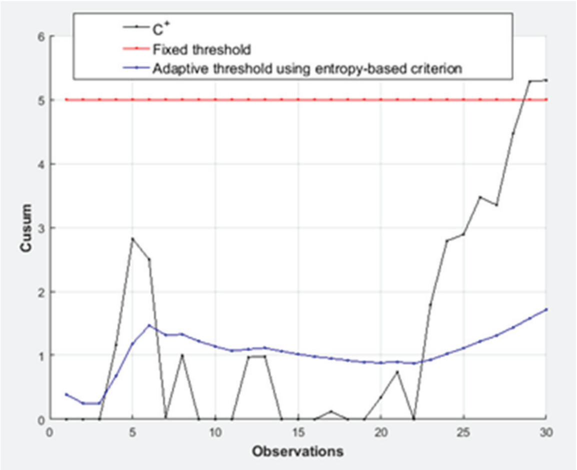


Fig. 3. Cusum results with fixed/adaptive threshold.

4. Application to Mobile Robotics

In this study, we consider a mobile robot (Turtlebot^{TM}) equipped with two differential wheels, and two freewheels for the stability enhancement (Fig. 4).

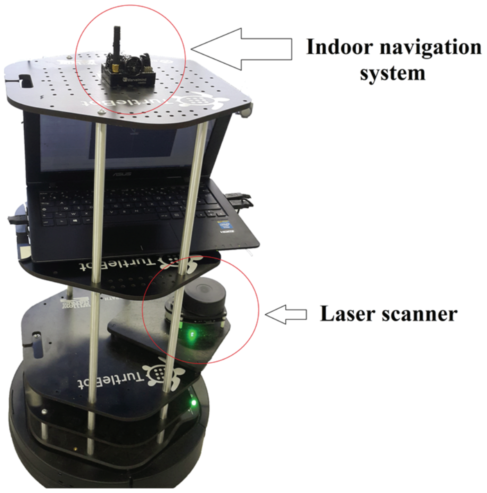


Fig. 4. Mobile robotic platform.

The robot is equipped with a set of sensors, such as wheel encoders, a 2D-laser scanner RPlidar^{TM} and an indoor navigation system Marvelmind^{TM}.

The main objective is to achieve the accurate pose of the robot by using measured data from the embedded sensors (that may be disturbed at any time).

The state of the robot at time step k is denoted as $X_k = (x \quad y \;\theta)_k^T$, where (x, y) and θ denote the robot's position and orientation, respectively, in a fixed global coordinate system.

The discrete-time motion model is given by:

$$X_k = f(X_{k-1}, U_k) + \omega_k \tag{6}$$

where f is a nonlinear function, and ω_k is the noise associated with the state model, considered as a white Gaussian noise (WGN) of zero mean value and covariance matrix Q_k. U_k is the input vector, i.e. the linear and angular speeds.

In the state estimation context, we propose to apply an extended Kalman filter (EKF) [13], which works in two steps:

First, consider the prediction step:

$$\begin{cases} X_{k/k-1} = f\left(X_{k-1/k-1}, U_k\right) \\ P_{k/k-1} = F_k P_{k-1/k-1} F_k^T + G_k (Q_u)_k {G_k}^T + Q_k \end{cases} \tag{7}$$

where:

- $P_{k/k-1}$ is the predicted (prior) covariance matrix. It is a measure of the accuracy of the predicted state $X_{k/k-1}$.
- $F_k = \frac{\partial f}{\partial X}|_{X_{k-1/k-1}}$ and $G_k = \frac{\partial f}{\partial X}|_{u_k}$ are the Jacobian matrices.
- $(Q_u)_k$ is the covariance matrix associated with measurement noise related to the input vector [14].

Consider now the update step:

The pose of the robot is obtained from the Indoor navigation system and from the Laser sensor using the ICP algorithm [15]. The observation vector can be written in the following form:

$$Z_k = H.X_k \tag{8}$$

Therefore:

$$\begin{cases} K_k = P_{k/k-1} H^T \left(H^T P_{k/k-1} H^T + R_k\right)^{-1} \\ X_{k/k} = X_{k/k-1} + K_k\left(Z_k - H.X_{k/k-1}\right) \\ P_{k/k} = (I - K_k H) P_{k/k-1} \end{cases} \tag{9}$$

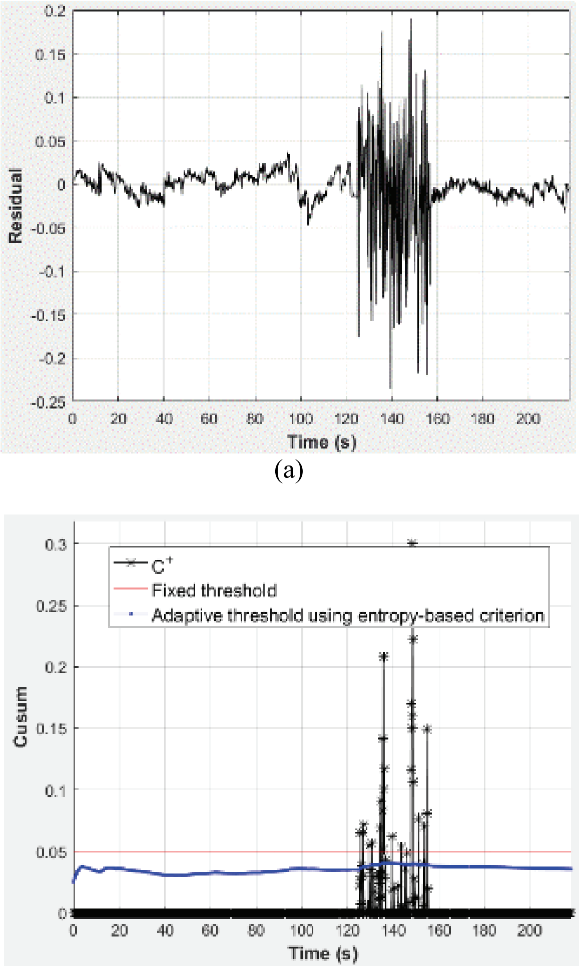
where $P_{k/k}$ is the updated (posterior) covariance estimate. It is a measure of the accuracy of the estimated (updated) state $X_{k/k}$. K_k is the near-optimal Kalman gain. And H is the observation matrix.

At each step k of the EKF, a residual is defined as follows:

$$Res_k = Z_k - H.X_{k/k-1} \tag{10}$$

The proposed change detection method applied to each component of Res_k makes it possible to detect a significant difference between the raw measurement given by the sensor and the datum issued from the predicted state. This faulty sensor is then excluded from the EKF.

The experiment lasts 220 seconds. The robot follows a simple circular trajectory with a diameter of 2 meters. The laser sensor data are disturbed by a WGN $\mathcal{N}(0.05; \sigma = 0.05)$ over the period 125-155 seconds. At each step k, the Cusum technique with/without adaptive threshold is applied to each residual in order to detect and isolate a sensor fault. Fig. 5 (a) represents the corresponding residual. Fig. 5 (b) shows the results obtained with the Cusum method using a fixed threshold (red) and using an adaptive threshold issued from entropy criterion (blue).



(a)

(b)

Fig. 5. (a) Laser sensor residual; (b) Cusum results for the data issued from the laser sensor.

Note that the fixed threshold technique missed some detections. One solution would be to decrease the threshold, but in this case, false alarms could be generated. On the other hand, our adaptive threshold makes it possible to avoid these missed detections, thus showing an obvious robustness of our approach.

Fig. 6 shows the Y-coordinates obtained from the laser sensor (red) and the estimated trajectory after detection and exclusion of the faulty sensor (black).

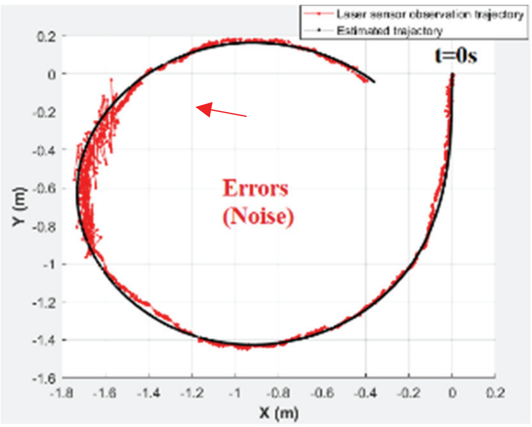


Fig. 6. Estimated trajectory - after detection and exclusion of the faulty sensor.

5. Conclusion

In this paper, we presented a reformulation of the change detection strategy using an entropy-based criterion. The advantage of our approach is to determine an adaptive threshold that can be used by any change detection technique based on the likelihood ratio, such as Cusum control charts.

Our method has been validated on a well-known example of the literature. The model changes are detected more efficiently (faster), and without prior knowledge of the detection threshold. In addition, this entropy-based change detection makes it possible to propose a fault-tolerant fusion methodology. This is illustrated from an EKF in the context of mobile robotics. We applied and compared the change detection method with a predetermined threshold and the proposed approach to detect and remove faulty sensors: our strategy is much more robust with respect to false alarms and missed detections.

References

[1]. W. A. Shewhart, Economic control of quality of manufactured product, *ASQ Quality Press*, 1931.

[2]. E. S. Page, Continuous inspection schemes, *Biometrika*, Vol. 41, No. 1/2, 1954, pp. 100–115.

[3]. M. Basseville and I. V. Nikiforov, Detection of abrupt changes: theory and application, Vol. 104, *Prentice Hall Englewood Cliffs*, 1993.

[4]. P. Granjon, The CuSum algorithm-a small review, 2013.

[5]. D. M. Hawkins and Q. Wu, The CUSUM and the EWMA Head-to-Head, *Quality Engineering*, Vol. 26, No. 2, Apr. 2014, pp. 215–222.

[6]. I. Y. Hoballah and P. K. Varshney, An information theoretic approach to the distributed detection problem, *IEEE Transactions on Information Theory*, Vol. 35, No. 5, 1989, pp. 988–994.

[7]. D. Pomorski, Entropy-based optimisation for binary detection networks, in *Proceedings of the 3rd International Conference on Information Fusion, (FUSION 2000)*, Vol. 2, 2000, pp. THC4–3.

[8]. R. C. Conant, Laws of information which govern systems, *IEEE Transactions on Systems, Man, and Cybernetics*, No. 4, 1976, pp. 240–255.

[9]. C. Desrousseaux and D. Pomorski, Optimisation entropique des systèmes de détection distribuée Entropy optimisation for distributed detection systems, *Traitement du Signal*, Vol. 16, 1999, No. 4.

[10]. D. Pomorski and C. Desrousseaux, Improving performance of distributed detection networks: An entropy-based optimization, *Signal Processing*, Vol. 81, No. 12, 2001, pp. 2479–2491.

[11]. D. Pomorski, Entropy-based optimisation for binary detection networks, in *Proceedings of the 3rd International Conference on Information Fusion*, 2000, Vol. 2, p. THC4/3-THC410.

[12]. D. C. Montgomery, Introduction to statistical quality control, *John Wiley & Sons,* New York, 2009.

[13]. S. J. Julier and J. K. Uhlmann, New extension of the Kalman filter to nonlinear systems, in *Signal processing, sensor fusion, and target recognition VI*, 1997, Vol. 3068, pp. 182–194.

[14]. R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza and R. C. Arkin, Introduction to Autonomous Mobile Robots, *MIT Press*, 2011.

[15]. A. Censi, An ICP variant using a point-to-line metric, in *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA 2008)*, 2008, pp. 19–25.

(30)

The Problem of High Confidence Mistakes of Deep Learning in Safety-critical Functions

E. Stensrud [1], **I. Myrtveit** [2]
[1] DNV GL, Veritasveien 1, 1363 Høvik, Norway
[2] Norwegian Business School, BI, Nydalsveien 37,0484 Oslo, Norway
Tel.: + 4748000857
E-mail: erik.stensrud@dnvgl.com

Summary: Deep learning, DL, is an old technique that has gained momentum since 2012 due to successes in image recognition competitions such as ImageNet. DL-based image recognition is also deployed in safety-critical functions of autonomous cars and ships. Car accidents have exposed DL's lack of robustness to irregular events like unusual image objects. Research has also exposed how easy it is to fool a DL with manipulated images or manipulated physical objects, and that it can also make high confidence mistakes. High confidence mistakes is of a particular concern to autonomous shipping where we foresee a remote, land-based human operator in the loop who can intervene if warned. A high confidence mistake will by definition not generate a warning to the human operator. This study presents examples of high confidence mistakes from the maritime sector, and then proceeds with a review of methods that might be applicable in trying to understand why DL systems make high confidence mistakes in image recognition.

Keywords: Deep learning, Safety-critical systems, Autonomous vehicles, High confidence mistakes, Autonomous ships, Functional safety.

1. Introduction

Autonomous systems are becoming an increasing part of our society. The development of autonomous vehicles promises economic benefits for society, for example driverless taxis and buses. Driverless cars may also reduce accidents since human drivers make mistakes. Likewise, autonomous shipping promises both cost savings and increased safety, but there are a number of challenges to be solved, including a reliable *situational awareness* function [1].

For autonomous ships, correctly classifying sailboats, kayaks and other objects at sea, is a challenge as they may have different colors, shapes, etc., and must be detected under varying weather and light conditions, including high waves, fog, rain, and dark.

In this paper, we first provide some maritime business context in Section 2. Section 3 presents some general problems with Deep Learning (DL), whereas Section 4 proceeds to show some examples of misclassifications from the maritime sector. Section 5 gives a short overview of techniques to understand and possibly debug DL-based object classification systems, with a focus on the emerging field of ”Explainable AI” but also mentioning techniques from other fields that may be relevant in the context of assuring safety-critical components. We conclude and propose some further work in Section 6.

2. Maritime Business Context

DNV GL is a provider of assurance services for safety-critical assets including ships, oil rigs, and energy transmission systems, ensuring functional safety of physical as well as software assets.

There are currently several maritime industrial projects seeking to pilot the implementation of remotely operated or autonomous ships with reduced or no manning [2, 3]. One of the major technical challenges for autonomous ships is to have a correct *situational awareness* to ensure safe navigational decisions and actions [1]. Correct classification of objects in images is one of the major inputs for situational awareness, and therefore it is a critical component. From this follows that assurance of the object classification technology component is crucial.

Currently, DNV GL is developing assurance services for of DL-based systems [4, 5], and the company is investigating several of the issues related to DL, including the lack of explainability [6], and the lack of prior knowledge [7].

3. DL Problems in Situational Awareness

Deep learning (DL) is deployed, or planned deployed, in safety-critical functions such as for *situational awareness* where image recognition systems is a critical component. DL is an old technique that has gained momentum since 2012 due to successes in image recognition competitions such as ImageNet [8].

However, the accident with the autonomous Uber car have also revealed the vulnerability of Deep Learning-based image recognition. The accident on May 18th 2018 with an Uber self-driving car, which resulted in the death of a 49-year-old female was such an incident. She was walking her bicycle, and recognized neither as a pedestrian nor as a cyclist [9].

Such accidents have exposed DL's lack of robustness to irregular, rare situations like unusual image objects (sometimes termed "out-of-distribution" [10]). Research has also exposed how easy it is to fool a DL with manipulated images or manipulated physical objects [11]. Fig. 1 is such an example. In the left photo, the face was correctly classified as the AI pioneer and late MIT Professor Marvin Minsky. In the manipulated image, the object was classified as "female entertainer". We observe that in the right photo, a smaller portion of the face was used to make the classification (white bounding box).

Finally, a DL also makes high confidence mistakes [12] which we explain more in the next section.

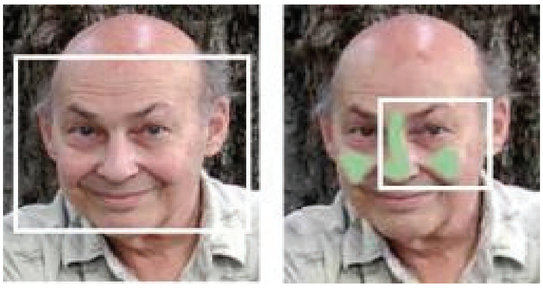


Fig. 1. Easy-to-fool example: Marvin Minsky, original and manipulated photos.

4. High Confidence Mistakes Examples

High confidence mistakes is of a particular concern to autonomous shipping where we foresee a remote, land-based human operator in the loop who can intervene if warned. A high confidence mistake will by definition not generate a warning to the human operator as it is not viewed as an uncertain or unrecognized object by the DL-system.

This section provides examples of high confidence mistakes, mostly from the maritime area. One of the crucial tasks in situational awareness is to detect and classify objects on the sea surface correctly.

Fig. 2 is an example of a high confidence mistake [13]. The reason of the misclassification in Fig. 2 of a kayak as a motor vessel has not been investigated, but we speculate that the "training" data was unbalanced, in the sense that the data contained many more images of motor vessels than of kayaks. We know from other studies in DNV GL that an image classifier will tend to classify an object towards the class of which there is most training data [14].

In another study, supervised by DNV GL, a Faster R-CNN convolutional neural network was developed as a single-class detector for detecting boats in images [15]. Some examples are shown below. Fig. 3 shows another example of a high confidence mistake where the building on the quay is classified as a boat with a 99 % score.

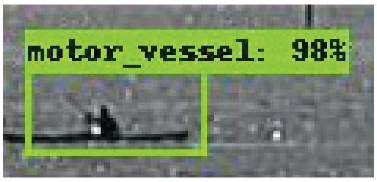


Fig. 2. High-confidence mistake example: a kayaker in a kayak classified as a motor-vessel with 98 % score.

The image in Fig. 3 is a single frame in a video stream. Fig. 4 shows that the confidence has dropped to 31 % in another of the video frames. The differences between these two frames is minor and due to the boat on which the video camera is mounted is moving forward. This points to another issue with situational awareness: it is quite unstable. Kamsvåg [16] found that most of the detections originated from the surroundings, such as from houses or structures in the vicinity, rather than from the actual targets, the boats. Kamsvåg suggested that these high confidence misclassifactions could be reduced by adding "background" classes such as *house, building, car* and so on. One issue we see with his suggested approach, is the "and so on". Which "background" classes and how many would we need to ensure a robust classification of boats?

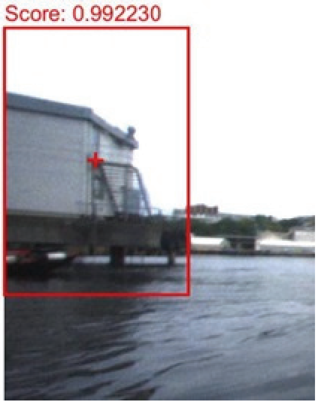


Fig. 3. High-confidence mistake example: a building on the quay classified as a boat with 99 % score.

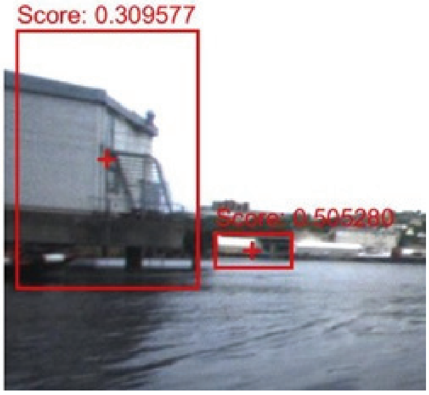


Fig. 4. Lower confidence example: the building on the quay is classified as a boat with 31 % score.

5. Techniques for Understanding a DL

High confidence mistakes give reason to distrust in DL systems. For safety-critical functions, understanding the working of the models and why they predict the values they do, is key to assessing and improving their trustworthiness. For example, in the case of the Uber self-driving car accident, we want to know why the woman killed was recognized neither as a pedestrian nor as a cyclist [9] in order to improve the image classification system and inspire confidence in it again. To verify a DL algorithm, it is therefore necessary to explain how it works.

The opacity of black box neural networks has been a major focus of discussion in the last few years [17] [18], with respect to the lack of understanding and therefore “debuggability”. This awareness has given rise to many initiatives aiming at mitigating this black-box problem, trying to understand the reasons for decisions taken by systems. These include the "ACM Statement on Algorithmic Transparency and Accountability", Informatics Europe's "European Recommendations on Machine-Learned Automated Decision Making" and the EU's GDPR regulation which introduces, to some extent, a right for all individuals to obtain "meaningful explanations of the logic involved" when automated decision making takes place. One of the first activities of the newly established European Commission’s High Level Expert Group on Artificial Intelligence has been to prepare a draft report on ethics guidelines for trustworthy AI [19]. The European Union has issued a “Draft Ethics Guidelines for Trustworthy AI” [26] that lists a number of requirements for trustworthy AI that include *robustness, safety, and transparency*.

We argue that explainability is key to fulfilling these three requirements. *Interpretability or explainability* is the degree to which a human can understand the cause of a decision [27]. This is a young and emerging field of research, with DARPA as a notable player, in Explainable Artificial Intelligence (XAI) [28].

In classical software engineering, the V-model[1] recommends several verification & validation steps at each stage of the software development for the different software artefacts. The design and the source code are two of the artefacts produced in software engineering, and verification of the design[2], and inspection of the source code, are two methods that give insight into the internal workings of the system, and thus contributes to explainability. When the behavior of the software is explicitly designed and coded, it is possible to understand the behavior by inspecting the design and code, and we can therefore use *white box* methods for the investigation.

Unfortunately, the behavior of software based on *statistical algorithms* is not explicitly designed and coded but rather is a result of the “training” data. A DL is a statistical algorithm that to a large extent is therefore a black box technology not lending itself easily to white box methods that help us understand the behavior.

There is a “grey box” zone between a black box and a white box. In a white box, all behavior is explicitly specified, designed, and coded. We have full transparency. With a black box system, all we know about the system behavior, is derived from the test cases. We input a stimulus and observe a response.

For image classification systems, we input a test image of a known class and observe the system’s classification. If the test image was a boat, and the system classifies it as a boat (true positive) the test is passed. If the boat was classified as a house (false negative), or if the test classifies a house as a boat (false positive), the test fails, but we want to know why it failed in order to know how to improve the classifier, other than the brute force approach which is to increase the training set and tune DL hyperparameters and then test again, but without any guidelines on *what kind of training data* or *what kind of tuning* that would improve the classifier. Finding out what kind of training data that is missing may be viewed as the equivalent of bug detection a traditional software system.

For black box models like DL’s, there are assessment methods that can help understand how it works and give some clues on the degree of confidence we might have in the system. A study in DNV GL evaluates a number of potentially useful tools, methods and practices that to some extent open up the DL black box into a more grey box, e.g. DeepLIFT [20], LIME [21], IOFP [22], SHAP, and GSM [23, 24], and more [5]. Lundberg and Lee introduce the framework SHAP and in addition provide a good overview of related approaches like LIME and LRP [25].

A key question in explainability of a DL-based *image* object classifier is how do we find out what are the *most important features*, e.g. the most important *pixels* (at the lowest feature level) used to classify the image object, and how can this information be used to debug and improve the algorithm and training data?

Visualisation of deep learning networks for image analysis is one approach and includes methods like heat maps, attention maps and the like [29, 30]. Heatmapping is one of the approaches that currently are gaining momentum. Bach et al. [31] developed the Layer-wise Relevance Propagation (LRP) technique.

Fig. 5 shows an example of heatmapping a handwritten digit. It was classified correctly as a ”3”, and the 2nd and 3rd choices were ”2” and ”8”, respectively. The red pixels indicate which pixels were most relevant or important in classifying it as a ”3”, and the blue pixels indicate which pixels are most *negatively* relevant. We observe that it is not the pixels within the shape of the ”3” that seem most important. There are few red pixels inside the shape of the ”3” that

[1] https://en.wikipedia.org/wiki/V-Model_(software_development).

[2] As an aside, verification of ship *design* drawings is a central part of the ship approval process.

match the white pixels in the left image. One interpretation is that the classifier puts importance to pixels that rather *differentiates* it from for example an ”8”. The red pixels in the left gap of the ”3” are not pixels within the white pixels constituting the ”3” but rather black pixels outside the ”3”. One interpretation of the blue pixels at the lower end of the digit is that this part *differs* from a typical shape of a ”3”.

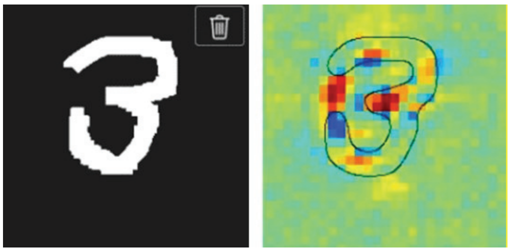


Fig. 5. Example of heat maps on image recognition of hand-written numbers (http://heatmapping.org/).

Montavon et al. [32] provide a tutorial on explanability methods applicable to DL that include many of the methods presented in Kazemi and Lindberg [5] but includes additional methods, e.g. Taylor decomposition, and backward propagation techniques like LRP and deconvolution.

The above heatmapping exercise of digits suggests that the DL emphasises features that differentiate between digits rather than features of the digit itself. This then seems to suggest that we need to know all classes beforehand that all object types encountered must be known. Otherwise, there might be a high risk of high confidence mistakes. The single-class classifier’s classification of the building on the quay as a ”boat” also supports this interpretation of how a DL ”sees” an object.

Heatmaps highlight which pixels of the input image most strongly support the classification decision, but then a question is how this knowledge can guide us in the ”debugging”, especially of high confidence mistakes. One obvious debugging is to ensure that all classes that might be encountered are represented in the training data.

Another and more difficult debugging is to find out how many more examples and variations within the class are needed of a given class for robust classification. Humans are good a generalising and abstracting from a few examples whereas DL’s are not. Rather, DL is data hungry.

As for *safety-critical systems*, the contribution of DL explainability methods like heatmapping need further scrutiny. Compliance with functional safety standards is one means of providing trust and confidence in a system. In the IEC 61508 functional safety standard [1], both white-box and black-box knowledge of the object to be certified is required. For example, IEC 61508, Table B.2, “Dynamic analysis and testing”, require white-box techniques such as “boundary value analysis” and “equivalence classes and input partition testing”. These techniques are currently not applicable to DL since the behavior is not explicitly programmed in the software code. Therefore, IEC61508 states that AI/ML is not recommended for safety-critical functions.

Compliance with safety standards is usually among the building blocks in a *safety case*, which is one of the most important deliverables for creating confidence in safety-critical systems, e.g. in industries like railway, aerospace, and oil&gas.

6. Conclusions and Further Work

We assume that general autonomous ships are far into the future, whereas semi-autonomous ships in controlled and restricted environments seem feasible even in short term, with a remote, land-based human operator in the loop who can intervene if warned.

There are several challenges related to a having a robust situational awareness. Correct object classification in images is one crucial challenge, and classifying wrongly with a high confidence (or high score/probability) puts safety at stake. To resolve high confidence mistakes and increase trust in such systems, we first need to understand why they are made in the first place to be able to correct the mistakes, to debug, that is.

The emerging field of *explainable AI* proposes already many techniques. In particular, we see some benefit in heat maps, and further work is to evaluate heat maps on our database of maritime video frames with classified objects.

Assurance of DL-based safety-critical autonomous systems will of course require many additional techniques from other fields such as safety engineering and software engineering.

References

[1]. A. Brandsæter, K. E. Knutsen, Towards a framework for assurance of autonomous navigation systems in the maritime industry, in *Proceedings of the European Safety and Reliability Conference (ESREL’18)*, Trondheim, Norway, 17-21 June 2018.

[2]. Kongsberg, «Autonomous Shipping», https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/597733F8A1B8C640C12580AC0049C134?OpenDocument

[3]. B. J. Vartdal, R. Skjong, A. L. St. Clair, Remote-controlled and autonomous ships in the maritime industry, *DNV GL Position Paper*, 2018.

[4]. DNV GL RP, Assurance of Data-Driven Algorithms and Models – Recommended Practice, DNV GL-RP-0510, *DNV GL Report,* 2018.

[5]. A. Kazemi, D.V. Lindberg. Explainable Machine Learning – Manual for Interpreting Machine Learning

[1] https://www.iec.ch/functionalsafety/

Models, Report No.: 2018-0795, Rev. 1, *DNV GL Technical*, 2018.

[6]. T. A. Pedersen, J. A. Glomsrud, N. Husteli, Exploring System Verification for Autonomous Applications, 2017-1245 Rev. 0, *DNV GL Report*, 2017.

[7]. S. Eldevik, et al., AI + safety: Safety implications for artificial intelligence. why we need to combine causal- and data-driven models, *DNV GL Position Paper*, 2018, https://ai-and-safety.dnvgl.com/

[8]. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, 2012, https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[9]. NTSB, Preliminary report highway HWY18MH010, *National Transportation Safety Board*, 25 May 2018.

[10]. D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, D. Mane, Concrete problems in AI safety, 2016, https://arxiv.org/abs/1606.06565

[11]. I. Goodfellow, P. McDaniel, N. Papernot, Making machine learning robust against adversarial inputs, *Communications of the ACM*, Vol. 61, Issue 7, 2018, pp. 56-66.

[12]. G. Bansal, D. S. Weld, A Coverage-based utility model for identifying unknown unknowns, in *Proceedings of the Thirty-Second Conference on Artificial Intelligence (AAAI'18),* 2018, http://aiweb.cs.washington.edu/ai/pubs/bansal-aaai18.pdf

[13]. Ø. Helgesen, E. Piene, A. L. Flåten, Sensors for Autonomous Navigation, *DNV GL Internal Report,* 2017 (unpublished).

[14]. J. Xie, G. Hamre, E. Stensrud, B. Raeissi, Automated crack detection for drone-based inspection using convolutional neural network, in *Proceedings of the 17th Int. Conference on Computer and IT Applications in the Maritime Industries (COMPIT'18)*, Pavone, 14-16 May 2018, http://data.hiper-conf.info/compit2018_pavone.pdf

[15]. E. J. Tangstad, Visual detection of maritime vessels, MD Thesis, *University of Science and Technology*, Norwegian, 2017.

[16]. V. Kamsvåg, Fusion between camera and lidar for autonomous surface vehicles, MSc Thesis, *NTNU*, 2018.

[17]. W. Samek, A. Binder, G. Montavon, G. Lapuschkin, K. R. Muller, Evaluating the visualization of what a deep neural network has learned, *IEEE Trans. on Neural Networks and Learning Systems*, Vol. 28, Issue 11, 2017, pp. 2660-2673.

[18]. M. T. Ribeiro, S. Singh, C. Guestrin, Model-agnostic interpretability of machine learning, *arXiv:1606.05386*, 2016.

[19]. A. Rauber, R. Trasarti, F. Gianottti, Transparency in Algorithmic Decision Making, *ERCIM News*, 2019, https://ercim-news.ercim.eu/en116/special/transparency-in-algorithmic-decision-making-introduction-to-the-special-theme

[20]. A. Shrikumar, P. Greenside, A. Kund, Learning important features through propagating activation differences, *CoRR*, *abs/1704.02685*, 2017.

[21]. M. T. Ribeiro, S. Singh, C. Guestrin, Why should i trust you?: Explaining the predictions of any classifier, in *Proceedings of the 22nd ACM SIGKDD Int. Conference. on Knowledge Discovery and Data Mining,* 2016, pp. 1135-1144.

[22]. J. A. Adebayo, FairML: ToolBox for diagnosing bias in predictive modeling, *Brigham Young University*, 2012.

[23]. L. S. Shapley, A value for n-person games, in Contributions to the Theory of Games, *Princeton University Press*, 1953.

[24]. C. Molnar, Interpretable Machine Learning, 2018. https://christophm.github.io/interpretable-ml-book/

[25]. S. M. Lundberg, S. Lee, A unified approach to interpreting model predictions, in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS'17)*, Long Beach, CA, USA, https://arxiv.org/pdf/1705.07874.pdf

[26]. N. Smuha, Draft Ethics Guidelines for Trustworthy AI, Directorate-General for Communication, *High-Level Expert Group on Artificial Intelligence, European Commission*, Brussels, 18 December 2018, https://ec.europa.eu/digital-single-market/en/high-level-expert-group-artificial-intelligence

[27]. T. Miller, Explanation in artificial intelligence: Insights from the social sciences, *CoRR*, *abs/1706.07269*, 2017.

[28]. D. Gunning, Explainable Artificial Intelligence (XAI), https://www.darpa.mil/program/explainable-artificial-intelligence.

[29]. J. Yosinski, J. Clune, A. Nguyen, T. Fucks, H. Lipson, Understanding neural networks through deep visualization, in *Proceedings of the Int. Conference on Machine Learning (ICML'15) Workshop on Deep Learning*, 2015.

[30]. P. Rajpurkar, et al., CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning, *arXiv:1711.05225v3,* Dec. 2017.

[31]. S. Bach, A. Binder, G. Montavon, F. Klauschen, K. R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, *PLoS ONE*, Vol. 10, Issue 7, 2015, e0130140.

[32]. G. Montavon, W. Samek, K. R. Müller, Methods for interpreting and understanding deep neural networks, *Digital Signal Processing*, Vol. 73, 2018, pp. 1-15.

(38)

Intelligent Diagnosis and Signal Processing of Vibration Signal from Rotating Machinery

Steven Y. Liang [1,2], **Yanfei Lu** [1] **and Rui Xie** [3]
[1] George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology,
Atlanta, GA 30332, USA
[2] College of Mechanical Engineering, Donghua University, Shanghai, 201620, China
[3] Department of Statistics at the University of Georgia, Athens, GA 30602, USA
Tel.: +1-404-894-8164, fax: +1-404-894-9342
E-mail: steven.liang@me.gatech.edu

Summary: The accurate diagnosis of the health conditions of rotating machineries remains a challenge to various industry and research facilities. One of the most common failure of the rotating machines is the bearing failure. The timely detection of the incipient fault and failure location are critical to achieve successful diagnosis and prognosis of rotating machinery. In traditional maintenance scheme, the components of the machine are inspected based on a prescribed interval. However, as the machinery becomes more complex, the time-based maintenance schedule cannot satisfy the demand of the industry. With the booming of the machine learning and artificial intelligence, more learning based techniques are combined with signal processing techniques to address the limitations of the non-adaptive parameter and improve the training efficiency. Some of the state-of-the-art bearing diagnostic algorithms are reviewed with suggestions of the direction of future research.

Keywords: Adaptive algorithm, Artificial intelligence, Ball bearing, Fault diagnosis.

1. Introduction

As the newly developed artificially intelligence and signal processing techniques emerge, an increasing number of research have been dedicated to implementing intelligent algorithms to prolong the lifespan of rotating machinery and prevent accidents. One of the major challenge machinery industry encounters frequently is the failure of bearing components. In the early days, the time-based preventive maintenance is widely implemented because of the advantages of ease of planning and reduction of catastrophic failure. However, a small amount of unforeseen failure could still occur before the maintenance interval. A preventive maintenance scheme is demanded across various industries. With the introduction of condition-based monitoring (CBM), the occurrence of catastrophic events and maintenance cost has effectively reduced by 60 %. The key to the successful implementation of CBM hinges on the accurate diagnosis of machinery fault. Over the last few decades, various signal processing techniques and machine learning algorithms have been implemented by industry and researchers to increase the diagnostic accuracy. The diagnosis of the vibration signal from the machinery is one of the most popular methods to use because of the low cost, accuracy, and wide applicability. The classical methods to perform the diagnosis can be divided into three categories: time domain analysis, frequency domain analysis, and time-frequency analysis. In general, the time domain signal reveals the trend of degradation, and the frequency domain signal indicates the fault location. By the combination of these two analyses, the time-frequency analysis is more preferable among the three domains. However, the existing methods without intelligent algorithms require human intervention in parameter tuning, and most of the algorithms are not self-adaptive to changes of the vibration signal. With the advancement of machine learning algorithms, such as neural network (NN), support vector machines (SVM), dictionary learning and Bayesian hierarchical modeling, were implemented to facilitate the diagnosis of machinery fault with big data. Various improvements based on existing methods such as empirical mode decomposition (EMD) and wavelet transform are also proposed. Those machine learning algorithm tackles the challenges where the traditional methods fail to address. By effectively extract features from the vibration signals, the machine learning algorithms learn the underlying structure and characteristics automatically from observed signals. Additionally, some of the techniques are more effective to a specific signal than the others. In this paper, the latest abovementioned methods will be reviewed to provide a general guideline of selecting appropriate diagnosis method. In addition, a few recently developed methods utilizing the advantages of intelligent algorithms will be included to provide new research directions in the implementation of intelligent methods in signal processing.

2. Various Techniques

Neural network (NN) is one of the earliest self-learning algorithm implemented to solve problems from complex systems. In the early documented research, neural network has been implemented as a classification tool to diagnose the health condition of machines [1]. However, the application is only limited to distinguish defective bearings from healthy bearing. In addition, the developed algorithm is very sensitive to the location of fault, types of bearings, and actual running condition of bearings. Gebraeel *et al.* and Huang *et al.* implemented the NN in predicting the remaining life of bearings in [2, 3]. The algorithm shows good prediction result for the same type of bearings tested under overloading condition as shown in Fig. 1.

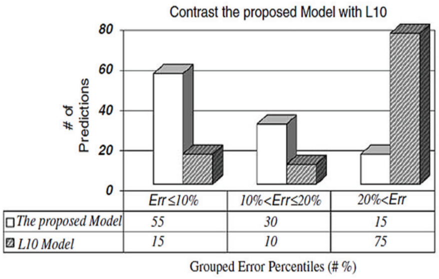
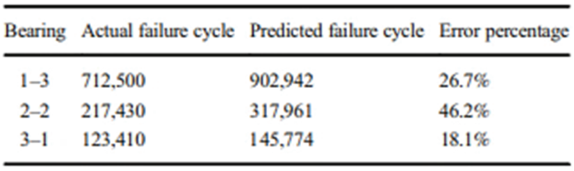


Fig. 1. Proposed model in comparison with tradition L_{10} model [3]

However, the generalization of this technique used for various type of bearings is uncertain. Most of the early documented research focuses on diagnose single fault on specific loading conditions. Later research focuses on combining newly developed preprocessing techniques with intelligent algorithms to improve the efficiency of the algorithm while minimizing the sacrifice of the accuracy of prediction. Prieto *et al.* implemented the curvilinear component analysis (CCA) as a selection tool to distinguish most important features extracted from bearing signal. By using the combined CCA and NN, they have extend the capability of the algorithm into identifying different types of fault under loading conditions with small variations [4]. Moura *et al.* implemented principal components analysis and NN to study the degradation severity of bearings with artificially introduced defect [5]. Two loading conditions and frequencies are considered in the validation of the proposed algorithm. However, the actual applicability of the proposed method has not been validated as the artificially introduced defect does not necessarily represent the actual defect in bearings. Bin *et al.* implemented empirical decomposition in combination with NN to identify failure mode of rotating machines [6]. The method is successful in diagnosing various type of failures. However, the multi failure scenario testing is not included during the study. Lu *et al.* developed a wavelet neural network to predict the remaining life of bearings [7]. The method is easy to implement, however, the error percentage could be large under certain operating conditions of the bearings as shown in Table 1.

A more detailed classification algorithm need to be added to improve the model performance in the future studies.

Table 1. Prediction result [7].



Bearing	Actual failure cycle	Predicted failure cycle	Error percentage
1–3	712,500	902,942	26.7%
2–2	217,430	317,961	46.2%
3–1	123,410	145,774	18.1%

In the recent years, deep learning NN has been applied to the various bearing fault diagnostic application to improve the signal to noise ratio and enhance the prediction accuracy. In addition, the automatic feature extraction features of deep learning algorithms significantly reduce the interruption required for offline diagnosis. With several algorithms aiming to improve the computation efficiency, the deep learning algorithm has become more favorable in analyzing vibration signals. Gan et al. developed a hierarchical network capable of predicting types of fault and defect severity [8]. The method overcame the ambiguous classification difficult resulted from traditional NN and support vector machines as show in Fig. 2.

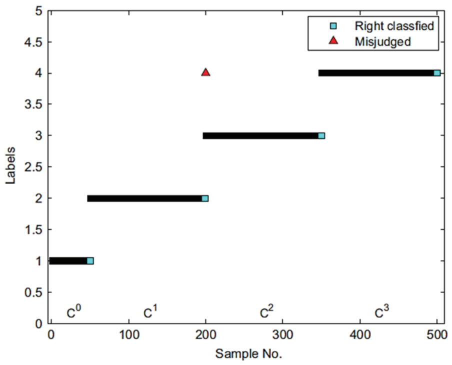


Fig. 2. Improved classification by deep learning [8].

Jia *et al.* proposed a deep learning algorithm that is capable of adaptation and solving highly non-linear problem in condition monitoring of rotating machinery [9]. The algorithm considers the fault location and defect size from bearing vibration data. In addition, improved accuracy is achieved by the added layers. Guo *et al.* proposed a deep convolution neural network for bearing fault classification and fault size identification in [10]. The method shows similar improved result in comparison to [9]. But it employed an adaptive learning rate and implemented more layers of the network. Other similar deep learning implement in bearing diagnosis can be found in [11-13]. The most challenging part of the deep learning algorithm is the parameter tuning. The future study will be focusing on improving computation efficiency, develop new algorithm to determine best network structure and parameters.

Dictionary learning, originally developed to process multi-dimensional data [14], is another efficient processing techniques for detecting the bearing fault at the early stage [15-17]. Its performance is highly dependent on the dictionary selection.

Although various algorithms exist to compute the dictionaries required to process the vibration signal, most of the algorithms are computationally heavy, which prevents the online application of dictionary learning. Adaptive algorithm combined with dictionary learning has been shown as an efficient signal processing technique of bearing diagnosis. Liu *et al.* implement dictionary learning for fault classification in [16]. The algorithm demonstrate good classification accuracy and robustness to varying loading conditions. Lu *et al.* proposed a dictionary updating scheme using the weighted least square to reduce the computation effort during the updating scheme of dictionary learnings in [17]. It can be seen from Fig. 3 that a dictionary without updating is not suitable to diagnose the added harmonics of the fault frequency. The adaptive algorithm replaces the tedious training stage of dictionary learning and can be used to update the dictionaries with newly acquired signal from the system.

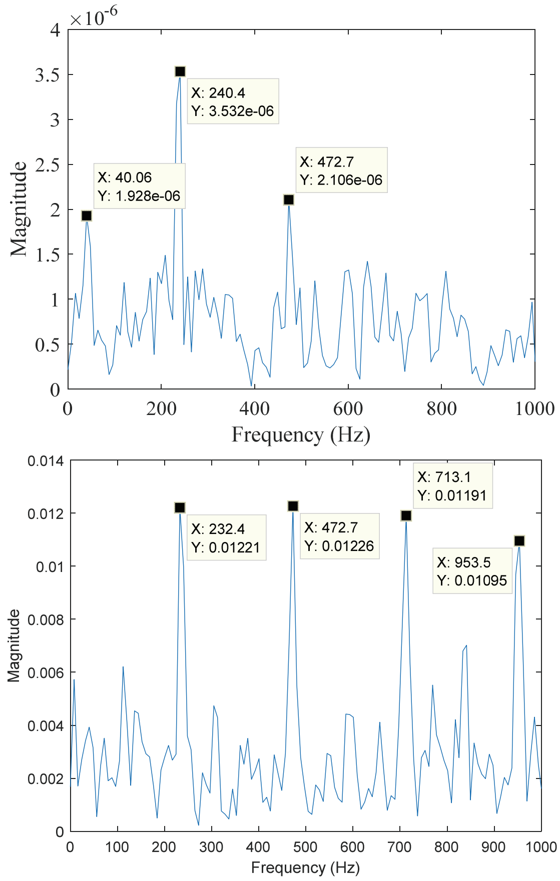


Fig. 3. Adaptive dictionary learning for bearing diagnosis [17].

Zhou *et al.* implemented a shift-invariant dictionary learning in combination with hidden Markov model to improve signal to noise ratio and distinguish failure mode [18]. Smith and Elad proposed a new dictionary updating algorithm in [18] to improve training speed and accuracy. More effective dictionary updating and parameter tuning algorithms still need to be developed to reduce the computation effort and improve the accuracy of diagnosis.

Empirical mode decomposition was initially developed to process nonlinear and non-stationary time series data [19]. It is still considered as one of the most popular diagnostic techniques nowadays [20]. Lei *et al.* wrote a comprehensive review for the evolution of the EMD algorithm in fault diagnosis in [21] with different variation of the EMD algorithm and their applications demonstrated. One of the most common issue with the EMD is the mode mixing. The complete ensembled empirical mode decomposition method (CEEMD) uses the white Gaussian noise to solve the mode mixing problem [22]. The reconstruction error of the original signal is significantly reduced as shown in Fig. 4.

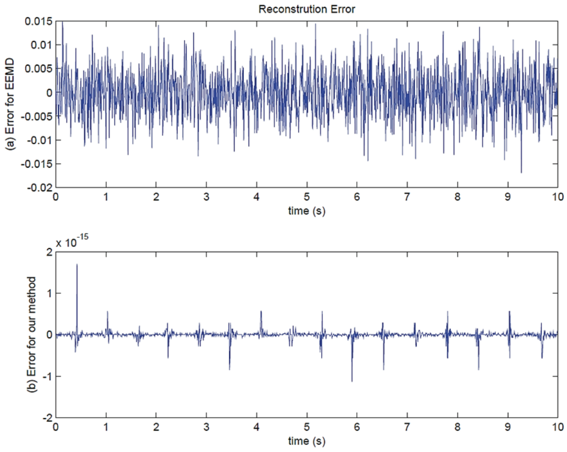


Fig. 4. Reconstruction error between EEMD and CEEMD [22].

The key to successfully implement the CEEMD method is selecting the right magnitude and standard deviation of the added white Gaussian noise. Bootstrap resampling method can be implemented to facilitate the parameter selection of the CEEMD method. Another requirement for the successful implementation of the EMD algorithm is the understanding of the system behavior. However, with increasing number of intelligent algorithm developed, such a requirement may not be necessary [23]. The last problem with the EMD algorithm is the relatively long computation time. The future research should be focusing on developing fast algorithms to achieve the decomposition of various modes.

Wavelet transform and decomposition are effective in the detection of weak fault signatures of rotating machineries [24]. Dual tree wavelet and empirical wavelet are of the few latest wavelets implemented in fault diagnostic applications [25-28]. The wavelet shape and parameter selection still remains as challenging topics. Sparsity guided techniques are implemented to enhance the selection of required parameters of different wavelet techniques [29]. The enery to Shannon entropy ratio is also shown as an effective approach in wavelet parameter selection [30].

The most suitable wavelet shape and parameters for specific fault diagnosis still remains undetermined.

Many other hybrid methods combining several signal processing and optimization algorithms are also considered effective in signal processing [31, 32]. However, the challenge for machinery diagnosis still remains. More accurate diagnostic algorithms which require less data while maintaining the accuracy in determining fault location, fault size and fault severity is always demanded.

3. Conclusion

With more advanced learning algorithm develops, the optimization of model parameters can be completed through accumulating experience data in the database and discovering adaptive algorithm. Different fault types will be better categorized and well understood. The detection accuracy can be improved through more efficient and intelligent techniques with minimum human intervention and knowledge of the mechanical systems to accommodate for various types and operating conditions of rotating machineries.

References

[1]. B. Li, et al., Neural-network-based motor rolling bearing fault diagnosis, *IEEE Transactions on Industrial Electronics*, 47, Vol. 5, 2000, pp. 1060-1069.

[2]. N. Gebraeel, et al., Residual life predictions from vibration-based degradation signals: a neural network approach, *IEEE Transactions on Industrial Electronics*, Vol. 51, Issue 3, 2004, pp. 694-700.

[3]. R. Huang, et al., Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods, *Mechanical Systems and Signal Processing*, Vol. 21, Issue 1, 2007, pp. 193-207.

[4]. M. D. Prieto, et al., Bearing fault detection by a novel condition-monitoring scheme based on statistical-time features and neural networks, *IEEE Transactions on Industrial Electronics*, Vol. 60, Issue 8, 2013, pp. 3398-3407.

[5]. E. De Moura, et al., Evaluation of principal component analysis and neural network performance for bearing fault diagnosis from vibration signal processed by RS and DF analyses, *Mechanical Systems and Signal Processing*, Vol. 25, Issue 5, 2011, pp. 1765-1772.

[6]. G. Bin, et al., Early fault diagnosis of rotating machinery based on wavelet packets – Empirical mode decomposition feature extraction and neural network, *Mechanical Systems and Signal Processing*, Vol. 27, 2012, pp. 696-711.

[7]. Y. Lu, Q. Li, S.Y. Liang, Physics-based intelligent prognosis for rolling bearing with fault feature extraction, *The International Journal of Advanced Manufacturing Technology*, 2018, pp. 1-10.

[8]. M. Gan, C. Wang, Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings, *Mechanical Systems and Signal Processing*, Vol. 72, 2016, pp. 92-104.

[9]. F. Jia, et al., Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data, *Mechanical Systems and Signal Processing*, Vol. 72, 2016, pp. 303-315.

[10]. X. Guo, L. Chen, C. Shen, Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis, *Measurement*, Vol. 93, 2016, pp. 490-502.

[11]. P. Tamilselvan, P. Wang, Failure diagnosis using deep belief learning based health state classification, *Reliability Engineering & System Safety*, Vol. 115, 2013, pp. 124-135.

[12]. M. He, D. He, Deep learning based approach for bearing fault diagnosis, *IEEE Transactions on Industry Applications*, Vol. 53, Issue 3, 2017, pp. 3057-3065.

[13]. H. Shao, et al., A novel deep autoencoder feature learning method for rotating machinery fault diagnosis, *Mechanical Systems and Signal Processing*, Vol. 95, 2017, pp. 187-204.

[14]. J. Mairal, et al. Online dictionary learning for sparse coding, in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML'09),* 2009, pp. 689-696.

[15]. Z. Feng, M. Liang, Complex signal analysis for planetary gearbox fault diagnosis via shift invariant dictionary learning, *Measurement*, Vol. 90, 2016, pp. 382-395.

[16]. H. Liu, C. Liu, Y. Huang, Adaptive feature extraction using sparse coding for machinery fault diagnosis, *Mechanical Systems and Signal Processing*, Vol. 25, Issue 2, 2011, pp. 558-574.

[17]. Y. Lu, R. Xie, S. Y. Liang, Adaptive online dictionary learning for bearing fault diagnosis, *The International Journal of Advanced Manufacturing Technology*, 2018, pp. 1-8.

[18]. H. Zhou, et al., Detection and diagnosis of bearing faults using shift-invariant dictionary learning and hidden Markov model, *Mechanical Systems and Signal Processing*, Vol. 72, 2016, pp. 65-79.

[19]. N. E. Huang, et al., The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 1998.

[20]. J. Dybała, R. Zimroz, Rolling bearing diagnosing method based on empirical mode decomposition of machine vibration signal, *Applied Acoustics*, Vol. 77, 2014, pp. 195-203.

[21]. Y. Lei, et al., A review on empirical mode decomposition in fault diagnosis of rotating machinery, *Mechanical Systems and Signal Processing*, Vol. 35, Isuue 1-2, 2013, pp. 108-126.

[22]. M. E. Torres, et al., A complete ensemble empirical mode decomposition with adaptive noise, in *Proceedings of IEEE International Conference on the Acoustics, Speech and Signal Processing (ICASSP'11),* 2011, pp. 4144-4147.

[23]. X. Qiu, et al., Empirical mode decomposition based ensemble deep learning for load demand time series forecasting, *Applied Soft Computing*, Vol. 54, 2017, pp. 246-255.

[24]. H. Qiu, et al., Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics, *Journal of Sound and Vibration*, Vol. 289, Issue 4-5, 2006, pp. 1066-1090.

[25]. Y. Lu, R. Xie, S. Y. Liang, Detection of weak fault using sparse empirical wavelet transform for cyclic fault, *The International Journal of Advanced Manufacturing Technology*, Vol. 99, Issue 5-8, 2018, pp. 1195-1201.

[26]. Y. Wang, Z. He, Y. Zi, Enhancement of signal denoising and multiple fault signatures detecting in rotating machinery using dual-tree complex wavelet transform, *Mechanical Systems and Signal Processing*, Vol. 24, Issue 1, 2010, pp. 119-137.

[27]. J. Qu, Z. Zhang, T. Gong, A novel intelligent method for mechanical fault diagnosis based on dual-tree complex wavelet packet transform and multiple classifier fusion, *Neurocomputing*, Vol. 171, 2016, pp. 837-853.

[28]. M. Kedadouche, M. Thomas, A. Tahan, A comparative study between empirical wavelet transforms and empirical mode decomposition methods: application to bearing defect diagnosis, *Mechanical Systems and Signal Processing*, Vol. 81, 2016, pp. 88-107.

[29]. G. Cai, X. Chen, Z. He, Sparsity-enabled signal decomposition using tunable Q-factor wavelet transform for fault feature extraction of gearbox, *Mechanical Systems and Signal Processing*, Vol. 41, Issue 1-2, 2013, pp. 34-53.

[30]. R. Yan, R. X. Gao, Base wavelet selection for bearing vibration signal analysis, *International Journal of Wavelets, Multiresolution and Information Processing,* Vol. 7, Issue 04, 2009, pp. 411-426.

[31]. Y. Gao, et al., Multi-Scale permutation entropy based on improved LMD and HMM for rolling bearing diagnosis, *Entropy*, Vol. 19, Issue 4, 2017, 176.

[32]. P. Shakya, M. S. Kulkarni, A. K. Darpe, Bearing diagnosis based on Mahalanobis–Taguchi–Gram–Schmidt method, *Journal of Sound and Vibration*, Vol. 337, 2015, pp. 342-362.

(39)

Improving Reliability Predictions by Stacking

W. You [1], **A. Saidi** [2], **A. Zine**[2] **and M. Ichchou** [1]
[1] Ecole Centrale de Lyon, LTDS, 36 Av. Guy de Collongue, 69134 Écully, France
[2] Ecole Centrale de Lyon, MI, 36 Av. Guy de Collongue, 69134 Écully, France
Tel.: + 33 4 72 18 62 30, fax: +33 4 72 18 91 44
E-mail: mohamed.ichchou@ec-lyon.fr

Summary: Reliability prediction plays a significant role in structural design and improvement process. Uncertainty in structural properties has made reliability predictions more difficult to achieve. In the machine learning perspective, the prediction task can be converted into a regression problem. Regressors such as tree methods always behave well, but further explorations should be carried out on improving the predictions. Stacking method is a way to build the prediction model in a hierarchical way, resulting in a meta learner inferred from a series of base learners. Our recent research shows that, with a relatively small price of CPU time, the Stacking method can easily improve reliability predictions by combining the individual base learners. Simulation results are provided and discussed on different settings of Stacking. The time complexity is also briefly analyzed.

Keywords: Structural reliability, Prediction, Machine learning, Stacking, Base learner.

1. Introduction

Structural reliability describes the probability that the object structure realizes its functions under given conditions for a specified time period [1]. For mechanical structures, an important task is to control the structural vibrations that may induce damages. In this aspect, tuned mass dampers (TMDs) are a widely used technique to reduce the risk of structure failures, in other words, increase the structural reliability. As a way to improve the quality of products, reliability prediction helps companies make product planning and implement preventive maintenance. Before predictions, a reliability model should be firstly determined. Due to complex factors (environmental, fatigue...), the structural properties become more related to uncertainties. To improve reliability predictions under structural uncertainties, growing attention has been paid to those non-parametric statistical learning approaches, such as Support vector machines [2] and Neural networks [3]. In a statistical learning view, the prediction can be seen as a regression problem. According to the current research state, we believe that the explorations are far from enough. As a way to reduce the errors of single models, Stacking (Stacked Generalization) [4] is a powerful machine learning method that combines different base models to improve the prediction results. Therefore, in this research, we introduce the application of Stacking into structural reliability predictions and explore its pros and cons with respect to individual base models. In the numerical simulations, the reliability modeling and prediction is carried out on a TMD-based passive control structure that is excited by seismic accelerations [5].

2. Structural Dynamic Model

A typical structure (see Fig. 1) consists of an N-DOF (Degree of Freedom) [6] main structure and an n-DOF TMD structure. The acceleration $\ddot{y}_b$ is provided by Kanai-Tajimi (KT) model [7]. Let S_0, ξ_f, ω_f and y_f be the intensity of the Gaussian white noise process [8], the damping ratio, the natural frequency and the relative response of KT model, respectively. By considering the displacements of all the DOFs of the structure, the global state space vector is built as

$$\mathbf{Z} = [y_{s_1},...,y_{s_N}, y_{T_1},...,y_{T_n}, y_f,$$
$$\dot{y}_{s_1},...,\dot{y}_{s_N}, \dot{y}_{T_1},...,\dot{y}_{T_n}, \dot{y}_f\ ]$$

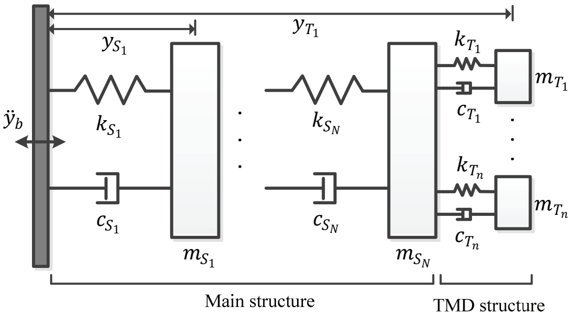


Fig. 1. Multi-DOF TMD system.

Then the motion of the system becomes

$$\dot{\mathbf{Z}} = \mathbf{AZ} + \mathbf{f}\ , \tag{1}$$

where $\mathbf{f} = [0, ..., 0, -w(t)]^T_{1\times(2N+2n+2)}$, $w(t)$ is the noise process. $\mathbf{A} = [\mathbf{O}_{N+n+1}, \mathbf{\Lambda}_{N+n+1}; \mathbf{K}_2, \mathbf{K}_3]$, $\mathbf{K}_2 = [\mathbf{K}(:,1:N+n)\ ,\ [\omega_f^2]_{(N+n)\times 1}; [0,0,...,0,-\omega_f^2]]$, $\mathbf{K}_3 = [\mathbf{K}(:,N+n+1:2N+2n)$, $[2\xi_f\omega_f]_{(N+n)\times 1}; [0,0,...,0,-2\xi_f\omega_f]]$. $\mathbf{K} = [\mathbf{K}_0, \mathbf{A}_1, \mathbf{C}_0, \mathbf{A}_2\ ;\ \mathbf{B}_1, \mathbf{K}_1, \mathbf{B}_2, \mathbf{C}_1]_{(N+n)\times(2N+2n)}$, $\mathbf{K}_0 = -\mathbf{M}_s^{-1}\cdot\mathbf{K}_s$; $\mathbf{C}_0 = -\mathbf{M}_s^{-1}\cdot\mathbf{C}_s$, $\mathbf{K}_1 = -\mathbf{M}_T^{-1}\mathbf{K}_T$, $\mathbf{C}_1 = -\mathbf{M}_T^{-1}\mathbf{C}_T$. $\mathbf{A}_1 = [\mathbf{O}_{(N-1)\times n}; 1/m_{S_n}\cdot[k_{T_1},...,k_{T_n}]]$, $\mathbf{A}_2 = [\mathbf{O}_{(N-1)\times n}\ ;\ 1/m_{S_N}\cdot[c_{T_1},...,c_{T_n}]]$; $\mathbf{B}_1 = [\mathbf{O}_{n\times(N-1)}, [k_{T_1}/m_{T_1}\ ;\ ..., k_{T_n}/m_{T_n}]^T]$, $\mathbf{B}_2 = [\mathbf{O}_{n\times(N-1)}, [c_{T_1}/m_{T_1}, ..., c_{T_n}/m_{T_n}]^T]$. $\mathbf{M}_s$, $\mathbf{M}_T$, $\mathbf{C}_s$, $\mathbf{C}_T$, $\mathbf{K}_s$, $\mathbf{K}_T$ are the mass matrix, damping matrix and stiffness matrix of the base structure and TMD structure respectively. The standard state space expressions are built to simulate the response process, i.e.

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}\end{aligned} \tag{2}$$

Here $\mathbf{B}\mathbf{u} = \mathbf{f}$, $\mathbf{C}$ is identity matrix $\mathbf{\Lambda}_{2N+2n+2}$, $\mathbf{D} = \mathbf{0}$.

3. The Proposed Reliability Model

3.1. Problem Statement

Figs. 2 and 3 display the example structures in deterministic and stochastic cases respectively. The two structures all have 1-DOF main structure and 1-DOF TMD structure. In deterministic case, all the structural properties are certain. The system has a base acceleration $\ddot{y}_b$ due to seismic excitation which is approximated by the KT model. Then, the base acceleration satisfies the following equations:

$$\begin{cases}\ddot{y}_f + 2\xi_f\omega_f\,\dot{y}_f + \omega_f^2 y_f = -w(t) \\ \ddot{y}_b = \ddot{y}_f + w(t) = -(2\xi_f\omega_f\,\dot{y}_f + \omega_f^2 y_f)\end{cases}, \tag{3}$$

where $w(t)$ is a stationary zero mean white noise process. It represents the excitation of the dynamic system. ω_f and ξ_f are the natural frequency and the damping ratio of the KT model, and y_f is its relative response [9]. With Monte-Carlo simulations, the reliability evaluation is carried out, resulting in a failure probability $P_f = p$.

However, when uncertainties are added to the base structure (see Fig. 3), the resulting P_f becomes uncertain. To manage this problem, we propose to build a reliability model that is actually a mapping relationship from the uncertainty space of structural properties to the uncertainty space of structure failure probability. This model can also be employed to predict failure probabilities for unknown points within the uncertainty space of structural properties.

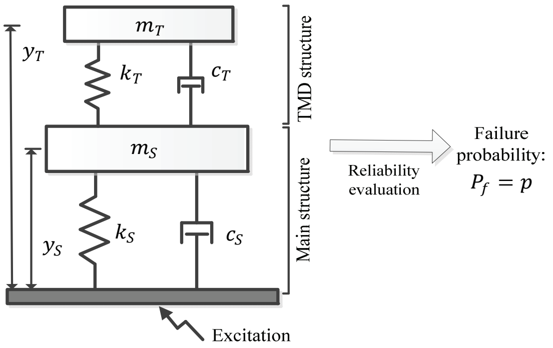


Fig. 2. 1-DOF main structure: deterministic case.

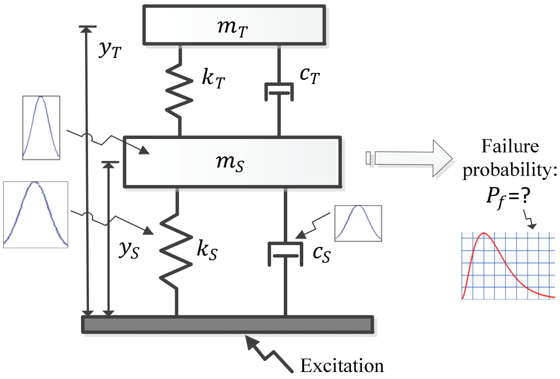


Fig. 3. 1-DOF main structure: stochastic case.

3.2. General Framework

Fig. 4 provides a framework that illustrates the procedures to build the reliability model as well as make predictions. We assume that the uncertainties only exist in the main structure. To prepare the reliability data, we firstly take samples of the structural properties according to their uncertainty characteristics [10]. For each sample, we apply Monte-Carlo simulations (MCS) [11] to obtain the corresponding failure probability.

3.3. MCS for Reliability Evaluation

The failure probability is calculated as the first-passage probability. In vibration analysis of structural systems subject to uncertain excitation modeled as stochastic process, the first-passage probability [12] is the probability that the system's response stays within safe, prescribed limits, within a specified time span. Mathematically, it is expressed as

$$P_f = P\big(Y(t) > c \mid t \in [0, T]\big), \tag{4}$$

where $Y(t)$ is the amplitude of the structure response at time t. T is the size of time span. c is the preset limit that denotes the safe region for structure response.

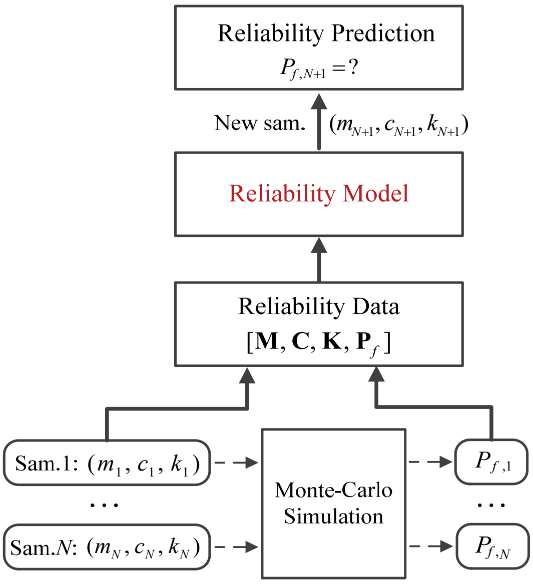


Fig. 4. Framework of structural reliability modeling & prediction: 1-DOF case. Here N samples are taken, each sample has three variables m, c and k.

In simulation methods, the stochastic excitation is specified as a certain number of input random variables [13], i.e. $\mathbf{x}=[x_1, x_2, ..., x_n]$. Theoretically, the failure probability can be expressed as the following integral

$$\begin{aligned} P_f &= \int_F f(\mathbf{x})d\mathbf{x} \\ &= \int ... \int_{G(\mathbf{x})<0} f_{\mathbf{x}}(x_1, ..., x_n)dx_1 dx_2 ... dx_n \end{aligned}, \quad (5)$$

where $G(\mathbf{x})$ is the state function of the structure performance; $G(\mathbf{x}) \le 0$ denotes the failure region in n-dimension space of input variables. It is practically impossible to directly calculate this integration. Thus, MCS method is employed to approximate the failure probability

$$\hat{P_f} = E\left(I_F(\mathbf{x})\right) = \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} I\left(G(\mathbf{x}_i) < 0\right), \quad (6)$$

where $I_F(\cdot)$ is the indicator function: $I_F(\mathbf{x})=1$ if $\mathbf{x} \in F$ and $I_F(\mathbf{x})=0$ otherwise; $\mathbf{x}_i$ is the ith sample taken from the joint distribution; N_{mc} is the number of samples used in the MCS. The failure criterion is defined as $y_{\lim}$ so that $y_{\max} > y_{\lim}$ means 'failure'. Here $y_{\max} = \max(|y_S|)$ is the maximum value of response among all degrees of the base structure. More intuitively, the Monte-Carlo estimator of failure probability is written as

$$\hat{P_f} = \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} I_F\left(Z^{(i)}\right), \quad (7)$$

where $\mathbf{Z}^{(i)}$ is the ith sample of the excitation process; $I_F(\cdot)=1$ if $y_{\max} > y_{\lim}$; $I_F(\cdot)=0$, otherwise. By MCS, the reliability data will be collected. These data will be used as training data to train the Stacking model.

3.4. Reliability Modeling by Stacking

In a machine learning perspective, the samples are the input feature values of the model, while the failure probability values are the output (target) values. Stacking is an ensemble learning technique that learns a meta model from multiple base models.

In a typical implementation of Stacking [14], a number of base (first-level) learners are generated from the training data by employing different learning algorithms. Then, based on the first-level learners, a meta-learner (second-level) is infered. Once the meta-learner is built, the predictions can be made on new inputs. See the schematic in Fig. 5.

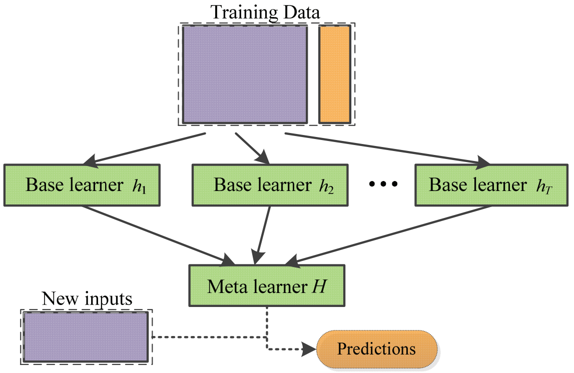


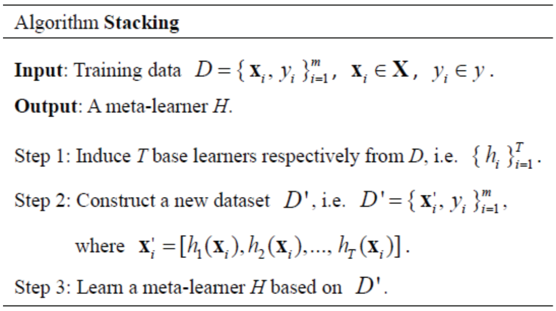
Fig. 5. Basic procedures of Stacking.

As is shown in Fig. 5, the training data consist of two parts, i.e. input (left part): samples of uncertain properties of the structure; output (right part): failure probabilities. We assume the training set has m cases, i.e. $D=\{\mathbf{x}_i, y_i\}_{i=1}^m$. A set of T base learners are induced from the data D. Then a meta-learner H is learned. See details in Table 1 and Fig. 6.

We can see that Stacking is a general framework. We can plug in different learning models even ensemble approaches to generate first/ second-level learners.

In Table 1, if we use the same data set D to train base learners and prepare the new training data D' for meta-learner, this may lead to over-fittings. To avoid this problem, cross-validation (CV) [16] method is incorporated in stacking. p-fold CV is mostly used technique. Fig. 7 illustrates an example to apply the CV (p = 5) procedures to create new features based on model h_1. The model is learned from 4 folds (i.e. 80 % of the training data), then it is employed to make predictions on the other one fold. All the predictions constitute the 1st column of **X′**.

Table 1. Pseudo-code of Stacking [15].



```
Algorithm Stacking
Input: Training data $D = \{\mathbf{x}_i, y_i\}_{i=1}^{m}$, $\mathbf{x}_i \in \mathbf{X}$, $y_i \in y$.
Output: A meta-learner $H$.
Step 1: Induce $T$ base learners respectively from $D$, i.e. $\{h_i\}_{i=1}^{T}$.
Step 2: Construct a new dataset $D'$, i.e. $D' = \{\mathbf{x}'_i, y_i\}_{i=1}^{m}$,
        where $\mathbf{x}'_i = [h_1(\mathbf{x}_i), h_2(\mathbf{x}_i), \dots, h_T(\mathbf{x}_i)]$.
Step 3: Learn a meta-learner $H$ based on $D'$.
```

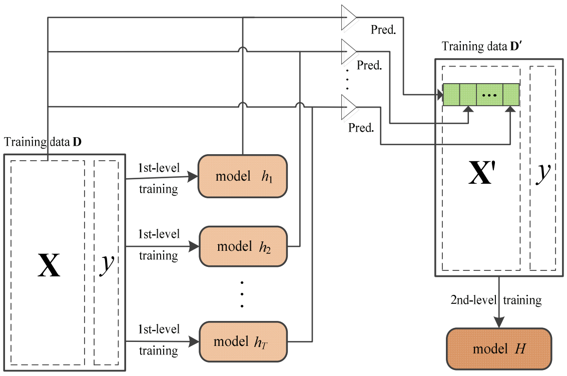


Fig. 6. Details of Stacking process.

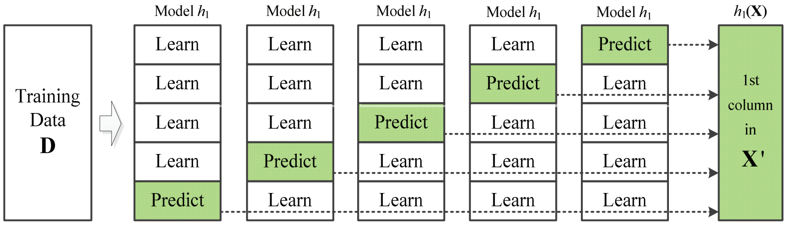


Fig. 7. Cross-validation (5-fold) to create new features.

4. Numerical Simulations

The Stacking method is tested on an object structure that consists of a 3-DOF main structure and a 2-DOF TMD. The uncertainties in the main structure only exist in the mass (*m*), damping factor (*c*), stiffness coefficient (*k*). The structure parameters are listed in Table 2. 'SD' means standard deviation.

The other parameters include $\omega_f = 8\pi$, $\xi_f = 0.4$, $S_0 = 0.031$. We have 20000 samples as training data and 1000 samples as test data. To show the evaluation results in a dynamic way, we change the size of training data by ranging it from 1000 to 20000 with step 1000. Then we train the base models as well as the meta models. Here the base learners are chosen as tree models such as Random Forest (RF) [17, 18], Extra-trees (ETs) [19] or Gradient Boosting (GB) [20, 21]. The meta model is GB. Root mean square error (RMSE) is employed to evaluate the model accuracy.

Table 2. Structure parameters ($i = 1, 2, 3$, $j = 1, 2$).

Index	m_{S_i}	c_{S_i}	k_{S_i}	m_{T_j}	c_{T_j}	k_{T_j}
Nominal	4.6	62	6500	1.38	1.83	39.0
SD	1	10	300	-	-	-

In Fig. 8, we compare the accuracy of different Stacking methods. We also compare the Stacking models with single models RF, ETs and GB. We see that the Stacking models are always outperforms the individual base learners especially when we have enough training data. The Stacking1 model always has smaller errors than its base learners RF and ETs. Similarly, the Stacking2 model is more accurate than its base learners RF, ETs and GB all the time. The two Stacking models take different sets of base learners. By comparison of the two Stacking models, it is clear that more base models will probably result in more accurate predictions. The three models RF, ET and GB are all based tree methods but different learning principles. The combination of them helps offset each other's weakness and improve the accuracy of the model. However, a large number of base learners may not be a wise choice, because the model building may be time-consuming. In view of this, we studied the time complexity [22] of Stacking method.

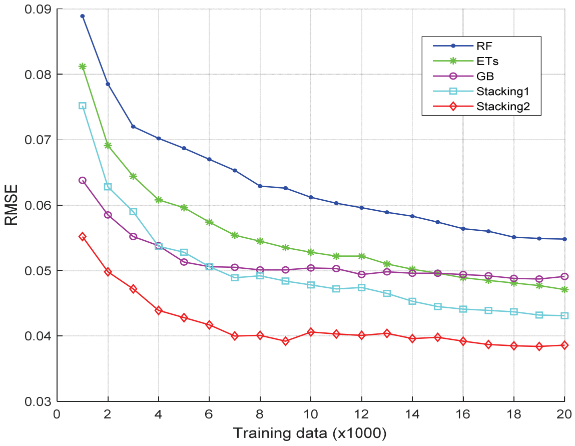


Fig. 8. Model evaluation results. Both Stacking1 and Stacking2 take GB as their meta-learner. The Stacking1 model takes RF&ETs as base learners; the Stacking2 model takes RF&ETs&GB as base learners.

Time complexity analysis

In the stacking model, we employed RF, ET and GB as the base learners, and GB as the meta-learner. To calculate time complexity, the following parameters are needed:
1) N: the number of samples in the training set;
2) The number of variables randomly drawn at each node: K_1(first-level), K_2(second-level);
3) The number of trees: M_1 (in RF), M_2 (ETs), M_3 (GB in 1st level) and M_4 (GB in the 2nd level);
4). N_1 is the number of cases in a bootstrap sample, so $N_1 \approx 0.63N$.

In the first-level training process, we train the RF, ETs and GB respectively. We apply 5-fold CV to make predictions on the training data. The time complexity is denoted as

$$T_1 = 5 \cdot (T_{RF} + T_{RF}^{'} + T_{ET} + T_{ET}^{'} + T_{GB} + T_{GB}^{'}) \,, \tag{8}$$

where T and $T^{'}$ are the time complexities to build the model and make predictions respectively. The six terms are calculated as follows [23],

$$\begin{aligned}
T_{RF} &= O(M_1 K_1 (0.8N_1) \log^2 (0.8N_1)),\\
T_{RF}^{'} &= O(0.2N_1 \cdot M_1 (0.8N_1)),\\
T_{ET} &= O(M_2 K_1 (0.8N) \log(0.8N)),\\
T_{ET}^{'} &= O(0.2N \cdot M_2 (0.8N)),\\
T_{GB} &= O\left(M_3 K_1 (0.8N) \log^2 (0.8N)\right) +\\
&\quad +(M_3 - 1) \cdot O\left(0.8N \cdot (0.8N)\right),\\
T_{GB}^{'} &= O(0.2N \cdot M_3 (0.8N)) =\\
&= (2.5M_1K_1 + 4M_3K_1 + M_4K_2) \cdot O(N \log^2 N) +\\
&\quad +(0.63M_1 + 0.8M_2 + 5M_3 + M_4 + 4M_2K_1 - 5) \cdot O(N \log N)\\
&= (2.5M_1K_1 + 4M_3K_1 + M_4K_2) \cdot O(N \log^2 N)
\end{aligned}$$

In the 2nd-level training process, we train the GB model using the data created by the 1st-level models

$$\begin{aligned}
T_2 &= O\left(M_4 K_2 N \log^2 N\right) +\\
&\quad +(M_4 - 1) \cdot O\left(N \cdot N\right)
\end{aligned} \tag{9}$$

Therefore, the total time complexity is denoted as

$$\begin{aligned}
T = T_1 + T_2 &= 5 \cdot O(M_1 K_1 0.8N_1 \log^2 (0.8N_1)) +\\
&+5 \cdot O(0.2N_1 \cdot M_1 \log(0.8N_1)) +\\
&+5 \cdot O(M_2 K_1 (0.8N) \log(0.8N)) +\\
&+5 \cdot O(0.2N \cdot M_2 \log(0.8N)) +\\
&+5 \cdot O(M_3 K_1 (0.8N) \log^2 (0.8N)) +\\
&+5(M_3 - 1) \cdot O(0.8N \cdot \log(0.8N)) +\\
&+5 \cdot O(0.2N \cdot M_3 \log(0.8N)) +\\
&+O(M_4 K_2 N \log^2 N) + (M_4 - 1) \cdot O(N_1 \log N)
\end{aligned} \tag{10}$$

We see that the time complexity T calculated above approximates a linear increment when N becomes large.

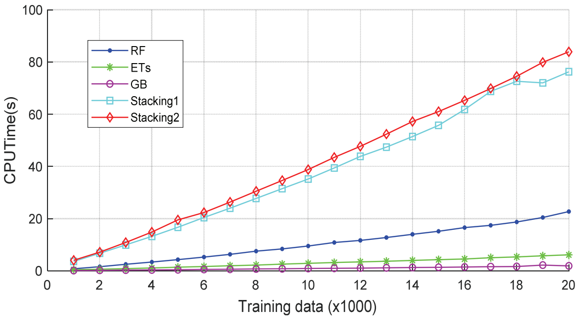


Fig. 9. Changes of CPUTime.

Fig. 9 shows the CPUTime spent to build the models involved in Fig. 9. In the process of learning a Stacking model, several base models need to be built, so it is reasonable that the Stacking models always take more time to build than single models do. The comparison of two Stacking models tells that the running time will increase when more base learners are added in the first-level training process. For each Stacking model, the running time increases in a close-to-linear way with respect to the training data. This is the experimental evidence for the assertation of time complexity made before.

5. Conclusions

In this study, Stacking method is explored for structural reliability modeling and predictions. The Stacking method builds the model in a hierarchical way, combining different base learners to produce a higher level learner that outperforms any of the base learners. The numerical simulations show that the Stacking method can easily outperform its base learners with a bearable CPUTime increment. In this study, Stacking models have effectively improved the accuracy of reliability predictions.

References

[1]. B. Bertsche, Reliability in Automotive and Mechanical Engineering: Determination of Component and System Reliability, *Springer Science & Business Media*, 2008.

[2]. M. Kutyłowska, Neural network approach for failure rate prediction, *Engineering Failure Analysis*, Vol. 47, 2015, pp. 41-48.

[3]. M. das Chagas Moura, E. Zio, I. D. Lins, E. Droguett, Failure and reliability prediction by support vector machines regression of time series data, *Reliability Engineering & System Safety*, Vol. 96, Issue 11, 2011, pp. 1527-1534.

[4]. D. H. Wolpert, Stacked generalization, *Neural Networks*, Vol. 5, Issue 2, 1992, pp. 241-260.

[5]. E. Mrabet, M. Guedri, M. Ichchou, S. Ghanmi, New approaches in reliability based optimization of tuned mass damper in presence of uncertain bounded

parameters, *Journal of Sound and Vibration*, Vol. 355, 2015, pp. 93-116.

[6]. R. S. Jangid, Optimum multiple tuned mass dampers for base-excited undamped system, *Earthquake Engineering & Structural Dynamics*, Vol. 28, Issue 9, 1999, pp. 1041-1049.

[7]. Y. K. Lin, Y. Yong, Evolutionary Kanai-Tajimi earthquake models, *Journal of Engineering Mechanics*, Vol. 113, Issue 8, 1987, pp. 1119-1137.

[8]. E. Mrabet, M. Guedri, M. N. Ichchou, S. Ghanmi, M. Soula, A new reliability based optimization of tuned mass damper parameters using energy approach, *Journal of Vibration and Control*, Vol. 24, Issue 1, 2018, pp. 153-170.

[9]. H. Yu, F. Gillot, M. Ichchou, Reliability based robust design optimization for tuned mass damper in passive vibration control of deterministic/uncertain structures, *Journal of Sound and Vibration*, Vol. 332, Issue 9, 2013, pp. 2222-2238.

[10]. I. Venanzi, Robust optimal design of tuned mass dampers for tall buildings with uncertain parameters, *Structural and Multidisciplinary Optimization*, Vol. 51, Issue 1, 2015, pp. 239-250.

[11]. E. Zio, The Monte Carlo Simulation Method for System Reliability and Risk Analysis, Vol. 39, *Springer*, 2013.

[12]. J. B. Roberts, First-passage probabilities for randomly excited systems: diffusion methods, *Probabilistic Engineering Mechanics*, Vol. 1, Issue 2, 1986, pp. 66-81.

[13]. S. K. Au, J. L. Beck, First excursion probabilities for linear systems by very efficient importance sampling, *Probabilistic Engineering Mechanics*, Vol. 16, Issue 3, 2001, pp. 193-207.

[14]. Z. H. Zhou, Ensemble Methods: Foundations and Algorithms, *Chapman and Hall/CRC*, 2012.

[15]. C. C. Aggarwal, Data Classification: Algorithms and Applications, *CRC Press*, 2014.

[16]. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, J. Vanderplas, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825-2830.

[17]. L. Breiman, Random forests, *Machine Learning*, Vol. 45, Issue 1, 2001, pp. 5-32.

[18]. A. Liaw, M. Wiener, Classification and regression by randomForest, *R News*, Vol. 2, Issue 3, 2002, pp. 18-22.

[19]. P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, *Machine Learning*, Vol. 63, Issue 1, 2006, pp. 3-42.

[20]. J. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning, *Springer*, 2001.

[21]. J. H. Friedman, Greedy function approximation: A gradient boosting machine, *Annals of Statistics*, 2001, pp. 1189-1232.

[22]. M. J. Kearns, The Computational Complexity of Machine Learning, *MIT Press*, 1990.

[23]. G. Louppe, Understanding random forests: From theory to practice, PhD Thesis, *University of Liège*, 2014.

(40)

Identifying Land Use Functions Based on POIs and Buildings Data

Yuhao Zhang [1], Xianhan Zeng [1], Zhicheng Liu [1], Yuming Qian [1], Junyan Yang [2] and Qiao Wang [1]

[1] School of Information Science and Engineering and Shing-Tung Yau Center, Southeast University, Nanjing, China

[2] School of Architecture, Southeast University, Nanjing, China

E-mails: yuhao_zhang@seu.deu.cn, xianhan_zeng@seu.edu.cn, zhichengliu@seu.edu.cn, yumingqian@hotmail.com, yjy-2@163.com, qiaowang@seu.edu.cn

Summary: We demonstrate in this paper a competitive method by using buildings data and POIs data to identify Land Use Functions (LUFs). Although the multi-sources data approach which includes social network data, taxi trajectories and remote sensing images, and so on, is successfully applied in characterizing mixed-use buildings, it only identifies few functions and the accuracy is not adequate. We provide a case study around analyzing Shenzhen with innovative machine learning framework, which attains excellent performance comparing to convenient multi-source data approach, in which 25 main functions of urban land use are all identified accurately with merely the buildings data and POIs data. In addition, we design a pipeline that can translate the massive and passive data to meaningful functional zones readily interpretable for urban planning and refined managing purposes. With shrinking labor and fiscal resources in the public sector globally, the method presented in this research can be used as a low-cost alternative for planning and managing agencies to identify the function of urban land zones, and provides targeted plans for future sustainable development.

Keywords: Urban functional zone identification, POI, KDE, Spatial join, TF-IDF, GBDT, Partial dependency.

1. Introduction

In many fields of city science, such as urban planning, human activity analysis and urban refined management, a basic problem is what function a certain urban region undertakes, namely, Land Use Functions (LUFs). Mapping land use and land cover can help urban planners analyze the value of urban land and dynamic patterns of urban space [1-4]. Generally, urban planners will make plan of land use for a city or region based on factors such as policy, environment, population, transportation, and future development. However, in actual urban development, regional functions are modified and adjusted due to spontaneous activities of people. The former is called the nature of planning land, and the latter is called the nature of the current land. On the other hand, the division of land is based not only on roads, but also on walls, trees, and even property rights. With the construction and transformation of the city, the flow of people, the replacement of the owners and the boundaries of the land are constantly changing. In this article, we focus on the automatic discovery of boundaries of the current land and the nature or function of the current land use. For identifying the function of each region, the usual method is to send surveyors to the streets for investigation every few years. This traditional method consumes considerable labor and material resources and cannot keep up with the speed of urban development. If the urban land use map always lags behind the reality, it will not be conducive to the government and enterprises to make scientific decisions, for example, the government cannot estimate whether the number of public facilities meets the needs of residents, the company miscalculates the expected return of investment, etc.

To-date many researches have tried to design ways to identify urban area functions automatically. Niu N [5] proposed a method to identify building functions using a variety of geographic data, with an accuracy rate of 65 %. Yao Y [6] took into account the effect of Point of Interest (POI) on regional functions, and used the Word2Vec model to identify urban land. It can identify about 15 land types with an accuracy of 0.87. Other research teams have discovered urban functions through unsupervised learning. For example, Yuan J [7, 8] used the LDA model to mine the potential behavior of people in taxi data to conduct an inductive analysis of the distribution of functional areas in Beijing; Sun Y [9] analyzed the subway site to explore the functional partition law of Shanghai through the Node2Vec model. However, no unsupervised models can give a good and intuitive clustering explanation, and the conclusions drawn are far from the actual classification in terms of quantity and type. Zhu Y [10] provides a new idea, which is using the neural network to classify the Flickr images in the building to better identify the function of the area. Xing H [11] considers both urban morphology and human activities to classify urban functional regions. Building-based and region-based landscape metrics derived from building-level blocks are delineated to measure urban morphology, while socioeconomic features are extracted from crowdsourced data related to human activities using a topic model and semantic scaling method. But [11] neglects the previous mining of the semantics of building structure, and does not make a better choice in the supervised model. The probabilistic multi-sources model [12, 13] does show a good performance in at most 10 functions, however, the number of functions identified is inadequate for practical application, since 25 main functions are necessary to be identified in most usual cases.

In this paper, we attempt to build an automatic algorithm framework to identify enough categories of function. To achieve the goal, we present a novel framework that integrates the existing data preprocessing part with the classification model part (see Fig. 1). It can process POI data and building data simultaneously, to achieve higher accuracy of land use classification. This framework must have sufficient generalization performance, that is, it can be effectively used in multiple cities, rather than just filling in the missing data of a city. The accuracy of classification of our model is satisfactory. Moreover, several enlightening conclusions are drawn through comparing different schemes and analyzing partial dependencies.

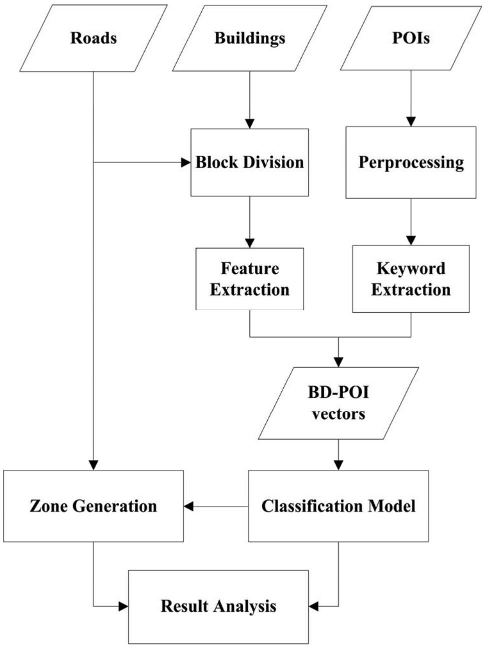


Fig. 1. Framework.

2. Data & Method

This section provides the details of our approach in the order of data selection, block division, preprocessing, feature extraction, keyword extraction, classification model, zone generation.

2.1. Data Selection

Firstly, geo-tagged data is numerous and jumbled and, in general, divided into static city data and dynamic city data. Static geo-tag data includes urban roads, points of interest, buildings, land prices, weather data, etc. Dynamic geo-tag data includes various time series data, such as mobile phone signaling data, GPS data, check-in data, driving record data, and the like. Relatively speaking, static data is relatively easy to obtain. Dynamic data often involves a lot of desensitization and time limitation due to personal privacy.

Since we aim at the sufficient generalization performance of this framework and the ability of being used in real-world scenarios, the data required by the framework must be available online and have a higher update frequency. At the same time, these data need to have sufficient coverage within the required range so that our framework can work throughout the city. Finally, the data should be able to contain enough information about the land use.

Check-in data is first ruled out. Although it is large in number and contains both geo-tags and explicit demographic information, it is clear that these sign-in data, such as Flickr and Twitter, cannot cover all land. This does not meet the data requirements above. Secondly, mobile phone signaling data and GPS data, which can accurately outline the moving trajectory of a person, are difficult to acquire at any time due to the need for privacy protection. Therefore, the full exploitation of static data such as POI data and buildings data should be the key to mapping land use. These data can be obtained from map service providers such as Google Map, Baidu Map, and AMap.

2.2. Block Division

Considering that the planning urban land use zone division is unknown, road network data is indispensable. In our experiment we use five levels of road network for cutting blocks (see Fig. 2 (a)) and the result is roughly consistent with the actual situation. Fig. 2 (a) shows the measurement of blocks, in which the regions in gray define the target area.

There are several zones inside the block. As is shown in Fig. 2 (b), a block is mapped with varying numbers of colors, each representing a zone. A zone`s function is formed by activities of people, and in normal cases the carriers of human activities are buildings (beige areas in Fig. 2 (c)). In our experiment we firstly identify the function of each building, and then generate zones by aggregating the buildings (the groud truth is shown in Fig. 2 (d)).

Buildings data is not enough for identifying the function of zones, we need to understand what kind of activities people are engaged in in a building. Thus we take use of Points of Interest (POIs) data (the black points in Fig. 2(e)).

2.3. Preprocessing

One of the key problem of our experiment is how to link POIs to buildings. To solve this problem, we design two schemes for preprocessing:

a) Slightly inflate each building with a given size (about 5-10 m), then use Spatial-Join method (see Fig. 2 (f)) to connect each poi to a certain building.

b) Diffuse the POIs into the whole space with Kernel Density Estimation (KDE) technique, and each building gets a numerical value of influence from a certain POI.

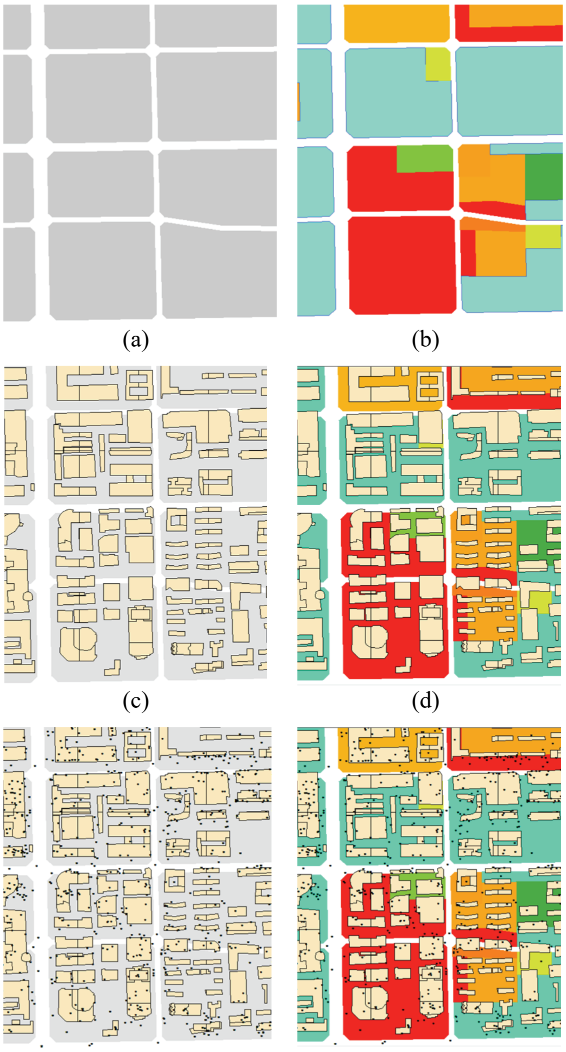


Fig. 2. Typical samples of (a) Blocks, (b) Buildings; (c) POIs; (d) Zones; (e) Buildings in Zones; (f) Data of Spatial Join

As a result, each building gets N additional features (N is the number of categories of POIs). In scheme a) the numerical value of each additional feature is the number of POIs belong to each category linked with each building, in scheme b) the numerical value of each additional feature is the sum of influence of POIs belong to each category. In the following experiment, we compare the performance of these two schemes, and make interpretation of the differences between them.

2.4. Feature Extraction

Feature extraction of buildings is one of the key steps of the framework. There are many ways to extract features. Some specific architectural features can be obtained through data acquisition and simple calculations, such as building projected area, building height, building orientation, and number of architectural projection corners. Furthermore, the Auto-Encoder constructed by convolutional neural network encodes the projection contour of the building to make the architectural projection form becomes a new feature dimension.

2.5. Keyword Extraction

Concerning the fact that distribution of POIs is not homogeneous, for instance, catering & food POIs own a much larger number than scenic spot POIs apparently, we employ term frequency – inverse document frequency (TF-IDF) technique to neutralize this inhomogeneity. Similarly, other types of keyword extraction algorithms can be applied in this step, such as the LDA topic model and its various variants. In our work, the term frequency (TF) is defined as the number of POIs linked to a certain building meanwhile belong to a certain category divided by the number of all POIs belong to this category on one hand, which indicates the normalized frequency of this POI category linked to this building. On the other hand, the inverse document frequency (IDF) is defined as a logarithm of the number of all buildings divided by the number of buildings linked with POIs belong to a certain category, a higher IDF means this POI category owns more significant features to distinguish the function of a building. To integrate these two indexes, TF-IDF is equaled with the product of TF and IDF, and consequently this process reduce the inhomogeneity of POIs in a remarkable level.

After integrating POIs information and buildings data, we obtain a training set of buildings with N+m features, in which the meaning of N features refers to the POI weighted by TF-IDF, and the m is the number of feature dimensions that are all extracted by feature extraction from building data. Finally we got the BD-POI vectors**.**

2.6. Classification Model

All of our pre-training experiments are implemented in the case of Shenzhen city. Thus we regard the ground truths of the functions of Shenzhen`s urban zones where a building in as the labels of buildings. To employ the Cross-Validation method, the raw data set is divided firstly to two parts: 80 % as train set and 20 % as test set, secondly we separate the train set into five equal parts and train our model five times, one of five equal parts of train set is regarded as validation set respectively each time. The classification model we use is Gradient Boosting Decision Tree (GBDT), which has been widely adopted recently because of it`s excellent performance in data mining and machine learning.

After the Cross-Validation and Grid-Search process, we gain the appropriate hyperparameters minimized the loss function, then we employ the trained model in previously divided test set to obtain the final result of our experiment.

2.7. Zone Generation

After properly identifying buildings, we segment the land where the building is located to generate zones. In each block, buildings of the same function are outlined in a geometric pattern. If there are too few buildings on a geometric pattern, appropriate modifications will be made according to the surrounding building types.

3. Result & Analysis

As Table 1 shows, our method derives a pleasantly surprised identifying result on 25 main categories of urban land use function. The final identifying accuracy of our method is 85.7 %, which is better than almost any previously research. On some representative functions, such as Resident zones (89 %) and Education zones (92 %), the result is accurate enough for practical application. Then we compare the two schemes of connection between POIs and buildings. We find that for some functions like Tourist Scenery zones and Government zones, the scheme used KDE perform better, while the other scheme used spatial-join perform better on functions like Resident zones and Shopping zones. A probable explanation for this phenomenon is that the functions identified better by the KDE scheme innate larger scope of influence with a slower decay than the functions identified better by the Spatial-Join scheme.

Table 1. Top 10 gategories of higheset accuracy, classification is based on Chinese National Land Classification Standard GB 50137-2011.

Land Use Zones	Number in Test Set	Accuracy
Physical Health Zone	406	94 %
Advanced Education Zone	349	93 %
4th Level Residential Zone	26122	91.7 %
Logistic Zone	289	89.3 %
2nd Level Industry Zone	3671	89 %
2nd Level Residential Zone	13396	88.3 %
New Type Industry Zone	758	87.9 %
1st Level Residential Zone	4696	87.1 %
Secondary Education Zone	316	86.6 %
Under Construction Zone	2282	86.5 %

Moreover, partial dependencies are analyzed in order to reveal the degrees of influence on each land use by each category of POIs and other feature dimensions. Take the certain function of Primary Education zones as instance, as Fig. 3 indicates. Primary Education function is mainly decided by the education POIs (see Fig. 3-16), which is a deserved fact. Nevertheless, from Fig. 3-11 we also find that medical care POIs influence Primary Education function in a remarkable level. In like manner, Fig. 3-13 reflects a similar phenomenon. When the number of entertainment and medical care POIs exceeds a threshold, the dependency between them and Primary Education zone declines rapidly, from which we can deduce the estimated value of the best number of entertaining places or clinics built nearby a primary school.

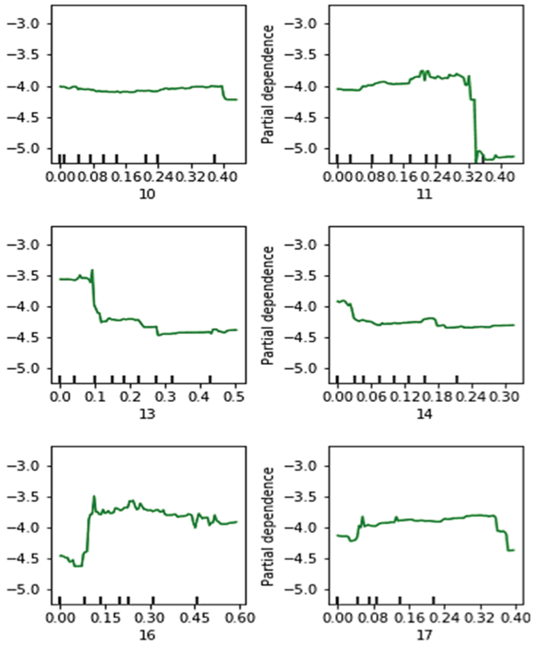


Fig. 3. Partial dependencies of Primary Education Zone on POIs, in which (10) are Automobile Service POIs, (11) are Medical Care POIs, (13) are Entertainment POIs, (14) are Hotel POIs, (16) are Education Organization POIs, (17) are Scenic Spot POIs.

4. Summary

The contributions of this research are as follows:

1) We propose a framework of method, which identify urban functional zones with a high accuracy of practical value, merely using road network data, building data and POIs data.
2) Our framework has good generalization performance and can be used in other cities.
3) Several enlightening conclusions about influence degree on land use zones by POIs have been drawn, which owns reference value in city science.
4) We design two schemes to integrate buildings and POIs, which reveal properties of POIs in different aspects.

5. Discussion

The in-depth exploration of the architectural form of buildings is going to play the key role in our future work. Unique architectural form contains the corresponding information about the function a building undertakes in usual case. Generally speaking, there are two forms of extraction of the architectural form. One is to disassemble the various dimensions in an architectural structure as new input dimensions of classification model, such as the orientation of the building, floor area ratio, development intensity, etc. The other is to regard the architectural form as image data, thus techniques of image feature extraction can be employed to extract the features of the architectural form. However, what is more critical is how the architectural forms of buildings located in the same block or even several blocks relate to each other, and how these connections affect the zone generating and the determination of architectural functions.

On the other hand, although our work proposes a reliable overall framework to improve classification accuracy, optimization can still be done in each step. For example, in the feature extraction of architectural form, there are many better solutions besides Auto-Encoder for architectural form. In addition, Street View data can provide more details about the facade of the building, and if necessary, add modules to the frame for further improvement of recognition accuracy.

Finally, the end-to-end classification framework is also a viable attempt. We can convert all the data into a certain channel of the image according to its two-dimensional coordinates, and then use the convolutional neural network to generate the regional boundaries and their categories directly.

Acknowledgements

This work was jointly supported by National Natural Science Foundation of China under Grant No. 51578128, National Science and Technology Major Project of China under Grant 2016ZX03001022-002 and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions under Grant No. 1101007003. This work has applied for national patent, and the patent application number is 201910072439.8.

References

[1]. Z. Liu, J. Yu, W. Xiong, J. Lu, J. Yang, Q. Wang, Using mobile phone data to explore spatial-temporal evolution of home-based daily mobility patterns in Shanghai, in *Proceedings of the International Conference on Behavioral, Economic and Sociocultural Computing (BESC'16)*, 2016, pp. 1-6.

[2]. Z. Liu, J. Cao, J. Yang, Q. Wang, Discovering dynamic patterns of urban space via semi-nonnegative matrix factorization, in *Proceedings of the IEEE International Conference on Big Data (BigData'17)*, 2017, pp. 3447-3453.

[3]. Z. Liu, S. Yan, J. Cao, T. Jin, J. Tang, J. Yang, Q. Wang, A Bayesian approach to residential property valuation based on built environment and house characteristics, in *Proceedings of the IEEE International Conference on Big Data (Big Data'18)*, 2018

[4]. J. Tang, Z. Liu, Y. Wang, J. Yang, Q. Wang, Using geographic information and point of interest to estimate missing second-hand housing price of residential area in urban space, in *Proceedings of the IEEE International Smart Cities Conference (ISC2'18)*, 2018.

[5]. N. Niu, X. Liu, H. Jin, et al., Integrating multi-source big data to infer building functions, *International Journal of Geographical Information Science*, Vol. 31, Issue 9, 2017, pp. 1871-1890.

[6]. Y. Yao, X. Li, X. Liu, et al., Sensing spatial distribution of urban land use by integrating points-of-interest and Google Word2Vec model, *International Journal of Geographical Information Science*, Vol. 31, Issue 4, 2017, pp. 825-848.

[7]. J. Yuan, Y. Zheng, X. Xie, Discovering regions of different functions in a city using human mobility and POIs, in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 186-194.

[8]. N. J. Yuan, Y. Zheng, X. Xie, et al., Discovering urban functional zones using latent activity trajectories, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 27, Issue 3, 2015, pp. 712-725.

[9]. Y. Sun, H. Deng, L. Zhao, et al., Discovering urban functional areas based on Node2Vec-taking Shanghai as an Example, in *Proceedings of the 11th International Conference on Intelligent Computation Technology and Automation (ICICTA'18)*, 2018, pp. 195-199.

[10]. Y. Zhu, X. Deng, S. Newsam, Fine-grained land use classification at the city scale using ground-level images, *IEEE Transactions on Multimedia*, Vol. PP, 2019.

[11]. H. Xing, Y. Meng, Integrating landscape metrics and socioeconomic features for urban functional region classification, *Computers, Environment and Urban Systems*, Vol. 72, 2018, pp. 134-145.

[12]. S. Gao, K. Janowicz, H. Couclelis, Extracting urban functional regions from points of interest and human activities on location-based social networks, *Transactions in GIS*, Vol. 21, Issue 3, 2017, pp. 446-467.

[13]. X. Liu, N. Niu, X. Liu, et al., Characterizing mixed-use buildings based on multi-source big data, *International Journal of Geographical Information Science*, Vol. 32, Issue 4, 2018, pp. 738-756.

(41)

Search for Exceeding Web Reviewers – Authorities and Trolls Using Genetic Programming

K. Machová [1] and M. Ledecký [1]
[1] Technical University of Košice, Faculty of Electrical Engineering and Informatics, Department of Cybernetics and Artificial Intelligence, Letná 9, 04200 Košice, Slovakia
Tel.: +421 55 6024142, fax: +421 55 6253574
E-mail: kristina.machova@tuke.sk

Summary: The paper is from the field of social media mining. It is focused on evaluation of web reviewers trying to distinguish credible authorities from uncreditable trolls. The work proposes a machine learning approach for estimation of a measure of authority or trollism of web reviewers. A goal of designed approach is to model a dependency of the authority or trollism (dependent variable) on independent variables representing structure as well as content of web discussions. The solution of this task is an important first step for an automatic recognition of antisocial behaviour during posting in on-line communities within web discussions. The paper contains also the results of experiments aiming at generation the estimation function using genetic programming approach.

Keywords: Recognition of antisocial behaviour, Measure of authority, Measure of trollism, Genetic programming, Estimation function.

1. Introduction

Social media are a solid part of today's human life. They bring a lot of positives, but unfortunately, also many negatives to our lives. From advanced countries to less developed countries, every nation uses the power of social media to improve life. Examples of the positive use of social media are: connectivity, learning, getting help or up-to-date information, propagation, charity activities, raising awareness, helping to fight crime, building communities, etc. Examples of the negative impact of social media on our modern society are: trolling, fake news, hoaxes, hacking, rumours, social spamming, hate speech etc [1].

Because of existence of some disinformative content in online discussions, some forms of regulation of social media contributions should be introduced. The community of informatics that has created social networks and social media has a debt to modern society. This debt can be compensated by proposing means for detecting these unhealthy web phenomena in an automatic way.

It is a big challenge that means a lot of work. But somewhere we need to start. In this work, we would like to focus on detecting trolls in contrast to honest authoritarian reviewers.

2. Disinformative Content

Disinformative content may be of a dual nature. First, it can represent information that will affect and manipulate its recipients (fake news, hoax). Second, it is misinformation, which is caused by misunderstanding without manipulation [2].The first is based on disseminating propaganda. Propaganda tries to relativize reality by generating arguments that distort the truth. Sometimes this truth distortion could be generated automatically using algorithms based on similarity measures [3].

There are various disinformation techniques. These techniques are discussed in [4], including consequences of their usage. In this work, there is presented a probability approach to detection of relativized statements.

An opinion sharing by product reviews is a part of online purchasing. This opinion sharing is often manipulated by fake reviews. The paper [5] presents an approach which integrates content and usage information in the fake reviews detection. The usage information is based on reviewers’ behaviour trails. In this way, a reviewer reputation is formed. The ability to define a reviewer reputation can help us to recognize authorities and also trolls in social media.

3. Authority Versus Troll

The basis of our further research is to understand the differences between features of authoritative contributing and features of trolling in online discussions. Therefore, we first attempt to describe and define authority as well as the the fenomenon of trolling.

3.1. Authority

In general, the term “authority” means the ability of people, companies or institutions to influence others positively and to lead others to achieve a certain goal. Important fact for us is that an authority is connected with the relations between people (for example reviewers in online discussions), positions and hierarchies [6].

Here are many kinds of authorities, but most often we recognize formal and informal authority. Our goal is to automatically detect the informal one - natural authority among social media reviewers.

The web authority has different attributes as the authority in real life. These attributes are related to the structure of web discussions and also to discourse content.

3.2. Troll

Troll is an internet slang term that identifies individuals who intentionally attempt to incite conflict or hostility by publishing offensive, inflammatory, provocative or irrelevant comments in online social communities. Their intent is to upset others and to produce a strong emotional response, mostly negative one. They use it as a bait for engaging new users in the discussion [7].

The activity of a troll is referred to as trolling in online discussions. We can furthermore say that trolling is an activity with the intent of fraudulent behavior, exploiting the sensitivity of society to spreading fanaticism, racism, hatred, or causing conflicts with bickering among others, often over very provocative topics. Despite attempts to limit it, trolling is becoming more and more widespread. We can meet him in all social networking settings, where people can freely express their opinions and thoughts.

Trolling is becoming a bigger problem, and it is important to find ways to detect and limit it. The most common types of trolls that we encounter in active online communities are [8]:

- Troll opponent,
- Provocative troll,
- Troll, who insults,
- Still insulted troll,
- Troll, who persists in the discussion,
- Troll, who knows everything,
- Troll, which uses Caps Lock,
- One word troll,
- Troll, which overcame,
- Troll, which is out of the theme,
- Unsolicited spammer.

No matter where we find trolls lurking, every troll tends to disturb communities in a very similar and often predictable way.

4. Attributes of the Authoritative Posting and of the Trolling

The authority and troll reviewer are totally different contributors to web discussions. However, approaches to authority and troll detection could be similar, based on the use of machine learning methods. In order to use any machine learning method, we need to define attributes specifically for authoritarian and troll contributions and to collect labeled training data for both these types of reviewers.

To achieve our goal – the value of Authority and Troll estimation, we need to:

1. Define attributes (variables, predictors) which values for each reviewer could be extracted from an online discussion,
2. Label those data,
3. Learn estimation function of the variable “Authority” and ”Troll” on the independent variables – attributes selected in the first step.

4.1. Attributes of an Authority

The very important first step is definition of those attributes of an online discussion which represent the authoritative contribution. These attributes will play a role of independent variables or predictors in a process of modelling of an estimation function for the prediction of a measure of authority. The values of these attributes could be extracted from the structure and content of the online social media. We have proposed the following attributes of the web discussion:

- *NP* is the number of posts of a given reviewer.
- *ANR* is the average number of responses per post of a given author.
- *AL* is the average number of all layers, on which posts of a reviewer are situated within the discussion tree. (For exaple, if a reviewer has 3 posts in layers 2,4 and 6, then his AL = 4).
- *NCH* is the average number of characters per one post, which represents the length of posts of an author and also his effort to explain something.
- *K* is karma of a reviewer in the form of a number from 0 to 200, which represents activity of the discussant from the last 3 months.
- *AE* is the average evaluation of the comment, which is available on the web discussion page. The range of the *AE* is the number from 0 to 80.

Maybe using just K and AE would be enough to get good predictive results, but these data may not be available on every online discussion site.

4.2. Attributes of a Troll

Trolling contribution has different attributes as authoritative contribution. We want to focus on the following attributes:

- *DSN* is an average degree of a sentiment negativity of all posts of the author – the troll.
- *DDS* is the degree of difference of polarity between the opinion of the reviewer and the whole discussion.
- *NWS* is a number of words from a special dictionary typical for trolling posting.
- *CL* is a measure of complexity of used language (can be represented by the average length of sentences of all posts of the troll).
- *NST* is a number of sensational words in the titles of posts of the reviewer.

- *CSC* is a cosine similarity of content. Trolls often repeat the same review text, because a new authoring content would be time consuming.

Some attributes of troll are the same as attributes of authority, for example:

- *NP* is the number of posts of a given contributor.
- *ANR* is the average number of responses per post of a given author.

5. Used Methods

5.1. Motivation for Methods Selection

We needed to define machine learning methods, including statistical learning to find the estimation function for the prediction of a measure of authority as well as a measure of trolling of a reviewer. As the first approach, we used linear regression with good results presented in Table 1.

The disadvantage of this approach is that before the beginning of the learning process we have to formulate the assumption of the exact form of the linear equation and only weights (constants) of the given equation are generated during the learning process. That is why we have decided for using genetic programming, the method that is also able to learn the form (structure) of the equation or what elementary functions it will be composed of.

Another advantage of using genetic programming is that all attributes may not be present in the generated function, and therefore the genetic program makes selection of the most important attributes, which is an important finding for future experiments.

5.2. Genetic Programming

The genetic programming is one type of genetic algorithms. Character strings represent chromosomes. These character strings are replaced by more difficult structures in the form of elementary functions. The process of symbolic regression is based on operations of crossover and mutation [9].

Two parents (syntactic trees) are selected for operation of crossover (see Fig. 1). The first parent is represented by function [(1+x)+x]*(x-x*x) and the second parent is represented by 2x*(x*x). The nodes selected for crossover operation are emphasized with circles. The emphasized nodes are called "points of crossover" and they are selected randomly. After changing these points of crossover, we will obtain the new generation. In the new generation the first parent will become [(x*x)+x]*(x-x*x) and the second one 2x*(1+x).

The mutation operation provides the replacement of a subtree of the parent tree with another subtree randomly generated. This means that the selected branch of the syntactic tree is randomly changed.

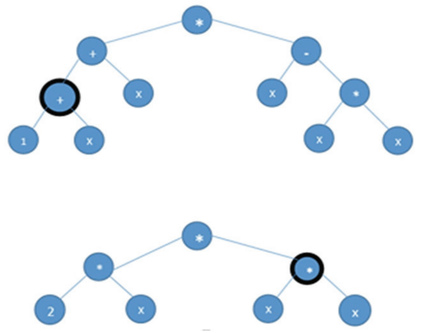


Fig. 1. Syntactic trees – the first parent (top tree) and the second parent (bottom tree) as the input to operation of crossover.

6. Tests Results

Our approach was tested on data collected from the portal "*www.sme.sk*". For each reviewer from 117 reviewers, the data were extracted in the form of values of the attributes NP, ANR, AL, NCH, K and AE (see 4.1 subsection). The data had to be labeled for machine learning methods purposes. We used two ways of labeling: expert-based and the wisdom of the crowd based, where the other participants of the online debate rated the reviewer.

Results in experiments with linear regression analysis are presented in Table 1. and they were published in more details in paper [10]. The prediction of Authority using generated estimation function was testing using the deviation measure. The prediction was transformed to classification using the following rules:
IF estimated value within <70,100> THEN Authority,
IF estimated value within <0,70> THEN Non-authority.

The classification was tested using two obvious measures - precision and recall.

Table 1. Average results for linear regression.

Labelling	Deviation	Precision	Recall
Expert	17.35	0.70	0.67
Crowd	3.29	0.98	0.80

New experiments with genetic programming were provided. For them a new toolbox was added to Matlab. The function "gplab" from the toolbox was used. We have prepared an environment for experiments with genetic programming. We have done some experiments on extracted data from the web page "*www.sme.sk*". The data were divided to three parts and three experiments were provided for both approaches - learning from an expert as well as learning from wisdom of the crowd. The results of experiments are promising. They are presented in Table 2. These experiments were provided for populations of 50 individuals and the number of generations equal to 50. Allowed mathematical functions were +, -, /, *, square and square root.

The results of generation of estimation function using genetic programming have brought new information about a real need and suitability of independent attributes of authority of the online discussion. The function generated using genetic programming need not contain all variables. Within 16 experiments, variable AE was used in 100 %, AL and ANR in 56,25 %. The other variables are not so important for forming estimated functions, because variable K and NCH were used in 43,75 % and variable NP only in 37,50 %. It means, that for future it will be better to consider only variables AE, AL, ANR and maybe some new variables.

Table 2. Results of experiments using genetic programming.

Labelling	Precision	Recall
Crowd 1	0.94	1.00
Crowd 2	0.97	1.00
Crowd 3	0.96	1.00
Crowd average	**0.98**	**1.00**
Expert 1	0.39	0.64
Expert 2	0.47	0.89
Expert 3	0.42	0.72
Expert average	**0.43**	**0.75**

7. Conclusions

The design of the approach to the problem of the authority and troll recognition in online diskussions is presented. This approach is based on the estimation function construction using genetic programming.

We have provided experiments with generating authority estimation function using genetic programming and we have compared results of these experiments with our previous experiments using regression analysis. We can see (Tables 1, 2) that the genetic programming is better technique for learning from data, which were labelled by crowd. On the other hand regression analysis is better on data labelled by expert mainly in precision. In addition, genetic programming can provide us with the information about a real need and suitability of attributes of the authority or troll.

For future, we would like to provide experiments with generation estimation function for the troll reviewer's detection task. Also ontologies approach could be considerd.

Authority or troll recognition can be used in many real situations. Mainly in the process of decision-making connected with an important things, it is very important to know whether we can or cannot believe (confide) in presented opinions and advises.

Acknowledgements

The work presented in this paper was supported by the Slovak Research and Development Agency under the contract No. APVV-017-0267 "Automated Recognition of Antisocial Behaviour in Online Communities" and the contract No. APVV-015-0731 "Cloud Based Human Robot Interaction".

References

[1]. B. Ahmad, 10 Advantages and Disadvantages of Social Media for Society, https://www.techmaish.com/advantages-and-disadvantages-of-social-media-for-society/

[2]. S. Kumar, Ř. West, J. Leskovec, Disinformation on the Web, in *Proceedings of the 25th International on Word Wide Web (WWW'16)*, 2016, pp. 591-602.

[3]. S. E. Wang, H. Garcia-Molina, Disinformation Techniques for Entity resolution, in *Proceedings of the 22nd ACM International Comnference on Information and Knowledge Management (CIKM'13), New York, USA,* 2013, pp. 715-720.

[4]. M. Řimnáč, Detection of a disinnformation content – Case study novičok in CR, in *Proceedings of the Conference Data a Znalosti (WIKT'18)*, 11-12.10.2018, Brno, Vysoké Učení Technické, 2018, pp. 65-69.

[5]. I. Dematis, E. Karapistoli, A. Vakali, Fake review detection via exploitation os spam indicators and reviewer behaviour characteristics, in *Proceedings of the 44th International Conference on Current Trends in Theory and Practice of Computer Science beyond Frontiers (SOFSEM'18)*, 2018, pp. 1-14.

[6]. K. Chavalkova, Authority of a teacher, *Proceedings of Philosophic Faculty of the University of Pardubice*, Pardubice, Czech Republic, 2011 (in Czech).

[7]. E. March, 'Don't Feed the Trolls' Really Is Good Advice – Here's the Evidence https://theconversation.com/dont-feed-the-trolls-really-is-good-advice-heres-the-evidence-63657

[8]. E. Moreau, 10 Types of Internet Trolls You´ll Meet Online, https://www.lifewire.com/types-of-internet-trolls-3485894

[9]. P. Návrat, et al., Artificial Intelligence: Metaheuristics and Evolutionary Algorithms, *Publishing House of STU*, FIIT STU, Bratislava, 2014, pp. 1-418 (in Slovak).

[10]. K. Machová, M. Mikula, M. Szabóová, M. Mach, Sentiment and authority analysis in conversational content, *Computing and Informatics*, Vol. 37, Issue 3, 2018, pp. 737-758 (current content).

(42)

Training Set Class Distribution Analysis for Deep Learning Model – Application to Cancer Detection

Ismat Ara Reshma [1,2], **Margot Gaspard** [3], **Camille Franchet** [3],
Pierre Brousset [2,3], **Emmanuel Faure** [1], **Sonia Mejbri** [1,2] **and Josiane Mothe** [1,2]
[1] IRIT UMR5505 CNRS, 118 Route de Narbonne, F-31062 TOULOUSE, France
[2] Univ. de Toulouse, TOULOUSE, France
[3] Univ. Cancer Institute Toulouse Oncopole, 1 Avenue Irène Joliot-Curie, 31059 TOULOUSE, France
E-mail: Josiane.Mothe@irit.fr

Summary: Deep learning models specifically CNNs have been used successfully in many tasks including medical image classification. CNN effectiveness depends on the availability of large training data set to train which is generally costly to obtain for new applications or new cases. However, there is a little concrete recommendation about training set creation. In this research, we analyze the impact of different class distributions in the training data to a CNN model. We consider the case of cancer detection task from histopathological images for cancer diagnosis and derive some useful hypotheses about the distribution of classes in the training data. We found that using all the training data leads to the best recall-precision trade-off, while training with a reduced number of examples from some classes, it is possible to inflect the model toward a desired accuracy on a given class.

Keywords: Medical information retrieval, Image segmentation and classification, Deep learning, Class-biased training.

1. Introduction

The huge success of deep learning models in visual recognition [1, 2] and specifically CNN, drove researchers to explore their use in computer-aided diagnosis system for cancer [1] detection from histopathological or whole slide images (WSIs) [3-5]. This paper contributes to this research topic with the objective to help physicians to detect metastasis by providing them the image regions in which there is a high probability of cancer and the regions where there is no cancer. In that purpose, we employ segmentation (i.e. pixel classification) of the WSIs. Segmentation facilitates readable separation of each class and eases image analysis [6].

One of the specificities of CNN is that they need a lot of training examples [7]. However, when dealing with biomedical images, precise annotation process needs both expertise and time. As a result, the image data sets that can be built are small. The problem of how to distribute the examples from the different classes to learn is not well studied.

In this paper, we tackle the challenge of deciding which types of examples would be needed to obtain the expected prediction from the trained model. Indeed, when considering segmentation problems, the trained model can be very effective on one class and poorly performing on another. Would the results be different if one class is "over-represented"? Is it better to have about the same number of examples in each class? This paper aims at answering these questions, all related to the balanced/imbalanced nature of the training set [8]. There are some studies in the literature that tackle this problem [8-12] as it has an adverse effect on classification accuracy.

Among the proposed methods to balance classes, the most straightforward and commonly practiced method is the oversampling of the minority class [8, 9]. However, it can lead to overfitting [11] and concrete studies on the effects are lacking. To address the class-imbalance in metastasis detection the existing methods usually adopt random sampling to select an equal number of positive and negative examples [4, 5] and thus generate a balanced training set. However, there is no analysis to answer whether this balanced distribution is the optimal one for this task. Kubat et al. [12] suggested downsampling.

In this paper, we consider both balanced and class-biased distributions of the training set and analyze the selection impact on the model accuracy. The result could serve for deciding which examples should be first added in the training set when there are costly to add. We study the impact on the model of a training biased by an over-represented class (what we call a class-biased training). We run a series of experiments in which we train the model in one hand with an over-representation of the cancer class and on the other hand with an over-representation of the non-cancer class. We also consider balanced sets. We found that balanced data not always lead to the best result and suggest solutions to optimize the model toward a specific accuracy on a target class.

The rest of the paper is structured as follows: we first present related work. We then describe the

[1] Throughout the paper, "cancer" and "metastasis" are interchangeably used.

experimental details with results. Finally, we conclude the paper with some future directions.

2. Related Work

During the last decades, several studies have been done to facilitate computer-aided diagnosis for metastasis detection from WSIs. Most of the methods used classical machine learning techniques [13]. Studies on utilizing deep learning on this topic are comparatively few and new. From 2015, Bejnordi et al. [3] are organizing a worldwide challenge named CAMELYON on this topic in which most of the participants use CNN-based methods. The winning team [4] utilized the 22 layers GoogleNet, employed rotation and random cropping for data augmentation and color normalization. Liu et al. [5] utilized the updated version of GoogleNet named Inception (V3) [14]. To avoid class-bias they selected normal and tumor classes with equal probability then extracted patch of that class from a WSI which was selected by uniformly at random; then it was followed by applying several data augmentation techniques including rotation, mirroring, and extensive color perturbation. Sonia M. et. al. [15] proposed a new data set for different types of breast cancer, and an end-to-end deep learning framework for multilabel tissue segmentation utilizing their data set, while network parameters were determined with a deep analysis.

One of the common problems in machine learning is imbalanced data. In the real world, the availability of some classes makes them the over-presented majority, while the scarcity of some classes makes them the under-presented minority. This imbalance of classes representation makes the classification task challenging for a classifier. A limited amount of studies on this topic is available in the literature, especially on deep learning perspective. Some studies suggest data level modification [16, 17], while other studies suggest network architecture level modification [10]. Buda et al. [8] present a comparative study of different methods. Oversampling of the minority class is the most prescribed solution [8, 9]. Kubat et al. [12] suggested downsampling.

To address the class-imbalance in metastasis detection task the existing methods usually adopt random sampling to select the equal number of positive and negative examples [4, 5]. However, comparative studies among the different distribution of classes in the training set are absence in this domain. In this paper, we consider this issue. The result could serve for deciding which examples should be first added in the training set when there are costly to add.

3. Experiments

3.1. Data Set and Setting

We use the “metastatic LN” data set from Toulouse Oncopole. The data set contains 61 WSIs (34 for training, 27 for test) of lymph nodes stained with hematoxylin and eosin (H&E), for which an expert pathologist has provided the ground truth segmented masks. The masks are annotated with 3 classes: *metastasis/cancer* (C), *lymph-node* (¬C), and *other* (O). Class O can be either background or histological structures not included in the first two classes C and ¬C, such as adipose or fibrous tissue. Metastasis of 16 primary cancer types and organs have been included in the data set.

Although many other parameters may influence the results, in this research we focus on analyzing the impact of the training examples and the classes they belong to. Although, the WSIs are very large in size, here we utilize them by 8 times downsampling in size to save time and memory resource during analysis. However, the full resolution images will give better result [5]. As a network architecture of CNN, we select U-net [18]. We implement the U-net architecture using Keras [19] on the TensorFlow backend. In all the experiments, 20 % of the training data is kept for validation. All data are normalized by scaling the pixel value from [0, 255] to [0.0, 1.0] by dividing 255. It makes the convergence of training faster [20]. We utilize Adam [21] as an optimizer. After empirical preliminary evaluation, we set the learning rate of Adam as 1e-05. We use the "categorical cross-entropy" (original U-net) as loss function.

We extract squared overlapping patches of dimension d^2 with stride d/2 pixels from each training WSI that correspond to our training examples; we use d = 384 pixels and extract 127,898 patches. We use usual recall, precision, and F-measure to evaluate the model; however, rather than considering the pixel-level evaluation, we consider non-overlapping patches of dimension 500^2 on the predicted test images’ masks and make a patch-based evaluation. We compute patch-based recall, precision, and F-measure for each test image separately, and finally, take the average result of 27 test images to evaluate the performance of models.

3.2. Experiment Design

The statistics in Table 1 depicts that most of the pixels (78 %) in the training WSIs belong to the class O.

Table 1. Report on the average number (in million and percentage) of pixels of each class in the training and test set.

	Pixel class	Mean in million	Mean in %
Training	C	15.2	11
	¬C	14.6	11
	O	107.4	78
Test	C	20	14
	¬C	9.8	7
	O	114.8	79

Since the class O is (i) over-represented, (ii) not the class the pathologists focus on, it imposes us to check the impacts of some other artificial distributions of classes in our training set. To create some other artificial distributions of classes, we need to separate the class examples. In that purpose, we define several patch categories based on the pixel classes they belong to. Throughout the paper, we use the term “class” to indicate pixel type, while, “category” to indicate patch type. The patches which pixels belong to 100 % class O, we define them as patch category O. The remaining patches belong to class C, ¬C, and both C and ¬C with an optional presence (0.0 to 99.1 %) of class O pixels. The details of each patch category in the training set are as follows:
- O (**only other**): 90,374 patches contain almost 100 % class O pixels.
- C (**metastasis/cancer**): 15,328 patches contain pixels labeled with class C and optionally class O.
- ¬C (**lymph node**): 17,274 patches contain pixels labeled with class ¬C and optionally class O.
- C&¬C (**mixed**): 4,922 patches contain pixels labeled with both class C and ¬C, and optionally class O.

These categories are used to design several experiments with different class distribution in the training set as follows:

- (All): done with all possible patches.
- (C&¬C): done with patches from the C&¬C category. Here, the three classes are balanced in terms of pixels, however, the number of training examples is fewer (4,922).
- (C, C&¬C): patches are from the C and C&¬C categories. By excluding the ¬C category, here we limit the presence of class ¬C. Thus the training set is class C biased.
- (¬C, C&¬C): patches are from the ¬C and C&¬C categories. This is the twin case of (C, C&¬C). The training set is class ¬C biased.
- (C, ¬C, C&¬C): patches categories C, ¬C, and C&¬C are used. Here, class C and ¬C pixels are almost balanced, however, class O pixels are downsampled compared to experiment (All) to make the all three classes pixels almost balanced.

3.3. Results

In Fig. 1 we report the results for both class C and class ¬C. The results are the average results computed from the results of 27 test WSIs. Since recall and precision varies in reverse order, it is important to report both. To evaluate the model performance by considering both recall-precision at the same time, we also report F-measure. Experiments are ordered according to the descending order of precision on class C.

From Fig. 1 we can see that recall is higher than precision for both classes; which implies that in this domain most of the error comes from false positive[1] rather than false negative[2]. Another noticeable thing is, unlike it has been reported in [22] for 20 different data sets from UCI machine learning repository [23], Statlog [24], and some private data sets, here balanced distribution i.e. experiments (C, ¬C, C&¬C) and (C&¬C) does not produce the best result.

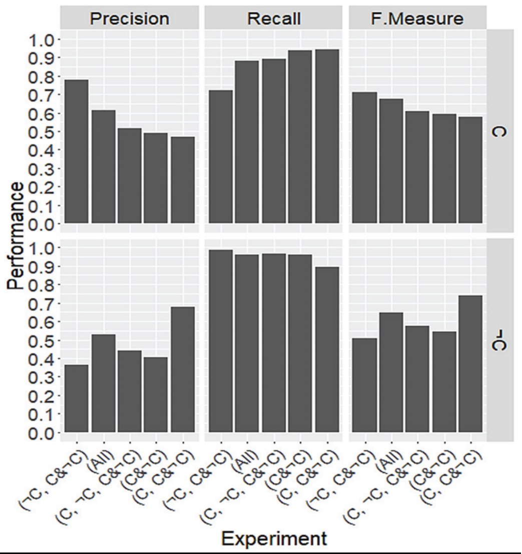


Fig. 1. Model performance for class C (top row) and ¬C (bottom row) when using different combinations of the patch categories as a training set, i.e. different distribution of the classes in the training set. Here, experiments are arranged according to the descending order of the precision on class C.

On the other hand, the natural distribution i.e. the one that has been used in the experiment (All) is the best trade-off maintaining distribution; it produces reasonable recall and precision for both classes C and ¬C at the same time, however not the best result producing distribution. Since in this research our main objective is helping pathologist in all cancer location detection with less false positive generation, the best trade-off maintaining distribution for both classes is not the desirable one, rather best result producing distribution for cancer (class C) class is the most desirable. For class C, the best precision is found for ¬C-biased training set i.e. for the experiment (¬C, C&¬C), while the best recall is found for C-biased training set i.e. for the experiment (C, C&¬C). However, while considering both recall and precision at the same time i.e. F-measure, ¬C-biased distribution (¬C, C&¬C) is the best distribution, and C-biased distribution (C, C&¬C) is the worst distribution for class C. According to this result it is obvious that, in the ¬C parts of WSIs, there are some regions which look like class C parts i.e. there are some inter-class similar regions, that is why the absence of enough ¬C examples compared to class C examples in the training set (e.g. experiment (C, C&¬C)) causes false positive for cancer (C) class during test.

[1] Actually belongs to negative, however, predicted as positive.

[2] Actually belongs to positive, however, predicted as negative.

When considering the class ¬C, this is the other way around: the best precision is found for C-biased training set i.e. for the experiment (C, C&¬C), while the best recall is found for ¬C-biased training set i.e. for the experiment (¬C, C&¬C). However, while considering both recall and precision at the same time i.e. F-measure, C-biased distribution is the best distribution for class ¬C.

In summary, for the cancer class **C**:
(1) Class C-biased training makes recall higher;
(2) Class ¬C-biased training makes precision higher;
(3) Balanced training causes an average result;
(4) The natural distribution i.e. training with the original distribution of the training set (experiment, (All)) makes the best trade-off in recall and precision (both are reasonable at the same time).

For class ¬C class:
(1) Most of the experiments give more than 0.95 recall;
(2) Class C-biased training gives the higher precision;
(3) Balanced training gives an average result;
(4) The natural distribution i.e. training with the original distribution of the training set (experiment, (All)) makes the best trade-off in recall and precision (both are reasonable at the same time).

In a nutshell for C (resp. ¬C), to increase precision we need ¬C (resp. C) biased training, while to increase recall, we need C (resp. ¬C) biased training. The class O is predicted well whatever the experiment is. Detailed results are presented in Table 2**.**

Table 2. Average results computed from the results of 27 test WSIs for the different experiments. Here, R, P, F means the recall, precision, and F-measure respectively.

Exp Name	**Class C**	**Class ¬C**	**Class O**	**Comment**
(All)	R:.882 P:.614 F:.675	R:.959 P:.526 F:.647	R:.999 P:.928 F:.960	Best trade-off
(C, C&¬C)	R:**.943** P:.468 F:.578	R:.894 P:**.675** F:**.740**	R:.997 P:.943 F:.968	Best R for **C**
(¬C, C&¬C)	R:.720 P:**.779** F:**.712**	R:**.984** P:.363 F:.506	R:.997 P:.932 F:.961	Best P and F for **C**
(C&¬C)	R:.939 P:.491 F:.592	R:.961 P:.406 F:.545	R:.996 P:.950 F:.972	Average for **C**
(C, ¬C, C&¬C)	R:.888 P:.516 F:.608	R:.965 P:.439 F:.574	R:.998 P:.932 F:.962	Average for **C**

4. Conclusions and Future Work

In this research, we analyzed the impact of class distribution in the training set for metastasis detection task from WSIs while using U-net deep learning architecture. We utilize our own data set, in which one class, the class O is over-represented compared to the two other classes C and ¬C. This class O-biased data leads us to do a series of experiments with two other artificially class-biased training data: C-biased and ¬C-biased data, and artificially balanced data as well. All these artificially created training data were created by downsampling the over presented class O and in some experiments either downsampling C or ¬C class.

We found that balanced data does not lead to the best result in this domain, rather imbalance data leads to the desired accuracy for a given class. On the other hand, while keeping all possible training examples i.e. keeping the natural distribution in the training set causes the best trade-off in recall-precision, however, does not give the best result either in diagnosis perspective. In fact, the imbalanced distribution gives the most desirable result in this domain. More specifically, for cancer class prediction, non-cancer biased training reduces the confusion due to the inter-class similar region between cancer and non-cancer class, thus produces less false prediction for cancer class. Although our analysis gives a preliminary flavor of the behavior of the model towards the different distribution of classes in the training set, it demands deeper analysis. Specifically, here the number of training examples was not the same for all experiments, we will solve this issue in our future work. Moreover, here we tested the class distribution for a fixed set of network parameters, in the future, we will test the same setting for different parameter settings.

References

[1]. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, Vol. 521, Issue 7553, 2015, pp. 436-444.

[2]. W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, *Neural Computation*, Vol. 29, Issue 9, 2017, pp. 2352-2449.

[3]. B. Bejnordi, M. Veta, P. Van Diest, B. Van Ginneken, N. Karssemeijer, G. Litjens, J. Van Der Laak, M. Hermsen, Q. Manson, M. Balkenhol, O. Geessink, Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer., *The Journal of the American Medical Association*, Vol. 318, Issue 22, 2017, pp. 2199-2210.

[4]. D. Wang, A. Khosla, R. Gargeya, H. Irshad, A. Beck, Deep learning for identifying metastatic breast cancer, *arXiv:1606.05718*, 2016, https://camelyon16.grand-challenge.org

[5]. Y. Liu, K. Gadepalli, M. Norouzi, G. Dahl, T. Kohlberger, A. Boyko, S. Venugopalan, A. Timofeev, P. Nelson, G. Corrado, J. Hipp, Detecting cancer metastases on gigapixel pathology images, *arXiv:1703.02442*, 2017.

[6]. G. Srinivasan, G. Shobha, Segmentation techniques for target recognition, *International Journal of Computers and Communications*, Vol. 1, Issue 3, 2007, pp. 313-333.

[7]. F. Dernoncourt, J. Lee, O. Uzuner, P. Szolovits, De-identification of patient notes with recurrent neural networks., *Journal of the American Medical Informatics Association*, Vol. 24, Issue 3, 2017, pp. 596-606.

[8]. M. Buda, A. Maki, M. Mazurowski, A systematic study of the class imbalance problem in convolutional neural

networks, *Neural Networks*, Vol. 106, 2018, pp. 249-259.

[9]. D. Masko, P. Hensman, The impact of imbalanced training data for convolutional neural networks, Independent Thesis, *School of Computer Science and Communication (CSC)*, 2015.

[10]. S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, P. Kennedy, Training deep neural networks on imbalanced data sets, in *Proceedings of the IEEE International Joint Conference In Neural Networks (IJCNN'16)*, Vancouver, Canada, 24-29 July 2016, pp. 4368-4374.

[11]. N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, Vol. 16, 2002, pp. 321-357.

[12]. M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: One-sided selection, in *Proceedings of the International Conference on Machine Learning (ICML'97)*, Tennessee, USA, 8-12 July 1997.

[13]. M. N. Gurcan, L. Boucheron, A. Can, A. Madabhushi, N. Rajpoot, B. Yener, Histopathological image analysis: A review, *IEEE Reviews in Biomedical Engineering*, 2009, Vol. 2, pp. 147-171.

[14]. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, Boston, MA, 2015, pp. 1-9.

[15]. S. Mejbri, C. Franchet, I.-A. Reshma, J. Mothe, P. Brousset, E. Faure, Deep analysis of CNN settings for new cancer whole-slide histological images segmentation: the case of small training sets, in *Proceedings of the 6th International Conference on Bioimaging (BIOSTEC'19)*, Prague, Czech Republic, 2019.

[16]. N. Jaccard, T. W. Rogers, E. J. Morton, L. D. Griffin, Detection of concealed cars in complex cargo X-ray imagery using deep learning, Journal of X-ray Science, *Technology*, Vol. 25, Issue. 3, 2017, pp. 323-339.

[17]. G. Levi, T. Hassner, Age, gender classification using convolutional neural networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'15)*, Boston, MA, 2015. pp. 34-42.

[18]. O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'15)*, Munich, Germany, 5-9 October 2015, pp. 234-241.

[19]. F. Chollet, et al., Keras, 2015, https://keras.io

[20]. Y. LeCun, L. Bottou, G. B. Orr, K.-R. Müller, Efficient backprop, *Neural Networks: Tricks of the Trade*, 2012, pp. 9-48.

[21]. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, *arXiv:1412.6980*, 2014.

[22]. R. C. Prati, E. A. Gustavo, P. A. Batista, D. F. Silva, Class imbalance revisited: a new experimental setup to assess the performance of treatment methods, *Knowledge and Information Systems*, Vol. 45, Issue 1, 2015, pp. 247-270.

[23]. A. Frank, A. Asuncion, UCI MACHINE LEARNING REPOSITORY, 2010, http://archive.ics.uci.edu/ml

[24]. D. Michie, D. Spiegelhalter, C. C. Taylor, (Eds.), Machine Learning, Neural and Statistical Classification, *Ellis Horwood*, 1994.

(43)

Analysis of Objects Evolution in Satellite Image Time Series Transformed with Neural Network Autoencoder

E. Kalinicheva [1], J. Sublime [1, 2] and M. Trocan [1]
[1] ISEP/LISITE - DASSIP Team, 10 rue de Vanves, 92130 Issy-les-Moulineaux, France
[2] LIPN - CNRS UMR 7030, 99 av. J-B Clément, 93430 Villetaneuse, France
E-mails: {ekaterina.kalinicheva, jeremie.sublime, maria.trocan}@isep.fr

Summary: Nowadays, the analysis of satellite image time series (SITS) is required in many applications. Given the high image resolution of such series, it is complicated in general to detect changes. As the representation of different objects is no longer limited to a simple pixel values, the changes further need the analysis of neighbouring textures. In this paper, we propose an unsupervised SITS change detection algorithm based on texture extraction with neural network autoencoders. The autoencoder first compresses the extracted feature maps and transforms the initial images in their encoded version and then performs the analysis of the pixel evolution corresponding to different objects. The proposed approach highlights the properties of objects behaviour through time and can further be used to create spatio-temporal clusters corresponding to different types of changes in SITS.

Keywords: Satellite image time series, Autoencoder, Change detection, Feature extraction, Unsupervised learning.

1. Introduction

Change detection in SITS, despite all the scientific efforts, remains a challenge nowadays. All change detection approaches can be divided into supervised and unsupervised. In the case of supervised image analysis, change detection is a relatively simple process. Using the reference data, we classify different objects presented on the images in order to create a change map [1].

However, for most remote sensing applications we do not dispose of labelled data due to the high cost of producing them for remote sensing images. In this case, unsupervised change detection algorithms come in handy. Most of the existing algorithms aim at detecting changes between two co-registered images. Therefore, the complexity of some of them might drastically increase when applied to TS.

The easiest unsupervised change detection approach is a bi-temporal image difference or ratio analysis [1]. However, this method is not suitable for high (HR) and very high resolution (VHR) images as different textures become distinguishable. In [2] the authors introduce a contrario approach based on Scale Invariant Feature Transform (SIFT) key points matching to identify changed objects between two images acquired by different satellites. A segmentation-based change detection method was proposed in [3]. In [4] authors use an autoencoder (AE) for detection of the permanent changes when learning a translation model from one image to another.

In this paper, we propose a neural network-based approach for change detection in SITS. This method has a low complexity compared with existing methods because the time series (TS) images are processed as a stack and not individually.

2. Proposed TS Change Detection Method

Let $Im_1, Im_2, \ldots, Im_S$ be a SITS of S images taken on dates $T_1, T_2, \ldots, T_S$. The algorithm steps are the following:

1. Transform SITS in its encoded features representation version $Im'_1, Im'_2, \ldots, Im'_S$ with a neural network AE.
2. Compute the difference on adjacent images Im'_i and $Im'_{i+1.}$
3. Perform clustering analysis of $S-1$ differentiated images in order to detect different spatio-temporal clusters of changes (no change areas, seasonal changes, permanent changes).
4. Do clustering analysis of encoded images to identify different types of objects presented in TS.

Among different neural network models, AEs have been used in many domains: in image processing, the autoencoders are mostly used for feature extraction, image segmentation, reconstruction, compression and reprojection.

All AEs involve an encoding and a decoding structure. In most applications, during the model training, the encoding pass transforms the initial data in its compressed version or a set of feature maps. Further, the encoded data is transformed back to the ‘initial’ one by the decoder. Finally, the reconstruction error is calculated and the model parameters are optimized.

In our work, we use a convolutional AE with fully-connected (FC) layers for a patch-wise image encoding (Fig. 1). To train the model, all the images of TS are clipped in $m \times n$ patches (m, and n, are the image width and height, respectively) of the size $p \times p$ and passed to the AE model. The encoding pass firstly performs the feature maps extraction with convolution layers and secondly compresses them with FC layers to a feature vector of size *f*. The decoding pass is symmetrical to the encoding one. Once the model is trained, the encoder transforms every patch of an original image to a feature vector.

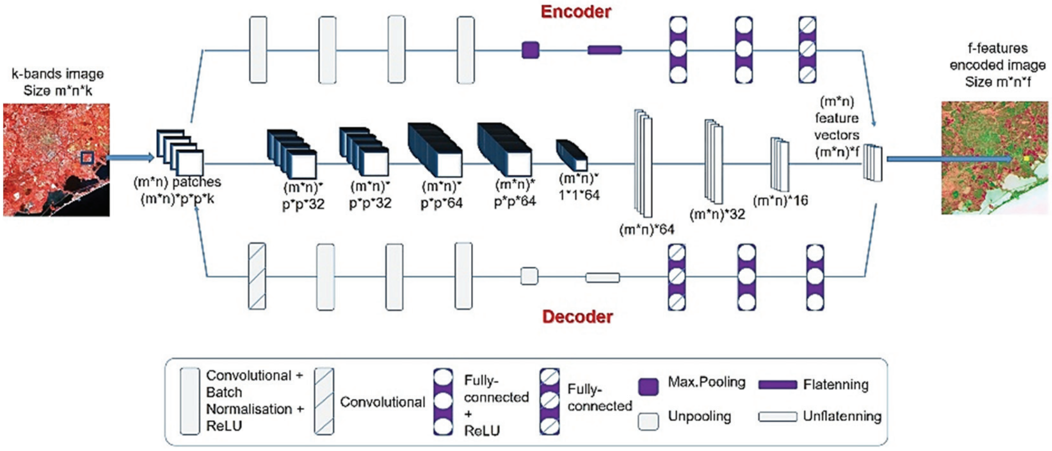


Fig. 1. Autoencoder model.

3. Experiments

Our dataset is made of 10 meters pixel resolution SPOT-5 satellite images of Montpellier, France, taken between 2002 and 2008, that belong to archive Spot World Heritage (available on https://theia-landsat.cnes.fr). In this section, we present the results obtained with 9 images from TS with green, red and NIR spectral bands. The original images were radiometrically normalized as described in [5].

To avoid overfitting, we sample $m \times n/9$ patches of size 5×5 pixels from every image of the time series to train the model with $f = 3$. After the training, the mean-squared reconstruction error of 6 pixel intensity values for 8-bits images was achieved. A stack of encoded images was then obtained.

To test the robustness of the proposed encoding approach for change detection, we choose the areas of interest that correspond to different types of objects. Among them are an urban area with no changes; an agricultural parcel that have seasonal changes; a vegetation zone with some minor seasonal changes, that was transformed in a construction site at T_7 and, finally, a building was constructed at T_9.

20 random pixels of every class were extracted to perform the analysis of their evolution in time. To visualize the evolution of pixels from 3-dimensional data, we transformed it in 1-coordinate dimension with PCA (Fig. 2). The analysis was performed for the original images of TS and for the encoded ones. We can state that the intra-class similarity is higher for the encoded images, hence, it facilitates the further change analysis. More importantly, using the encoded images, we can easily detect the beginning of the construction on T_7, thus proving the efficiency of our method to detect changes.

4. Conclusion

In this paper, we have presented a new approach to encode a SITS in its texture representation version. A new encoded TS is easier to interpret, and therefore easier to use for the change detection analysis.

In the future works, an algorithm to automatically reveal different clusters of objects and different types of changes in a SITS will be elaborated. It will be based on the pixel evolution analysis of encoded TS.

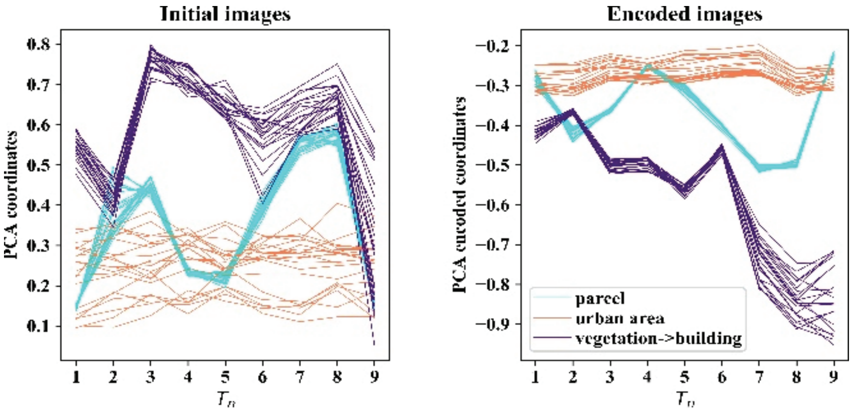


Fig. 2. Analysis of pixel evolution for the stack of images (initial – on the left, encoded – on the right).

References

[1]. Xu, L., S. Zhang, Z. He, Y. Guo. The comparative study of three methods of remote sensing image change detection, in *Proceedings of the 17th International Conference on Geoinformatics*, 2009, pp. 1-4.

[2]. F. Dellinger, J. Delon, Y. Gousseau, J. Michel, F. Tupin, Change detection for high resolution satellite images, based on sift descriptors and an a contrario approach, in *Proceedings of the IEEE Geoscience and Remote Sensing Symposium (IGARSS)'14)*, 2014, pp. 1281-1284.

[3]. D. Peng, Y. Zhang, Object-based change detection from satellite imagery by segmentation optimization and multi-features fusion, *International Journal of Remote Sensing*, Vol. 38, Issue 13, 2017, pp. 3886-3905.

[4]. Y. Xu, S. Xiang, C. Huo, C. Pan, Change detection based on auto-encoder model for VHR images, *Proceedings of SPIE*, Vol. 8919, 2013, 02

[5]. M. El Hajj, A. Bégué, B. Lafrance, O. Hagolle, G. Dedieu, et al., Relative radiometric normalization and atmospheric correction of a SPOT 5 time series, *Sensors Journal*, Vol. 8, Issue 4, 2008, pp. 2774-2791.

(44)

Methods for Minimizing and Understanding E-commerce Returns

A.K. Seewald [1], **T. Wernbacher** [2], **A. Pfeiffer** [2], **N. Denk** [2], **M. Platzer** [3], **M. Berger** [3] **and T. Winter** [4]

[1] Seewald Solutions, Lärchenstraße 1, A-4616 Weißkirchen a.d. Traun, Austria
[2] Donau-Universität Krems, Dr.-Karl-Dorrek-Straße 30, A-3500 Krems, Austria
[3] yVerkehrsplanung, Brockmanngasse 55, A-8010 Graz, Austria
[4] Attribu-i, Nibelungengasse 32d, A-8010 Graz, Austria
E-mails: alex@seewald.at, {thomas.wernbacher, alex.pfeiffer, natalie.denk}@donau-uni.ac.at, {mario.platzer,martin.berger}@yverkehrsplanung.at, thomas.winter@attribui.com

Summary: The importance of e-commerce including the associated freight traffic with all its negative consequences (e.g. congestions, noise, emissions) is constantly increasing. Already in 2015, a European market volume of 444 billion Euros at an annual growth of 13.3 % was achieved [8]. However, online commerce will only have a better footprint than buying in the local retail shop under optimal conditions. Next to frequent single deliveries, CO^2 intensive and underutilized transport systems, returned goods are the main problem of online shopping. The last is currently estimated at up to 50 % [4, 5]. Our research project Think!First tackles these problems in freight mobility by using an unique combination of gamification elements, persuasive design principles and machine learning. Here we list several methods to reduce and understand returns than have proven useful within our project.

Keywords: Machine learning, Visualization, Data mining, Rule learning, E-commerce, Returned goods.

1. Introduction

One important issue in retail e-commerce is the inherently high returns rate of up to 50% [4, 5]. Combined with a higher number of offers for shorter delivery times this results in a corresponding increase of freight traffic and therefore CO^2 emissions at a time when a reduction of these emissions is needed. On the other hand an increasing number of online shoppers want to buy in a sustainable way [2, 3]. Actual shopping decisions are however less sustainable due to ignorance, laziness and missing incentives. Our research project Think!First [7] aims to optimally inform and motivate online customers. A large Austrian shopping chain with online shop was our main project partner.

2. Initial Data Survey

From the data warehouse of our project partner we received a set of a few million samples from a time period of several years, containing detailed information on customers, orders, products, deliveries and returns during this time period. In total there were 312 features – 202 numeric features and 110 nominal features with up to 10 values (avg. 2.75) per feature. Overall returns over all products (including clothing) within this time period were within the ranges reported in the literature.

One trivial way to reduce returns would be to delist products or ban customers which are responsible for a large proportion of returns. To achieve a 10 % reduction in returns, for this dataset, we would need to ban 0.17 % of customers at the cost of selling 4.31 % less products overall; or delist 1.13 % of products at the cost of selling 13.46 % less products overall. Both alternatives were deemed unsatisfactory by our project partner.

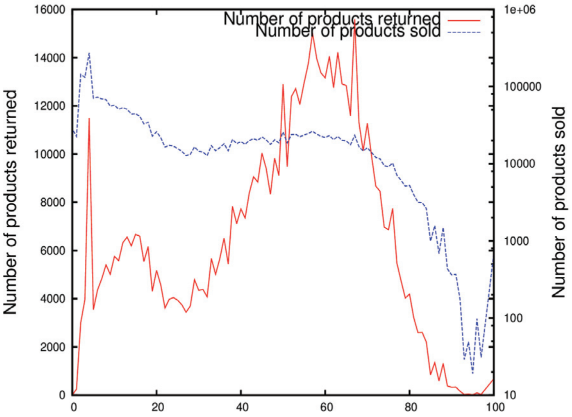


Fig. 1. Products sold vs. products returned (Y axis) vs. returns rate (X axis), grouped by products. See text.

We then decided to visualize all relevant parameters in a single graph. We binned returns rate in percent into 100 bins at the X axis. For each bin we computed both the number of products sold and the number of returns. Absolute number of products returned (linear scale) and sold (log scale) are shown on the left and right Y axis.

Fig. 1 shows the results. Here, we computed returns rate on products level (i.e. each product was assigned its average returns rate over all its sales). The bimodal distribution of returns indicates that returns are driven by two approximate normal distributions: a smaller peak of products which sell high numbers at low relative returns rate - which still translates to high absolute returns - and a larger peak of products which sell lower numbers at higher relative returns rate. The latter group contributes about six times the absolute number of returns of the former.

3. Characterizing Returns

We then aimed to characterize returns by a well-known rule learning algorithm, JRip, which is an open source implementation of RIPPER [1] within the data mining suite WEKA. We chose RIPPER for its ability to produce small concise rule sets that are easy to interpret. We also considered Logistic Regression but found that too many features had high weight resp. odds ratio, severely impairing interpretability of the trained model. We still used it to choose features for exclusion (see below). We down sampled a 20 % subset of the original dataset to 1:1 class distribution between returns and non-returns and evaluated the model on the remaining 80 % data that was not subsampled plus the nonreturns removed from the training data during subsampling. For initial evaluation we used twofold cross validation.

One major problem was that the data dictionary had not been updated for some time. Therefore from 312 features only about a third were actively used and well-known. To prevent inadvertently using features that are retroactively changed when returns are entered into the system – which yields results that are too good to be true – we chose two mitigations: 1) training the model on one set of data and evaluating on a later data warehouse export (ongoing), 2) successively removing highly predictive attributes – either by high odds value in logistic regression or by appearing often in the first 5-10 rules from RIPPER. The final choice which attributes to remove was mainly based on these two feature sets as little or no feedback on these attributes was available. The cut-off was manually chosen by visual inspection. Table 1 shows the results for the second mitigation. It can be seen that the removal of highly predictive attributes does not always reduce the performance of RIPPER but in some cases even proves beneficial. The addition of products base data from tr7 onward clearly proved beneficial at first and it will turn out that many final rules make use of those features.

Initially we ran RIPPER on the complete data from all years containing a few million samples (tr3-tr11). However, the data warehouse format had been changed at least three times during the last several years in which the data was collected causing some variables' interpretation to be changed and some new variables to be added, both causing RIPPER to return large sets of around a hundred rules to account for the additional data variance. So we chose to retrain it using only the latest data from 2017 and 2018 (tr11-17) containing a few hundred thousand samples. This led to a small set of eleven rules which predict returns on the independent test set with a precision of 0.495 and recall of 0.784 (balanced F-measure: 0.607), comparable to models trained on the whole dataset. More details see [6].

One problem with RIPPER was that according to its internal heuristics its default class was sometimes returns and sometimes non-returns. However since rules for non-returns are much harder to interpret than rules for returns and we are also much more interested in the latter, we chose to adapt RIPPER to force it to use only non-returns as default class and therefore only output rules that predict returns. This modified RIPPER was used from tr7 onwards.

Table 1. Successive removal of predictive features as estimated by two-fold cross validation.

Dataset	#F.	Prec.	Rec.	F_0	AUC
tr3	154	0.813	0.821	0.817	0.840
tr4	150	0.866	0.59	0.706	0.790
tr5	149	0.798	0.772	0.785	0.820
tr6	145	0.822	0.862	0.842	0.857
tr7	303	0.865	0.898	0.881	0.917
tr8	290	0.784	0.827	0.805	0.830
tr9	283	0.765	0.793	0.779	0.792
tr10	282	0.765	0.794	0.779	0.794
tr11	237	0.768	0.789	0.778	0.789
tr11-17	237	0.767	0.775	0.771	0.769

Concluding, we note that the most features of these rules were drawn from products base data followed by orders and only the last (non-default-)rule contains customers base data. This may indicate the relative importance of these feature subsets. Mainly, a more detailed analysis of these rules may lead to be a better understanding of the reasons for returns.

Acknowledgements

This project was funded by the Austrian Research Promotion Agency (FFG) and by the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT) as project Think!First (Proj. no. 859099)

References

[1]. W. W. Cohen, Fast effective rule induction, in *Proceedings of the Twelfth International Conference on Machine Learning (ICML'95)*, San Francisco, CA, 1995, pp. 115-123.

[2]. H. Hagemann, Umweltrelevante Produktinformationen im E-Commerce: Chancen für nachhaltigen Konsum, *Umweltbundesamt*, Germany, 2015.

[3]. J. Halbach, E. Stüber, M. Piepke, Nachhaltigkeit im Online-Handel – Die Rolle von Ausgestaltung und Kommunikation, *IFH Institut f. Handelsforschung*, 2015.

[4]. L. Hofacker, C. Langenberg. E-Commerce Markt Österreich/Schweiz, *EHI Retail Institute*, 2015.

[5]. K. Kristensen, N. Borum, L.G. Christensen, et al., Towards a next generation universally accessible online shopping-for-apparel system, *Human-Computer Interaction*, *Lecture Notes in Computer Science*, Vol. 8006, 2013, pp. 418-427.

[6]. A. K. Seewald, et al., Towards minimizing e-commerce returns for clothing, in *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (SCITEPRESS'19)*, 2019.

[7]. T. Wernbacher, et al., Minimizing returns through gamification, persuasive design principles & machine learning, in *Proceedings of the 11th European Conference on Games Based Learning (ECGBL'17)*, Oct. 5-6 2017, Austria, 2017.

[8]. R. Willemsen, J. Abraham, R. van Welle, Global B2C E-commerce Report, *Ecommerce Foundation*, The Netherlands, 2016.

(45)

Sensors Fusion for Head Tracking Using Particle Filter in a Context of Falls Detection

I. Halima [1,2], J. M. Laferté [1], G. Cormier[3], A. J. Fougères [1] and J. L. Dillenseger [2]
[1] ECAM Rennes, Campus Ker Lann, Bruz, 35091 Rennes, France
[2] Univ Rennes, Inserm, LTSI - UMR 1099, F-35000 Rennes, France
[3] Neotec Vision ltd, F-35740 Pacé, France
Tel.: + 33299058048
E-mail: imen.halima@ecam-rennes.fr

Summary: In the context of ageing societies, assessing risk factors and detecting falls for the elderly is becoming a crucial issue. In this paper, we propose an iterative head tracking method based on particle filtering using the fusion of low cost thermal and depth sensors for home environments whilst preserving privacy. The iteration process begins by segmenting the head in the depth image to calculate the depth coefficients and the thermal coefficients used for updating the particle weights. The method was tested on several sequences, with or without depth-thermal fusion: results show its robustness and accuracy, and also demonstrate that fusion improves tracking, namely when fast motion occurs (in case of a fall for instance) or when segmentation is erroneous.

Keywords: Head tracking, Sensor fusion, Particle filter, Thermal sensor, Depth sensor, Fall detection.

1. Introduction

Due to the ageing population and the fact that falls are the second cause of accidental deaths worldwide (according to WHO[1]), providing efficient fall detection systems of elderly is becoming crucial.

In previous work [1], we proposed a visual system solution based on the collaboration between a depth sensor and a low resolution thermal sensor to detect elderly falls. In the first study, we developed a fall detection system based only on the depth sensor. This system has been tested in 2 senior citizens' homes during 1 year but has produced too many false alarms caused mainly by occlusions and lack of shape tracking. For these reasons, we chose to add a thermal sensor. In this paper, we aim to develop a person tracking algorithm in order to improve the accuracy and the sensitivity of the system proposed in [1] and to reduce the number of false alarm.

This tracker will also be used to estimate the people's trajectories and to analyze their activities in order to prevent falls. This system is low cost, works day and night, can be easily installed in a room and moreover preserves privacy.

In our framework, instead of tracking the whole body, we chose to solely track the head because it is non-deformable, the hottest, highest and least hidden part of the body which can easily be approximate as an ellipse with only few parameters. The motion head is also a significant marker for the fall detection.

Among tracking methods, particle filtering (PF)-based ones have proven to be very flexible and to more accurately model the dynamics of the object motion [2]. PF is often applied for tracking in color images. In this paper we present an adaptation of PF to track people using both depth and thermal information.

In general, when the tracker is based only on a signal feature, the result can be wrong due to the complexity of the environment and the process of tracking [3]. In order to improve the tracking algorithm, multi modal data can be used to increase the performance. The main contributions of our paper are the using of different observations from the 2 sensors and the form to integrate these measures which increases significantly the performance of the tracking method.

The rest of the paper is organized as follows. In Section 2, we define calibration inter sensor and head segmentation steps. In Section 3, we illustrate particle filter algorithm on depth and thermal images. In Section 4, we report the experimental results and provide some discussions. Finally, we summarize our paper and point out some further directions in Section 5.

2. Calibration Inter Sensors and Head Segmentation Method

2.1. Depth and Thermal Calibration

Our system is based on a thermal sensor (FLIR lepton 2.5, Focal length: 5 mm, Thermal Horizontal Field of View T_HFOV: 51º, Thermal Vertical Field of View T_VFOV: 37.83º, T_Xres: 80 pixels and T_Yres: 60 pixels) and a depth camera (Microsoft Kinect, Focal length: 6.1 mm, Depth Horizontal Field of View D_HFOV: 58º, Depth Vertical Field of View

[1] World Health Organization (WHO)

D_VFOV: 45°, D_Xres: 640 and D_Yres: 480) which are aligned horizontally (see Fig. 1).

In order to match a point from depth image with a point in thermal image, a calibration step is required to calculate the transformation parameters (see Fig. 2).

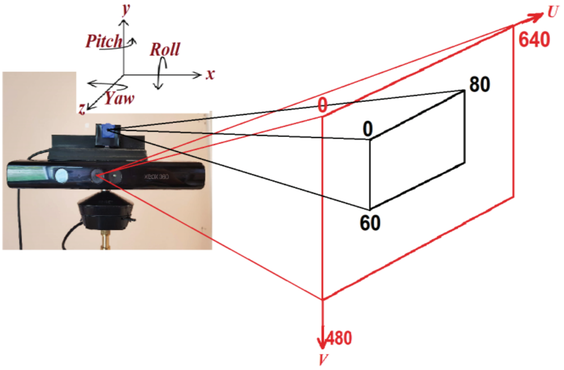


Fig. 1. Sensors coordinate systems.

As input we have depth and thermal images. u_i, v_i are the pixel coordinates in the images and w_d the depth information (i.e. the distance of the object to the depth optical center). The estimation of the relationship between these 2 coordinate systems needs 3 steps (see Fig. 2): **1)** The estimation of the transform of the depth image (u_d, v_d, w_d) to the coordinate system (x_d, y_d, z_d) of the depth sensor. This can be done analytically from the intrinsic parameters of the depth camera:

$$\begin{cases} x_d = \left(u_d - \frac{D\_X_{Res}}{2}\right) \frac{2w_d \, tan\left(\frac{D\_HFOV}{2}\right)}{D\_X_{Res}} \\ y_d = -\left(v_d - \frac{D\_Y_{Res}}{2}\right) \frac{2w_d \, tan\left(\frac{D\_VFOV}{2}\right)}{D\_Y_{Res}} \\ z_d = w_d \end{cases} \qquad (1)$$

2) The transformation between the coordinate system (x_d, y_d, z_d) of the depth sensor to this (x_{th}, y_{th}, z_{th}), of the thermal sensor. It can be obtained from the extrinsic parameters, in our case a rotation and a translation matrix:

$$\begin{pmatrix} x_{th} \\ y_{th} \\ z_{th} \end{pmatrix} = T + R \begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix},$$
$$R = \begin{pmatrix} cos(\alpha) & -sin(\alpha) & 0 \\ sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} ** \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix} ** \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \text{ and } T = \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}, \qquad (2)$$

α, θ and β are the Roll, Pitch and Yaw angles [1], and **3)** the transformation between the coordinate system (x_{th}, y_{th}, z_{th}) of the thermal sensor to the thermal image coordinates (u_{th}, v_{th}). This can be done analytically from the intrinsic parameters of the thermal camera:

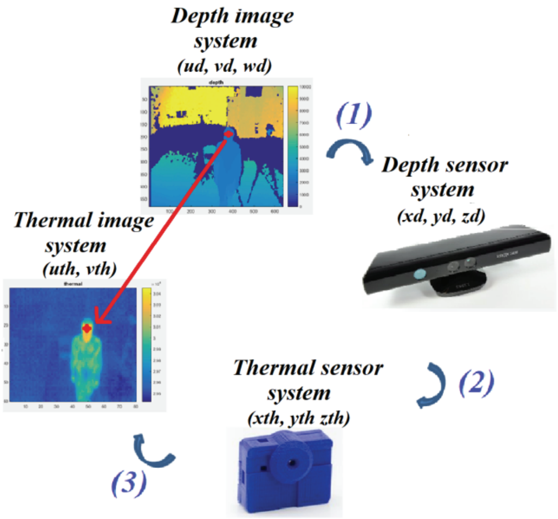


Fig. 2. Calibration system.

$$\begin{cases} u_{th} = \frac{T\_X_{Res}}{2z_{th} \, tan\left(\frac{T\_HFOV}{2}\right)} x_{th} + \frac{T\_X_{Res}}{2} \\ v_{th} = -\frac{Y_{Res}}{2z_{th} \, tan\left(\frac{T\_VFOV}{2}\right)} y_{th} + \frac{T\_Y_{Res}}{2} \end{cases} \qquad (3)$$

In our case, the intrinsic parameters are the values given by the constructor. So the purpose of the calibration is to estimate 6 parameters of equation (2) and then to generate a one to one pixel correspondence of the depth and thermal images.

2.2. Segmentation

In order to track the head, we need to extract the head position from the image. After calibration, we extract this position according to segmentation stage using the following framework (Fig. 3). The first step is to create a reference depth map by averaging the N first depth images without any moving objects. This defines the static background of the scene. From each frame at time N+t, we subtract the static background from this frame and detect so the moving objects we called as foreground. Then, we filter the foreground noise and we extract the silhouette and model it with an ellipse. Finally, the head size and pose are estimated from the silhouette ellipse. The head is modeled as a smaller ellipse with the same orientation as the silhouette ellipse but 3 time smaller. The center C of the head ellipse is set at 1/3 of the major axis from the upper part B (see Fig. 4): SC = 2/3 SB with S the silhouette ellipse center.

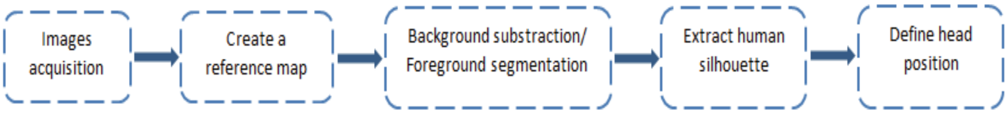


Fig. 3. Segmentation framework.

We define the depth head position C from this ellipse as another ellipse located at the 1/3 of the upper part $\overrightarrow{SC} = \frac{2}{3}\overrightarrow{SB}$ where S is the silhouette center and B is the upper point of silhouette. The head size and pose are extracted from the silhouette ellipse. They have the same orientation but the major axe of the head ellipse is 1/3 of silhouette major axe (see Fig. 4). In thermal image, we apply the matching to get the thermal head position.

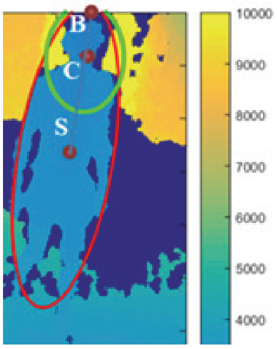


Fig. 4. Head position.

3. Particle Filter Algorithm on Depth and Thermal Images

The PF tracking process is based on a hidden state vector x_t which is defined by the center, orientation and the size of the ellipse around the head in the depth image. x_t is predicted from x_{t-1} and the observation vectors, $Z_t = \{z_1, .., z_t\}$ and $H_t = \{h_1, .., h_t\}$ obtained from depth and thermal sensors, respectively. PF uses a sample of N particles $S_t = (S_t^1, \dots, S_t^N)$ to approximate the conditional probability $p(x_t / Z_t, H_t)$. Each particle S_t^n can be seen as a hypothesis about x_t (an ellipse model) and is weighted by $\pi_t(n)$ which are normalized. Particles are resampled according to their weights and are updated according to new observations ($\boldsymbol{new_{observ}}$) [3].

3.1. Tracking Method

For each frame (time step t) t, we have x_t which represent the head. We sample this vector on N particles. Each particle S_t^n is defined by the same parameters of (x_t) and has a weight $\pi_t(n)$. Next, we predict x_{k+1} according to the propagation of particles based on:

$$S_t^n = A * S_{t-1}^n + w_t, \tag{4}$$

where A is the transition model matrix and w_t is a Gaussian noise. Finally, we update the particle weight according to observation vectors where we combine depth and thermal information in $\boldsymbol{new_{observ}}$ and resample particles to prevent the problem of particles degeneration (see Fig. 5) [4].

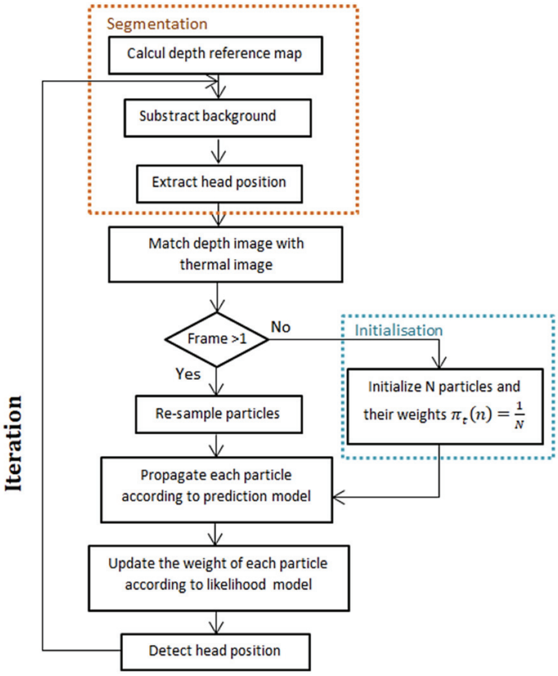

Segmentation
Calcul depth reference map
Substract background
Extract head position
Match depth image with thermal image
Frame >1
No
Yes
Initialisation
Initialize N particles and their weights $\pi_t(n) = \frac{1}{N}$
Re-sample particles
Propagate each particle according to prediction model
Update the weight of each particle according to likelihood model
Detect head position
Iteration

Fig. 5. Tracking method flowchart.

Thus, the steps of iterative PF tracking algorithm are:

1) ***Initialization***: Generate a sample of N particles $S_1 = (S_1^1, \dots, S_1^N)$ based on the probability of the state vector $p(x_1)$, and initialize the weight of each particle by 1/N,
2) ***Resampling***: Re-sample particles to prevent the problem of particles degeneration, if frame > 1,
3) ***Prediction***: Propagate particles according to prediction model to predict the state vector x_t,
4) ***Updating***: Update the particle weight at frame t.

$$\pi_t(n) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}\boldsymbol{\pi\sigma}}} \boldsymbol{e}^{\boldsymbol{new_{observ}(n)/2\sigma^2}}, \tag{5}$$

where σ was fixed empirically.
Then normalize the weight:

$$\pi_t(n) = \frac{\pi_t(n)}{\sum_{k=1}^{N} \pi_t(k)}, \tag{6}$$

and return to step 2.

3.2. Depth-thermal Fusion

Updating particle weights is a key point of PF and is specific for each application (see [5] for color information).

In our framework, the weights are computed from the following three coefficients which combine depth and thermal information: a distance coefficient C_d, a depth gradient coefficient C_{g_d} and a thermal gradient coefficient C_{g_th}. For each particle, C_d is the distance between the center of the particle and the center of the segmented head in depth image. C_{g_d} and C_{g_th} are based on the gradients along the head ellipse in the depth image and the thermal image respectively along the ellipse, as inspired by [6].

In this work, we considered two models to combine coefficients and use them in equation (5) for the updating step. Comparisons between these models are given in subsequent sections of this paper.

The first weighing model (M1) uses only 2 coefficients ($\mathrm{C_d}$ and $\mathrm{C_{g\_th}}$):

$$C_{fusion}(n) \;=\; \boldsymbol{\alpha}\, C_d + (1 - \boldsymbol{\alpha})\, C_{g\_th}, \tag{7}$$

$\boldsymbol{\alpha}$ is an Importance Factors (IF) and n is the particle number.

The second weighing (M2) model combines all the 3 coefficients:

$$C_{fusion}(n) \;=\; \boldsymbol{\alpha}\, C_d + \boldsymbol{\beta}\, C_{g\_d} + \boldsymbol{\gamma}\, C_{g\_th}, \tag{8}$$

α and β are the IF of depth information and γ = 1-α-β is the IF of thermal information. In order to estimate the impact of each coefficient in the particle filter, we tested several combination of static IF at each test (see Table 1).

Table 1. Importance factor IF values.

Test	α	β	1-α-β
C1	1/3	1/3	1/3
C2	1/4	1/2	1/4
C3	1/4	1/4	1/2
C4	1/2	1/4	1/4
C5	3/8	1/4	3/8
C6	3/8	3/8	1/4

Particle weight is updated according to equation (5) by replacing $new_{observ}$by the weighing model:

$$\pi_t(n) \;=\; \frac{\mathbf{1}}{\sqrt{\mathbf{2}\boldsymbol{\pi\sigma}}}\, e^{C_{fusion}(\boldsymbol{n})/2\sigma^2} \tag{9}$$

4. Experimental Results

In this section, we demonstrate the performance of the proposed algorithm. We have performed several sequences of people moving in a room with co-calibrated static depth and thermal cameras which were fixed in the ceiling. We have tested our system with the following objectives: (1) compare our proposal work with segmentation only and depth tracking method, (2) evaluate the performance of the fusion algorithm, (3) evalute each weighing model, and (4) compare IF values.

In all tests, we used the following parameters: N = 1000 particles, $\sigma\ =\ 0.25$ and transition model matrix A $= I_4$. We fixed the acquisation frequency to 8 Hz.

The ground truth (GT) was established manually by setting an ellipse on each frame (white ellipse in Fig. 6, 7) and the processing was performed using Matlab on Intel(R) Core(TM) i7-6700HQ CPU, 2.6 GHz.

We used 2 quantitative measurement to assess the accuracy of the specific versions: the localization error which is defined as the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truths, and the overlap score which is the overlap of the ground truth area and the tracking area [7].

Fig. 6 illustrates the results of a) The head segmentation only, b) The depth version, c) The M1 model. Comparing these results, we can see that the head segmentation is totally wrong, the depth version estimates an area bigger than the head and the fusion model is able to track the head more accurately. These results shows that segmentation is erroneous in critical situations and depth sensor is useless on its own.

As illustrated on Fig. 9, a comparaison between the 6 tests performed with the second model (M2) that shows the impact of the cofficient IF to estimate the new head position.

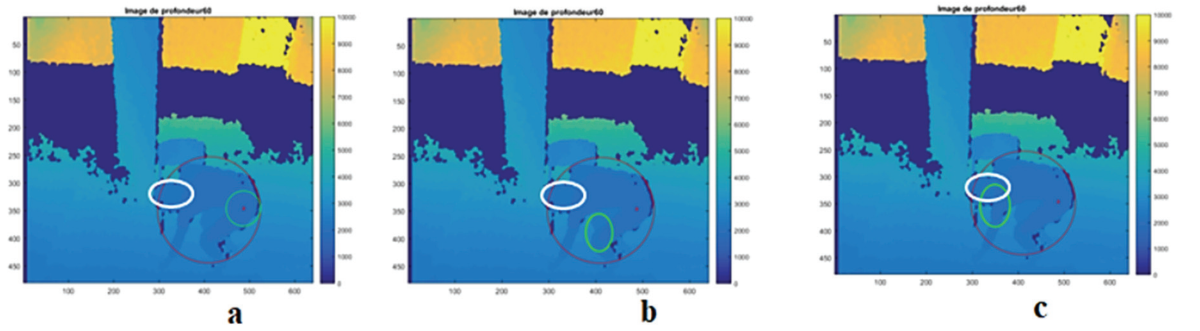


Fig. 6. Tracking results on one frame of a sequence: a) Segmentation only; b) Depth version; c) M1 model. Tracking results are in green, silhouette ellipse is red and GT ellipse is white.

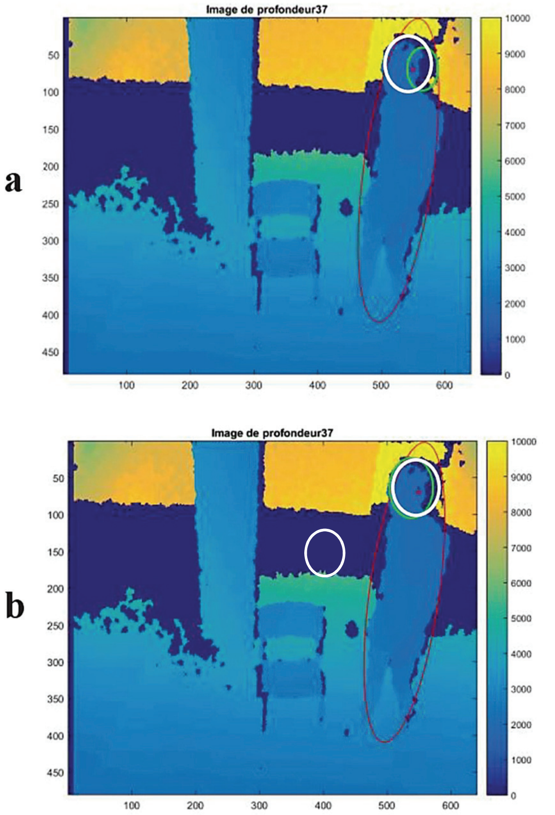


Fig. 7. Tracking comparison results between a) M1 model and b) M2 model. Tracking results are in green, silhouette ellispe is red and GT ellipse is white.

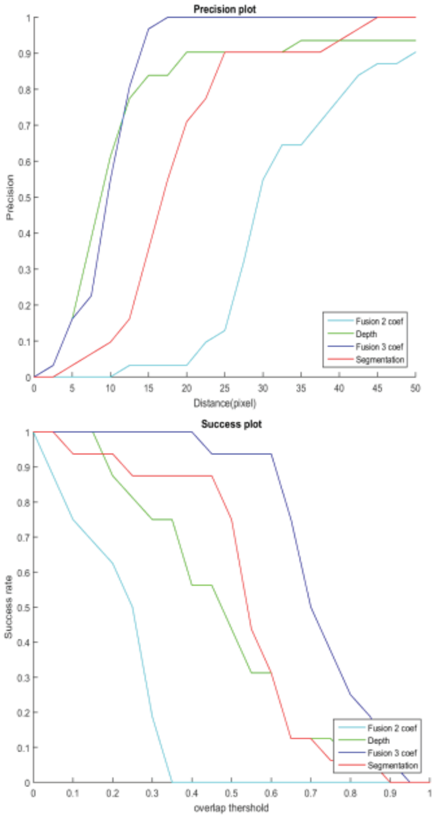


Fig. 8. Quantitative measurements over a sequence. Localization error (a) and the overlap score (b) using the segmentation (red), the depth version (green) the M1 model (cyan) and the M2 model (blue).

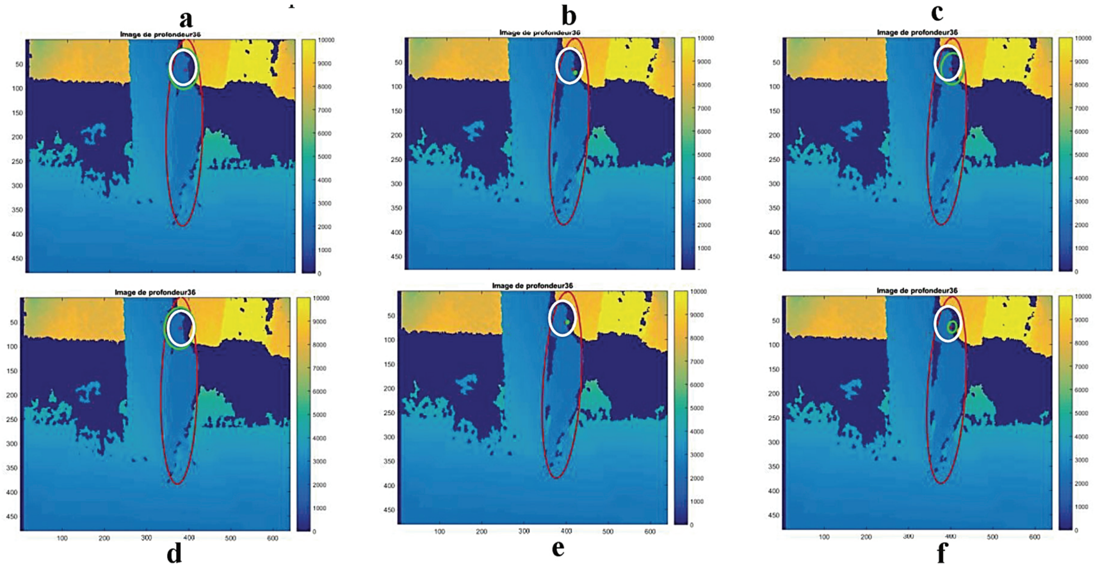


Fig. 9. Tracking results of 6 IF tests a) C1 test; b) C2 test; c)C3 test; d) C4 test; e) C5 test and f) C6 test, Tracking results are in green, silhouette ellispe is red and GT ellipse is white.

When we evalute these results using the 2 quantitative measurement, we observe in (Fig. 10) a clear difference between the performance of the C4 test that assign more improtance to distance coefficient and C2 that assigns more importance to depth gradient coefficient. For instance, when the person is walking in front of something, the depth gradient coefficient decreases the results and the distance coefficient gets the results better.

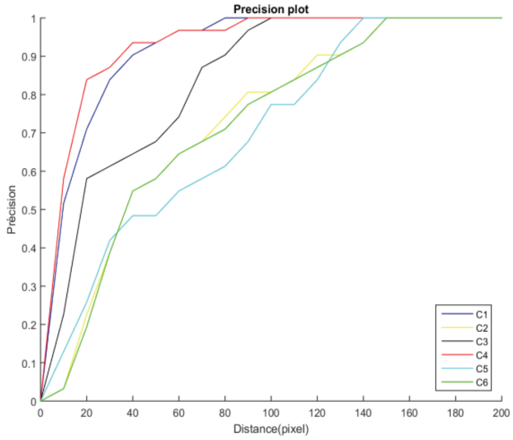


Fig. 10. Quantitative IF measurements over a sequence. Localization error of C1 test (blue), C2 test (yellow), C3 test (black), C4 test (red), C5 test (cyan) an C6 test (green).

5. Conclusions

In this paper, we have presented a head algorithm based on particle filters that fuse the information of depth and thermal of a person in an indoor environment using the position, the orientation and the size of the ellipse around the head. The scene was acquired by two co-calibrated depth and thermal cameras. We have presented calibration system that can be implemented on any depth thermal system.

We have evaluated our proposal work in several situations with different models and compared it with other methods. Moreover, results have shown that our system gave the most accurate tracking results even in critical situations with very low resolution images. The combination of information provided by both cameras improve the tracking. For future work we plan: 1) To modify the constant importance factors α β and γ to dynamic values according to the influence of each coefficient; 2) To make an automatic GT with Vicon Systems in living labs; 3) To compare the method with other robust methods like Deep Learning based methods and 4) to add the velocity in the state vector.

Acknowledgements

This work is funded under the PRuDENCE project (ANR-16-CE19-0015-02) which has been supported by the French National Research Agency.

References

[1]. G. Cormier, Analyse statique et dynamique de cartes de profondeurs: application au suivi des personnes à risque sur leur lieu de vie, PhD Thesis, *Université Européenne de Bretagne*, 2015.

[2]. E. Erdem, et al., Particle filter based visual tracking by fusing multiple cues with context sensitive reliabilities, *Pattern Recognition,* Vol. 45, 2012, pp. 1948-1959.

[3]. M. Isard, et al., Condensation – Conditional Density Propagation for Visual Tracking, *International Journal of Computer Vision*, Vol. 29, Issue 1, 1998, pp. 5-28.

[4]. A. Sanjeev, M. Simon, G. Neil, C. Tim, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing,* Vol. 50, Issue 2, 2002, pp. 174-188.

[5]. K. Nummiaro, et al., An adaptive color based particle filter, *Image and Vision Computing,* Vol. 2, Issue 1, 2003, pp. 99-110.

[6]. C. Rougier, et al., 3D head tracking for fall detection using a single calibrated camera, *Image and Vision Computing*, Vol. 31, Issue 3, 2013, pp. 246-254.

[7]. Y. Wu, et al., Online object tracking: A benchmark, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*, 2013, pp. 2411-2418.

(49)

Intelligent Models and Methods Decision Support System for Management of ICT-industry Development in Regions

N. I. Yusupova [1], O. N. Smetanina [1], E. Yu. Sazonova [1], A. G. Ionis [1], A. M. Morozov [2], T. I. Fabarisov [1]

[1] Ufa State Aviation Technical University, Ufa, Russia

[2] Dresden University of Technology, Dresden, Germany

E-mails: yussupova@ugatu.ac.ru, smoljushka@mail.ru, rassadnikova_ekaterina@mail.ru, solomonarmkeys@gmail.com, andrey.morozov@tu-dresden.de, flatagir@gmail.com

Summary: Authors consider intelligent models and methods the decision support system for management of the ICT-industry development in the regions. This article considers identification of implicit knowledge on a complex data. The task statement and the methodology of solution are given. Data mining technologies are applied for identifying hidden common factors in data. Details considered data analysis tools, as like factor analysis and clustering.

Keywords: Neural network technology, Data mining, Decision support, Factor analysis, Information and communications technology.

1. Introduction

Nowadays, the fact that information and communication technologies play a primary role for the modern person is beyond doubt. And this is connected not only with everyday life. Innovation policy in the field of information and communication activities is associated with the general recovery of the economy, the country's competitiveness. At the same time, when investigating the development of this industry, there is a problem associated with the constant emergence of new technologies and opportunities, because ICT industry is one of the fastest growing industries both in the world and in Russia. But despite the rapid growth, Russia still lags behind the level of world powers. This is due, among other things, to the fact that the one of the main problem in the development of the ICT market in Russia is a meaning digital divide between the 85 regions that make up the country. According to statistics for 2017, there is an explicit lag in a number of indicators even within one federal district, for example, between Moscow and the Bryansk region. Thus, there is the problem of digital inequality of the regions of Russia, associated with geographical, economic and managerial differences. One of the solutions to this problem can be the modeling of the level of ICT development at the regional level, the identification of similarities between regions, taking into account the values of indicators characterizing the level of development and building a rule base based on the interpretation results.

2. State of the Problem

All research in this area somehow affects the Solow paradox, which suggests that information technology does not affect labor productivity, and attracting investment in computerization of production entails only more investment. However, after a sharp rise in productivity in the 1990s, most researchers and scientists agreed that the ICT industry was the cause of such a sharp jump. Since the middle 2000s, the results of research by many experts [1-7] show that there is a correlation between the development of information and communication technologies and the level of economic development in the country. Thus, according to the analytical service of The Economist magazine [7, 11], there is a mutual merger between ICT level and economic development in developed countries, while for developing countries this effect is not observed or is insignificant. Among modern researches, there is to single out the use of the production function method for testing the hypothesis about the low value of the innovative potential of the country in the field of ICT for support economic development, presented by authors [5, 6], these researchers use methods of correlation and regression analysis to identify the relationship between innovations in ICT and small business development. However, all these studies have a number of significant shortcomings, which are confirmed by the results of analytical reports. One of the most successful is the empirical study of the Economist Intelligence Unit [11], which took as a basis not only the level of ICT use, but also the state of the business environment, as well as conditions conducive to the introduction of new technologies from the ICT area. The resulting model allowed not only to differentiate countries depending on the level of development, but also to explore the effects of interaction. But the question of the reasons for the influence of the ICT industry on the level of economic development in some countries still remains unresolved. In this connection, the hypothesis was proposed that the ICT industry begins to influence the country's economic growth, expressed in GDP only when the so-called critical mass, that is, the saturation point, is reached. In this case, the methods and models proposed in the works, as a rule, take into account only the relationship between the level of ICT development and the level of economic development. However, it makes sense to take into account both the contribution of the ICT industry in the country's GDP and the development of the ICT industry itself. The development of the ICT industry in each region of the country is seen as fundamental, contributing to the level of development of the country's industry as a whole. In this case, the rating of regions is determined annually.

3. Statement of the Problem

The authors proposed to develop a methodology for identifying similarities between objects (regions) based on the most significant factors, which influence the industry development (taking into account the rating), and with using new knowledge to formulate recommendations for managing the industry development.

Problem statement can be formulated as follows: Formulate recommendations to improve the level of the development level based on analysis of the initial data presented by a set of vectors, which describes the characteristics of the regions of the Russian Federation in the ICT-industry development.

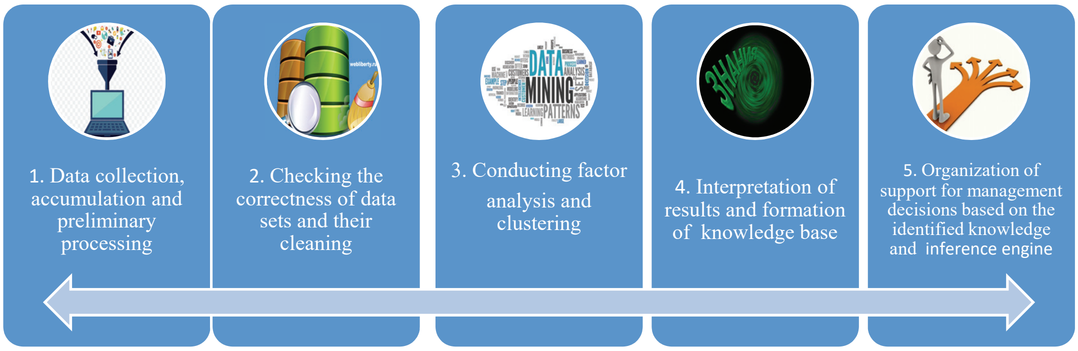
Mathematical statement of the problem of the research is formulated as follows.

Given X - a set of vectors represented by the characteristics of the regions in the region of X_i, where $X_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$, here $x_{i1}, x_{i2}, \dots, x_{in}$ – region information, where $i = \overline{1,85}$, $j = \overline{1,n}$ – j factor taken into account that influences the level of development (total n factors)

Find: $F: X_i \rightarrow Y_c$, function that maps data to knowledge. This function will make it possible to assign a region with take into account its characteristics to one of clusters that unite regions with similar characteristics, and to construct RULE rules, using which it is possible to formulate recommendations for the industry development.

4. The Technique for Solving the Problem

The technique for solving the problem (Fig. 1) includes five stages. The first two stages are aimed at preparing data for analysis. At the first stage, statistical information is collected, the characteristics of the objects are selected for further research. At the second stage, preliminary data processing is carried out. Pre-processing is necessary to control the invalid values of selected indicators, as well as to detect, correct and delete erroneous records. At this stage, using the approximation functions and the means of the existing application programs, the existing gaps in the characteristic values are filled, individual criteria are standardized, for example, to normalize indicators taken in monetary terms, it is necessary to impose an index deflator. At the third stage, a factor analysis is carried out to identify the most significant factors. The choice in favour of this method is due to the fact that the factor analysis tool allows you to: determine the data structure, reduce the dimensionality of the attribute space without losing information (relevant for data with a high level of correlation), and also highlight the most significant characteristics for regulating the process under study. At the fourth stage, the neural network apparatus, in particular the Kohonen network, is used to identify similarities through clustering. In this study, clustering is necessary because it allows you to select one or more attributes as a basis for generalized groups of regions, and also to show what their similarities are and how the ranges of values are coordinated. A feature of this method is also that clustering can work in “both sides”.



1. Data collection, accumulation and preliminary processing

2. Checking the correctness of data sets and their cleaning

3. Conducting factor analysis and clustering

4. Interpretation of results and formation of knowledge base

5. Organization of support for management decisions based on the identified knowledge and inference engine

Fig. 1. Solution technique.

When analyzing, it can be assumed that there is a cluster at a certain point and then to test this hypothesis using the existing criteria values, on the other hand, taking into account the available input data, to identify various properties and processes occurring in the objects. At the fifth stage, the revealed new knowledge in the form of characteristics of similar objects by interpreting the data makes it possible to formulate recommendations for the development of the industry in the region. At this stage of solving the problem, in addition to the results of cluster analysis, it is proposed to use the knowledge of experts.

The comprehensive analytical Deductor Studio platform (steps 1-3) and the shell, which allows to realize a production system (stages 4-5) are used to implement the solution techniques.

5. The Results of the Study

Interpretation of the factor analysis results allows to be drawn the following conclusions: the ICT-level and ICT-industry development in the regions is significantly affected by the organization's internal resources (the cost of research and its share of gross product; additional training for staff (human capital increase); in fixed capital; the share of computer use in the organization); the proportion employed in ICT; development of communication networks and data transmission; the share of graduates of IT areas (specialties) in the total graduation; the development level of technology; labor market index.

The obtained clustering results (Fig. 2) are interpreted and allow identifying similar objects with both high and low ratings (Table 1).

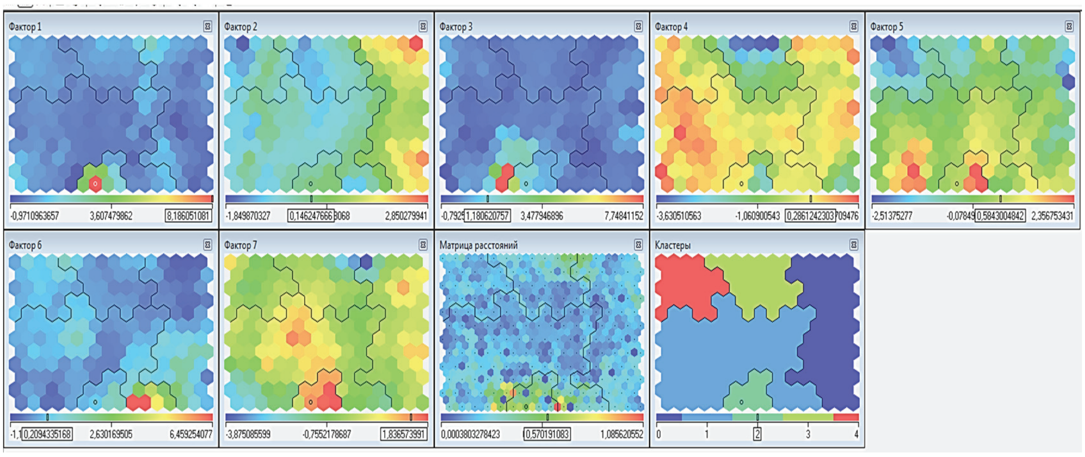


Fig. 2. Clustering results.

Table 1. The fragment of clustering results.

Cluster Factor value	0	1	2	3	4
Factor 1	Below the average	Low	Medium and high	Below the average	Below the average
Factor 2	Average, above the average	Below the average	Medium	Below the average	Average and low
Factor 3	Low	Below the average	Medium	Low	Below the average
Factor 4	Above the average	Average, above the average	Average, above the average	Low, average	Average, above the average
Factor 5	Average	Average, above the average	Above the average and high	Average	Average and below the average
Factor 6	Average, above the average	Average, above the average	Average	Below the average	Low
Factor 7	Average and above average	Average and above average	High	Average	Average
Regions (class in rating)	Novosibirsk Oblast, Yaroslavl Oblast, Perm Territory, Mari El Republic, Chuvash Republic, etc. (2-6)	Belgorod, Omsk, Irkutsk, Arkhangelsk, Leningrad regions, Krasnodar Territory, etc. (7-12)	Moscow, St. Petersburg (1)	Tverskaya, Ivanovo regions, the Republic of Ingushetia, etc. (9,13,17)	Republics of Altai, Tyva, Zabaykalsky Krai, etc. (14-16)

Specific values of the factors "translated into terms of the corresponding names of factors of linguistic variables" for later use in the formation of fuzzy knowledge base. At this stage of the research, interpretation results are laid down in production rules. In order to orient regions on higher rating, it is necessary to increase / decrease the values of some parameters.

As examples, the following production rules can be given:

Rule 1: IF cluster = 0 AND 8 ≤ rating ≤ 15, THEN IT companies need to find internal resources for additional training of employees, for scientific research and for hardware upgrades.

Rule 2: IF cluster = 4 AND rating = 14, THEN IT companies need to find internal resources for additional training of employees, for scientific research, hardware upgrades, AND to increase the level of science and technology development by creating innovation centers.

Rule 3: IF cluster = 1 and 9 ≤ rating ≤ 12, THEN it is necessary to increase the proportion of people employed in IT, for example, by creating new IT companies.

Rule n: IF cluster = 0 and 1 ≤ rating ≤ 15, THEN there are need to increase the output of IT specialists in universities by attracting commercial students, working with existing students, strengthening career guidance, increasing the admission plan.

5. Conclusions

Any development strategy for the ICT industry in the region are guided by human capital; innovation potential; institutional and infrastructural environment; ICT infrastructure and access; information industry; information security; use of ICT for development, etc.

The study of indicators characterizing information and communication activities, ranked by constituent entities of the Russian Federation, revealed a strong differentiation between the central part of the country and the other federal districts. The indicators of Moscow, the Moscow region are markedly different from the all-Russian (they are anomalous observations for all the ICT indicators studied).

The proposed concept is focused on identifying similar objects by conducting cluster analysis. The analysis results are interpreted according to the semantics of the application domain. Authors describes the tools develop recommendations that provide information support for decision-making. In the future, authors propose the use of neuro-fuzzy tools to solve the problem.

Acknowledgements

The research is supported by Grants RFBR 19-07-00709-a and 18-07-00193-a.

References

[1]. E. V. Presnyakov, Impact of information and communication technologies on the global economy, *Izvestiâ Sankt-Peterburgskogo Gosudarstvennogo Èkonomičeskogo Universiteta*, 2011, pp. 132-134.

[2]. A. A. Erumbanab, D. K. Das, Information and communication technology and economic growth in India, *Telecommunications Policy*, Vol. 40, Issue 5, May 2016, pp. 412-431.

[3]. D. K. Khakimdzhanova, The ICT development and impact on the economy, *Young Scientist*, Vol. 7, 2017, pp. 290-293, https://moluch.ru/archive/141/39836

[4]. V. I. Ivanova, E. P. Mitrofanov, T. N. Polkanova, The importance and role of information and communication technologies in the modern economy and innovation activities, *Vestnik Chuvashskogo Universiteta*, Vol. 2, 2013, pp. 253-257.

[5]. O. Valieva, Small Innovative Business in Russia: Problems and Prospects, 2012.

[6]. S. A. Moskalonov, A. G. Lvov, Analysis of the innovative potential of the Russian economy: The method of production functions, 2010.

[7]. K. V. Zimin, A. V. Markin, K. G. Skripkin, The Impact of Information Technology on the Performance of a Russian Enterprise; Methodology of Empirical Research, 2012, http://ecsocman.hse.ru/data/2012/06/06/1271384056/6.pdf.

[8]. Brynjolfsson and Hitt, Information, Technology, and Productivity, Brynjolfsson and Hitt, 2005.

[9]. Brynjolfsson and Yang, Exploring the Relationship Between Information Technology and Business Process Reengineering, 1996.

[10]. S. Baller, S. Dutta, B. Lanvin, The global Information Technology Report 2016, http://www3.weforum.org/docs/GITR2016/WEF_GITR_Full_Report.pdf

[11]. The Economist intelligence Unit, Industry Analysis, http://www.eiu.com/home.aspx

[12]. The Next Level, Annual Report 2017, https://ict.eu/wp-content/uploads/2018/03/ICT-Annual-Report-2017.pdf

[13]. Ministry of Communications of the Russian Federation, http://www.mskit.ru/

[14]. Federal State Statistics Service, http://www.gks.ru;

[15]. Information and Legal Portal Garant, http://base.garant.ru/12148555/

[16]. Consultant Plus Company, http://www.consultant.ru

[17]. V. A. Andreev, Factors determining the commercial success of research and development of various levels of novelty in the real sector of the Russian economy.

(50)

Advanced Analysis of Online Handwriting in a Multilingual Cohort of Patients with Parkinson's Disease

J. Mucha [1], **J. Mekyska** [1], **M. Faundez-Zanuy** [2], **P. Sanz-Cartagena** [3], **Z. Galaz** [1,5], **V. Zvoncak** [1], **T. Kiska** [1], **Z. Smekal** [1], **K. Lopez-de-Ipina** [4] **and I. Rektorova** [5]

[1] Department of Telecommunications and SIX Research Centre, Brno University of Technology, Technicka 10, 61600 Brno, Czech Republic
[2] Escola Superior Politecnica, Tecnocampus Avda. Ernest Lluch 32, 08302 Mataro, Barcelona, Spain
[3] Hospital de Mataró, Neurology Unit, Consorci Sanitari del Maresme, Mataró, Spain
[4] Department of Systems Engineering and Automation, University of the Basque Country UPV/EHU, Av de Tolosa 54, 20018 Donostia, Spain
[5] Applied Neuroscience Research Group, Central European Institute of Technology, Masaryk University, Kamenice 5, 62500 Brno, Czech Republic
Tel.: + 421 915 420 540, +420 54114 6968
E-mail: muchajano@phd.feec.vutbr.cz

Summary: The majority of Parkinson's disease (PD) patients suffer from handwriting abnormalities commonly called as Parkinsonic dysgraphia. Several approaches of PD dysgraphia analysis exist, e.g. based on online handwriting processing. However, a small and unilingual cohort of PD patients is often an issue in quantitative PD dysgraphia analysis studies. Therefore, in this work, we aim to perform a discrimination analysis in a multilingual cohort of 73 PD patients and 48 healthy controls (Spanish and Czech). For this purpose, we extracted advanced handwriting features based on fractional order derivatives (FD). Discrimination power of the advanced FD-based features was evaluated by Mann-Whitney U test and random forests classifier. We reached 82 % classification accuracy (86 % sensitivity, 77 % specificity) in the multilingual cohort. In addition, we observed high discrimination power of the FD-based parameters and proofed the high impact of online handwriting processing in cross-cultural PD dysgraphia analysis studies.

Keywords: Parkinsonic dysgraphia, Micrographia, Online handwriting, Fractional order derivative, Fractional calculus, Multilingual cohort.

1. Introduction

Parkinson's disease (PD), as the second most frequent neurodegenerative disorder, affects approximately 1.5 % of the world population aged over 65 years [1]. A rapid degeneration of dopaminergic cells in substancia nigra pars compacta emerged as the most important biological finding accompanying the disease [2]. Considering the cardinal motor symptoms of PD (tremor in rest, bradykinesia and rigidity) in conjunction with cognitive, perceptual and motor requirements of handwriting, the disrupted handwriting of PD patients may be used as a significant biomarker for PD diagnosis [3]. The most commonly observed handwriting abnormality in PD patients is micrographia (progressive decrease of letters amplitude) [4], which may be noticed even before the onset of PD motor symptoms in approximately 5 % of PD patients.

Nowadays, by utilizing digitizing tablets, which brings an ability to acquire x and y position with temporal information, we have the opportunity to process online handwriting signals. Therefore, we are not limited to analyze the spatial features only, but we are able to quantify more manifestations of PD appearing in patients handwriting data (temporal, kinematic or dynamic), generally named as *PD dysgraphia* [5].

The impact of quantitative PD dysgraphia analysis employing several handwriting or drawing tasks (e. g. characters, loops, sentences, figures) has been explored in [6-9]. Researchers usually use kinematic, temporal, spatial or dynamic handwriting features in PD dysgraphia analysis. However, more advanced parameters (based on entropy, energy operators or empirical mode decomposition) have been reported too. PD dysgraphia classification accuracies reported by recent works vary in the range of 85 and 97 %. In our previous works [6, 10, 11], we proposed and evaluated a new advanced approach of kinematic analysis based on fractional order derivatives (FD). Using this approach, we were able to identify PD with almost 90 % accuracy employing only 5 basic kinematic features.

The most common issue in PD differential analysis (cause by complicated and time-consuming patient examination process), which researchers are encountering with, is a small and unilingual cohort of patients. This may result into poor generalization. Especially, the size of examining dataset has a significant influence on results reliability. The smallest the dataset is, the more misleading results may be. Therefore, in this study, we aimed to analyze a multilingual cohort involving two PD handwriting databases (Czech and Spanish) in order to train a more robust classification model. To our best knowledge, this is the first study considering multilingual cohort in PD dysgraphia analysis.

2. Datasets and Methodology

2.1. Datasets

For the purpose of this study, we used two PD handwriting databases. The Czech (PaHaW [8]) database consists of 37 PD patients and 38 healthy controls (HC). It includes 9 different handwriting tasks (Archimedean spiral, repetitive loops, repetitive letter *l*, syllable, words and sentence). The Spanish database (recorded in Mataró Hospital, Spain) consists of 36 PD patients and 10 HC. It includes 2 handwriting tasks (repetitive and continuously written letter *l* and sentence). Demographic and clinical data of both cohorts can be found in Table 1. All patients were examined on their regular dopaminergic medication approximately 1 hour after the L-dopa dose. All participants were right-handed, and all participants signed an informed consent form approved by the local ethics committees.

Table 1. Demographic and clinical data of all participants.

Cohort	Number	Age [y]	PD dur [y]
Parkinson's disease patients			
Czech	37	69.21 ± 11.10	8.70 ± 4.82
Spanish	36	68.25 ± 10.46	6.10 ± 3.78
Healthy Controls			
Czech	38	62.50 ± 11.70	-
Spanish	10	57.50 ± 6.36	-

[1] y – years; dur - duration

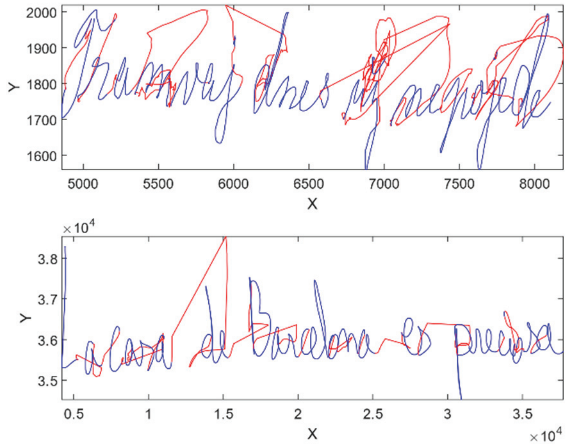


Fig. 1. PD patient's sentences examples. Czech sentence in the upper part and Spanish in the bottom part of the figure. On-surface (blue) and in-air (red) movement are visualized.

For the purpose of this study, sentence handwriting task was selected from the databases. Even the tasks are different due to language, we hypothese that pathological characteristics in the handwritten signals will be similar. Sentences in their original language and with resulting English translations are listed below:

a) Czech: *"Tramvaj dnes už nepojede"*.
   English: The tram will no longer go today.
b) Spanish *"La casa de Barcelona es preciósa"*.
   English: *The house in Barcelona is beautiful.*

Samples of PD patients' sentences can be found in Fig. 1. In Fig. 2, descriptive statistics of both datasets are visualized. Handwriting data were acquired using a digitizing tablet Wacom Intuos 4M (both datasets). Following time sequences were sampled with frequency fs = 150 Hz: x and y coordinates (x[t], y[t]); time-stamp (t); in-air/on-surface status (b[t]); pressure (p[t]); azimuth (az[t]); and tilt (al[t]).

2.2. Methodology

Firstly, each handwritten signal was split into on-surface and in-air movements [12] (see Fig. 1). Next, basic kinematic features such as velocity, acceleration and jerk were extracted. Instead of conventional differential derivative, we utilized FD as an advanced approach of kinematic features calculation. For this purpose, the Grünwald-Letnikov approximation was used [13, 14]. The advantage of FD is based on their extensive range of settings and several approaches of approximation. Moreover, we also applied FD on pen pressure, azimuth and tilt signals. All features were extracted for different values of α (order of FD). In the frame of this work, a range from 0.1 to 1.0 with a step of 0.1 was used. Finally, statistical properties of the features were described by: mean, median, standard deviation (std), and maximum (max). Altogether, 1188 handwriting features were extracted for each dataset.

We were considering 3 following feature sets: Czech, Spanish and multilingual (mixed – 73 PD, 48 HC). In order to identify features that discriminate HC and PD we employed the Mann-Whitney U test. The significance level was set to α = 0.001.

Next, to evaluate the discrimination power of handwriting features, we performed multivariate classification analysis based on random forests (RF) [15]. In order to reduce the number of handwriting features entering into the classification analysis, we designed fast and efficient 2-stage feature selection. Firstly, each feature set was reduced by minimum redundancy maximum relevance [16] (mRMR) feature selection algorithm to 50 best features. Secondly, to obtain the most appropriate combination of the features, the sequential floating forward selection [17] (SFFS) algorithm was employed. To achieve the most accurate results for each dataset, we used different types of model validation techniques. In the case of Czech and Spanish feature sets we used leave-one-out cross-validation (due to small sample size). For the multilingual feature set, 10-fold cross-validation with 20 repetition was used. Classification performance was evaluated by the Matthew's correlation coefficient [18] (MCC), classification accuracy (ACC), sensitivity (SEN) and specificity (SPE).

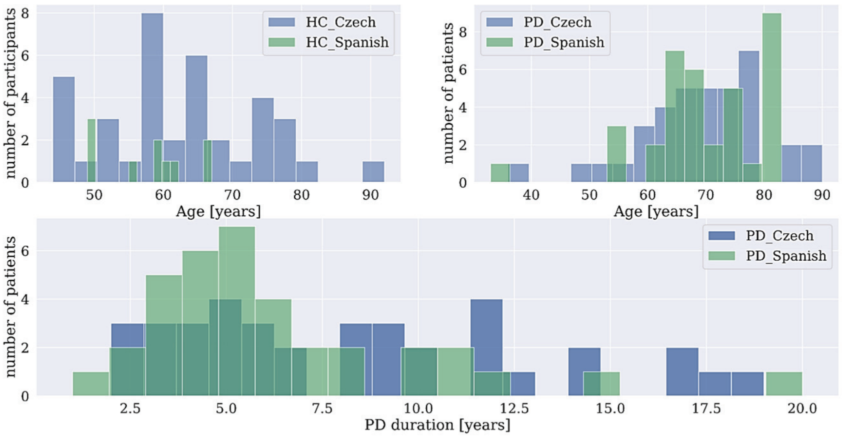


Fig. 2. Descriptive statistics of examined datasets. In the top left part of the figure, the HC age distribution is visualized. The PD age distribution is in the top right part and in the bottom part, the distribution of PD duration is shown.

3. Results

The results of the Mann-Whitney U test can be found in the upper part of Table 2. Three most discriminative features which passed through the test are reported for each feature set. Features are sorted by significance level p, while all reported features obtained $p < 0.0001$. The most discriminative feature from the Spanish feature set is velocity (on-surface). In the case of Czech and multilingual feature set, it is its vertical variant, which is probably linked with the vertical micrographia [19]. As can be noticed, the results of the Czech and multilingual feature sets are quite similar, in comparison with Spanish one. This is probably caused by the size of the Spanish HC cohort (10 participants).

Table 2. Results of Mann-Whitney U test and classification analysis.

Mann-Whitney U test			
Feat. Set	**Feature Name**	**α**	***p***
Spanish	velocity^{s} (median)	0.1	0.000069
	velocity^{s}(median)	0.2	0.000069
	velocity^{a} (mean)	0.1	0.000077
Czech	vertical velocity^{s} (mean)	0.2	0.000012
	vertical velocity^{s} (mean)	0.2	0.000014
	vertical velocity^{s} (median)	0.4	0.000014
Multi-lingual	vertical velocity^{s} (mean)	0.1	0.000001
	vertical velocity^{s} (median)	0.4	0.000001
	vertical velocity^{s} (median)	0.3	0.000001

Multivariate classification analysis					
Feat. Set	**N**	**MCC**	**ACC [%]**	**SEN [%]**	**SPE [%]**
Spanish	2	0.87	95.65	97.22	90.00
Czech	9	0.71	85.33	89.19	81.58
Multi-lingual	8	0.63	82.29	85.99	77.22

[1] Feat. Set – feature set; α – order of FD; p – significance level; ^{s} – on-surface movement; ^{a} – in-air movement; N – number of features

Next, the results of the multivariate classification analysis can be found in the bottom part of Table 2. The highest classification performance was obtained in the Spanish feature set (ACC = 95.65 %), nevertheless, due to the imbalanced cohort (36 PD patients and 10 HC), these results may be misleading. Number of HC in the Spanish database is 3.6 times lower than number of PD patients. By mixing the Spanish and Czech (well balanced) databases we have reduced the imbalance of the Spanish one (PD ≈ 1.5 × HC). Also, the distribution of PD duration for the Czech cohort is more uniform (see Fig. 2). In the Spanish cohort, patients with shorter disease duration (less than 6 years) outweigh, however the distribution of PD patient's age is quite similar for both cohorts. Thus, by combining the datasets, we also improved non-uniformity of the final cohort. Although the accuracy of the multilingual feature set is the lowest one (82.29 %), credibility of the results may be considered as higher in comparison to the Spanish feature set.

4. Conclusions

This study deals with the advanced analysis of PD dysgraphia in a multilingual cohort. First of all, since the most significant features identified in the Mann-Whitney U test and features selected by the SFFS have a non-integer value of the FD order, we suppose that the FD based parameters play significant role in PD dysgraphia quantification. Next, we achieved more than 80 % classification accuracy in all scenarios, which suggests the high impact of online handwriting processing in cross-cultural clinical studies focused on PD dysgraphia diagnosis.

This study has several limitations and suggestions for further research. Firstly, the Spanish dataset is not balanced (PD/HC, PD duration). In addition, the overall sample size is not big. On the other hand, to the best of our knowledge, it is the first and therefore the biggest multilingual online handwriting PD dataset that has ever been analyzed. Finally, the FD-based features may be more explored and extended (e.g. by Caputo approximation approach). To sum it up, this study has a pilot character and further research should be done to be able to generalize the results.

Acknowledgements

This work was supported by the grant of the Czech Ministry of Health 16-30805A (Effects of non-invasive brain stimulation on hypokinetic dysarthria, micrographia, and brain plasticity in patients with Parkinson’s disease), grant of the Czech Science Foundation 18-16835S (Research of advanced developmental dysgraphia diagnosis and rating methods based on quantitative analysis of online handwriting and drawing) and the following projects: LO1401, FEDER and MEC, TEC2016-77791-C4-2-R, from the Ministry of Economic Affairs and Competitiveness of Spain. For the research, infrastructure of the SIX Center was used.

Reference

[1]. D. Berg, R. B. Postuma, C. H. Adler, B. R. Bloem, P. Chan, B. Dubois, T. Gasser, C. G. Goetz, G. Halliday, L. Joseph, et al., MDS research criteria for prodromal Parkinson's disease, *Movement Disorders,* Vol. 30, 2015, pp. 1600-1611.

[2]. A. Elbaz, L. Carcaillon, S. Kab, F. Moisan, Epidemiology of Parkinson's disease, *Revue Neurologique*, Vol. 172, 2016, pp. 14-26.

[3]. C. De Stefano, F. Fontanella, D. Impedovo, G. Pirlo, A. S. Freca, Handwriting analysis to support neurodegenerative diseases diagnosis: A review, *Pattern Recognition Letters*, 2018.

[4]. J. E. McLennan, K. Nakano, H. R. Tyler, R. S. Schwab, Micrographia in Parkinson's disease, *Journal of the Neurological Sciences*, Vol. 15, 1972, pp. 141-152.

[5]. Letanneux, J. Danna, J.-L. Velay, F. Viallet, S. Pinto, From micrographia to Parkinson's disease dysgraphia, *Movement Disorders*, Vol. 29, 2014, pp. 1467-1475.

[6]. J. Mucha, V. Zvoncak, Z. Galaz, M. Faundez-Zanuy, J. Mekyska, T. Kiska, Z. Smekal, L. Brabenec, I. Rektorova, K. Lopez-de-Ipina, Fractional derivatives of online handwriting: A new approach of Parkinsonic dysgraphia analysis, in *Proceedings of the 41st International Conference on Telecommunications and Signal Processing (TSP’18)*, 2018.

[7]. E. Nackaerts, S. Broeder, M. P. Pereira, S. P. Swinnen, W. Vandenberghe, A. Nieuwboer, E. Heremans, Handwriting training in Parkinson's disease: A trade-off between size, speed and fluency, *PLoS ONE*, Vol. 12, 2017, e0190223.

[8]. P. Drotar, J. Mekyska, I. Rektorova, L. Masarova, Z. Smekal, M. Faundez-Zanuy, Evaluation of handwriting kinematics and pressure for differential diagnosis of Parkinson's disease, *Artificial Intelligence in Medicine*, Vol. 67, 2016, pp. 39-46.

[9]. Kotsavasiloglou, N. Kostikis, D. Hristu-Varsakelis and M. Arnaoutoglou, Machine learning-based classification of simple drawing movements in Parkinson's disease, *Biomedical Signal Processing and Control*, Vol. 31, 2017, pp. 174-180.

[10]. J. Mucha, J. Mekyska, M. Faundéz-Zanuy, K. Lopez-de-Ipina, V. Zvončák, Z. Galáž, T. Kiska, Z. Smékal, L. Brabenec. I. Rektorová, Advanced Parkinson's disease dysgraphia analysis based on fractional derivatives of online handwriting, in *Proceedings of the 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT’18)*, 2018.

[11]. J. Mucha, J. Mekyska, Z. Galaz, M. Faundez-Zanuy, K. Lopez-de-Ipina, V. Zvoncak, T. Kiska, Z. Smekal, L. Brabenec, I. Rektorova, Identification and monitoring of Parkinson’s disease dysgraphia based on fractional-order derivatives of online handwriting, *Applied Sciences*, Vol. 8, 2018, 2566.

[12]. E. Sesa-Nogueras, M. Faundez-Zanuy, J. Mekyska, An information analysis of in-air and on-surface trajectories in online handwriting, *Cognitive Computation*, Vol. 4, Issue 2, 2012, pp. 195-205.

[13]. I. Podlubny, Fractional Differential Equations an Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications, *Academic Press*, San Diego, 1999.

[14]. A. A. Kilbas, H. M. Srivastava, J. J. Trujillo, Theory and Applications of Fractional Differential Equations, Vol. 204, *Elsevier*, 2006.

[15]. L. Breiman, Random forests, *Mach. Learn*., Vol. 45, 2001, pp. 5-32.

[16]. H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Trans. Pattern Anal.*, Vol. 27, 2005, pp. 1226-1238.

[17]. J. Pohjalainen, O. Rasanen, S. Kadioglu, Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits, *Comput. Speech Lang*., Vol. 29, Issue 1, January 2015, pp. 145-171.

[18]. B. W. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, *Biochim. Biophys. Acta (BBA)*, Vol. 405, 1975, pp. 442-451.

[19]. M. Thomas, A. Lenka, P. Kumar Pal, Handwriting analysis in Parkinson's disease: Current status and future directions, *Movement Disorders Clinical Practice*, Vol. 4, pp. 806-818.

(51)

An Upper Bound for the Maximal α -quasi-clique-community Problem

Patricia Conde-Cespedes [1]
[1] LISITE-ISEP-10 rue de Vanves, 92130 Issy-les-Moulineaux, France
Tel.: +33-01.49.54.52.41
E-mail: patricia.conde-cespedes@isep.fr

Summary: Community detection in large complex networks is important to understand their structure and to extract features useful for visualization or prediction of various phenomena like information diffusion. In some situations, one can be interested in the community of a specific node of interest in the network, for instance, for applications dealing with huge networks, when iterating through all nodes would be impractical or when the network is not entirely known. We call this task *local community detection* of a specific node. A community consists in a set of densely connected nodes. In graph theory, for a given $\alpha\,(0<\alpha<1)$, an $\alpha-$ quasi-clique is a group of nodes where each member is connected to more than a proportion α of the other nodes. Then, an $\alpha-$ quasi-clique is a community of density greater than α . The size of an $\alpha-$ quasi-clique is limited by the degree of its nodes. In complex networks whose degree distribution follows a power law, usually $\alpha-$ quasi-cliques are small sets of nodes for high values of α . This led to the problem of finding *the maximal $\alpha-$ quasi-clique community (or quasi-clique) of a given node*. This problem is *NP-hard*. Some heuristics and upper bounds on the optimal solution were recently proposed. In this paper, we propose an improved version of the upper bound on the optimal solution in order to evaluate the heuristics. The proposed upper bound will be evaluated experimentally on real and computer generated networks.

Keywords: Community detection algorithms, Maximal $\alpha-$ quasi-clique; Local community detection, Upper bound on the optimal solution.

1. Introduction

In this paper, we will propose an upper bound for the so-called *the maximal α -quasi-clique community (or quasi-clique) of a given node*. In the next Section we will introduce in the notations as well as the problem formulation. Next, we discuss about the upper bound.

2. Definitions and Notations and Problem Formulation

A graph $G=(V,E)$, is defined by V the set of vertices or nodes, and E the set of edges or links, formed by pairs of vertices. We consider only undirected graphs, where the edges are not oriented. The neighborhood $\Gamma(u)$ of a node u is the set of nodes v such that $(u,v)\in E$. The degree of a node u, denoted $d(u)$, is the number of its neighbors, i.e. $d(u)=|\Gamma(u)|$. An $\alpha-$ quasi-clique is defined as follows[1]:

Denition 1. $\alpha-$ quasi-clique or $\alpha-$ consensus community.

Given an undirected graph $G(V,E)$, and a parameter α with $0<\alpha<1$, an $\alpha-$quasi-clique or $\alpha-$consensus community is the subgraph induced by a subset of the node set $C\subseteq V$ if the following condition holds:

$$|\Gamma(n)\cap C|>\alpha(|C|-1),\forall n\in C \tag{1}$$

Equation (1) implies that each node in the quasi-clique C must be connected to more than a proportion α of the other nodes. In the following, we will call Equation (1) the **rule of an $\alpha-$ quasi-clique.** This rule constitutes a lower bound on the minimal internal connections of each node. Notice that for $\alpha=1$ an $\alpha-$quasi-clique is a complete clique. Notice also that an $\alpha-$ quasi-clique has a density greater than α .

Then, the resulting communities are robust, contain strongly connected nodes and have an edge-density greater than α .

Since the size of an $\alpha-$quasi-clique is limited by the degree of its nodes, for complex networks whose degree distribution follows a power law, mining for the $\alpha-$ quasi-clique community of specific nodes with low degree can lead to trivial solutions, such as pairs of nodes or triangles. Indeed, those are $\alpha-$ quasi-cliques or even complete cliques as they achieve maximal density. Such trivial communities are not interesting for applications. Therefore, the purpose is to find quasi-cliques of maximal size.

[1] In the literature, one can find other definitions of $\alpha-$ quasi-clique (some of them are quite similar), see for instance [7], or [2]. Other definitions are mostly based on the density, see [1, 8].

In this paper, we focus on the problem of finding an $\alpha-$quasi-clique of maximal cardinality containing a given node of interest. That is, the local community of type $\alpha-$quasi-clique for a given α. This problem can be formulated as follows:

Problem 1. The maximal $\alpha-$ quasi-clique community (or quasi-clique) of a given node problem.

Given a node n_0 of a graph $G(V,E)$ and a parameter α ($0<\alpha<1$), the purpose is to find the biggest $\alpha-$quasi-clique or $\alpha-$consensus community $C(n_0)$ containing n_0, mathematically:

$$\begin{array}{ll} \underset{C}{\text{maximize}} & |C| \\ \text{subject to} & n_0 \in C \\ \text{and} & |\Gamma(n_i) \cap C| > \alpha(|C|-1), \forall n_i \in C \end{array} \tag{2}$$

The Problem 1 was first tackled in [3] where the authors proposed an algorithm called *RANK- NUM-NEIGHS (RNN)*. Nearly similar (but not exactly the same) problems were tackled in [7] and [2] where the authors proposed the *QUICK* and *RLS-DLS* methods respectively. We can also mention the *Louvain method* adapted to Zahn-Condorcet and Owsiński-Zadrożny criteria, which was proved to output $\alpha-$quasi-cliques (see [4] and [5] for proof).

The Problem 1 being *NP-hard*, all of these methods are heuristics. In [3] the authors proposed an upper bound on the optimal solution, in order to evaluate how far was the obtained results to the optimal solutions. This bound is briefly described in the next section.

3. Upper Bound on the Optimal Solution

Consider a node n_0 with degree $d(n_0)$, then the size of its maximal $\alpha-$quasi-clique community can not exceed the following quantity, denoted $B_0(n)$:

$$B_0(n_0) = \left\lceil \left(\frac{d(n_0)}{\alpha} \right) \right\rceil \tag{3}$$

Based on this principle, in [3], the authors proposed an upper bound on the oprimal solution of Problem 1. The proposed bound showed to work well in most situations. However, one drawback of this bound is that it does not perform well when nodes have a high degree and have at least one neighbor with high degree. In the present paper, we will present an improved version of the existing upper bound.

4. Experimental Results

The Fig. 1 shows an example of the results for two real networks *The Zachary Karate Club* (see [9]) and *Polbooks* [6]. The figure presents the histogram of the difference between the upper bound and the experiemental results found in [3].

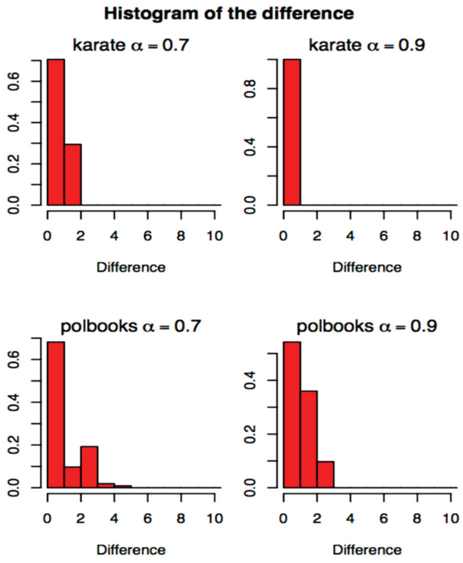


Fig. 1. Histogram of Difference between the upper bound and the experimental value for two real networks.

References

[1]. J. Abello, M. G. C. Resende, S. Sudarsky, Massive quasi-clique detection, in *Proceedings of the 5th Latin American Symposium on Theoretical Informatics (LATIN'02)*, London, UK, 2002, pp. 598-612.

[2]. M. Brunato, H. H. Hoos, R. Battiti, On effectively finding maximal quasi-cliques in graphs, *LION*, *Lecture Notes in Computer Science*, Vol. 5313, 2007, pp. 41-55.

[3]. P. Conde-Céspedes, B. Ngonmang, E. Viennet, An efficient method for mining the maximal alpha-quasi-clique- community of a given node in complex networks, *Social Network Analysis and Mining*, Vol. 8, December 2018, 20.

[4]. P. Conde-Céspedes, J. F. Marcotorchino, E. Viennet, Comparison of linear modularization criteria using the relational formalism, an approach to easily identify resolution limit, Revue des Nouvelles Technologies de l'Information, *Extraction et Gestion des Connaissances*, 2015.

[5]. P. Conde-Céspedes, J. F. Marcotorchino, E. Viennet, Comparison of linear modularization criteria using the relational formalism, an approach to easily identify resolution limit (extensions), *Advances in Knowledge Discovery and Management (AKDM-6)*, 2017, pp. 101-120.

[6]. V. Krebs. Books about US politics, *Unpublished*, 2004.

[7]. G. Liu, L. Wong, Effective pruning techniques for mining quasi-cliques, *Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science*, Vol. 5212, 2008, pp. 33-49.

[8]. J. Pattillo, A. Veremyev, S. Butenko, V. Boginski, On the maximum quasi-clique problem, *Discrete Applied Mathematics*, Vol. 161, 2013, pp. 244-257.

[9]. W. W. Zachary, An information flow model for conflict and fission in small groups, *Journal of Anthropological Research*, Vol. 33, Issue 4, 1977, pp. 452-473.

(53)

Macrocura: A Medical Diagnosis Assistant System for Traditional Chinese Medicine Primary Care Practices

Shasha Wu, [1], **Qiang Su,** [2] **and Wenli Fan** [2]
[1] Spring Arbor University, Spring Arbor, Michigan, United States
[2] Macrocura, Ltd., Hangzhou, Zhejiang, China
Tel.: + 1-517-750-6698
E-mail: shasha.wu@arbor.edu

Summary: Traditional Chinese medicine (TCM) has been practiced in China and surrounding countries like Japan and Korea for over two thousand years. It is still widely accepted as an important health care practice in those countries. In spite of the widespread use of TCM in Asia and the West, rigorous scientific evidence of its effectiveness is limited. TCM can be difficult for researchers to study because its treatments are often complex and are based on ideas very different from those of modern western medicine [1]. The Macrocura TCM diagnosis assistant system for primary care is proposed in this article to address these issues. A set of standard representations is predefined in the system to normalize the description of the TCM clinical symptoms, diagnosis and treatment. The image and standard representations are suitable for data processing and machine learning. With the deployment of the system in primary care clinics, it produces large amounts of high quality labeled clinical images and standardized symptom data, which are key materials in areas of research, such as image processing, machine learning, and medical analysis.

Keywords: Medical diagnosis assistant, Clinic data processing, Health information system, Macrocura.

1. Introduction

Traditional Chinese Medicine (TCM) has been widely practiced in China and surrounding countries like Japan and Korea for over two thousand years. It still plays an important role in the modern Chinese health care system. In the year of 2017, more than 588 million patients received medical treatment in 54,243 clinics by 448,716 certified Chinese medicine clinicians [2]. Many of these clinicians provide primary care services to the patients.

Primary Care Practitioners (PCPs) manage a wide range of complex and diverse conditions through one or more relatively brief encounters [3]. The average Chinese medicine primary care encounter takes less than 10 minutes. Within that time, the clinician was trained to follow a set of sophisticated rules and principles to diagnose the patients based on the patient's description and clinician's observation. The result of the diagnosis will lead to one or several established TCM prescriptions as the treatment. TCM uses different combinations of over 1,500 medical herbs to form about 300 established prescriptions that were experimentally proven in history to be effective for specific sicknesses and presenting symptoms.

2. Challenges and Opportunities

Although huge amount of Chinese medicine primary care are practiced every day, most of the practices are still recorded by paper and pen with unstructured plain text descriptions. Few of the practices resulted in feedback from the patients for the clinicians to review the correctness of the diagnosis and the effectiveness of the treatment. The lack of centralized high quality clinical data and method of collecting timely feedback from the patient prevents the scientific study of Chinese medicine.

Beyond that, most of the rules and principles of TCM are based on Chinese philosophical concepts, which are abstract and difficult to be measured quantitatively. Therefore, those rules and principles are difficult to simulate and challenging to verify. However, the patients' symptoms and corresponding treatments are both solid and measurable. This gives us the opportunity to study the mechanisms and effectiveness of treatments with the help of modern health information technology.

Health information technology (HIT) systems have the potential to reduce delayed, missed, or incorrect diagnoses. Unfortunately, progress in diagnostic HIT has been slow and incremental with few significant "game-changing" approaches having emerged in the last decade [4]. With recent advances in artificial intelligence, cloud computing, mobile devices, and wearable devices, we believe it's the time to change the situation. For example, mobile applications can be used to collect patients' feedback for researchers to evaluate the effectiveness of TCM diagnoses and treatments. Image processing and convolutional neural networks (CNN) can be used to detect abnormalities through patients' face or tongue images for disease screening. Decision tree algorithms can help clinicians decide whether to rule in or rule out specific TCM treatments. Deep neuronal networks can be trained to classify a set of presenting symptoms to a set of established TCM prescriptions.

3. Implementation

The four traditional Chinese medicine diagnosis steps are observing, listening, questioning, and checking patient's pulse patterns. We proposed and implemented the Macrocura TCM diagnosis assistant system [5] hoping not just to help the TCM clinicians in their daily practices, but also to provide high quality modern clinic data to facilitate information technology researches in the areas of signal processing, machine learning, and medical analysis.

The Macrocura system provides a procedure of five steps to process daily primary care information. In the first step shown in Fig. 1, the system provides a graphic user interface for the clinician to take normalized pictures of the patient's face and body, record the voice of the patient as audio files, and help the clinician label observed symptoms and pulse patterns. In step two, the system analyzes the patient's chief presenting symptoms and classifies these standard symptoms based on the TCM rules and principles. If it successfully finds the pattern, it will list the recommended TCM treatments for the clinician. In step three, the clinician will adjust the treatment if necessary and generate a final prescription. System's recommendation and clinician's final decision will both be recorded for future medical study and machine learning experiments. In step four, a patient can revisit the clinic or run the Macrocura mobile app (Fig. 2) on his/her smart phone to report his/her progress and feedback to the clinic. In step five, the clinician will review the progress and feedback report in the system to confirm or adjust the original diagnosis and treatment. All information will be saved in a cloud storage for future analysis and research.

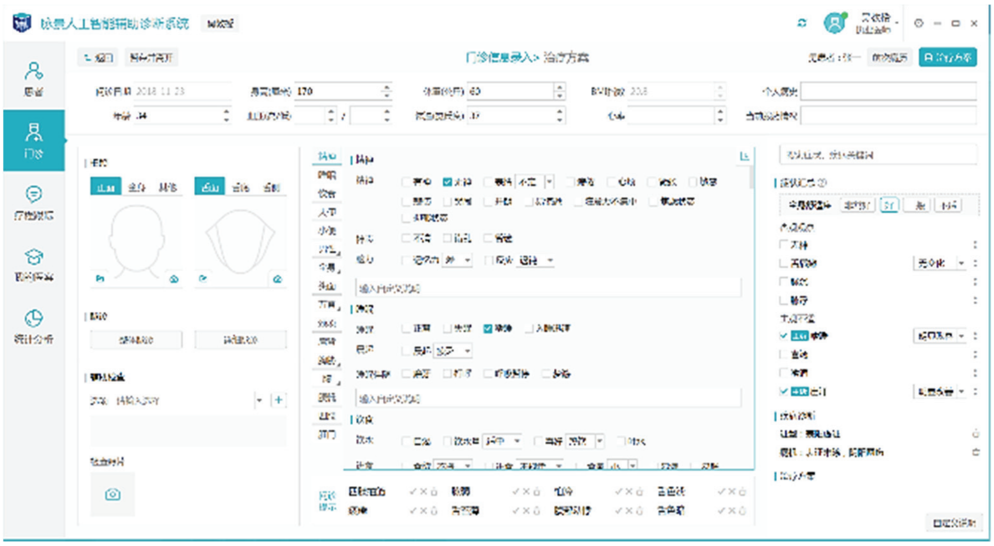
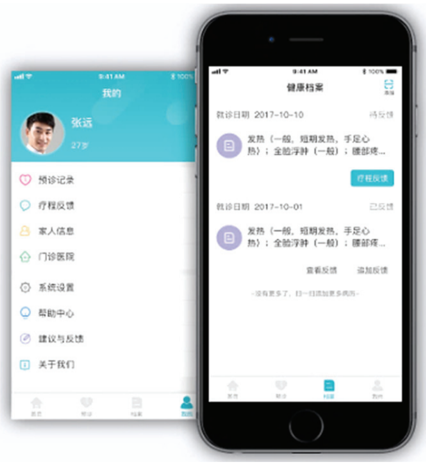


Fig. 1. Screen shot of one user interface of Macrocura TCM diagnosis assistant system for primary care.



就诊日期 2017-10-10 待反馈

发热（一般，短期发热，手足心热）；全脸浮肿（一般）；腰部疼...

疗程反馈

就诊日期 2017-10-01 已反馈

发热（一般，短期发热，手足心热）；全脸浮肿（一般）；腰部疼...

查看反馈 追加反馈

Fig. 2. Screen shot of Macrocura patient feedback mobile application.

The Macrocura system predefined more than a thousand commonly seen primary care symptoms for clinicians to choose from. The design of standardized symptom description facilitates machine processing and supports multilingual environment. The ultimate goal for this design is to make the predefined descriptions specific enough to manage the standard situations, broad enough to encompass the common exceptions, and flexible enough to allow separate decisions for the rare [6].

4. Outputs and Future Work

The Macrocura TCM diagnosis assistant system was first adopted by two sponsoring Chinese medicine clinics in April 2018. It was officially purchased and deployed by three hospitals and health care providers for their daily Chinese Medicine primary care practices after July 2018. By the mid of November 2018, the system recorded more than 30,000 clinic visits for more than 20,000 patients. The automatically generated TCM prescription recommendations reached an 80 % correctness comparing to human TCM expert's prescription in an internal test.

The Macrocura system shows the potential to help clinicians in their daily practices. The structured real life clinic data provides a lot of potentials for information technology research and medical analysis. Although the system was designed for TCM clinics, the method it demonstrated and the data it collected are meaningful to all medical analysis and researches.

References

[1]. Traditional Chinese Medicine: In Depth, *U.S. Department of Health and Human Service*, https://nccih.nih.gov/health/whatiscam/chinesemed.htm

[2]. China Statistical Yearbook of Chinese Medicine, *National Administration of Traditional Chinese Medicine*, 2017, http://www.satcm.gov.cn/2017tjzb/start.htm

[3]. M. M. Hardeep Singh, M. M. Traber, D. Giardina, P. Ashley, N. D. Meyer, M. M. D. Samuel, N. Forjuoh, M. Michael, D. Reis, M. M. Eric J. Thomas, Types and origins of diagnostic errors in primary care settings, *Journal of the American Medical Association Intern Medicine*, 2013, pp. 418-425.

[4]. R. El-Kareh, O. Hasan, G. D. Schiff, Use of health information technology to reduce diagnostic errors, *BMJ Quality & Safety*, Vol. 22, 2013, pp. 40-51.

[5]. Macrocura Ltd., https://www.macrocura.com/index.html

[6]. A. R. Feinstein, An analysis of diagnostic reasoning, III. The construction of clinical algorithms, *Yale Journal of Biology and Medicine*, Vol. 1, 1974, pp. 5-32.

(54)

Using Neural Networks for the Classification and Clustering of Multicomponent Alloys

N. I. Yussupova [1], O. S. Nurgayanova [1], T. I. Fabarisov [1], A. Morozov [2]
[1] Ufa State Aviation Technical University, Ufa, Russia
[2] Technische Universität Dresden, Germany
Faculty of Informatics and Robotics, Ufa, Russia
E-mail: onurgayanova@yandex.ru

Summary: The issues of classification and clustering of multicomponent Ni-based superalloys with monocrystallic structure used in the aviation industry for the manufacturing of gas turbine engine blades are considered. The goal of alloy clustering is to generate a "composition-property" dependencies and to predict the characteristics of an alloy based on its composition. Problem statement of clustering and classification problems is provided in paper. As a solution for described problems artificial neural networks are proposed in the article. For the clustering problem it is proposed to use Kohonen network and for the classification problem it is proposed to use "k-nearest neighbors" method using two proximity metrics: Euclidean distance and Manhattan distance. Comprehensive and details of the algorithms and an example of program implementation are presented in the paper. And, in conclusion, a discussion of the results is provided.

Keywords: Classification/clustering and forecasting tasks, Artificial neural networks, Kohonen network, Multilayer perceptron, Displacement neuron, Ni-based superalloys.

1. Introduction

The development of new materials with higher properties is a relevant challenge of modern industry. Especially in such industrial domains as aircraft and space engine manufacturing as well as in deep drilling and in mineral resource and fossil fuel extraction and processing.

To reach the highest possible energy conversion efficiency ratio of engines and drilling rigs it is necessary to increase the operating temperature of the system lower the heat transfer. The most obvious solution for this problem is to use heat-resistant superalloys which can withstand temperatures of up to 1500-2000 °C [1]. For the aerospace investment casting and die casting of the drilling rig parts requires using an alloy with elevated-temperature capability. Turbine outlet temperature is restricted by the heat resistant properties of the source material. The usage of cooled aircraft gas turbine blade made from the high-performance alloy allowed to increase the turbine outlet temperature up to 1400-1500 °C for aircrafts, especially for interceptor aircraft whose engine longevity are low, and up to 1050-1090 °C for stationary gas turbine plants that intended for a long-term operation [1, 2]. Thus, it is obvious that the field of application of a material brings its own adjustments.

The development of new materials is often associated with the experience and intuition of the researcher and requires considerable material and time expenditures. Among those are multiple smelting of prototypes, use of expensive equipment, subsequent activities related to their testing for strength characteristics as well as conducting various kinds of analyzes (X-ray diffraction, physico-chemical, etc.) [2]. When choosing materials for the manufacturing of a particular part, developers rely not only on the technical requirements for it, but also follow the principle of minimum costs. On the basis of information about chemical composition and properties of already known and used high-temperature alloys, they can be classified according to certain characteristics, which will further narrow the search for new compositions and increase the efficiency of the selection process [3].

2. State of the Art

Studies by well-known scientists [1- 3] showed that the high-temperature strength of alloys depends on the strength of interatomic bonds as well as on structure and state of grain boundaries. The decisive role in achieving high heat resistance belongs to the nature of the interaction of coexisting phases. However, despite all revealed principles of doping high-temperature alloys, it is not yet possible to obtain specific quantitative recommendations for choosing the composition of new alloys, as well as attributing them to a certain class of heat resistance [2, 3]. Earlier [1], the problem associated with predicting the properties of alloys was solved using a multilayer perceptron with the number of input neurons equal to the number of chemical elements, trained by the method of back propagation of error.

By summarizing and systemizing current state of the high-temperature alloys synthesis problem and problem of automation of its classification, it is clear that this task is related to the Data Mining class of problems, particularly, to the cluster analysis and statistical classification problems [4, 5]. Generally, the solving of these tasks can be divided to following subtasks:

1) Generation of training set that would consist of: alloy name, its chemical formula and percent composition of main 16 elements, 100-hour long-term strength at 1000 °C. This training set will be used as input data for system learning.
2) Generation of training dataset for classifier which would consist of: name of alloy, percent composition of its chemical formula.
3) Formalization the level of impact of different alloying elements to the heat resistance of alloy.
4) Classification method selection. It was decided to test different approaches for solving classification of high-temperature alloys problem. Next step will be to choose the classification method. Two data mining approaches will be tested: k-nearest neighbors search and Kohonen network.
5) Classification/clustering parameters selection: choice the number of neighbors, distance metric between classes/clusters for both methods.
6) Classification/clustering of the alloys with chosen method: associating alloys from classification set of: k-nearest neighbors search method with alloys from the clustering set for the Kohonen network.
7) Analysis and formulating of recommendation for use of achieved classification/clustering results.

Let's formulate clearer definition of terms used above:

- The **classification** problem is the problem of identifying the class of object by his properties. In this problem set of classes is known [3].
- In **clustering** problem grouping of a set of objects to a clusters that prior are unknown. Dividing and clustering of a set of objects is performed concurrently.

Classification problem is considered as an instance of supervised learning. There are two main types of classification [4, 5]:

- Artificial classification - that is, performed based on external properties and is used to put a set of objects in required order;
- Natural classification - that is, performed based on essential properties that characterize internal commonality of objects and events. It is a result and an important asset of scientific research, because it suggests and consolidate results of the pattern research, including hidden patterns, of classifying objects.

In this article considered task is related to the multidimensional natural classification problem, i.e., carried out on several attributes. The goal of classification is to build a model using predicting properties as input parameters and depended attribute. Classification is a process of dividing a set of objects by one or several criteria.

Clustering, on the other hand, is related to the unsupervised learning. The goal of clustering is to discover already existing structures. Clustering is a descriptive procedure that provides an opportunity to carry out the exploratory data analysis and research the "structure of data", viz. to discover the main alloying elements and type of hardening phases [1, 5].

3. Problem Statement

Problem 1. Systematize and summarize the results of research on the issue of alloy synthesis. This problem is related to classification/clustering problems [4, 5].

Classification Problem:

Given: A set of alloys (unclassified elements *N*), each of which has its own chemical composition and properties (a set of attributes *P*), a set of classes (groups into which they are close in a set of attributes *C*). A training sample M is set for which the map C^* is specified: $M \rightarrow C$ taking into account the attributes of *P*.

Find: the distribution of elements of the set *N* among the elements of the set *C* taking into account the criteria (of attributes *P*).

Clustering tasks:

Given: A set of alloys – unclassified elements of *N*, which have a set of attributes *P*.

Find: the number of clusters and split the set *N* into non-intersecting cluster subsets so that the elements that fall into one cluster are as close as possible on the set of attributes *P*.

Problem 2. Predict the properties of new alloys using a trained neural network.

Problem of predicting the properties of alloys for a given chemical composition:

Given: The set of vectors *N* of the type $x_i(x_{i1}, x_{i2}, \ldots, x_{in})$ is the mass fractions of chemical elements, and the corresponding values of heat resistance y_i.

Find: F(X) = Y, such that for $\forall x^*$ it is possible to define the property $y^* = f(x^*)$.

4. Proposed Solutions

The article [4] presents the results of computational experiments on a set of *Ni*-based superalloys with a single-crystal structure, for which the chemical composition and 100-hour long-term strength at 1000 °C were known.

4.1. Application of the "k-nearest Neighbors" Method for the Classification of Alloys

The specifics of the method is such that each alloy classified in the course of the algorithm operation expands the training sample, and each subsequent unidentified alloy is recognized based on all previous ones as well as on new ones. The algorithm is rather resistant to anomalous outliers, since the probability of getting such a record among the k-nearest neighbors is small, but even in this case, the effect of outliers on the weighted vote will be insignificant, which is explained by the very nature of the algorithm [4].

Application of the "k-nearest neighbors" method for the high-temperature alloys classification.

Given:

1) Representative training set N that consists of n alloys with known heat resistance values;
2) Each element of training set is an alloy that consist an information percent chemical composition of main 16 elements;
3) For each of these 16 elements the coefficient of the influence on summarized heat resistance value is defined;
4) Range of heat resistance values for three classes: maximum, medium and minimum heat resistance;
5) Sample M of m alloys with unknown heat resistance value;
6) Each element in a sample of unclassified alloys is an alloy that consist an information percent chemical composition of main 16 elements.

Required to determine heat resistance class for each alloy in sample *M* based on the analysis of sample *N* using “k-nearest neighbors” method [5].

Description of algorithm:

Step 1. Analyze and divide training set into heat resistance classes based on given ranges.

Step 2. Set the number of *k*-nearest neighbors.

Step 3. Randomly choose one element from the set *M* – vector A_s, declare it as a center of class and find the distances to each element in training set *N* using one of the following proposed proximity measures:

1) Euclidean proximity measure considering element’s level of influence, i.e. weighted Euclidean distance:

$$d_i = \sqrt{\sum_{j=1}^{C} Z_j * (A_{sj} - B_{ij})^2}, \qquad (1)$$

where d_i is the distance between two objects: unclassified alloys from the set A_s and an alloy from the training set B_i, $i = 1 \dots N$; C is the number of elements which characterizing the object – chemical elements that composes an alloy; Z is a list of the level of influence coefficients of each chemical element, $j = 1 \dots C$.

2) Manhattan proximity measure considering element’s level of influence, i.e. weighted Manhattan distance:

$$d_i = \sum_{j=1}^{C} Z_j * |A_{sj} - B_{ij}|, \qquad (2)$$

where d_i is the distance between two objects: unclassified alloys from the set A_s and an alloy from the training set B_i, $i = 1 \dots N$; C is the number of elements which characterizing the object – chemical elements that composes an alloy; Z is the list of the level of influence coefficients of each chemical element, $j = 1 \dots C$.

Step 4. Choose from the list of distances *D k* shortest one and determine by them *k*-nearest neighbors.

Step 5. Carry out the weighted vote on *k*-nearest neighbors by criteria being the heat resistance class. In this particular case to avoid uncertainty when several classes receive same number of votes, weighted vote will consider not only abovementioned criteria but also distance to this neighbor. The smaller the distance to the «neighbor», the more significant is his voice:

$$votes(class) = \sum_{i=1}^{N} \frac{1}{d_i^2}, \qquad (3)$$

where d_i^2 is the squared distance from unclassified alloy A_s to the alloy from the training set B_i, that belongs to the class $class$.

Step 6. Class that will receive the most votes is assigned to alloy A_s, which is then being added to the set of classified alloys *N* and being deleted from the sample *M*.

Repeat Steps 2-6 until sample *M* is not empty.

Using the method of "k-nearest neighbors", Ni alloys were divided into three classes – with high, medium and low heat resistance. With the number of neighbors equal or close to the number of chemical elements and the measure of proximity being Euclidean distance, the classification error was 7 %. With the same number of neighbors and a measure of proximity being Manhattan distance, the classification error was 10 %. As a rule, these were alloys having a boundary value of heat resistance.

The method can be applied in the situations when a new alloy is synthesized, its composition is known, but the heat resistance is unknown, in which case it can be classified into one of three classes.

4.2. Application of Kohonen Networks for Alloy Clustering

The principle of the Kohonen network is to introduce information about neuron location into the neuron’s learning rule, which, in turn, simplifies the multidimensional network structure and displays the multidimensional vector, alloy composition and heat resistance in a two-dimensional map [6-8].

Application of Kohonen networks for the high-temperature alloys clustering.

Given:

1) Representative training set N that consists of n alloys with known heat resistance values;
2) Each element of training set is an alloy that consist an information percent chemical composition of main 16 elements;
3) For each of these 16 elements the coefficient of the influence on summarized heat resistance value is defined;
4) Sample M of m alloys with unknown heat resistance value;
5) Each element in a sample of unclassified alloys is an alloy that consist an information percent chemical composition of main 16 elements.

Required to divide all alloys from sample *M* into clusters based on the proximity measure of chemical composition using Kohonen neural network trained on training sample *N*.

Description of algorithm:

Step 1. Initialization of neural network. Neural network is a set of neurons - vectors of weights. Vector of weights of neuron is equal to the number of dimensions of input vectors. In our case, number of dimensions of vector u is 16, i.e. a number of quantity of chemical elements in alloy. Number of neurons in the network was set as number of dimensions of vector q, which is optimal for this task.

Initialization of network is to assign to the vectors of weights initial values that would be corrected during the algorithm. Initial values of the weight vectors are set randomly [9, 10].

As the result, after first step we got a neural network - a set of neurons *Neuro*, where each neuron is a weights vector:

$$\omega_i = (\omega_{i1}, \dots \omega_{iu}), i = 1 \dots u \tag{4}$$

Step 2. Training of neural network, i.e. adjustment of the weight coefficient of neurons. Training is to subsequent adjustments of weight coefficients. On each training step from initial training set randomly one vector is considered, which is being declared as a center of a cluster. Then the search of the most similar vector is carrying out. The most "similar" neuron is considered as a winner. After the winner have found, an adjustment of its weight coefficients is carrying out, as well as its neighbors that are in the range of training, where range of training is equal or less than the calculated proximity measure [6, 7].

Step 2.1. Randomly choose one element (alloy) from the training set N (vector A_s) and find the distances between it and each neuron in neural network using one of the following proximity measures:

1) Euclidean proximity measure considering element's level of influence, i.e. weighted Euclidean distance:

$$d_i = \sqrt{\sum_{j=1}^{C} Z_j * (A_{sj} - \omega_{ij})^2}, \tag{5}$$

where d_i is the distance between two objects: alloy A_s from training and weight vector of i-neuron $\omega_i, i = 1 \dots q$; C is the number of elements which characterizing the object - chemical elements that composes an alloy; Z is the list of the level of influence coefficients of each chemical element, $j = 1 \dots C$.

2) Manhattan proximity measure considering element's level of influence, i.e. weighted Manhattan distance:

$$d_i = \sum_{j=1}^{C} Z_j * |A_{sj} - \omega_{ij}|, \tag{6}$$

where d_i is the distance between two objects: alloy A_s from training and weight vector of i-neuron $\omega_i, i = 1 \dots q$; C is the number of elements which characterizing the object – chemical elements that composes an alloy; Z is the list of the level of influence coefficients of each chemical element, $j = 1 \dots C$.

Step 2.2. Choose from the list of distances D the shortest one and therethrough determine the neuron-winner. Determine neighbors of the neuron-winner that will be in a training range.

Step 2.3. Determine the coefficients of training speed:

1) Calculate the speed of the training [6]:

$$\sigma(k) = \sigma_0 * \exp(\frac{-k}{\tau_1}), \tag{7}$$

where *k* is the number of the training iteration, σ_0 is the speed of training, which is equal to the cluster radius, $\tau_1 = \frac{1000}{\log \sigma_0}$ is a time constant.

2) Calculate a value of a neighbor function [6]:

$$h(d, k) = \exp(\frac{-d^2}{2\sigma^2(k)}), \tag{8}$$

where *k* is the number of the training iteration, σ is the speed of training at the training iteration *k*, $d_{w,i}^2$ is the squared distance between neuron-winner with a weight vector ω_w and neuron ω_i, which is in the training range that calculates using Euclidean proximity measure.

3) Calculate a value of speed of training function [6]:

$$a(k) = \sigma_0 * \exp(\frac{-k}{\tau_2}), \tag{9}$$

where *k* is the training iteration, $\sigma_0 = 0{,}1$ is the speed of training, which is equal to the cluster radius, $\tau_2 = 1000$ is a time constant.

Step 2.4. Adjust the weights of neuron-winner and its neighbors according to the following rule:

$$\omega_i = (\omega_{ij} + h(d_{w,i}, k) * a(k) * \\ * (A_{sj} - \omega_{ij})), \tag{10}$$

where ω_i is the weight vector of adjustable neuron (which is a neighbor of the neuron-winner or is a neuro-winner) $i, j = 1 \dots n$, $h(d_{w,i}, k)$ is a neighbor function of adjustable neuron and neuron-winner (see Step 2.3), $a(k)$ is a speed of the training (see Step 2.3), k is a training iteration, A_s is an input vector-alloy for which this neuron-winner was chosen.

When for every alloy from the training set *N* will be carried out Steps 2.1-2.4 then the first training iteration is over. Iterations will continue until error will be equal to acceptably low value.

Step 2.5. After every training iteration an error function value must be calculated:

$$E = \frac{1}{n} \sum_{i=1}^{n} ||A_i - \omega_{iw}||, \tag{11}$$

where *n* is the number of alloys from training set *N*, A_i is an alloy from the training set N, ω_i is the weight vector of the neuron-winner for alloy A_i.

Acceptably low value of error for this task is 900, which justified by the large amount of input data and its number of dimensions.

After the error function value will become acceptably low the neural network can be considered as trained.

Step 3. Clusterize a set of objects *M*.

Choose an input vector from the set *M* (alloy $B_l, l = 1 \dots m$), find the distances from it to every neuron of trained network using one of the way for calculating a distance (Manhattan (5) and Euclidean distances (6)) [11]. And therethrough generate a list of distanced *D*.

Step 4. Find in the list of distances *D* the shortest one, i.e. determine neuron-winner. Assign output signal of neural-winner as a number of cluster of alloy B_l. Output signal of neuron in this case will be its index [5].

With the help of the Kohonen network, a map of alloy distribution was obtained with the proximity of the Manhattan distance consisting of seven clusters that made up alloys, usually of the same manufacturer, and were close in heat resistance; and a map with six clusters with the proximity of the Euclidean distance; in this case, the clusters were alloys of similar chemical composition, not of heat resistance.

In the general case, the number of clusters depends on the training set, but cannot exceed the number of neurons in the network, i.e., the number of chemical elements in the alloy. The larger and more diverse is the sample, the better the network will be trained.

The method may be useful in exploration analysis; for example, when a new alloy is being synthesized, unlike any of the samples, then the network trained on known alloys will not be able to attribute it to one cluster and thereby will reveal its novelty.

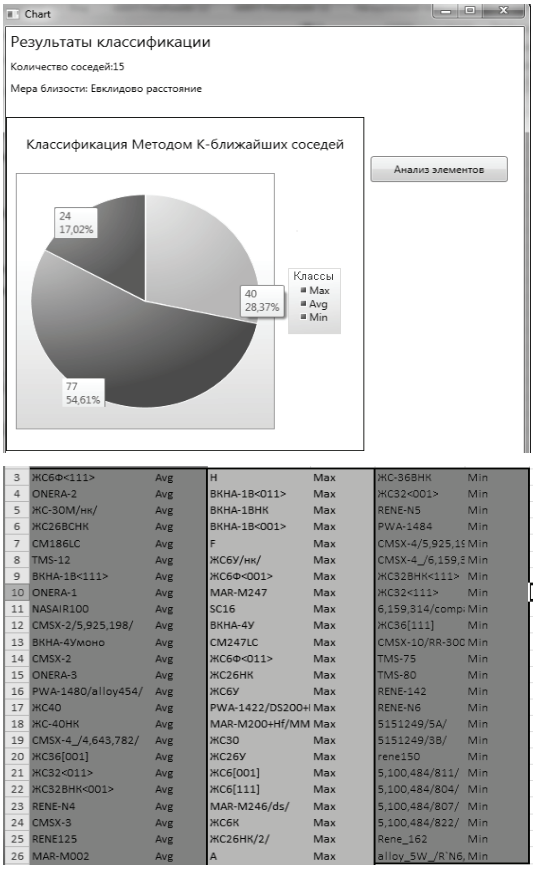
The advantage of the obtained classes and clusters is that they will probably allow, in the future, to improve the compositions of already known alloys, and could be used for modeling the composition-property dependencies in less noisy samples [3, 7].

5. Results

The algorithms for solving the first task described above were implemented as a desktop application in C # using WPF technology. The program allows you to choose a measure of proximity for classes/clusters.

On the Fig. 1 results of a classification using the nearest neighbor method are presented. Chemical compositions of alloys are considered as neighbors, Euclidean distance was chosen as the proximity measurement. As the result three classes were formed of the alloys with maximum, medium and minimum heat resistance. Some of the alloys divided to three classes are presented in the list below on the Fig. 1.

On the Fig. 2 results of a clustering using the Kohonen networks are presented. Manhattan distance was chosen as the proximity measurement. As the result seven clusters were formed. Alloys divided to seven clusters are presented in the list at the bottom of the Fig. 2.



Chart

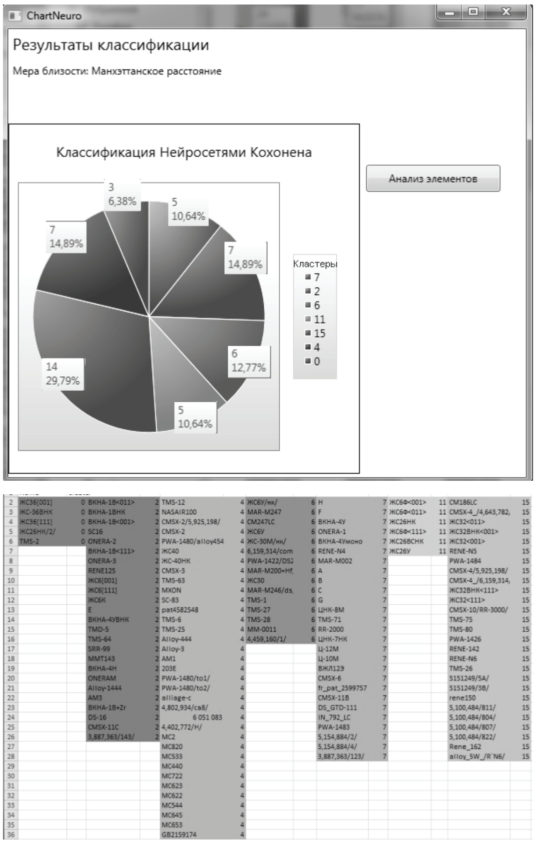
Результаты классификации

Количество соседей:15

Мера близости: Евклидово расстояние

3	ЖС6Ф<111>	Avg	H	Max	ЖС-36ВНК	Min
4	ONERA-2	Avg	ВКНА-1В<011>	Max	ЖС32<001>	Min
5	ЖС-30М/нк/	Avg	ВКНА-1ВНК	Max	RENE-N5	Min
6	ЖС26ВСНК	Avg	ВКНА-1В<001>	Max	PWA-1484	Min
7	CM186LC	Avg	F	Max	CMSX-4/5,925,1!	Min
8	TMS-12	Avg	ЖС6У/нк/	Max	CMSX-4_/6,159,3	Min
9	ВКНА-1В<111>	Avg	ЖС6Ф<001>	Max	ЖС32ВНК<111>	Min
10	ONERA-1	Avg	MAR-M247	Max	ЖС32<111>	Min
11	NASAIR100	Avg	SC16	Max	6,159,314/comp	Min
12	CMSX-2/5,925,198/	Avg	ВКНА-4У	Max	ЖС36[111]	Min
13	ВКНА-4Умоно	Avg	CM247LC	Max	CMSX-10/RR-300	Min
14	CMSX-2	Avg	ЖС6Ф<011>	Max	TMS-75	Min
15	ONERA-3	Avg	ЖС26НК	Max	TMS-80	Min
16	PWA-1480/alloy454/	Avg	ЖС6У	Max	RENE-142	Min
17	ЖС40	Avg	PWA-1422/DS200+I	Max	RENE-N6	Min
18	ЖС-40НК	Avg	MAR-M200+Hf/MM	Max	5151249/5A/	Min
19	CMSX-4_/4,643,782/	Avg	ЖС30	Max	5151249/3B/	Min
20	ЖС36[001]	Avg	ЖС26У	Max	rene150	Min
21	ЖС32<011>	Avg	ЖС6[001]	Max	5,100,484/811/	Min
22	ЖС32ВНК<001>	Avg	ЖС6[111]	Max	5,100,484/804/	Min
23	RENE-N4	Avg	MAR-M246/ds/	Max	5,100,484/807/	Min
24	CMSX-3	Avg	ЖС6К	Max	5,100,484/822/	Min
25	RENE125	Avg	ЖС26НК/2/	Max	Rene_162	Min
26	MAR-M002	Avg	A	Max	alloy_5W_/R`N6,	Min

Fig. 1. Results of classification obtained using «k-nearest neighbors» algorithm.


ChartNeuro

Результаты классификации

Мера близости: Манхэттанское расстояние

2	ЖС36[001]	0	ВКНА-1В<011>	2	TMS-12	4	ЖС6У/нк/	6	H	7	ЖС6Ф<001>	11	CM186LC	15
3	ЖС-36ВНК	0	ВКНА-1ВНК	2	NASAIR100	4	MAR-M247	6	F	7	ЖС6Ф<011>	11	CMSX-4_/4,643,782,	15
4	ЖС36[111]	0	ВКНА-1В<001>	2	CMSX-2/5,925,198/	4	CM247LC	6	ВКНА-4У	7	ЖС26НК	11	ЖС32<011>	15
5	ЖС26НК/2/	0	SC16	2	CMSX-2	4	ЖС6У	6	ONERA-1	7	ЖС6Ф<111>	11	ЖС32ВНК<001>	15
6	TMS-2	0	ONERA-2	2	PWA-1480/alloy454	4	ЖС-30М/нк/	6	ВКНА-4Умоно	7	ЖС26ВСНК	11	ЖС32<001>	15
7			ВКНА-1В<111>	2	ЖС40	4	6,159,314/com	6	RENE-N4	7	ЖС26У	11	RENE-N5	15
8			ONERA-3	2	ЖС-40НК	4	PWA-1422/DS2	6	MAR-M002	7			PWA-1484	15
9			RENE125	2	CMSX-3	4	MAR-M200+Hf,	6	A	7			CMSX-4/5,925,198/	15
10			ЖС6[001]	2	TMS-63	4	ЖС30	6	B	7			CMSX-4_/6,159,314,	15
11			ЖС6[111]	2	MXON	4	MAR-M246/ds,	6	C	7			ЖС32ВНК<111>	15
12			ЖС6К	2	SC-83	4	TMS-1	6	G	7			ЖС32<111>	15
13			E	2	pat4582548	4	TMS-27	6	ЦНК-8М	7			CMSX-10/RR-3000/	15
14			ВКНА-4УВНК	2	TMS-6	4	TMS-28	6	TMS-71	7			TMS-75	15
15			TMD-5	2	TMS-25	4	MM-0011	6	RR-2000	7			TMS-80	15
16			TMS-64	2	Alloy-444	4	4,459,160/1/	6	ЦНК-7НК	7			PWA-1426	15
17			SRR-99	2	Alloy-3	4			Ц-12М	7			RENE-142	15
18			MMT143	2	AM1	4			Ц-10М	7			RENE-N6	15
19			ВКНА-4Н	2	203E	4			ВЖЛ12Э	7			TMS-26	15
20			ONERAM	2	PWA-1480/to1/	4			CMSX-6	7			5151249/5A/	15
21			Alloy-1444	2	PWA-1480/to2/	4			fr_pat_2599757	7			5151249/3B/	15
22			AM3	2	alliage-c	4			CMSX-11B	7			rene150	15
23			ВКНА-1В+Zr	2	4,802,934/ca8/	4			DS_GTD-111	7			5,100,484/811/	15
24			DS-16	2	6 051 083	4			IN_792_LC	7			5,100,484/804/	15
25			CMSX-11C	2	4,402,772/H/	4			PWA-1483	7			5,100,484/807/	15
26			3,887,363/143/	2	MC2	4			5,154,884/2/	7			5,100,484/822/	15
27					MC820	4			5,154,884/4/	7			Rene_162	15
28					MC533	4			3,887,363/123/	7			alloy_5W_/R`N6/	15
29					MC440	4								
30					MC722	4								
31					MC623	4								
32					MC622	4								
33					MC544	4								
34					MC645	4								
35					MC653	4								
36					GB2159174	4								

Fig. 2. Results of clustering obtained using Kohonen neural network.

Since the problem in question is quantitative, the program implements the selection of the following metrics: Euclidean distance and Manhattan distance. In most cases, the latter leads to results similar to calculations of the Euclidean distance; however, for this measure, the effect of individual emissions is less than when using the Euclidean metric, since in this case the coordinates are not squared. Alloys in one class / cluster are displayed as a table.

6. Conclusions

Further research directions are related to the development of more efficient algorithms for learning neural networks, introducing neurons into the network architecture by “protection variables” and displacement neurons [6]. A separate area of research is the study of the phase composition of alloys using the methods of Data Mining and chemoinformatics.

Acknowledgements

The study was partially supported by the RFBR grant 18-07-00193-a.

References

[1]. C. T. Sims, N. S. Stoloff, W. C. Hagel, Superalloys II: High-Temperature Materials for Aerospace and Industrial Power, 2nd Ed., *John Willey & Sons Inc.*, 1987.

[2]. M. M. Rakhmankulov, V. M. Parashchenko, Technology of Casting Superalloys, *Intermet Engineering Publ.,* Moscow, 2000.

[3]. O. S. Nurgayanova, Design *Ni*-based Superalloys with Artificial Intelligence Methods, *USATU*, Ufa, 2006.

[4]. J. T. Tou, R. C. Gonzalez, Pattern Recognition Principles, London-Amsterdam-Dom Mills, *Addison-Wesley Publishing Company*, Ontario-Sydney-Tokyo*,* 1974.

[5]. H. Solomon, Dependent clustering techniques, in Classification and Clustering (J. Van Ryzin, Ed.), *Academic Press*, New York, 1977, pp. 155-174.

[6]. S. S. Haykin, Neural Networks and Learning Machines, *Pearson Education*, Upper Saddle River, NJ, 2009.

[7]. O. S. Nurgayanova, Artificial neural networks in the classification problems of superalloys, in *Proceedings of the 6th All-Russian Scientific Conference Information Technologies for Intelligent Decision Making Support*, Vol. 3, May 28-31, Ufa- Stavropol, Russia, 2018

[8]. T. Kohonen, Self-organizing Maps, *BINOM, Knowledge Lab*, 2008.

[9]. M. J. Zaki, W. Meira Jr., Data Mining and Analysis. Fundamental Concepts and Algorithms, *Cambridge University Press*, 2014.

[10]. T. Rashid, Make Your Own Neural Network. *CreateSpace*, 2016.

[11]. S. A. Ayvazian, Data Analysis, Applied Statistics and General Classification Theory, Data Analysis Methods, *Finances and Statistics*, 1985.

(55)

Data Analysis on Powered Two Wheelers Riders' Behaviour Using Machine Learning

**M. U. Ahmed [1], A. Boubezoul [2], G. Forsström [1], N. Sherif [1], D. Stenekap [1], S. Espié [2],
A. Sundström [1] and R. Södergren [1]**
[1] School of Innovation Design and Engineering (IDT), Mälardalen University, Sweden
[2] UPE-IFSTTAR/TS2/SIMU&MOTO, F-77447 Marne la Vallée Cedex, France
Tel.: + 46736620804; + 33181668321
E-mails: mobyen.ahmed@mdh.se; abderrahmane.boubezoul@ifsttar.fr

Summary: Analyzing powered two-wheeler rider behavior, i.e. classification of riding patterns based on 3-D accelerometer/gyroscope sensors mounted on motorcycles is challenging. This paper presents machine learning approach to classify four different riding events performed by powered two wheeler riders' as a step towards increasing traffic safety. Three machine learning algorithms, Random Forest (RF), Support Vector Machine (SVM) and Artificial Neural Network (ANN) have been used to classify riding patterns. The classification is conducted based on features extracted in time and frequency domains from accelerometer/gyroscope sensors signals. A comparison result between different filter frequencies, window sizes, features sets, as well as machine learning algorithms is presented. According to the results, the Random Forest method performs most consistently through the different data sets and scores best.

Keywords: Machine learning, Random forest (RF), Support vector machine (SVM), Artificial neural network (ANN), Powered two-wheeler, Classification of riding patterns, Accelerometer/gyroscope.

1. Introduction

Research has shown, people who travel on motorbikes, Scooters and other powered two wheelers (PTW) to have about 20 times larger risk of a fatal accident than people using other types of vehicles on the road [1]. An inertial measurement unit (IMU) can be placed on humans, vehicles and even skateboards, as a way to classify skateboard tricks [8], the possibilities are many and the requirements are as little as a smart-phone. To understand PTW rider's behavior the study needs to perform in-depth investigation of specific riding situations, i.e. left turn (LT), right turn (RT), straight line (SL), roundabout (RA), and stop (ST), which is a challenging task [2]. In order to classify different patterns, the most commonly used data are from gyroscope and accelerometer measurements. They are used in [3-5] to classify daily human activities like walking, running, going upstairs, going downstairs and more. Research to classify different riding events has been done before in [6] and [7] but for other moving objects than PTW. Supervised machine learning is a good approach when the data is labelled. Frequently used machine learning algorithms include Random Forest (RF) [3, 11, 14], Neural Network [11, 7 13] and Support Vector Machine (SVM) [4, 8, 11, 16, 13, 12].

The goal for this study is to create a consistent model for the classification that works for multiple algorithms and is able to identify which feature sets that contribute the most for each machine learning algorithm. Thus, this paper aims to classify basic riding situations i.e. LT, RT, SL, RA, and ST as an application of machine learning algorithms. The classification is based on data from an IMU which contains gyroscopes and accelerometers that are used to acquire movements during the motorcycle rides. Some pre-processing of data is considered to eliminate high frequency noise in order to improve the results, thus low pass filtering is introduced. The machine learning classifiers such as RF, SVM and ANN, are investigated with extracting features from the signals, both from the time domain and the frequency domain.

2. Materials and Methods

In this study 10 data channels sampled at 1000 Hz were collected from 5 drivers driving a prescribed route [9]. Out of the original 10 channels only data from the gyroscope and accelerometer were used, giving a total of 6 dimensions. The velocity, the steering angle, the break status and the distance travelled of the PTW were not used. The measurements have been manually analyzed together with a synced video so the data could be tagged with the detected class in 1 second resolution. The tagged labels are not split equally over the different classes: 1) SL: 1730 2) LT: 175 3) RT: 430, 4) RA: 291.

The step-by-steps system overview is shown in Fig. 1, where the boxes indicate programs, and the cylinders indicate stored data in files. The noise from the accelerometer and gyroscope signals was removed by a low pass filter [9], i.e. the 6th order Butterworth filter as this was the best candidate from a Signal-to-Noise Ratio (SNR) point of view and as in this off-line analysis the problem of time lag is not relevant. To find the best approach both FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) filters at multiple frequencies have been analyzed. Also, normalization was implemented by setting the mean to 0, and the variance to 1 for each of the measurement channels. The technique of Sliding Windows was used, as it emphasizes the importance of time locality of the data. The first estimate was a window length of 0.5-3 seconds. This was assumed to be a good starting point considering the physical behavior of a PTW in city traffic where an event lasted at least four seconds according to the measurement data.

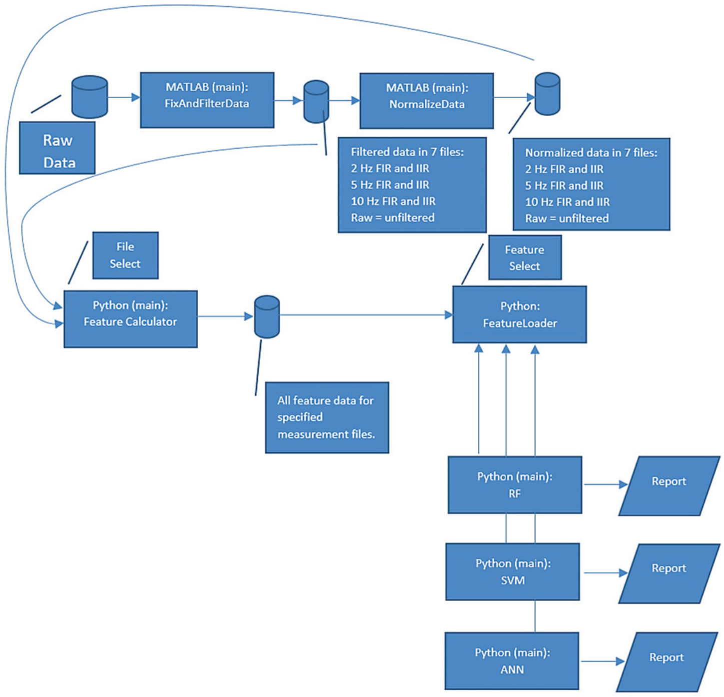


Fig. 1. System overview.

Table 1. List of all features.

Feature	Index
Mean [Ax, Ay, Az, Rx, Ry, Rz]	[0, 1, 2, 3, 4, 5]
STD [Ax, Ay, Az, Rx, Ry, Rz]	[6, 7, 8, 9, 10, 11]
Var [Ax, Ay, Az, Rx, Ry, Rz]	[12, 13, 14, 15, 16, 17]
Median [Ax, Ay, Az, Rx, Ry, Rz]	[18, 19, 20, 21, 22, 23]
Skew [Ax, Ay, Az, Rx, Ry, Rz]	[24, 25, 26, 27, 28, 29]
Kurtosis [Ax, Ay, Az, Rx, Ry, Rz]	[30, 31, 32, 33, 34, 35]
Iqr [Ax, Ay, Az, Rx, Ry, Rz]	[36, 37, 38, 39, 40, 41]
Min [Ax, Ay, Az, Rx, Ry, Rz]	[42, 43, 44, 45, 46, 47]
Max [Ax, Ay, Az, Rx, Ry, Rz]	[48, 49, 50, 51, 52, 53]
RMS [Ax, Ay, Az, Rx, Ry, Rz]	[54, 55, 56, 57, 58, 59]
ZQR [Ax, Ay, Az, Rx, Ry, Rz]	[60, 61, 62, 63, 64, 65]
Cor [(Ax,Ay),(Ax,Az),..., (Rx,Rz),(Ry,Rz)]	[66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80]
Cov [(Ax,Ax),(Ax,Ay),(Ax,Az)..., (Ry,Rz),(Rz,Rz)]	[81,82,83,84,85,86,87,88,89,90,91,92,93,94,95, 96,97,98,99,100,101]
FFT Mean (Re) [Ax, Ay, Az, Rx, Ry, Rz]	[102,103,104,105,106,107]
FFT Cor (Re) [(Ax,Ay),(Ax,Az),(Ax,Rx),(Ax,Ry)..., (Rx,Rz),(Ry,Rz)]	[108,109,110,111,112,113,114,115,116,117,118,119,120,121,122]
FFT Mean (Im) [Ax, Ay, Az, Rx, Ry, Rz]	[123,124,125,126,127,128]
FFT Cor (Im) [(Ax,Ay),(Ax,Az),(Ax,Rx),..., (Rx,Rz),(Ry,Rz)]	[129,130,131,132,133,134,135,136,137,138,139,140,141,142,143]

Based on a study which included a meta study of work with feature extraction from an IMU [10], *Mean*, *Standard deviation*, *Max*, *Min*, *Median*, *Skewness*, *Kurtosis*, *Interquartile range*, *Root mean square*, *Zero crossing rate*, *Correlation* and *Covariance* were selected as features from the time domain. Further *Mean* and *Correlation* were also selected from the frequency domain to increase the diversity of features, [5]. The total feature set is large with 145 different features presented in Table 1. However, RF and ANOVA approaches have been used in which both were set to select the 15 best features. A total of 5 feature sets have been developed and considered for classification analysis, as presented in Table 2.

Table 2. Feature sets.

Sets	Description	Features
a	RF selection	5, 22, 24, 27, 40, 69, 78, 87,88, 94, 110, 112, 123, 126, 132
b	All 145 features	0, 1, 2, 3,..., 143, 144
c	Rz Instrument-Only	5, 11, 17, 23, 29, 35, 41, 47, 53, 59, 65, 107, 128
d	Mean value for each channel excluding Az	0, 1, 3, 4, 5
e	ANOVA selection	4, 5, 23, 47, 59, 110, 115, 117,119, 120, 129, 133, 134, 141, 143

Three machine learning methods were used i.e. ANN, RF and SVM and compared the classification accuracy. ***ANN***: Keras is a Python library for Neural Network providing a simplified front-end for user friendliness. The back-end in this implementation is TensorFlow. This is the library used to build the feed-forward Neural Network. To avoid over fitting four methods have been considered: a) stratified k-fold cross-validation, b) L2-regularization c) dropout with a 20 rate and d) early stopping on validation loss (0.005). Further the tests are run with the activation functions with 2 to 4 hidden layers. The hidden layers dept 2-4 corresponds to the parameter x, y, z. ***SVM***: In this work the SVM tools included in scikit-learn where used to evaluate the method on a data set that had been pre-processed in different ways. After the pre-processing and saving the data into separate files, certain files were selected together with features and hyper-parameters to train and validate the algorithm. The hyper-parameters that were experimented on were the kernel function where three different kernels were used, Polynomial with degree two (param. x), Radial Basis Function (param. y) and Sigmoid (param. z). ***RF***: The RF is implemented through the scikit-learn library. The RF is defined with a chosen set of hyper-parameters like the number of trees in the forest, min samples of leafs to split an end node and the number of features in each split. The settings are, MinLeaf: 1 and Dept: none corresponds to parameter x, while MinLeaf: 2 and Dept: 50 corresponds to y and z. The forest is then built at random using a built-in function from scikit-learn's RandomForestClassifier library operating on the training data.

The metrics used are *mean accuracy* and its *standard deviation* (from cross-validation), *Global F_1-score*, *Class wise F_1-score*, and *unweighted mean of Class wise F_1-score*. For the evaluation a test set was created with the intent to test these different settings, and different methods against each other.

3. Results

The 15 combinations that were analyzed using the different machine learning algorithms can be found in Table 3. Settings 1-3 and 4-6 were used to evaluate the impact of normalization, and the filter cut-off frequency. The four different window sizes were compared in 6-9. Three different parameters for the machine learning algorithms were analyzed in 8, 10 and 11. Finally five different feature sets were compared between 11-15. The Params is use to provide different input to machine learning algorithms, such as for SVM it is a hyper-parameter that was experimented on were the kernel function where three different kernels were used; Polynomial with degree two (param. x), Radial Basis Function (param. y) and Sigmoid (param. z). The results for all the best settings for the machine learning variants can be found under Tables 4-6.

Table 3. Experimental settings for the ML algorithms.

Nr	Norm	filter (Hz)	Window size	Params	Features
1	Yes	2	1000	params {x}	Set {a}
2	Yes	5	1000	params {x}	Set {a}
3	Yes	10	1000	params {x}	Set {a}
4	No	2	1000	params {x}	Set {a}
5	No	5	1000	params {x}	Set {a}
6	No	10	1000	params {x}	Set {a}
7	No	10	500	params {x}	Set {a}
8	No	10	2000	params {x}	Set {a}
9	No	10	3000	params {x}	Set {a}
10	No	10	2000	params {y}	Set {a}
11	No	10	2000	params {z}	Set {a}
12	No	10	2000	params {z}	Set {b}
13	No	10	2000	params {z}	Set {c}
14	No	10	2000	params {z}	Set {d}
15	No	10	2000	params {z}	Set {e}

Table 4. Best ANN results and corresponding results for RF and SVM.

Run 12	F_1 score/class				F_1 Global score	Accuracy
	SL	LT	RT	RA		
ANN	95	75	87	87	92	90±2
RF	96	90	89	83	93	91±2
SVM	85	18	0	12	72	67±3

Table 5. Best RF results and corresponding results for ANN and SVM.

Run 9	F_1 score/class				F_1 Global score	Accuracy
	SL	LT	RT	RA		
ANN	95	0	81	65	88	90±2
RF	97	75	90	80	94	94±1
SVM	95	43	93	38	87	91±1

Table 6. Best SVM results and corresponding results for RF and ANN.

Run 8	F_1 score/class				F_1 Global score	Accuracy
	SL	LT	RT	RA		
ANN	93	0	81	58	85	86±2
RF	95	84	81	74	91	89±2
SVM	94	67	81	74	90	88±1

Using the full feature set, 'run 12' illustrated in Table 4, gave best results for the ***ANN***. Both activation functions achieved high mean accuracy on this set running with 4 hidden layers. The feature selection methods got poor results when training the neural networks, especially in detecting left turns (LT). Moving forward with an iterative search/selection method 4 based on the weight values on the input nodes, a feature set of 18 features was found. This subset preformed close to the full set in that it can still detect left turns and it gets a F_1 Global score of 88 %, see Table 7. According to Table 5, ***RF*** performed the best on run 9 were it uses feature set "a". It could correctly classify 94 % of the data. The second best was 'run 12' where it used the whole feature set from both the time domain and the frequency domain. These inputs gave an accuracy of 91 %. The difference from 'run 9' with 15 features that the accuracy for LT was higher. Another reduced feature set for RF is found in Table 7. The best results shown in Table 6, for the ***SVM*** were obtained from run 8 which used feature set "a". It achieved an accuracy of 88 % and could classify SL and RT but had some troubles with LT and RA. Using feature selection on SVM an optimal subset of 19 features achieved a score of 90 %, see Table 7.

Table 7. Selected SVM, RF and ANN feature set from 10 HZ IIR filtered data with 2 Second sliding window.

ML algorithms	Features	F_1 score/class				F_1 score	Accuracy
		SL	LT	RT	RA		
SVM	[4; 5; 23; 47; 53; 59; 110; 115; 117; 119; 120; 129; 133 - 135; 138; 141 - 143]	94	60	87	72	90	89±4
RF	[2; 15; 20; 26; 27; 55; 57; 66; 83; 86; 90; 96;130; 141]	95	95	83	77	92	91±4
ANN	[0; 18; 22; 34; 38; 42; 49; 54; 57; 59; 60; 61;65; 69; 73; 105; 109; 124; 126]	93	63	76	76	88	87±1

4. Discussion

During early evaluation using the low pass cut-off frequencies 1 Hz, 2 Hz and 5 Hz indicated that the 5 Hz gave the best Machine Learning performance. Normalization was tried both on raw data and on extracted featurs. However, the results were inconclusive about whether normalization had any effect. Due to minimal resolution of 1 sec for the data labels, any window size above 2 sec introduces a risk of data loss due to current implementation where frames are not recorded if they contain multiple labels. This problem can decrease the diversity of the data which may lead to a sampling bias. From comparing the results 6 run-9 run, greater size window increases the accuracy and score of the classification. However, this increase is questionable since the increase in score and accuracy might come due to the loss of data for the events occurring with lesser frequency. The main difficulty lies in classification of the LT and RA and to a lesser degree RT events, which all suffer from a larger proportional loss of labels compared to the SL event.

In Table 2 five different feature sets are presented. Feature set "a" comes from the RF selection method and performed well for RF. The second set "b" (All features) was chosen as a comparison to the others. It performed very well, as was expected. Feature set "c" was loosely based on a similar classification problem but for a car and consists of features only extracted from the Rz channel. For SVM and for ANN it gave poor results (0 % LT). One reason could be that a PTW leans more in curves then a car would. This also indicates that for classification of the LT event, other features are required. Feature set "d" gave really poor results for all algorithms except for the RF which gave 89 %, which is considered to be OK. Even if the Az channel did not have so much impact on the data, the other features did. The last feature set "e" was selected using the SVM's ANOVA method. It did give a good overall accuracy for the SVM but unfortunately it could not classify the RA class.

The overall results indicate that the RF method gives the best results. A reduced feature set with good performance and robustness is suggested in Table 5. At first glance, the F1 score is not as strong as the subset that is presented in Table 5, however considering that the feature set contains one less feature, the window size is 2s instead of 3s and that the Class F1-score of LT is 20 % better while only losing 2 % on the Global F1-score, this feature set is surly the better choice. Some effort was made at finding a connection between features and events, however, this attempt was unsuccessful. The authors believe that this is mainly a time constraint problem, the connection would be found if more time was invested in finding it. The examined data set was strongly biased towards SL for each of the 9 tours. Only about 5 % of the events were left turns and almost 75 % were straight line which means that the models could achieve high *Global F_1-score* by predicting SL at every event even though they missed all LT.

5. Conclusions

In this paper three different machine learning algorithms were trained on IMU data from a PTW to classify four different activities: SL, LT, RT and RA. The steps taken before training the algorithms include filtering, feature extraction and feature selection. Feature selection is useful for all methods. RF especially benefits by reducing computation efforts without impairing the accuracy or score. SVM also benefited from feature selection in the same way as RF but the accuracy also improved. ANN loses more accuracy and score compared with the other models however it gains a lot in training speed. The results also indicate that modularization of the parameter choices for filtering, window size and feature set is not possible. When comparing features from the reduced feature sets the results indicate that features from accelerometer x-y axes and features from gyroscope x-z axes were selected. Also, features from the frequency domain like covariance and correlation performed best.

References

[1]. F. Attal, A. Boubezoul, L. Oukhellou, S. Espié, Riding patterns recognition for powered two-wheelers users behaviors analysis, in *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC'13)*, 2013.

[2]. N. Baldanzini, Y. Hurth, M. Regan, I. Spyropoulou, N. Eliou, P. Lemonakis, Literature Review of Data Analysis for Naturalistic Driving Study, Deliverable 4, 2BESAFE Project, *European Commission*, Belgium 2009.

[3]. L. Yiyan, Z. Fang, S. Wenhua, L. Haiyong, An hidden Markov model based complex walking pattern recognition algorithm, in *Proceedings of the 4th International Conference on Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS'16)*, 2017, pp. 223–229.

[4]. Z. Ouyang, J. Niu, M. Guizani, Improved vehicle steering pattern recognition by using selected sensor data, *IEEE Transactions on Mobile Computing*, Vol. 14, Issue 8, 2017, pp. 1-1.

[5]. Y. Tian, W. Chen, MEMS-based human activity recognition using smartphone, in *Proceedings of the 35th Chinese Control Conference (CCC'16)*, Chengdu, 2016, pp. 3984-3989.

[6]. N. Thepvilojanapong, K. Sugo, Y. Namiki, Y. Tobe, Recognizing bicycling states with HMM based on accelerometer and magnetometer data, in *Proceedings of the SICE Annual Conference*, 2011, pp. 831-832.

[7]. V. D. Lecce, M. Calabrese, NN-based measurements for driving pattern classification, in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference (I2MTC'09)*, 2009, pp. 259-264.

[8]. B. H. Groh, T. Kautz, D. Schuldhaus, IMU-based trick classification in skateboarding, in *Proceedings of the KDD Workshop on Large-Scale Sports Analytics*, 2015.

[9]. A. Boubezoul, S. Espié, B. Larnaudie, S. Bouaziz, A simple fall detection algorithm for powered two wheelers, *Control Engineering Practice*, Vol. 21, Issue 3, 2013, pp. 286-297.

[10]. I. M. Pires, N. M. Garcia, N. Pombo, F. Flórez-Revuelta, S. Spinsante, Pattern Recognition Techniques for the Identification of Activities of Daily Living using Mobile Device Accelerometer, *arXiv:1711.00096*, 2015.

[11]. C. Shen, Y. Chen, G. Yang, On motion-sensor behavior analysis for human-activity recognition via smartphones, in *Proceedings of the IEEE International Conference on Identity, Security and Behavior Analysis (ISBA'16)*, 2016.

[12]. S. Kim, J. Chun, A. K. Dey, Sensors know when to interrupt you in the car, in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15),* 2015, pp. 487-496.

[13]. G. Bhatia, S. Rani, Disease recognition and classification from movement patterns, in *Proceedings of the International Conference on Computing for Sustainable Global Development (INDIACom'16)*, 2016, pp. 3682-3687.

[14]. V. Silva, et al., A wearable and non-wearable approach for gesture recognition – Initial results, in *Proceedings of the 9th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT'17)*, 2017, pp. 185-190.

[15]. I. M. Pires, G. Nuno, N. Pombo, F. R. Francisco, S. Susanna, Pattern recognition techniques for the identification of activities of daily living using mobile device accelerometer, computers and society, data analysis, statistics and probability, arXiv:1711.00096, 2017.

(57)

Drivers' Sleepiness Classification Using Machine Learning with Physiological and Contextual Information

S. Barua [1], M. U. Ahmed [1] and S. Begum [1]
[1] School of Innovation Design and Engineering (IDT), Mälardalen University, Sweden
Tel.: + 46736620804
E-mails: {shaibal.barua, mobyen.ahmed, shahina.begum}@mdh.se

Summary: Analysing physiological parameters together with contextual information of car drivers to identify drivers' sleepiness is a challenging issue. Machine learning algorithms show high potential in data analysis and classification tasks in many domains. This paper presents a use case of machine learning (e.g. k-nearest neighbour (k-NN), support vector machines (SVM), case-based reasoning (CBR), and random forest (RF)) approach for drivers' sleepiness classification. The classifications are conducted based on drivers' physiological parameters and contextual information. The sleepiness classification shows receiver operating characteristic (ROC) curves for k-NN, SVM, and RF were 0.98 on 10-fold cross-validation and 0.93 for leave-one-out (LOO) for all the classifiers.

Keywords: Sleepiness, Machine learning, Electroencephalography, Contextual information.

1. Introduction

Futuristic driving paradigm focuses on detection of the driver states for current naturalistic driving and also for highly automated driving system. Impaired driving is dangerous and factors such as sleepiness, often cause impaired driving that leads to vehicle crashes and traffic injuries [1, 2]. In literature, more than 90 % of traffic crashes are assigned to the drivers [3]. Domain knowledge from sleepiness can be used with machine learning for detection and classification of the level of sleepiness. To develop a classification scheme requires understanding and knowledge discovery from the physiological signals and other relevant data [4].

This paper presents the use of standard sources of information to detect sleepiness. That is, a classification approach for drivers' sleepiness using machine learning with physiological and contextual information. Here, four machine learning classifiers namely, k-nearest neighbour (k-NN), support vector machines (SVM), case-based reasoning (CBR), and random forest (RF) have been investigated. It also explores the improvement of detection and classification accuracies by adding contextual information. Physiological signals, i.e., electroencephalography (EEG) and electrooculography (EOG) used with machine learning algorithms to detect sleepiness. Contextual features such as road condition, light condition and sleepiness model based on sleep/wake pattern included as features in the feature engineering process. A 10-fold cross-validation and leave-one-out (LOO) approaches have been investigated by the all classifiers and considers both binary and multi-class classification approaches. Moreover, sensitivity, specificity, accuracy and receiver operating characteristic (ROC) were compared between the classifiers.

2. Materials and Methods

The study on drivers' sleepiness classification was conducted under the research project VDM[1] [5]. The dataset contains recording from 30 participants acquired while driving in a high-fidelity moving-base car driving simulator at VTI [2]. Three simulated scenarios: (1) a rural road with a speed limit of 80 km/h in daylight, (2) the same rural road in darkness and (3) a suburban road in daylight is conducted. Data collected from the participants in six sessions which were a combination of 'alert' and 'sleep' derived condition and the duration of each scenario was 30 minutes and a speed limit of 80 km/h. In total the dataset holds recoding from 540 drives (30 drivers × 6 occa-sions × 3 scenarios). Physiological signals, i.e. Electroencephalography (EEG) signals were recorded using 30 channels EEG settings based on the 10-20 system. Electrooculography (EOG) as horizontal with electrodes at the outer canthi, and vertical with electrodes above/below the left eye were also acquired. In the study, sleepiness was measured using the Karolinska Sleepiness Scale (KSS) [6] rating, where participant rated their sleepiness status every fifth minute through-out the drives [7, 8]. Three sleepiness levels were defined based on the KSS: *alert* (KSS 1 – 5), *somewhat sleepy* ($6 \leq$ KSS ≤ 7) and *sleepy* (KSS ≥ 8).

Drivers habitually engage in a series of actions during driving such as turning the steering wheel, changing gears, scanning the forward road scene,

[1] http://www.es.mdh.se/projects/326-VDM___Vehicle_Driver_Monitoring

[2] https://www.vti.se/en/

looking at a side and rear-view mirrors and looking over the shoulder. Besides, a sleepy driver usually changes position, yawn, and lean the head against the headrest [9]. These activities naturally contaminate the EEG signals with muscle, motion, and ocular artefacts [10]. Therefore, before feature extraction, the acquired EEG signals were artefacts handled using the ARTE tool [11]. ARTE is an in-house developed tool which has been evaluated qualitatively by an expert in neurophysiology as well as by several quantitative measures.

From each EEG signal, power spectral density (PSD) of particular frequency bands (Borghini, et al., 2012; Craig, et al., 2012; Papadelis, et al., 2006), δ(<4 Hz), θ(4-7 Hz), α(8-12 Hz), β(12-30 Hz), γ (31-50 Hz), and the ratio such as (θ+α)/β, α/β, (θ+α)/(α+β), and θ/β (Jap, et al., 2009) were extracted as features. Two features: (a) blink durations (Jammes, et al., 2008) and (b) PERCLOS (Wierwille, et al., 1994) were extracted from the EOG signal. Further, three contextual features were: (a) sleep/wake predictor (SWP) [12, 13], (b) road condition in the scenario i.e., rural/urban road and (c) light condition in the scenario i.e., daylight/darkness. In total, features obtained from 9310 one-minute segments, out of 312 drives and the feature vector consisted of 275 features: 270-features from 30 channels EEG, 2-features from EOG and 3-contextual features. The dataset split into training and test datasets, where 70 % of the observations in the training set and 30 % of the observations in the test dataset.

The feature vector mostly consisted of features obtained from the EEG signals, where many of which recorded from the adjacent electrodes. Hence there might be some overlapping among the features, and by reducing redundant and irrelevant features, a classifier can improve its performance and generalisability. Therefore, the optimal subset of features selected from the 275 features using a univariate feature selection method [14]. The univariate feature selection was performed based on the ratio of the sum of squared differences between-groups and within-groups (BSS/WSS) of the classes, for each feature. During feature selection, the training dataset was further divided into two sets, with 80 % for feature selection and 20 % of the training data as a validation set.

Later, four classifiers namely, k-NN, SVM, CBR, and RF were separately trained using the training dataset, and later classification performance evaluated using the test dataset.

3. Results

Features selected by BSS/WSS essentially consisted of clusters or groups containing similar information. For example, eye closure-related EOG features and $(\theta + \alpha)/(\alpha + \beta)$ from ten different frontal electrode sites, Fp1, Fp2, Fpz, F3, F4, F7, F8, FC1, FC6, and Fz. Further, the SWP was the top-ranked feature by the BSS/WSS, whereas light condition and road condition was ranked 43rd and 47th respectively [7].

One of the objectives of feature selection was to investigate the importance of contextual features in classifying sleepiness. Hence, comparisons were made between classifications with and without contextual features. To investigate the influence of contextual information, feature subsets obtained from the BSS/WSS were used with and without contextual features to classify sleepiness level for both multi-class and binary classification using an SVM classifier. The best classification performance on the validation dataset is presented in Table 1; notably, performance increased when contextual features were included.

Table 1. Comparison of SVM evaluation on the subset of features obtained by BSS/WSS. Results show best performance when contextual features were included and excluded using the validation dataset.

Criteria	Multi-class classification		Binary classification	
	Including contextual features	Excluding contextual features	Including contextual features	Excluding contextual features
Sensitivity	0.80	0.76	0.95	0.94
Specificity	0.85	0.80	0.94	0.91
Accuracy	0.92	0.87	0.95	0.93

For the sleepiness classification, both multi-class and binary classification is performed using a) 10-fold cross-validation and b) leave-one-out (LOO) validation. *Multi-class classification* using 10-folds cross-validation on training dataset shows 78 % accuracy for k-NN, 80 % accuracy for SVM, 77 % accuracy for RF, and 33 % accuracy for CBR. Fig. 1 shows the prediction performance of multi-class classification on test dataset considering accuracy, sensitivity, and specificity. On the test dataset, SVM showed better performance (79 % accuracy) with 10-fold cross-validation and RF showed better performance (55 % accuracy) using the LOO validation.

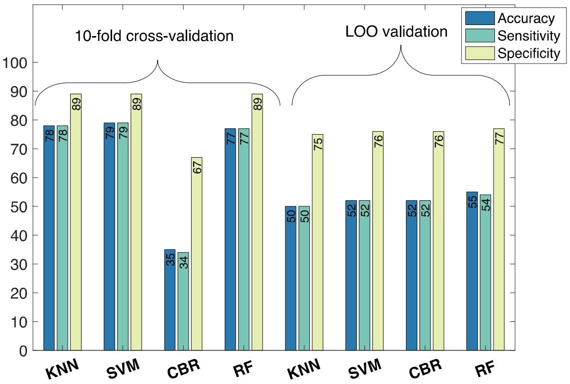


Fig. 1. Multi-class classification performance of k-NN, SVM and CBR and RF on the test dataset, where the models were trained using 10-fold cross-validation and LOO.

Binary classification excluding the somewhat sleepy group with 10-fold cross-validation on training dataset obtains classification accuracy of 93 % for k-NN, 94 % for SVM 94 %, 54 % for CBR and 93 % for RF. For the LOO validation, the results are 88 % accuracy with k-NN; 89 % accuracy with SVM; 91 % accuracy with CBR; and 88 % accuracy with RF. Fig. 2 shows the prediction performance of binary classification excluding *somewhat sleepy* group on test dataset considering accuracy, sensitivity, and specificity. Fig. 3 shows the corresponding ROC curve of binary classification excluding somewhat sleepy group on the test dataset. On the test dataset, SVM showed better performance (94 % accuracy) with 10-fold cross-validation and CBR showed better performance (91 % accuracy) using the LOO validation.

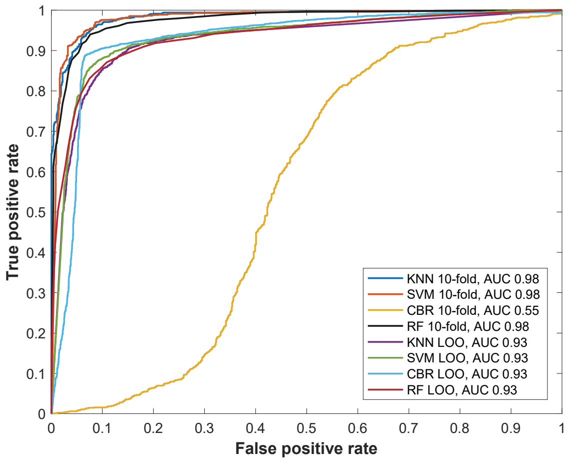


Fig. 2. Binary classification performance of k-NN, SVM and CBR and RF on the test dataset, where the models were trained using 10-fold cross-validation and LOO.

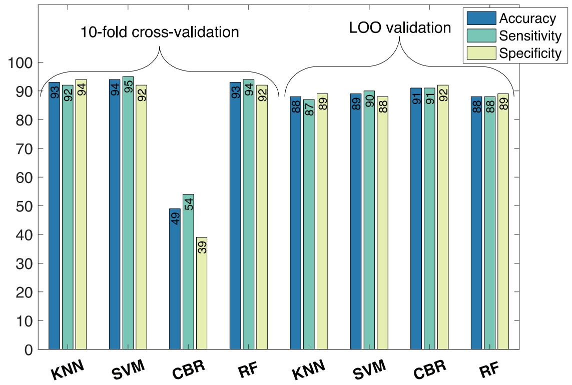


Fig. 3. ROC curves for k-NN, SVM and CBR and RF classification on the test dataset, where the models were trained using 10-fold cross-validation and LOO.

4. Discussion

The importance of contextual information for sleepiness classification is presented in this paper, in which the classification accuracy improved by 5 % in multi-class classification and by 2 % in binary classification. The improvement may appear small; however, the performance can be further improved with advanced sampling methods for imbalanced datasets. Moreover, real-world driving is much more complex than simulated driving. Driving states are often subjective experiences, in which many other contextual data can play a vital role. The work described in this thesis indicates further investigation in this area, as suggested in [15, 16]. However, the problem with this approach is that these features only represent a portion of the driving conditions, whereas probabilistic measures, as presented in [15] are not generalised but subject- and environment-dependent; nevertheless, no significant improvement in ROC and AUC was found by McDonald, et al. [16]. The use of SWP as features may appear unreasonable. Modern vehicles are already equipped with a wireless network and can synchronise with smart devices. In the future, personalised information such as sleeping hours and waking time can be integrated with smart devices and synchronised with vehicles. The limitation of SWP is that it functions with group mean data, and for different sleep patterns (e.g., those of shift workers), model parameters must be modified [17]. This shortcoming represents an open challenge for identifying appropriate contextual features from different road conditions, which also requires an extensive study design.

Compared to the other classifiers, at first it may seem strange that somewhat contradictory results obtained with CBR, with increasing performance in LOO compared to 10-fold cross validation. Historical cases are the foundation of a case library in CBR to solve a problem [18], and it requires a sufficiently large case library to perform well. The case library becomes larger when LOO was chose for validation and thus covers a larger problem space, giving CBR a better likelihood of finding similar cases. An advantage with CBR is its inherent ability to continuously incorporate new cases into the case library, thus facilitating a system that adapts to the individual over time.

Another issue observed was with the multi-class classifications: although KSS is a validated method for measuring sleepiness [6], there are difficulties in rating sleepiness in the transitional state between alert and sleepy. This issue with uncertainty in the target values may be alleviated by redistributing the ‘somewhat sleepy’ group into the ‘alert’ and ‘sleepy’ groups, which indeed led to improved performance.

5. Conclusions

In this paper, driver sleepiness classification approach using EEG, EOG and contextual information is presented. Compare to the binary classification the multi-class classification found to be challenging to classify. The reason might be that KSS is a subjective measure and the somewhat sleepy group, i.e., neither alert nor sleepy is difficult is discriminate. Thus the future research indicates a) more focus on multi-class classification, b) what other contextual information can be included in the classifier design and c) continuous develop and adapt to the current driver as new unlabelled data becomes available in the system.

Acknowledgment

The authors would like to acknowledge VINNOVA (Swedish Governmental Agency for Innovation Systems) for supporting the Vehicle Driver Monitoring (VDM) research project. The authors would also like to acknowledge our project partners Anna Anund, Christer Ahlström and Carina Fors (VTI) as well as Emma Nilsson, Per Lindén and Bo Svanberg (Volvo Car Corporation).

References

[1]. P. Philip, *et al.*, Fatigue, sleep restriction and driving performance, *Accident Analysis & Prevention,* Vol. 37, Issue 3, 2005, pp. 473-478.

[2]. J. Horne, L. Reyner, Vehicle accidents related to sleep: A review, *Occupational and Environmental Medicine,* Vol. 56, Issue 5, 1999, pp. 289-294.

[3]. S. Singh, Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey, in Traffic SafetyFacts Crash. Report No. DOT HS 812 115*, National Highway Traffic Safety Administration*, Washington, DC, 2015.

[4]. G. Yang, Y. Lin, P. Bhattacharya, A driver fatigue recognition model using fusion of multiple features, in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC'05)*, Vol. 2, 2005, pp. 1777-1784.

[5]. E. Nilsson, *et al.*, Vehicle Driver Monitoring: Sleepiness and Cognitive Load, in VTI Rapport, 2017, http://urn.kb.se/resolve?urn = urn:nbn:se:vti: diva-11835

[6]. T. Åkerstedt, A. Anund, J. Axelsson, G. Kecklund, Subjective sleepiness is a sensitive indicator of insufficient sleep and impaired waking function, *Journal of Sleep Research,* Vol. 23, Issue 3, 2014, pp. 242-254.

[7]. S. Barua, M. U. Ahmed, C. Ahlström, S. Begum, Automatic driver sleepiness detection using EEG, EOG and contextual information, *Expert Systems with Applications,* Vol. 115, 2019, pp. 121-135.

[8]. C. Ahlstrom, S. Jansson, A. Anund, Local changes in the wake electroencephalogram precedes lane departures, *Journal of Sleep Research,* 2017.

[9]. A. Anund, C. Fors, D. Hallvig, T. Åkerstedt, G. Kecklund, Observer rated sleepiness and real road driving: An explorative study, *PLoS ONE,* Vol. 8, Issue 5, 2013, e64782.

[10]. S. Barua, S. Begum, A review on machine learning algorithms in handling EEG artefacts, in *Proceedings of the Swedish AI Society Workshop (SAIS'14)*, Stockholm, Sweden, 22-23 May 2014.

[11]. S. Barua, M. U. Ahmed, C. Ahlstrom, S. Begum, P. Funk, Automated EEG artefact handling with application in driver monitoring, *IEEE Journal of Biomedical and Health Informatics,* Vol. PP, 2017, pp. 1-1.

[12]. T. Åkerstedt, J. Connor, A. Gray, G. Kecklund, Predicting road crashes from a mathematical model of alertness regulation – The Sleep/Wake Predictor, *Accident Analysis & Prevention,* Vol. 40, Issue 4, 2008, pp. 1480-1485.

[13]. D. Sandberg, T. Åkerstedt, A. Anund, G. Kecklund, M. Wahde, Detecting driver sleepiness using optimized nonlinear combinations of sleepiness indicators, *IEEE Transactions on Intelligent Transportation Systems,* Vol. 12, Issue 1, 2011, pp. 97-108.

[14]. S. Dudoit, J. Fridlyand, T. P. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, *Journal of the American Statistical Association,* Vol. 97, Issue 457, 2002, pp. 77-87.

[15]. R. Fu, H. Wang, W. Zhao, Dynamic driver fatigue detection using hidden Markov model in real driving condition, *Expert Systems with Applications,* Vol. 63, 2016, pp. 397-411.

[16]. A. D. McDonald, J. D. Lee, C. Schwarz, T. L. Brown, A contextual and temporal algorithm for driver drowsiness detection, *Accident Analysis & Prevention,* Vol. 113, 2018, pp. 25-37.

[17]. T. Åkerstedt, S. Folkard, The three-process model of alertness and its extension to performance, sleep latency, and sleep length, *Chronobiol Int.*, Vol. 14, Issue 2, 1997, pp. 115-123.

[18]. I. Watson, F. Marir, Case-based reasoning: A review, *The Knowledge Engineering Review,* Vol. 9, Issue 4, 1994, pp. 327-354.

(64)

A Feature Selection Framework and a Predictors Study for Internet Traffic Classification

Santiago Egea Gómez [1], Luis Hernández-Callejo [2], Belén Carro Martínez [1] and Antonio Sánchez-Esguevillas
[1] University of Valladolid, Castilla y León, Valladolid, Spain
[2] University of Valladolid, Campus Universitario Duques de Soria, Soria, Spain
Tel.: + 34 983423980
E-mail: santiago.egea@alumnos.uva.es

Summary: Network traffic classification (NTC) has attached attention due to its relevance for traffic control in enabling technologies, taking a significant prominence approaches based on Machine Learning (ML). Being Feature Selection (FS) an essential for ML classification, a new FS framework is presented here for efficient early NTC. Our proposal combines filter and wrapper methods to increase the diversity in attribute selection, these strategies independently produce predictor rankings that are used in a final subset selection. The proposal is tested against datasets extracted from a quite challenging network scenario. The proposal was validated against different NTC datasets extracted from various data sources contained in packet headers. The presented results probe that our FS method is effective in reducing the problem dimensionality preserving or even improving classifier performances. Furthermore, we discuss the predictive power of different sets of predictors for early NTC. Finally, we performed a minor experiment to assess the effect of port evasion when port numbers are included in the predictive model.

Keywords: Machine learning, Feature selection, Network traffic classification, Supervised learning, Network management.

1. Introduction

Enabling technologies rely on underlying Internet networks as means of communication. Smart Cities and Internet of Things envision facilities (such as: control of critical infrastructures, assistance in emergencies and smart transportation) requiring traffic control for availability, privacy and security [1]. In this vein, NTC constitutes a key piece to detect service decays and cyber threats.

ML has attached a relevant prominence for NTC, since it enables accurate traffic classification evading the handicaps of previous approaches [2]. An essential in ML is FS. Through FS, the best predictors are selected for training, meanwhile irrelevant ones are discarded leading to efficient classification models. Three FS approaches exist according to their selection schemes: (1) Filter methods, which assume an information metric to assess the quality of predictors; (2) Wrapper methods, which evaluate the importance of attributes using learning algorithms; and (3) Embedded methods, which are integrated in learning algorithms while training. In this paper, an FS framework is proposed for efficient NTC. The predictors are separately ranked by several filter and wrapper strategies, and afterwards the rankings are combined to select the final subset. Furthermore, we analyze several sets of predictors from different raw information contained in packet headers.

This article is structured as follows. Section 2 reviews previous works in FS for NTC. Section 3 describes the methodology followed and presents our FS framework. The experimental results are presented and discussed in Section 4, and finally we draw the conclusions.

2. Previous Works

FS for efficient ML-based NTC is not unexplored, and many authors have proposed their solutions. In [3], A. Fahad et al. presented an FS method called Global Optimization Approach (GOA) to find a stable set of predictors over time. GOA combines a preselection phase based on filters and a subsequent wrapper scheme based on Random Forest. A class-oriented FS algorithm (COFS) and an ensemble classifier were proposed in [4]. First, COFS selects a preliminary subset of attributes for each class, and redundant attributes are removed according to Weighted Symmetric Uncertainty (WSU).

A feature extraction and selection approach were presented in [5]. Feature extraction is based on Wavelet Multifractal transformation on raw packet-header information. As for attribute selection, Principal Component Analysis (PCA) is used to filter out the irrelevant components, and K-Means to cluster the features that are optimal or redundant. M. Shafiq et al. [6] presented a wrapper method to select the best predictors for imbalanced NTC. The proposed method filters out the irrelevant attributes using Weighted Mutual Information metric, and a learning algorithm is used to assess the AUC-ROC for each predictor.

Finally, Efficient Feature Optimization Approach (EFOA) is proposed [7] to confront Class Imbalance and concept of drift in NTC. This proposal includes feature generation using Deep Belief Networks and a subsequent selection based on WSU.

Our FS framework presents relevant differences respecting previous approaches, since we combine several information metrics and performance metrics to improve the diversity for predictor selection.

3. Methods and Materials

3.1. Network Environment and Datasets

The traffic data employed here was captured in a backbone at an ISP network, which constitutes a challenging scenario. The traffic was sniffed in a high-speed link supporting transmission rates up to 7 Gbps, where connections are susceptible to packet losses and multipath effect. We have employed two different datasets for training and validating the NTC models. The training data comprises 12 Gb collected on 17/1/2017 for 5 minutes, meanwhile the validation consists of 35.62 GB captured for almost 10 minutes on 23/3/2017.

The first five packet-headers were processed for each flow to create the classification objects and following the concept of early NTC [8]. There are mainly four information sources available in IP and transport layers: packet sizes, timestamps, window sizes and others parameters (duration, directions, ports and so on). For the three first of them, a set of 47 predictors were computed and merged with the original raw information, as Fig. 1 illustrates. The collection of predictors includes statistics (means, maximum and minimum values …) and FFT transform components. In the case of “Others” predictors, we considered ports, directions, % of packets in each direction and packet counts. Finally, we merged all those datasets resulting in a whole dataset with a total of 172 attributes. The different datasets are denoted as follows: SIZES, IATS, WINSIZES, OTHERS and WHOLE.

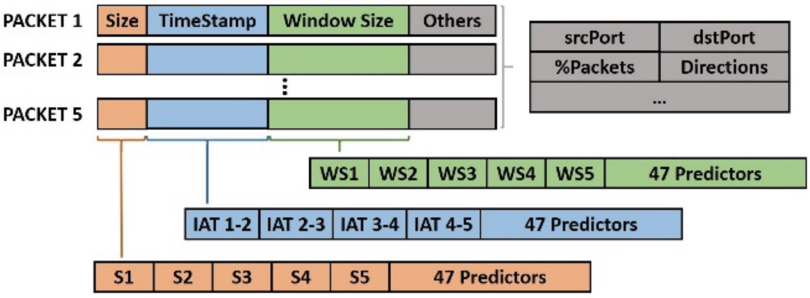


Fig. 1. Collection of attributes and datasets employed for our experiments.

3.2. FS Framework Based on Rankings

Fig. 2 presents the proposed FS framework. As it is shown, three phases are involved: (1) Filter, (2) Wrapper, and (3) Final Ranking. During (1), several independent attribute rankings are computed using different filters, meanwhile wrapper strategies rank predictors in (2). The more relevant the attribute, the topper the position it occupies in the rankings, so that the attribute positions are used as scores and combined in means to select the final subset in (3).

Through combining filters, the attributes are assessed against different information-based metrics, therefore increasing the diversity in the selection. In (2), several learning algorithms and performance metrics are assumed leading to more rich selection criteria. This framework follows a flexible design that allow extending it including more selection strategies.

For our experiment, we selected the following nine filter methods using different information metrics: MRMR [9], CIFE [10], CMIN [11], ICAP [12], MIFS [13], DISR [14], JMI [15] and MIM [16]. The most of the filters selected are implemented in the Python library, with the exception of FCBFiP, which was used in [17] and is available in [18]. Regarding Wrapper Ranking, we have evaluated three performance metrics for three learning algorithms. As Decision Tree algorithms have shown as a promising approach for NTC due to its excellent ratio amongst precision and speed, we have selected the CART Decision Tree and two ensemble algorithms implemented in Scikit-learn [19]. The OutputCode and Bagging algorithms were the ensemble techniques considered in wrapper ranking. In (2), our FS framework ranks the attributes according to the performance metrics they produce when they are used as unique predictor for each learning algorithm. The selected performance metrics in this phase are: Overall Accuracy (OA), Byte Accuracy (BA) and geometric mean (GM).

4. Experimental Results

In this section, we present and discuss the results obtained from applying our FS framework to the different datasets. Firstly, we present a preliminary experiment to assess predictive power of the different sets of predictors (SIZES, IATS, WINSIZES and OTHERS) and confirm the efficiency of our selection approach. In a second experiment, we applied our FS framework to the WHOLE dataset to assess which are the most relevant attributes when several information sources are combined. Furthermore, we performed a minor experiment to evaluate how port evasion may affect to performance metrics when port numbers are included as attributes in traffic classifiers.

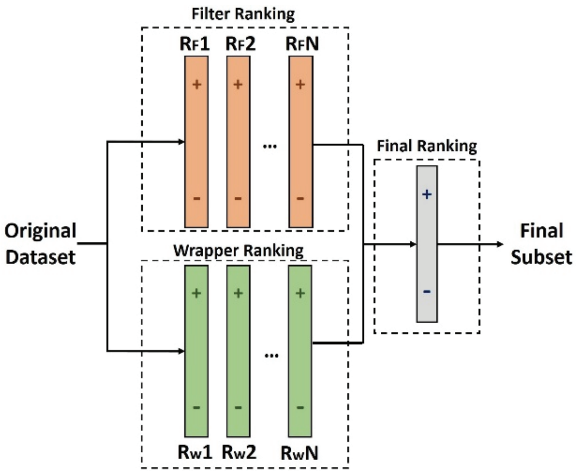


Fig. 2. FS Framework based on rankings.

4.1. Preliminary Results

Through this experiment, we pretend to validate our FS framework and study which family of predictors are the most relevant for early NTC. The final ranking for each dataset is employed to sequentially vary the number of attributes selected, and a Decision Tree is trained to evaluate the model performances according to OA, BA and GM. Fig. 3 presents the results obtained for the different models, and the top ten attributes are shown in Table 1 for each case.

From Fig. 3, we find that the best-performing attributes in terms of OA are OTHERS, with which we were able to identify more than 82 % of connections. Conversely, the WINSIZES and SIZES obtained OAs around 58 % and 51 % respectively. While on BA, SIZES clearly outperformed the other sets of predictors accomplishing BAs up to 94.7 %. The rest of datasets did not produced as positive BAs as SIZES, and the second best BAs were obtained by WINSIZES with values around 80 %. Focusing on GM, we observe that OTHERS is anew the set of attributes producing the best results reaching GMs greater than 65 % for certain subset sizes. In the case of SIZES, the GM overcame 23 % for 7 and 8 predictors; on contrast to IATs and WINSIZES datasets that yielded null GMs. OTHERS and IATS produced quite weak results for this performance metric. Note also that the subsets computed by our FS framework accomplished similar performances to that using all predictors for each datasets, which confirms the effectiveness of our proposal in reducing the dimensinality space without performance decays.

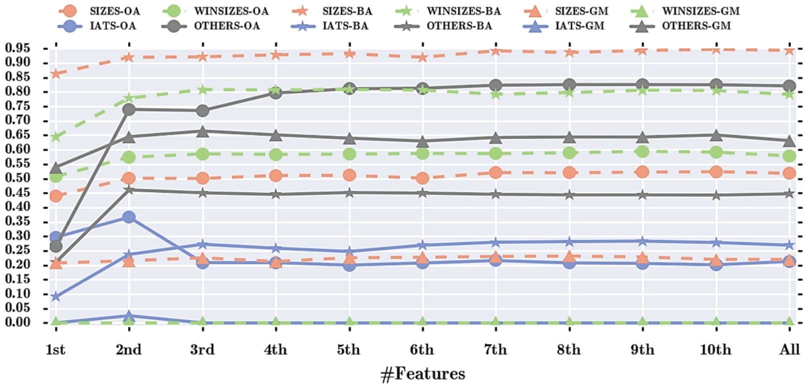


Fig. 3. Performances for different subet sizes.

If we observe Table 1, we find that only there is one FFT component (8th) amongst the best ten features for SIZES. On the contrary, raw packet sizes (2nd, 7th and 9th) and statistical components (such as: means root mean square and maximum values) have a notable presence in the selected subset (1st, 3rd, 4th, 5th, 6th and 10th). While on IATS, FFT-related predictors (such as module and phase of FFT components) were selected (1st, 6th, 7th, 9th and 10th) jointly with an important number of statistics (2nd, 3rd, 4th and 5th), meanwhile only one raw attribute was selected (8th). Focusing on WINSIZES, we observe that two raw predictors (2nd and 8th) and two FFT phases (4th and 10th) were selected for the final subset, in contrast to statistical attributes that mainly composed the top ten subset (1st, 3rd, 5th, 6th, 7th and 9th). Unlike the previous datasets, OTHERS dataset contains predictors of other nature. Interestingly, we find that source and destination ports (1st and 2nd) have an important impact on the final subset resulting from our FS framework. Furthermore, we find that packet-size counts exhibited notable predictive power for NTC according to their positions in the ranking (3rd, 4th, 5th, 6th and 10th). Conversely, packet directions (6th and 8th) and percentage of exchanged packets (7th and 9th) were not as important as the former attributes.

Generally, the poor performances exhibited by IATS for all metrics considered might be caused by the instability of this family of predictors. IATS are quite susceptible to the operation phase of the Internet network. When the workload is very high packet forward slows down, which produces variations in these types of attribute leading to performance losses.

In the case of WINSIZES, something similar might happen. TCP window sizes are variable parameters that protocols adjust depending on the network workload, so that attributes related to this parameter could vary on time. Another important handicap of these kinds of attribute is that window sizes are not useful for UDP connections.

As its high GM reveals, OTHERS attributes provided predictiveness to identify different types of application. However, port numbers had an important role for this collection of predictors. As port numbers are configurable parameters, models including these parameters are quite susceptible to port evasion, which is a severe handicap.

In the case of SIZES, these attributes are independent from the transport protocol. Similarly to OTHERS, the GM obtained for SIZES indicates that this type of attributes are useful to identify different family of applications, in contrast to the rest of datasets that obtained almost null GMs.

Table 1. Top ten predictors for the different datasets.

		Top ten predictors
SIZES		
	1st	Root Mean Square of packet sizes in both directions.
	2nd	Size of the 1st packet exchanged.
	3rd	Maximum packet size in the 1st direction.
	4th	Root Mean Square of packet sizes in the 1st direction.
	5th	Maximum packet size in both directions.
	6th	Mean of packet sizes in the 1st direction.
	7th	Size of the 2nd packet exchanged.
	8th	Phase of the 1st FFT component on sizes in the 1st direction
	9th	Size of the 5th packet exchanged.
	10th	Mean of packet sizes in both directions.
IATS		
	1st	Phase of the 1st FFT component on IATs in both directions.
	2nd	Percentage of IATs in the 1st direction.
	3rd	Maximum IAT in both directions.
	4th	Root Mean Square of IATs in both directions.
	5th	Root Mean Square of IATs in the 1st direction.
	6th	Module of the 5th FFT component on IATs in both directions.
	7th	Phase of the 1st FFT component on IATs in both directions.
	8th	IAT of the 2nd packet.
	9th	Module of the 3rd FFT component on IATs in the 1st direction.
	10th	Module of the 2nd FFT component on IATs in the 1st direction.
WINSIZES		
	1st	Root Mean Square of window sizes in both directions.
	2nd	Window size of the 1st packet exchanged.
	3rd	Mean of window sizes in the 1st direction.
	4th	Phase of the 1st FFT component on Window sizes in the 1st direction.
	5th	Maximum window size in the 1st direction.
	6th	Maximum window size in both directions.
	7th	Root Mean Square of window sizes in the 1st direction.
	8th	Window size of the 2nd packet exchanged.
	9th	Percentage of window sizes in the 1st direction
	10th	Phase of the 3rd FFT component on Window sizes in the 1st direction.
OTHERS		
	1st	Destination port number.
	2nd	Source port number.
	3rd	Number of packets with packet sizes between 128 and 64 bytes.
	4th	Number of packets with packet sizes between 10 and 20 kilobytes.
	5th	Number of packets with packet sizes greater than 64 bytes.
	6th	Direction of the 5th packet.
	7th	Percentage of packets in the 1st direction.
	8th	Direction of the 2nd packet.
	9th	Percentage of packets in 1st direction.
	10th	Number of packets with packet sizes greater than 20 kilobytes.

4.2. Final Subset

In this section we present and discuss the results obtained from applying our FS framework to the WHOLE dataset. Table 2 contains the top ten predictors selected by our FS method, and the performance metrics obtained when a Decision Tree is trained including each one.

Observing Table 2, we find that the five top predictors yielded better outcomes that the original dataset according to OA and GM, on contrast to BA that was slightly lower than using all attributes. The highest OA was reached when the 9th predictor was included in the predictive model identifying accurately the 98.5 % of connection flows. In the instance of BA, the best value was obtained when eight attributes are selected, although subsets achieved the same BA as the whole dataset when more than five predictors were selected. While on GM, the best performance was achieved when six or seven attributes are selected for training. GM accomplished 84 % for these subset sizes, meanwhile the whole datasets got a score of 73.5 %. The notable increase in this performance metric reveals that a smaller dataset better identifies a range of diverse applications, providing better outcomes in presence of Class Imbalance.

Table 2. The ten best-scored predictors and performance metrics for the WHOLE dataset.

Top ten predictors		OA	BA	GM
1st	Destination port number.	.265	.209	.539
2nd	Size of the 1st packet exchanged.	.829	.848	.607
3rd	Source port number.	.961	.969	.697
4th	Maximum packet size in the 1st direction.	.956	.989	.720
5th	Window size of the 1st packet exchanged.	.983	.995	.824
6th	Root Mean Square of packet sizes in both directions.	.981	.996	**.840**
7th	Phase of the 1st FFT component on packet sizes in the 1st direction.	.981	.996	**.840**
8th	Maximum packet size in both directions.	.980	**.997**	.817
9th	Phase of the 1st FFT component on Window sizes in the 1st direction.	**.985**	.996	.820
10th	Maximum window size in both directions.	.984	.996	.811
Whole dataset		.979	.996	.735

Through this experiment, we have probed that our FS framework is able to reduce the training subsets increasing some performance metrics for early NTC. We also found that the combination of predictors computed using different parameters from packet headers yielded much better results than independently using them. Note that the collection of top ten predictors (Table 2) includes source and destination port numbers (1st and 3rd). As aforementioned, these parameters are susceptible to evasion, since port numbers are a configurable parameter by users or applications. Below, we present the results of an experiment during which port evasion was simulated to assess performance losses due to this effect. In addition to port numbers, the top ten subset contains five predictors computed from packet sizes (2nd, 4th, 6th, 7th and 8th) and three Window-size related attributes (5th, 9th and 10th).

4.3. Masking Connections Behind Random Port Numbers

Finally, we provide a minor experiment to probe that ML-based traffic classifiers are susceptible to port evasion when port numbers are included in the predictive model. For this purpose, we have selected the subset with six features from Table 2, and port evasion was simulated by randomly modifying the source and destination port numbers for specific percentage of samples. Fig. 4 contains the results obtained in this experiment.

Generally, all performance metrics are negatively affected when the connections are masked behind other ports. When the percentage of masked connections increases, the performances of the classifier notably decreases. In the case of masking the 5 %, all metrics decreased in around 4 %. Through this simple experiments, we have probed as port evasion can decrease the performance of ML-based traffic classifiers when port numbers are included in the predictive model.

5. Conclusions

Through this work, we have presented an FS framework for selecting a reduced dataset for early NTC. Our framework combines two parallel ranking phases in which filter methods and wrapper strategies are employed to independently rank predictors according to their relevance in the predictive model. We have validated our FS scheme on different NTC datasets containing predictors computed from the different parameters in packet headers (such as: packet sizes, inter-arrival times, window sizes and so on).

As a result of our experiments, we have found that our FS framework is able to notably reduce the attribute space preserving and even improving the performances of the classifier. Additionally, we have analyzed the different collections of attributes to find out the predictive power of each one for NTC. Finally, we have applied our FS framework to a whole dataset that combines the former datasets. As result, we found that combining predictors computed from different network parameters provides better results than employing them separately. As source and destination port numbers were ranked as one of the most relevant attributes, we performed a minor experiment to assess the effect of port evasion on classifier performances. Our results reveal that port evasion decreases the classifier accuracy, and it should be considered when port numbers are included in the predictive model.

Acknowledgements

This work has been partially funded by the Ministerio de Economía y Competitividad del Gobierno de España and the Fondo de Desarrollo Regional (FEDER) within the project "Inteligencia distribuida para el control y adaptación de redes dinámicas definidas por software, Ref: TIN2014-57991-C3-2-P", in the Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento. Finally, we would like to thank the ISP for the real network traffic captures and the resources shared with us for this work.

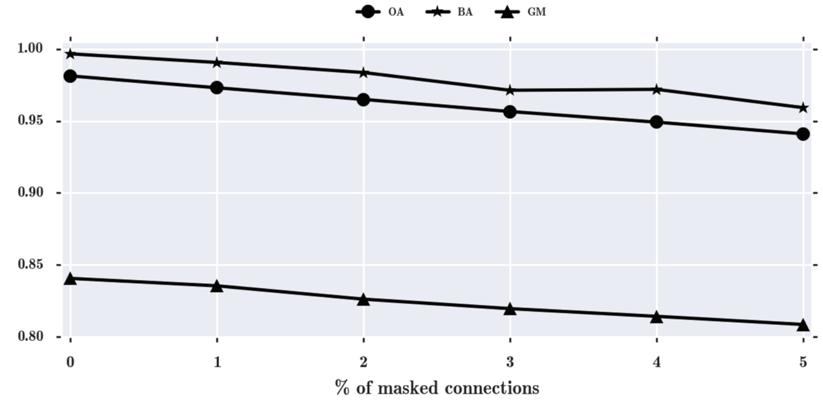


Fig. 4. Effect of port evasion on model performances.

References

[1]. T. K. L. Hui, R. S. Sherratt, D. Díaz, Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies, *Futur. Gener. Comput. Syst.*, Vol. 76, 2017, pp. 358-369.

[2]. T. Nguyen and G. Armitage, A survey of techniques for internet traffic classification using machine learning, *IEEE Commun. Surv. Tutorials*, Vol. 10, Issue 4, 2008, pp. 56-76.

[3]. A. Fahad, Z. Tari, I. Khalil, A. Almalawi, A. Y. Zomaya, An optimal and stable feature selection approach for traffic classification based on multi-criterion fusion, *Futur. Gener. Comput. Syst.*, Vol. 36, 2014, pp. 156-169.

[4]. Z. Liu, R. Wang, M. Tao, X. Cai, A class-oriented feature selection approach for multi-class imbalanced network traffic datasets based on local and global metrics fusion, *Neurocomputing*, Vol. 168, 2015, pp. 365-381.

[5]. H. Shi, H. Li, D. Zhang, C. Cheng, W. Wu, Efficient and robust feature extraction and selection for traffic classification, *Comput. Networks*, Vol. 119, 2017, pp. 1-16.

[6]. M. Shafiq, X. Yu, A. Kashif, B. Hassan, N. Chaudhry, D. Wang, A machine learning approach for feature selection traffic classification using security analysis, *J. Supercomput.*, Vol. 74, Issue 10, 2018, pp. 4867-4892.

[7]. H. Shi, H. Li, D. Zhang, C. Cheng, X. Cao, An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification, *Computer Networks*, Vol. 132, 2018, pp. 81-98.

[8]. L. Bernaille, R. Teixeira, K. Salamatian, Early application identification, in *Proceedings of the ACM Conex. Conference*, 2006, p. 6:1-6:12.

[9]. H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 27, Issue 8, Aug. 2005, pp. 1226-1238.

[10]. D. Lin, X. Tang, Conditional Infomax Learning: An Integrated Framework for Feature Extraction and Fusion, *Springer*, Berlin, Heidelberg, 2006, pp. 1531-1555, 2004, pp. 68-82.

[11]. F. Fleuret, Fast binary feature selection with conditional mutual information, *J. Mach. Learn. Res.*, Vol. 5, 2004, pp. 1531-1555.

[12]. A. Jakulin, Machine learning based on attribute interactions, PhD Thesis, *University of Ljubljana*, 2005, pp. 1-252.

[13]. R. Battiti, Using mutual information for selecting features in supervised neural-net learning, *IEEE Trans. Neural Networks*, Vol. 5, Issue 4, 1994, pp. 537-550.

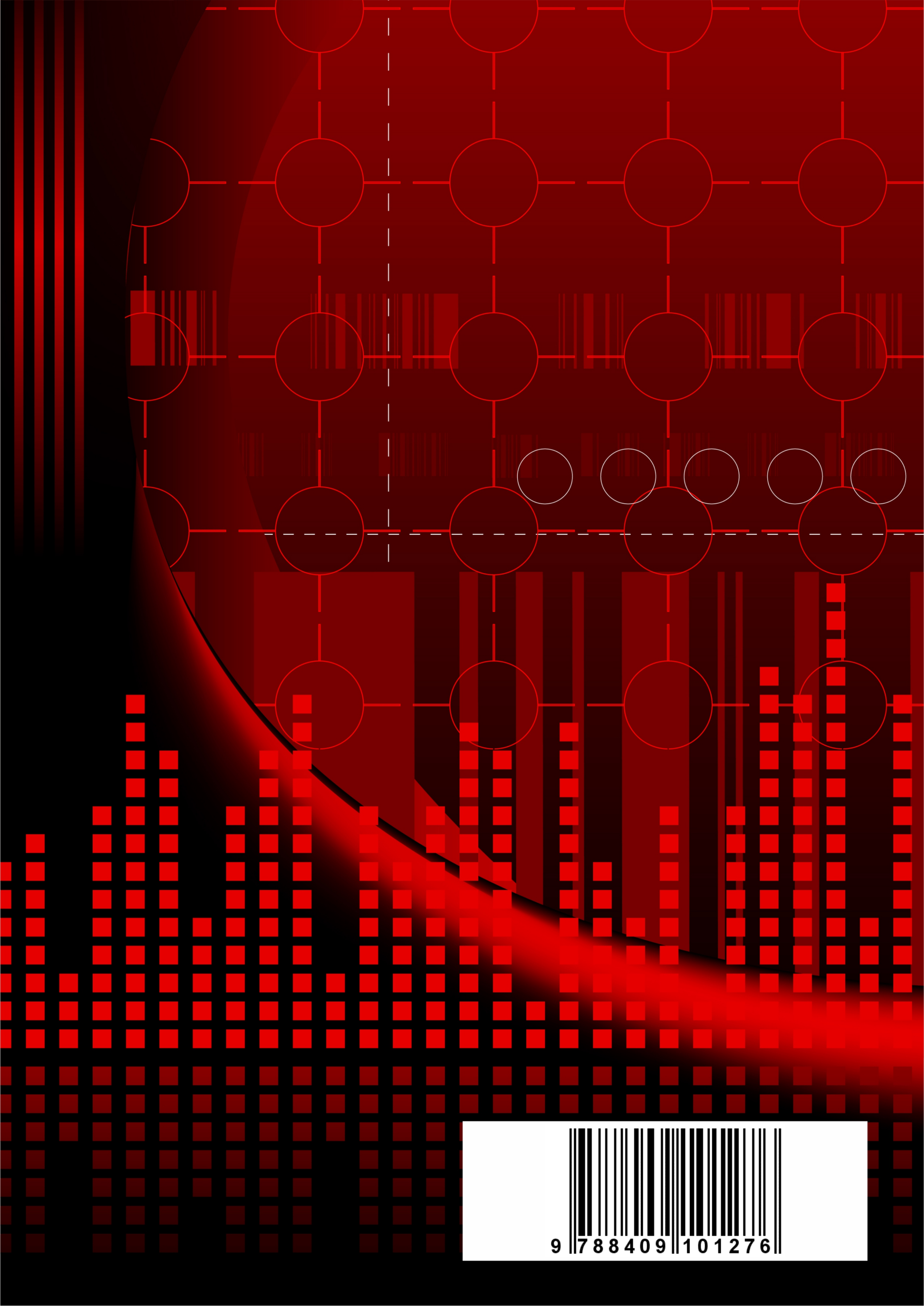
[14]. P. E. Meyer, C. Schretter, G. Bontempi, Information-theoretic feature selection in microarray data using variable complementarity, *IEEE J. Sel. Top. Signal Process.*, Vol. 2, Issue 3, 2008, pp. 261-274.

[15]. H. H. Yang, J. Moody, Data visualization and feature selection: new algorithms for nongaussian data, *Adv. Neural Inf. Process. Syst.*, Vol. 12, Issue Mi, 1999, pp. 687-693.

[16]. D. D. Lewis, Feature Selection and Feature Extraction for Text Categorization, in *Proceedings of the Workshop on Speech and Natural Language (HLT'91)*, 1992, pp. 212-217.

[17].S. E. Gómez, B. C. Martínez, A. J. Sánchez-Esguevillas, L. Hernández Callejo, Ensemble network traffic classification: Algorithm comparison and novel ensemble scheme proposal, *Comput. Networks*, Vol. 127, 2017, pp. 68-80.

[18]. S. E. Gómez, FCBF Module, 2018, https://github.com/SantiagoEG/FCBF_module



9 788409 101276