Sensors & Transducers, Vol. 263, Issue 4, December 2023, pp. 36-44

www.sensorsportal.com

Sensors & Transducers

Published by IFSA Publishing, S. L., 2023
http://www.sensorsportal.com

IMU Data-based Recognition for Sports Exercises: An Enhanced Distance Optimization Approach for Repetition Counting across Activities

[1, *] Pascal Krutz, [1] Matthias Rehm, [1] Holger Schlegel, [2] Justyna Patalas-Maliszewska and [1, 3] Martin Dix

[1] Chemnitz University of Technology, Institute for Machine Tools and Production Systems, Reichenhainer Strasse 70, 09126 Chemnitz, Germany
[2] University of Zielona Góra, Institute of Mechanical Engineering, 65-417 Zielona Gora, Poland
[3] Fraunhofer Institute for Machine Tools and Forming Technology IWU, Reichenhainer Strasse 88, 09126 Chemnitz, Germany
E-mail: pascal.krutz@mb.tu-chemnitz.de

Received: 4 September 2023 Accepted: 1 December 2023 Published: 20 December 2023

Abstract: In the field of human activity recognition (HAR), inertial measurement units (IMUs) are a commonly used method to record movement patterns. In the study presented in this paper, IMU captured seven different sports exercises performed by 21 participants. In the preliminary data analysis phase, an exercise classification was conducted using the Long Short Term Memory (LSTM) Network and Temporal Convolutional Network (TCN). The LSTM achieved an accuracy rate of 94.2 % for training and 90.8 % for testing. Similarly, the TCN demonstrated rates of 95.5 % for training and 91.6 % for testing. The subsequent stage was centered on quantifying the number of completed repetitions. A distance value was derived which showed promising results for exercise-independent counting without the need for manual feature selection. For further improvement, a range-to-mean ratio of the standard deviation was calculated and used for feature selection. Combined with a local extrema analysis of the modified distance values the accuracy of counting repetitions was significantly improved, especially for exercises that show irregularities in the signal course.

Keywords: Human activity recognition, Inertial sensors, Artificial neural networks, Repetition counting, Distance, Local extrema analysis

1. Introduction

Inertial Measurement Unit (IMU) sensors, in combination with broadband technologies and high-performance algorithms, offer great potential in the detection, monitoring and improvement of sports exercises. Especially in times of a lack of trained specialists and the demands resulting from the increasing individualization of personal leisure time, digital therapy and training methods are needed for both the physiotherapy and fitness markets. Typically, methods of supervised learning are predominantly used in the field of human activity recognition (HAR). The most common approaches in the literature are those based on the Support Vector Machine [1-3], followed by Naive Bayesian [4, 5] and k-nearest Neighbour [6]. Less common are deep learning approaches using Convolutional Neural Networks (CNN) [7] and Long Short Term Memory (LSTM) Networks [8]. Unsupervised learning methods are the exception and have so far only been used to count the number of repetitions, for example in tennis strokes

[9]. The work is based on preliminary work [10, 11] on supervised learning of three sports exercises, where high accuracy was achieved with both the classifiers and the neural networks used. The workflow included data acquisition and preprocessing, the training of Artificial Intelligence (AI) models, the optimization of model parameters and a final evaluation of model performance.

This study aims to choose, optimize and compare model structures and parameters for supervised learning using an extended database. The database consists of recordings of seven different sports activities performed by 21 participants. Another focus of this paper is the development of a universal counting algorithm suitable for counting repetitions of any sports activity.

2. Data Generation and Preprocessing

Within the study with 21 participants, 7 different exercises were completed at one sports device as successive set. The execution of the different activities is shown in Fig. 1.

Fig. 1. recorded exercises on the sports device, 1-support, 2-squats, 3-dips, 4-lunges, 5-pull ups, 6-sit ups, 7-push ups.

The exercises support, squats, dips, lunges, pull-ups, sit-ups and push-ups were recorded as continuous recordings with IMUs at the positions of chest, hand and foot. With a set sampling rate of 30 Hz, the quantities of linear axial accelerations, rotation rates, orientations (Euler angles, quaternions, magnetic flux densities), absolute pressure and the Cartesian position in space were captured. The time series were labelled during the recording process by switching the subjects between pause and activity intervals using a wearable device. In total, around 230 minutes of exercise data were recorded, divided into 24 exercise sets, which also included intervals of stretching and loosening exercises, labelled as pauses. Signal interruptions that occurred were linearly interpolated during data pre-processing and the time series of three IMUs were synchronized. The distribution of the generated motion data to the different classes is shown in Table 1. As can be seen from this, about half of the total amount of data belongs to the pause intervals. Despite the comparatively high volume of seemingly unnecessary data, it was fully utilized for model training to achieve the ability to distinguish between pause and active intervals.

Table 1. Distribution of classes in the used database.

Class	**Lunges**	**Dips**	**Pull-ups**	**Squats**
# Samples	38997	22675	21736	32288
percentage [%]	9.39	5.46	5.23	7.77
Class	**Push-ups**	**Pauses**	**Sit-ups**	**Support**
# Samples	21958	220290	32544	24816
percentage [%]	5.29	53.04	7.84	5.98

3. Training, Optimization and Comparison of KNN

The supervised models were trained in MATLAB R2022b. At first, the number of input variables was reduced to 36 by using only the linear accelerations, rotation rates, magnetometer data and Euler angles. Also, the data were divided by segmentation to be able to process the time series as sequential data in MATLAB. To ensure a diverse distribution of unbalanced data across the training and holdout validation sets, a segment length of 500 samples was chosen (corresponding to approximately 16 seconds of workout time). The holdout ratio was designated as 0.3, resulting in the generation of 582 sequence blocks for training data and 249 blocks for validation data. Subsequently, a LSTM- and a CNN network were trained and an optimization of the hyperparameters was carried out. The CNN network can be described as a temporal convolutional network (TCN), which consists of several CNN blocks connected in series. The time series are scanned in the form of a receptive field, the extent of which is influenced, for example, by the number of blocks. [12] The most accurate models are compared in Table 2 with the accuracies achieved and the training times required. The TCN could be trained more than three times as fast as the LSTM with slightly better accuracy.

Fig. 2 shows the obtained confusion chart for validating the TCN. One possible reason for incorrect assignments after the optimization of the hyperparameters can be justified with the applied methodology for labelling the data. In this study, the

test persons independently initiate the beginning and end of an exercise interval by operating a wearable, implying that the activity ranges are not uniformly defined. This can be observed in the false positives and false negatives elements of the pause class, which contain significantly more samples than the other classes. This effect was already reduced by the implementation of a three-second countdown between the actuated switch-over point and the beginning of the exercise execution. However, deviations remain, especially in exercises where the starting position can only be assumed after the wearable has been activated (e.g., pull-ups and dips).

Table 2. Comparison of the trained LSTM and TCN structures with the highest accuracies.

Model	Normalized training time	Accuracy		Worst F1-score (class)
		Training	Validation	
LSTM	100 %	94.2	90.8	83.8 (pull-ups)
TCN	28 %	95.5	91.6	83.4 (pull-ups)

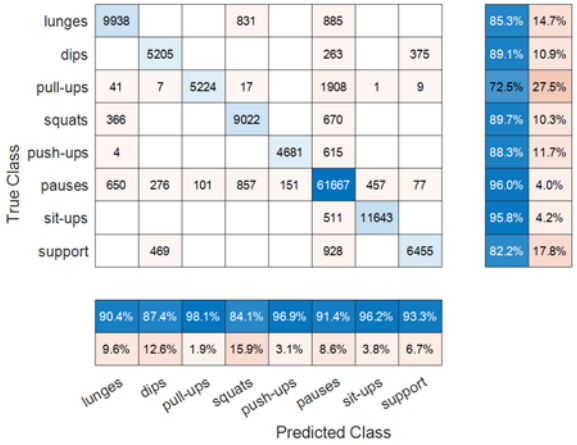

Fig. 2. Confusion matrix for applying the trained TCN on the validation data.

Furthermore, it must be taken into consideration that the subjects were largely unrestricted in their exercise execution. Different variations were tolerated, e.g., squats with stretched or bent arms and simplified executions (e.g., pull-ups in inclined posture). Even though some exercises have a high degree of similarity (e.g., lunges/squats, dips/support), inter-activity mismatch is limited. Fig. 3 shows the TCN model predictions for a section of the validation dataset. There are occasionally false assignments, which were filtered out by post-processing. In the smoothed prediction that is also shown, a sliding window of 50 samples was used to determine the dominant class in each window, which slightly improves the overall accuracy of the validation by additional 0.4 %.

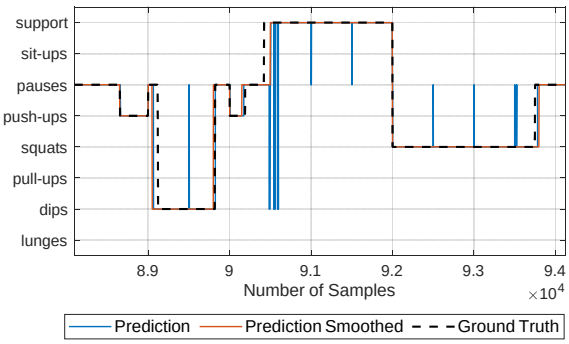

Fig. 3. Time course of the TCN model predictions for a segment of the validation data, comparing model prediction (non-smoothed/smoothed) and ground truth.

4. Repetition Counting

The execution of repetitive exercises is visible as a periodic progression of individual measured variables, which differ in type and curve progression depending on the respective type of exercise. To reduce the data load and to improve the generalizability of the implemented algorithms, the strategy was to use only variables that are independent of the orientation and position of the subjects. Therefore, only the values of the axial accelerations and rotation rates were considered. Consequently, 18 measurement variables were used to determine the number of completed repetitions. As the value ranges of the inertial measurement variables are widely distributed, symmetrical normalization was applied to all variables in the interval [-1...+1].

To enable an evaluation of the counting algorithm without the influence of an upstream model for classification, ground truth labels were used for the development of the counting algorithm. It is also important to mention that only the activity intervals in their chronological order from the recording were used as the data basis for the analysis. This order of data would also be present in a potential interconnection of the model and counting algorithm when the latter receives sequences of predictions by the classification model. Consequently, 24 sets in total were analyzed in series by the counting algorithm.

The special requirement in this work was to find a parameter that shows an oscillating course for a variety of different sports exercises. Such an exercise-independent parameter can be calculated with the distance *Dis* between a pattern segment *A* and every segment *B* of the respective activity as described in equations (1) and (2). *C* is the difference between *A, B* and *Dis* represents the summed amounts of *C*. *A* and *B* are matrices with the dimension *m* x *n*, where *m* defines the number of samples, also called segment length and *n* is the number of features for one segment.

$$C = A - B = \begin{bmatrix} a_{11} - b_{11} & \cdots & a_{1n} - b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} - b_{m1} & \cdots & a_{mn} - b_{mn} \end{bmatrix}, (A, B \in \mathbb{R}^{m \times n}), \quad (1)$$

$$Dis(A,B) = \sum_{i=1,j=1}^{i=m,j=n} |c_{ij}| = \sum_{i=1,j=1}^{i=m,j=n} |a_{ij} - b_{ij}| \quad (2)$$

The pattern segment was selected in the centre of each exercise interval to avoid the influence of transition areas between other classes because, as previously mentioned, it was not possible to precisely define the start and end points of an exercise execution. After the segment length was set to 30 samples, the distances for all segments of an activity interval were calculated with a step size of one sample. The obtained distance values were smoothed by a moving average filter with a bandwidth of 15. Additionally, the median was calculated for each interval of smoothed distance values. The courses of unfiltered/filtered distances and the median values are presented in Fig. 4 for an exemplary execution of six pull-ups. The oscillating distance curves represent the number of completed repetitions. A characteristic attribute is that the distance increases significantly at the borders of an interval to be examined. In addition, the distance in the area of the pattern segment becomes zero. To count the repetitions, the number of intersections at the falling edge of smoothed distance values and the median were analyzed (red crosses in Fig. 4).

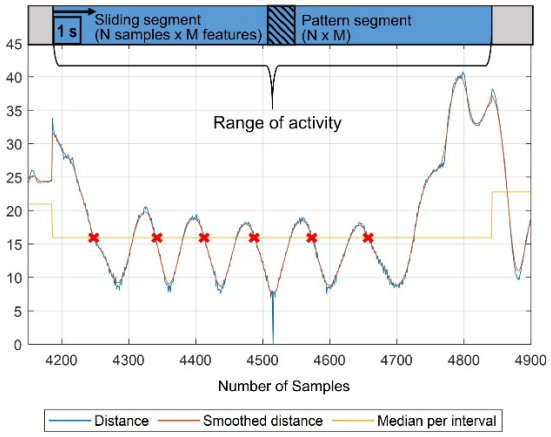

Fig. 4. Calculated distance values for the execution of six pull-ups, marking the counted intersections with the median at the falling flank (N = 30, M = 18).

To evaluate the accuracy of the counting algorithm, the deviation rate and the class-wise deviation rate were used, as described in (3) and (4). The deviation rate δ is signed and thus enables the assessment of whether too many or too few repetitions Rc have been counted in an exercise set. When determining the class-wise deviation rate δ_{class} for n records, the amounts of the individual deviation rates are calculated. All deviation rates are relativized by multiplying with the true number of repetitions Rt.

$$\delta = \frac{Rc-Rt}{Rt}, \quad (3)$$

$$\delta_{class} = \frac{\sum_{i=1}^{i=n} Rt_i \cdot |\delta_i|}{\sum_{i=1}^{i=n} Rt_i} \quad (4)$$

To obtain a final value for the overall performance of the counting algorithm for a larger number of trials, δ_{SUM} was calculated with the weighted average of δ_{class}, considering the overall distribution of the m activity classes and shown in equation (5). There n_m is the number of samples of each activity class and n_A is the number of all activity samples.

$$\delta_{SUM} = \frac{\sum_{i=1}^{i=m} \delta_{class} \cdot n_m}{n_A} \quad (5)$$

Table 3 shows the results of experiments for optimizing the counting algorithm by varying the features and the segment length for distance calculation. First, the features were varied, using a segment length of 30 samples. The deviation rates listed (class-wise and weighted average) suggest that the accelerations (acc) have a significantly higher influence on successful repetition counting than the gyroscope data (gyro). The lowest δ_{SUM} was determined with a combination of rotation rates and accelerations and was therefore defined as the feature setting for further investigations. The lower part of Table 3 shows the results of varying the segment length, i.e., the number of samples used to calculate the distance. The range of samples was adjusted from 20 to 70, and the most optimal outcome was achieved using the previously employed 30 samples. There was a slight incremental enhancement of 0.7 % for δ_{SUM}, observed when utilizing a segment length of 50 samples.

Table 3. Results of optimization experiments, determine the deviation rates for the activities (1-support, 2-squats, 3-dips, 4-lunges, 5-pull ups, 6-sit ups, 7-push ups) and the overall deviation.

	δ_{class} in %							δ_{SUM} in %
	1	2	3	4	5	6	7	
variable feature selection, segment length = 30								
acc + gyro	21.2	2.8	19.3	15.0	9.1	2.9	7.9	**10.8**
acc	18.6	8.3	14.6	13.4	7.5	5.2	9.0	10.9
gyro	38.3	3.9	34.8	58.1	28.6	11.7	10.7	27.5
variable segment length, features: acc + gyro								
20	19.7	4.4	21.5	23.5	12.1	7.3	9.0	14.0
40	23.9	3.0	19.3	14.2	8.5	2.6	7.3	10.8
50	23.9	3.0	15.9	12.4	9.6	2.3	7.3	**10.1**
60	23.1	3.0	16.7	12.4	14.6	2.6	7.9	10.8
70	26.1	2.8	19.3	13.7	14.6	3.5	9.0	12.0

For the detailed consideration of the class-wise differences, these two mentioned tests are considered (highlighted grey in Table 3). In both cases, the δ_{class}

was most significant for the support class with 21.2 % and 23.9 % respectively. In contrast, the classes squats and sit-ups were the most reliably counted by far with δ_{class} between 2.3 % and 3 %.

To analyze possible causes for the partly very different values of δ_{class}, the deviations for each of the 24 recorded sets were considered for the upper, highlighted parameterization in Table 3. Occurring δ with a modulus greater than 0.3 were examined in more detail. Typical scenarios for exceeding and falling below the true number of repetitions were identified from different recordings and are presented below.

In Fig. 5 top left, the performance of squats is shown where the exercise execution was varied within the interval. Consequently, the median value in the immediate vicinity of the pattern segment is notably elevated, leading to a lack of intersections with falling edges within this specific area. Similar characteristics are observed in the upper right portion of the figure, where multiple repetitions of dips toward the end of the interval were also not detected any longer. In the lower part of Fig. 5, too few dips were recorded in the left example. Here, the recognizable periodic progression of the distance values was comparatively short, which suggests only a short execution within the labelled active interval. In the fourth example (bottom right), the two completed repetitions of the support exercise cannot be identified in the progression. One interpretation is that the exercise was performed very irregularly, it is also possible that the test person was physically overloaded. Finally, it can be stated about the evaluation of the faulty intervals that the implemented methodology is particularly susceptible to errors when exercises are performed irregularly. The observation that irregular execution is caused by physical overstress among the test subjects is supported by a dependable analysis of relatively simple exercises, like squats and sit-ups. A special status is occupied by the lunges, whose execution (step position of left and right foot alternately or identical for several repetitions) was not exactly defined, which indicates that the sensor attached to only one foot does not contribute to a periodic progression of the distance values for all movements.

Fig. 5 indicates that achieving an accurate counting of repetitions in an interval is critically dependent on a consistent execution of the exercise, which is not always given in reality. Further refinement of the current approach is required to count varied exercise performances in an interval. The analysis of axial sensor data reveals that only a subset of inertial measurement variables is affected by variations in the exercise execution. For example, in Fig. 6 two different acceleration signals are shown while performing 14 squats (example top left in Fig. 5). While the z-acceleration of the chest sensor *za_1* shows a homogeneous course, clearly different movement patterns were recorded for the x-acceleration of the hand sensor *xa_2*, which can be justified by changing the guidance of the arms during exercise execution. In the context of universal exercise monitoring, these in-between executions should also be counted correctly. Highly irregular signals have to be identified depending on the exercise and should be excluded for an improved distance calculation. The standard deviation was determined for the measurements of *za_1* and *xa_2*. The window size for calculation was set to 90 samples with an overlap of 50 % to have the best representation of the differences between the signals.

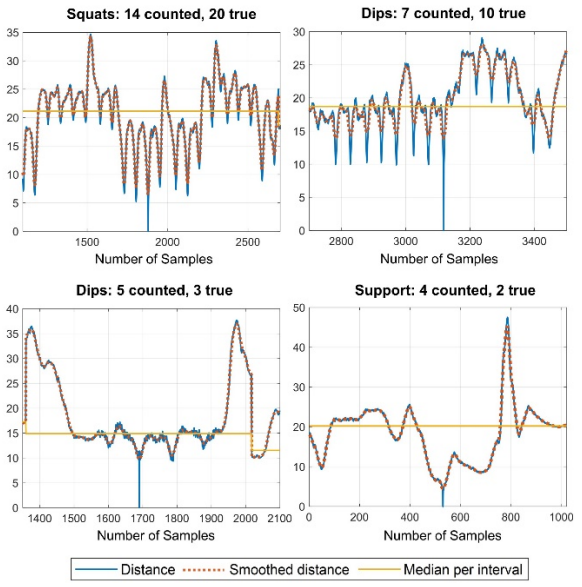

Fig. 5. Progressions of the distance values for incorrectly counted repetitions, top: lower deviation to the true number of repetitions ($\delta < 0$), bottom: exceeding true repetitions ($\delta > 0$); respective indication of the type of exercise with true and counted repetitions.

In the lower part of Fig. 6, the calculated standard deviation of *xa_2* shows a much more fluctuating course compared to *za_1*. In the context of automatic sorting of the measured variables according to their homogeneity, the range-to-mean ratio k_{R2M} was introduced for the description:

$$k_{R2M} = \frac{\max(data) - \min(data)}{mean(data)} \tag{6}$$

Calculating k_{R2M} for the standard deviation of the two measured variables to be compared yields around 0.3 for *za_1* and 1.7 for *xa_2*. The significantly larger value for *xa_2* describes the fluctuating signal course, which means that k_{R2M} is well suited for the interval-wise assessment of a measured variable. If, in comparison, only the standard deviations of the measured variables were determined over the entire interval, they would differ significantly less from each other with 0.0687 for *za_1* and 0.0717 for *xa_2*.

To examine the trend in distance values for reduced feature settings, k_{R2M} of all 18 features were determined for each interval and the features were sorted by descending k_{R2M}. In Fig. 7, the considered

squat interval is shown comparatively for 18 features and a feature set reduced to 12 and 6, respectively, where the features with the lowest k_{R2M} were selected for the reduction. Also plotted are the determined repetitions by the number of intersections with the median at the falling flank. It is apparent that with an increasing exclusion of features with a high k_{R2M} factor, a homogenization of the distance progression is achieved. As a result, the repetitions located in the middle section are aligned to the level of the median and are consequently captured via the intersections that occur. Thus, the delta to the 20 performed squats is reduced to +/- 1 recorded repetition.

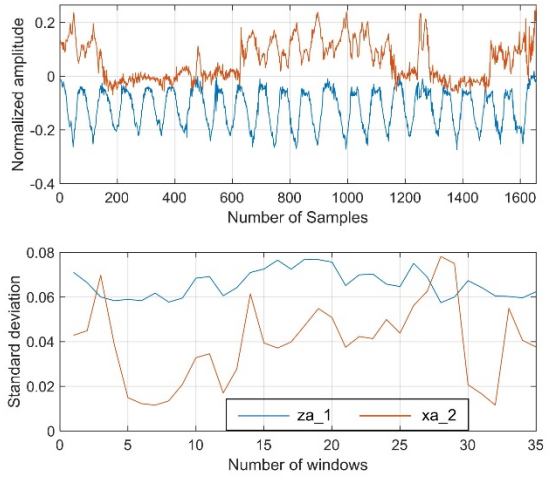

Fig. 6. Comparing the acceleration signals for a miscounted squat interval (top) and their standard deviations for a window size of 90 samples and 50 % overlap (bottom).

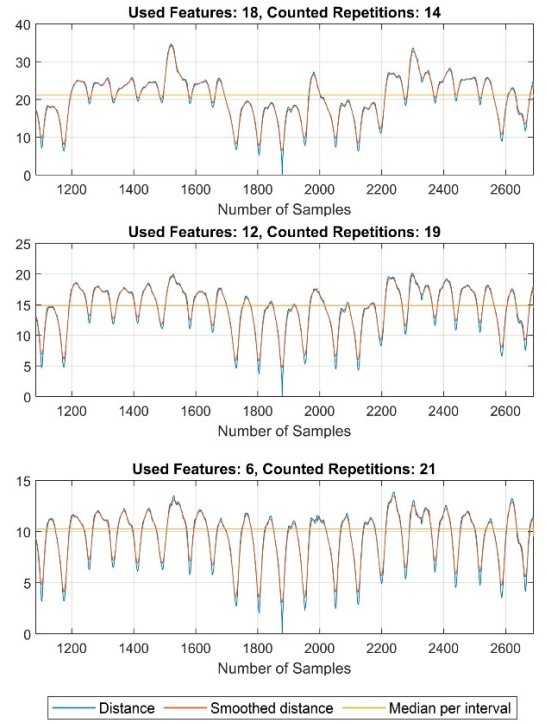

Fig. 7. Homogenizations of the distance with feature reduction based on an ascending sorting according to its range-to-mean ratio.

Table 4 presents the mean deviation rates for the reduction of features from 18 to 6 stepwise. The improvement in performance for counting squats (2) can be achieved except for $N_{Features} = 15$. Likewise, the deviation rates of sit ups (6) and push ups (7) remain at a low level. However, a significant decrease in counting performance is observed for the other exercises, which also increases δ_{SUM}. In the set-by-set analysis of the incorrect classes, it was recognized that the deviations were almost entirely due to exceeding the true repetitions. The described smoothing of the distances causes all motion patterns to be homogenized. However, the homogenization does not always ensure that the calculated median is at an optimal level for counting the intercepts. This effect can also be observed in Fig. 5 and occurs frequently with poorly counted activities, as a result of too many intercepts being detected with the median.

Table 4. Deviation rates for a feature reduction based on an ascending sorting according to its range-to-mean ratio.

$N_{Features}$	δ_{class} in %							δ_{SUM} in %
	1	2	3	4	5	6	7	
18	21.2	2.8	19.3	15	9	2.9	7.9	10.8
15	23.1	3.2	22.3	16.5	11.6	3.2	7.3	12
12	28.4	1.4	17.2	25.6	14.1	2.9	7.6	13.9
9	28.4	2.5	17.6	23	21.6	4.1	6.8	14.5
6	37.1	1.9	21.9	26.9	24.6	3.8	8.8	17.3

At this stage of processing, it was checked whether other characteristic points in the course of the distance can be used as incremental conditions for counting. When considering the distance values, it is obvious to check the number of the local extrema points. MATLAB offers the possibility to evaluate the number of local minima points via the command *islocalmin*. [13] The number of local minima can be limited by the parameter minimum separation and was set to 20 samples for the present data, which requires at least 0.67 s in between two repetitions.

In addition to the locations identified as local minima, the prominence is also an output for each minimum. In topography, the prominence describes the independence of a summit. [14] In the present application, it is used to depict the relevance of one local minimum in comparison to other local minima. To calculate the prominence, it is necessary to identify adjacent intervals, which involves applying a horizontal auxiliary line to the local minimum. Then, in both directions, the intersection points of the auxiliary line are defined as boundaries of the upper and lower intervals. These can be intersections with other minima or, if none exist, the start/end points of the signal. Finally, the highest peaks in both intervals are determined, and the smaller one is defined as the reference level. The distance between the reference

level and the local minimum finally describes the value of the prominence.

Using the prominence metric allows for additional refinement in the identification of minima, potentially by setting a threshold that determines the minimum value the prominence must surpass, thus further constraining the number of identified minima. For this purpose, the approach of a scalable limit was used by scaling the maximum of the prominences of an interval with a value smaller than 1. For a scaling factor of 0.3, the local minima shown in red in the following Fig. 8 were obtained. These represent the iterative execution of dips in the example shown. In contrast, without the limiting prominence threshold, too many minima (shown in blue) are detected and would not result in a good counting performance. In other words, by limiting with a scale factor of 0.3, the number of minima counted was reduced from 9 to 4 in the shown dip exercise section of Fig. 8. Based on the preliminary considerations, the counting performance for different parameterizations of the prominence scaling factor in combination with varying feature settings, based on the preselection by increasing k_{R2M}, was subsequently tested. Table 5 summarizes the results for the obtained δ_{SUM}.

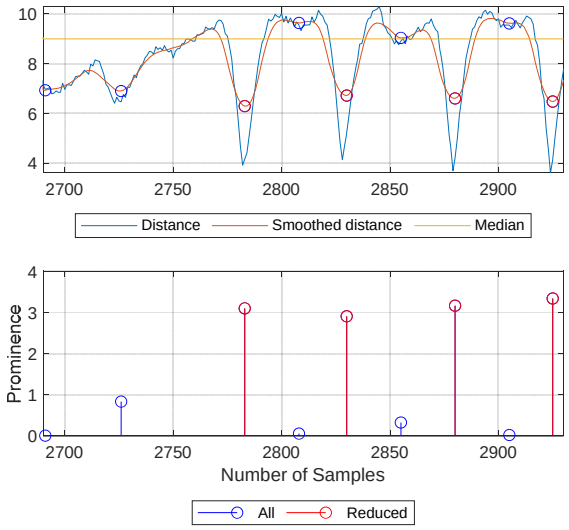

Fig. 8. Segment of a dip interval with feature-reduced distance (NFeatures = 9), indicated local minima and their according prominence (minSeparation = 20) and reduced number of minima with prominence >= 0.3·max (prominence).

First, the impact of the scaling factor was assessed in the range of 0.2...0.5 for $N_{Features} = 9$. It was detected that a value of 0.2 was the most effective setting, which was also verified for $N_{Features} = 12$. When the value was < 0.2, too many minima were used for counting, while with a scaling factor > 0.2, there was a tendency towards too few counted repetitions. Once the optimal scaling factor was determined, various feature settings with a size of 6 to 18 features were tested. 18 features correspond exactly to the size of the measurement variables selected at the beginning, without additional feature selection based on k_{R2M}. To ensure comparability, the original feature setting, including axial rotation rates and accelerations, was retained.

Table 5. Overall deviation rates for different scale factors and feature sizes.

Scale factor	$N_{Features}$	δ_{SUM} in %	Comments
0.1	9	16.1	Testing different scale factors for constant feature setting
0.2	**9**	**8.8**	
0.3	9	10.1	
0.4	9	14.2	
0.5	9	20.3	
0.2	12	10.2	Validation of scale = 0.2 for another feature setting
0.3	12	10.9	
0.2	6	11.1	Testing different feature settings with constant scale factor
0.2	12	10.2	
0.2	15	8.7	
0.2	18	10	
0.3	15	14.3	NFeat = 15 showed unfavourable performance

The results showed reduced variability, hovering approximately at 10 %. Furthermore, testing 15 features revealed a remarkably low deviation rate of 8.7 %, in addition to the nine features that had been tested previously. Nevertheless, a different test revealed a substantially worse performance with 15 features and a scaling factor of 0.3. Consequently, it can be assumed that of the parameters tested, $N_{Features} = 9$ and a scaling factor of 0.2 give the most reliable results.

When considering the class-specific deviation rates, a comparable influence of the scaling factor on almost all classes was found, which is shown below in Fig. 9.

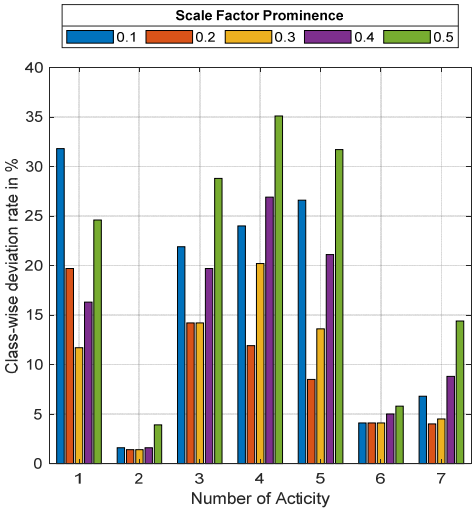

Fig. 9. Class-wise deviation rates for varying scale factor between 0.1 and 0.5.

The deviation rates show a minimum at a scaling factor of 0.2 to 0.3 and increase accordingly for other values. Squats and sit ups are the exercises that have the lowest deviation rates and consequently show a more robust behaviour with a wider range of scaling factor values. The impact of the number of features was also taken into account, but no distinct trends were evident in this analysis. It is reasonable to assume that with an increased quantity of input variables, discernible trends might emerge. Nonetheless, this aspect was not further explored to maintain consistency with the previous results for the sake of comparison.

5. Discussion and Conclusion

In the presented work, recorded IMU data consisting of seven sports activities were examined. The first step was to classify the completed exercises using supervised machine learning methods. Two different network structures (LSTM and TCN) were trained and the hyperparameters of the models were iteratively optimized. The TCN model gave a good performance with 91.6 % validation accuracy using significantly shorter training times. One way to improve the accuracy of the prediction is by post-processing the labels generated during recording. This will enable a more accurate determination of the neat duration of the exercise, as it has been shown that the dominant percentage of misclassifications is caused by confusion of the break class with other activities. For further investigations, the interconnection of several classification levels would be another possible approach to further increase the model accuracy. For post-processing of the model predictions, a smoothing algorithm was applied that determined the dominant prediction class using a filter width of 50 samples. Although the overall accuracy of the prediction was only slightly improved to 0.4 %, the described procedure can optimize the user experience in possible applications for real-time recognition.

The second crucial aspect of the study involved developing an algorithm to count completed exercise repetitions. By calculating distance values for a pattern segment defined within an exercise interval, the number of repetitions could be made detectable for all activities based on this one parameter, which made the class-specific analysis of individual signal variables unnecessary. To count the repetitions, the intersections of the distance values with the median of all distances of a single exercise phase were evaluated. For performance evaluation with different parameterizations (segment length, input features), the deviation rate was introduced. The implemented algorithm performed well when counting regular executions and less effectively if the distance of an interval does not show a homogeneous, periodic course. In order to accurately quantify these exercises, the measured variables were arranged based on their regularity using a range-to-mean ratio of their standard deviation. Subsequently, the distances were recalculated with reduced feature compositions. A significant improvement in counting performance was finally achieved by switching to the evaluation of local minima. Through additional parameters such as the prominence, the average total deviation rate could be reduced to 8.8 %. Nevertheless, even with the optimizations carried out, irregularly performed exercises cannot be counted correctly in every case. If, however, the uniform execution is observed within an interval, the optimized distance value method provides reliable counting results. The key advantage of the counting algorithm is its ability to load a new pattern segment as a reference for each new exercise interval, resulting in a highly robust algorithm capable of accommodating different people and their specific types of exercise execution.

Acknowledgements

Supported by:
ZIM Zentrales Innovationsprogramm Mittelstand
Federal Ministry for Economic Affairs and Climate Action
on the basis of a decision by the German Bundestag

This Project is supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) on the basis of a decision by the German Bundestag.

References

[1]. T. Kautz, et al., Activity recognition in beach volleyball using a deep convolutional neural network, *Data Mining and Knowledge Discovery*, Vol. 31, 2017, pp. 1678-1705.

[2]. H. Brock, Y. Ohgi, Assessing motion style errors in ski jumping using inertial sensor devices, *IEEE Sensors Journal*, Vol. 99, 2017, pp. 1-11.

[3]. I. Pernek, et al., Recognizing the intensity of strength training exercises with wearable sensors, *Journal of Biomedical Informatics*, Vol. 58, 2015, pp. 145-155.

[4]. D. Connaghan, et al., Multi-sensor classification of tennis strokes, in *Proceedings of the SENSORS Conference*, 2011, pp. 1437-1440,.

[5]. D. Schuldhaus, et al., Inertial sensor-based approach for shot/pass classification during a soccer match, in *Proceedings of the 21st ACM KDD Workshop on Large-Scale Sports Analytics*, 2015, pp. 1-4.

[6]. B. H. Groh, et al., Classification and visualization of skateboard tricks using wearable sensors, *Pervasive and Mobile Computing*, Vol. 40, 2017, pp. 42-55.

[7]. L. Jiao, et al., Multi-sensor Golf Swing Classification Using Deep CNN, *Procedia Computer Science*, Vol. 129, 2018, pp. 59-65.

[8]. A. Rassem, et al., Cross-country skiing gears classification using deep learning, *ArXiv Preprint*, 2017, arXiv:1706.08924,2017.

[9]. M. Kos, I. Kramberger, A wearable device and system for movement and biometric data acquisition for sports applications, *IEEE Access*, Vol. 5, 2017, pp. 6411-6420.

[10]. J. Patalas-Maliszewska, et al., Inertial Sensor-Based Sport Activity Advisory System Using Machine Learning Algorithms, *Sensors*, Vol. 23, Issue 3, 2023, 1137.
[11]. I. Pajak, et al., Sports activity recognition with UWB and inertial sensors using deep learning approach, in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'22)*, 2022, pp. 1-8.
[12]. Mathworks: Sequence-to-Sequence Classification Using 1-D Convolutions, https://de.mathworks.com/help/deeplearning/ug/sequence-to-sequence-classification-using-1-d-convolutions.html
[13]. Mathworks: Find Local Minima, https://de.mathworks.com/help/matlab/ref/islocalmin.html
[14]. J. Griffié, et al., Topographic prominence as a method for cluster identification in single-molecule localisation data, *Journal of Biophotonics*, Vol. 8, Issues 11-12, 2015, pp. 925-934.