Pascal KRUTZ[*]*, Matthias REHM*[*]*, Holger SCHLEGEL*[*]*,*
Martin DIX [*],[**]*

RECOGNITION OF SPORTS EXERCISES USING INERTIAL SENSOR TECHNOLOGY

Abstract

Supervised learning as a sub-discipline of machine learning enables the recognition of correlations between input variables (features) and associated outputs (classes) and the application of these to previously unknown data sets. In addition to typical areas of application such as speech and image recognition, fields of applications are also being developed in the sports and fitness sector. The purpose of this work is to implement a workflow for the automated recognition of sports exercises in the Matlab® programming environment and to carry out a comparison of different model structures. First, the acquisition of the sensor signals provided in the local network and their processing is implemented. Realised functionalities include the interpolation of lossy time series, the labelling of the activity intervals performed and, in part, the generation of sliding windows with statistical parameters. The preprocessed data are used for the training of classifiers and artificial neural networks (ANN). These are iteratively optimised in their corresponding hyper parameters for the data structure to be learned. The most reliable models are finally trained with an increased data set, validated and compared with regard to the achieved performance. In addition to the usual evaluation metrics such as F1 score and accuracy, the temporal behaviour of the assignments is also displayed graphically, allowing statements to be made about potential causes of incorrect assignments. In this context, especially the transition areas between the classes are detected as erroneous assignments as well as exercises with insufficient or clearly deviating execution. The best overall accuracy achieved with ANN and the increased dataset was 93.7 %.

1. INTRODUCTION

The exponential increase in data volumes, favoured by factors such as lower costs for end devices, storage media and servers, as well as the steadily expanding use of smart devices requires methods for automated data analysis (Brühl, 2019). One of such methods is supervised learning as a sub-discipline of machine learning (ML), in which knowledge is

---

[*] Professorship for Production Systems and Processes, Chemnitz University of Technology, Chemnitz, Germany, pascal.krutz@mb.tu-chemnitz.de, matthias.rehm@mb.tu-chemnitz.de, holger.schlegel@mb.tu-chemnitz.de
[**] Fraunhofer Institute for Machine Tools and Forming Technology IWU, Chemnitz, Germany, martin.dix@mb.tu-chemnitz.de

generated from labeled data sets and applied to previously unknown data. The described approach is also suitable for the human activity recognition (HAR), in particular for the recognition of sports exercises, which are characterised by a periodic execution of movement patterns. Inertial measurement units (IMUs) are especially suitable for the sensory detection of dynamic movements. The use of IMUs as a tracking method enables to record the movements of several people simultaneously in almost any environment. Compared to optical tracking methods, this allows a more robust data collection without the requirement for visual markers (Helten, 2013). With the inertial signal characteristics, statements can be formulated about the type and duration of the activity (type and length of the periodic signal patterns), as well as the speed of execution (signal frequency) and the intensity (signal amplitude) (Schuldhaus, 2019).

Movement classification of exercises in the fitness sector enables a better analysis, monitoring and correction of performed sports exercises. This favours new possibilities in the field of training plan development. An example is the application field of physiotherapy. Here the initial conditions of rehabilitation patients are precisely detectable through the additional use of sensors. Based on this, individual training plans can be created and training progress can be continuously measured without the need for the trainer's constant presence. As a result, health risks resulting from improper exercise are minimised and training potential can be better realised. An increased motivation caused by regular feedback during training is also an additional incentive for establishing tracking systems in the fitness sector (Schuldhaus, 2019). In the following, some recent work in the field of HAR is described in a comparative way.

As shown in Tab. 1 there is a wide range of implementations available, from data acquisition to data processing and classification. A frequently used development environment for the implementation of AI models is Python, due to its open source character. At research institutes, however, Matlab is a widespread development environment that also provides comprehensive toolboxes for machine learning, which are particularly suitable for a fast entry without in-depth programming knowledge.

The workflow in this work is developed uniformly in Matlab, from data acquisition to classification. A special feature of the measurement system used in this work is the additional detection of absolute position data. These can improve the accuracy of the classification, especially when performing exercises at several stations.

The main focus of the work described below can be summarised as follows:
– recording and pre-processing of inertial data during the execution of sports exercises;
– training of classifiers and artificial neural networks (ANN), iterative optimisation of model hyperparameters;
– performance comparison of data sets with varying numbers of subjects;
– analysis of the classification processes and interpretation of faulty assignments.

Tab. 1. Selection of current publications in the field of HAR, comparing the applied sensors and the achieved accuracies

Type of activity	Sensors used for detection	Accuracies achieved with applied AI	Ref.
event detection in rugby	position and inertial data, sensor placement on the upper back of the player	up to 0.972 (balanced) with classifiers	(Kautz, 2017)
automatic activity detection in beach volleyball, based on deep neural networks	IMU, attached to the dominant hand of a player	overall recall 0.99 overall precision 0.35, obtained with classifiers	(Kautz, 2017)
recognition of daily activities (washing, sitting, sweeping…)	4 IMUs, attached on chest, wrist, hip and foot	hierarchical classification system 0.891 (balanced)	(Schuldhaus, 2019)
recognition of soccer activities	accelerometer, gyroscope in the near of ankle and integrated in soccer shoe	combination of peak detection and type classification lead to 0.85 (balanced) for kick detection	(Schuldhaus, 2019)
classification of 42 gym exercises	tri-axial accelerometer at chest	up to 0.91 for overlapping data and separate LSTM-models for each muscle group	(Hussain et al., 2022)
HAR for five activities with thermal vision sensors	IoT thermal vision sensor captured movements for three different perspectives	neural networks (CNN integrated in LSTM network), obtained acc. up to 1	(Polo-Rodriguez et al., 2022)
recognition of seven different strokes in badminton	three acceleration/gyroscope sensors, attached to wrist, upper arm and the rackets grip	CNN lead to 0.99 with combination of accelerometer and gyroscope data	(Steels et al., 2020)
daily life activities	smartphone accelerometer, located in the right pocket	0.93 while using only two axes of accel-erometer, classification with multilayer neural network	(Javed et al., 2020)
walking activity identification	ankle-worn accelerometer and gyroscope, additional use of pulse sensor	0.97 overall accuracy distinguishing mimicked and real walking activities, based on CNN	(Chakraborty & Mukherjee, 2022)
recognition and counting of physical exercises	two smartwatches worn on wrist and ankle provided acceleration, gyroscope and orientation data	exercise recognition: 0.99; repetition counting: error of ±1 for 0.91 of sets; CNN each for classification and counting	(Soro et al., 2019)

2. METHOD

The performance of the ML models is evaluated with the usual metrics accuracy and F1 score. The accuracies were calculated for training and validation data. When validating the trained models, a confusion matrix as well as the trace of ground truth and prediction were plotted with Matlab®. The advantage of the confusion matrix is that it provides an overview of the relation between true and predicted classes. With the matrix, the class-related true/false positives and negatives can be determined, from which further performance values such as the F1 score are calculable. However, the confusion matrix does not provide any information about the temporal course of the assignment. For this reason, a trace function is additionally applied for showing the progression of ground truth and prediction over the ascending number of samples. This allows hypotheses for correct and incorrect assignments by including the experimental protocols. Fig. 1 shows an example of a generated performance plot for sports activity classification. The confusion matrix shows that there are no misclassifications between the activities, only between the noExercise-class and the three sports exercises. This can be seen in the strong occupancy of the major and weak occupancy of the minor diagonals of the Confusion Matrix, shown in Figure 1 above. Below, the comparison of ground truth, i.e. the actual time course of the individual states, with the predicted states shows the few differences between model and reality, especially for the squats.

The applied measurement system by Pozyx (developer kit) implies six stationary anchors, three mobile sensors, so called tags, and a local gateway for system controlling tasks. By distributing the anchors spatially around the training zone, the position of the individual tags in space is measured by trilateration, which describes the special characteristic of the employed measurement system. The individual tags periodically transmit signals with their ID. Therefore, the ultra wide band (UWB) range is used, which enables a high temporal resolution for precise distance measurements, even indoors. The anchors distributed in the room receive the ID message of the tag with a minimal time offset, which is in the nanosecond range. Based on a highly precise temporal synchronisation of the individual anchors, the gateway calculates the absolute positions of the tags in the room.

The measurement data are accessible in the local network using the MQTT protocol. The recording rate for exercise acquisition was set to 30 Hz per tag. In addition to the absolute 3D position, 3D acceleration, 3D rotation rate, 3D magnetic field strength, 3D euler angles, quaternions and absolute pressure are transmitted. This results in a total of 20 measured variables that are transmitted 30 times per second from each tag. The tags are attached to the chest, right hand and right ankle of the subjects.
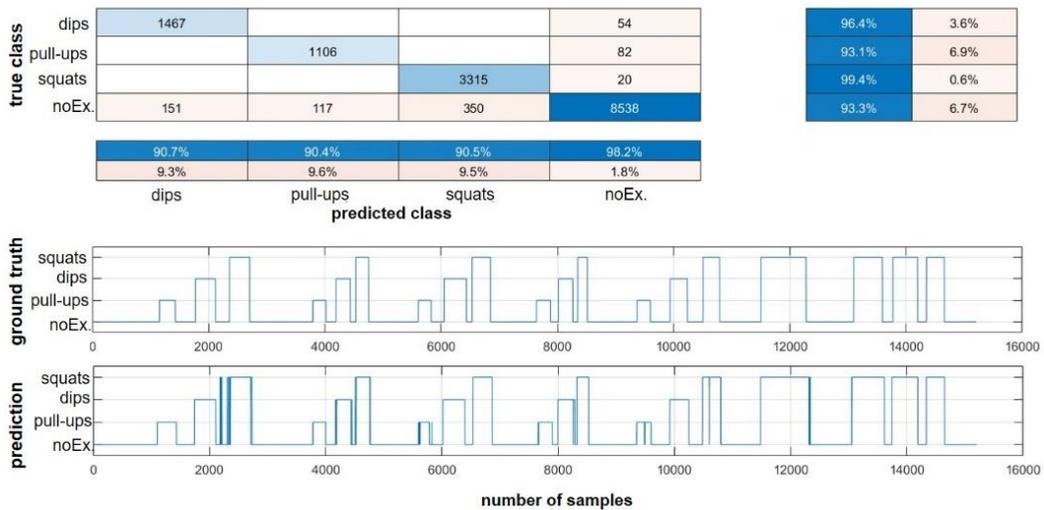
Fig. 1. Performance graph consisting of a confusion matrix and the ground truth as well as prediction curve for a multiclass problem consisting of dips, pull-ups, squats, and no exercise (noEx.)

Once the system is up and running, the accuracy of the 3D position detection is verified using static tests. The manufacturer's specifications were confirmed with an average deviation of 10 up to 30 cm. This is predominantly interesting if the exercises execution is located at several stations, for example in a fitness studio. A classification with positioning areas would then contribute to an improvement of the exercise classification, as exercises often are linked to specific sports equipment distributed in the tracking area.

In the described work, three activities should be recognized – named dips, pull-ups and squats. The exercises were recorded as combined sets with breaks in between. Each combined set has an average duration of about 60-75 s. In total, the sets were repeated 5 to 10 times per proband. Within the pauses, stretching and relaxing exercises were performed, to generate further movement patterns. Also the speed and orientation during the execution of the exercises were varied, which additionally contributes to a diversification of training and validation data. The subjects determined the number of repetitions according to their own fitness level and the experiment instructor documented course and number of finished exercises. As an example, the signal of one Euler angle (pitch) is shown in Fig. 2, linked with the performed sport exercises.

The recording of inertial data was done with Matlab® R2020b. Before the model training, a data preprocessing has to be carried out. As first step, a linear interpolation of lossy sensor data was implemented. Temporary signal losses during the data transmission can occur due to obstacles such as metallic objects, especially for the foot-mounted tag. The interpolation leads to a uniform sampling rate of the three sensors, which allows in the next step the synchronization of the data recorded by different sensor placements. The interpolated data are stored in an array of the dimension N x 61. N corresponds to the number of time stamps of a measurement, for which the synchronized sensor data are stored rows-wise. For three tags with 20 measurement variables in each tag and the conclusive label column, the total number of columns results to 61.
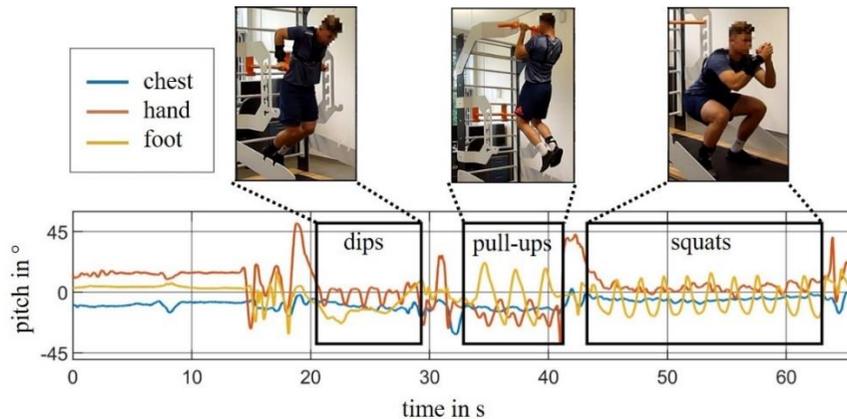
Fig. 2. Signal of the recorded pitch euler angles when performing four dips, three pull-ups and ten squats, sensors located at chest, hand and foot

For model training two different datasets were used – a Single Subject Dataset (SSD) as a compressed dataset with data recorded by just one subject and a Multiple Subjects Dataset (MSD) consisting of records from eight subjects. For the experimental search for suitable hyperparameters, the minor SSD dataset was used, since it allows a high number of tests with comparatively less computational effort. After optimizing the hyperparameters, the models were trained again with the MSD set to show effects in model behavior when increasing the data variety. The database was splitted according to the principle of holdout validation: Full sets of several subjects were used either as training or validation data. Here, a complete set can be understood as a continuous recording of a subject, in which for example pull-ups, dips and squats are performed one after the other with pauses in between. In average, one subject completed between eight to ten combined sets.

Fig. 3 shows the class distribution within the generated training and validation datasets. In addition to the no exercise class, the squats class is dominant cause this activity could be performed most easily by all subjects. In the conducted hyperparameter studies, two different basic structures were trained – ANN and classifiers. While ANN were trained with the preprocessed raw data, the classifiers were trained with additional extracted features and not with the original raw data. For this purpose, each measurement variable was segmented according to the sliding window method. This method describes the generating of overlapping windows with defined window width w and overlap o. Due to the segmentation, the original number of labels (N labels for N samples) gets reduced. The reduced number of samples respectively the sample rate is calculated by $N/(w-o)$. Subsequently, with the segmented data were calculated six different statistical values (mean, median, standard deviation, variance, skewness and kurtosis) for each window.
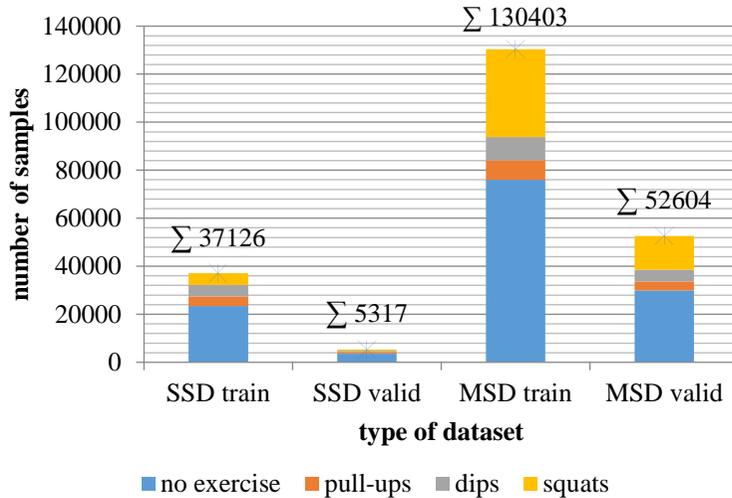
Fig. 3. Class distribution of the generated training and validation sets

3. RESULTS AND DISCUSSION

For the training of classifiers the classification learner toolbox of Matlab® was used, which provides 31 different types of classifiers. The initial configuration for generating sliding windows was set to a window size w of 30 and an overlap o of 10 samples. With this window parameterization, all available model types were trained and a reduction to four different model types with the highest validation accuracy was made. These four models were trained repeatedly and validated with the SSD data and different window parameterizations (w = [30, 50], o = [0, 10, 15, 25]). The achieved validation accuracies with the tested window parameterizations are shown in Fig. 4. The best results were obtained with the Gaussian Naive Bayes classifier, which provided a completely correct mapping of the validation data in one experiment.

ANN were also trained with the SSD set to determine suitable training settings and hyperparameters. First, the type of input layer was investigated, which can be defined as feature input or sequence input. After several trials, it was concluded that for the given data structure, the feature input layer achieved a higher validation accuracy (92.6 % vs. 90.6 %) after a significantly shorter training time (39 s vs. 980 s).

For subsequent investigations two different networks were used as examples, which are given in the Matlab® documentation (Sequence-to-Sequence Classification Using Deep Learning, n.d.) and which will be referred henceforth as LSTM and CNN. For CNN it is important to mention that it does not contain a convolutional layer due to the structure of the input data, but it has further typical components of the typical CNN (batch normalization layer, reluLayer). Both basic structures can be seen in Fig. 5.
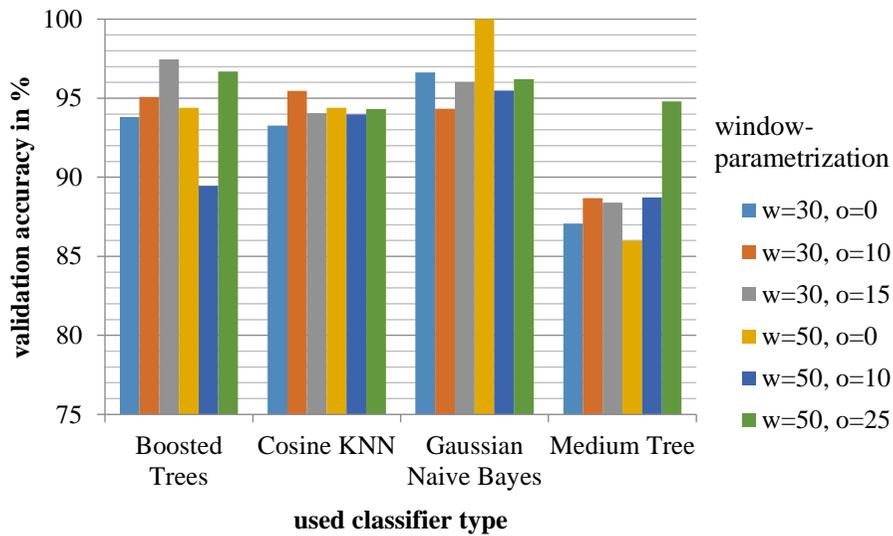
Fig. 4. Obtained validation accuracies of four classifier types with six different window parameterizations, small data set (SSD) used
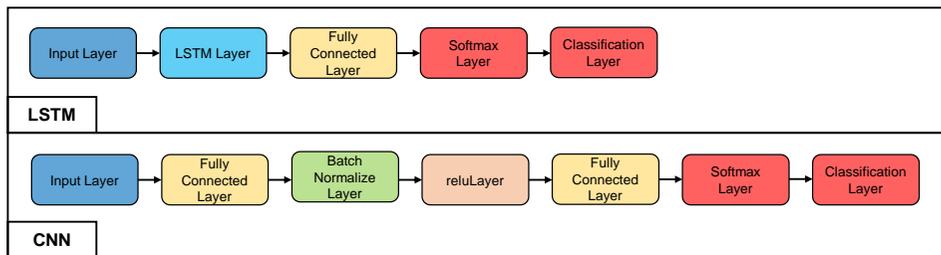

Fig. 5. Basic ANN structures adapted from Matlab® examples (Sequence-to-Sequence Classification Using Deep Learning, n.d.; Train Network with Numeric Features, n.d.)

As next step investigations followed on the type of input layer normalization and the number of training epochs that ANN passes through. With apply of the feature input layer, it was determined that the validation accuracies are already at a high level after three training epochs and that no further improvements are achieved by an increased number of epochs. Therefore, the number of epochs for subsequent experiments was set to three.

As another important training parameter, the parameter miniBatchSize was investigated. The hyperparameter describes the number of samples considered in one iteration to set the network weights of the neurons by applying the solver. Similar to the number of epochs, miniBatchSize has an effect on the required training time. An increased miniBatchSize leads to an exponential decrease of the training time. The parameter was varied in the range from 1 to 240 and the required training times as well as the achieved model accuracies were documented, which is shown in Fig. 6. It is evident that for a setting of miniBatchSize greater than 3, both networks provide high validation accuracies. For subsequent investigations, the values 15, 50 and 200 for minibatch were used to increase the statistical reliability.
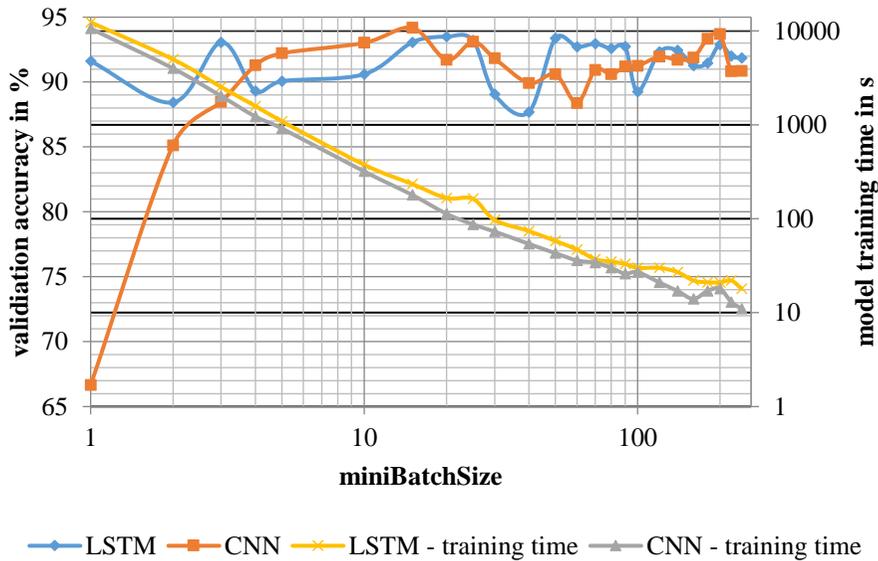
Fig. 6. Progress of validation accuracy and training time while changing the training parameter miniBatchSize

The optimization of further network and training parameters was continued in an iterative manner. The validation accuracy was always used as performance criterion. Finally, the determined parameter settings were tested with seven different network structures. This ANN structures were formed by adding individual layers to the basic structures of the LSTM and CNN network examples. Their layered structure and the obtained validation accuracies are presented in Fig. 7. The network "LSTM_2" achieved the best result with an average validation accuracy of 93.36 %.

The classifiers and ANN were finally trained and validated with the optimized hyper-parameters and the MSD dataset. Tab. 2 shows the achieved validation accuracies in compare with those of the SSD sets. The differences between MSD and SSD accuracies are also calculated. Essentially, it becomes clear that the classifiers achieve good results especially with small available datasets. Otherwise it could be observed for ANN that the performance remains on a similar level or even increases with larger data sets in most cases. Another advantage of ANN is the straight usability of sensor sizes without the preprocessing step of feature extraction. Likewise, the temporal resolution of the predictions of ANN corresponds exactly to the transmission rate of the sensors, while classifiers have a lower resolution (e.g. w = 50 and o = 45 lead to a label rate of only 6 Hz).
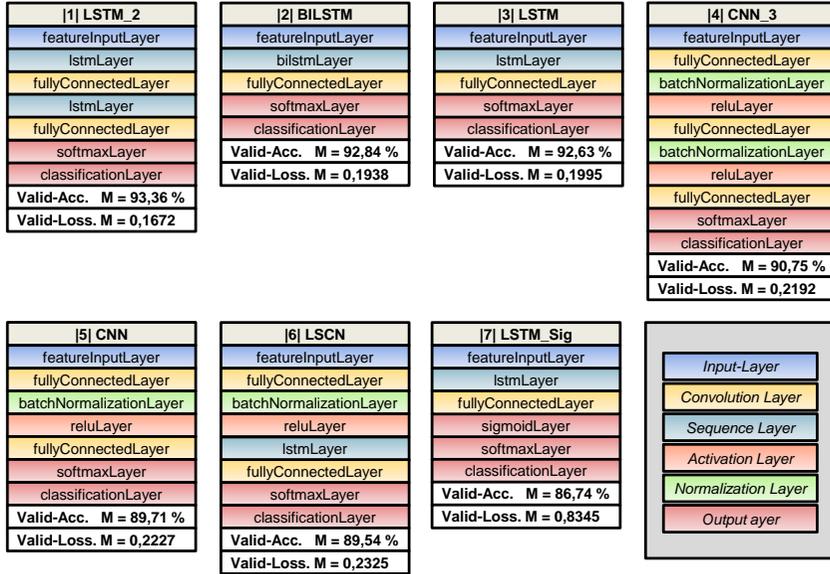
|1| LSTM_2
- featureInputLayer
- lstmLayer
- fullyConnectedLayer
- lstmLayer
- fullyConnectedLayer
- softmaxLayer
- classificationLayer
- **Valid-Acc.   M = 93,36 %**
- **Valid-Loss. M = 0,1672**

|2| BILSTM
- featureInputLayer
- bilstmLayer
- fullyConnectedLayer
- softmaxLayer
- classificationLayer
- **Valid-Acc.   M = 92,84 %**
- **Valid-Loss. M = 0,1938**

|3| LSTM
- featureInputLayer
- lstmLayer
- fullyConnectedLayer
- softmaxLayer
- classificationLayer
- **Valid-Acc.   M = 92,63 %**
- **Valid-Loss. M = 0,1995**

|4| CNN_3
- featureInputLayer
- fullyConnectedLayer
- batchNormalizationLayer
- reluLayer
- fullyConnectedLayer
- batchNormalizationLayer
- reluLayer
- fullyConnectedLayer
- softmaxLayer
- classificationLayer
- **Valid-Acc.   M = 90,75 %**
- **Valid-Loss. M = 0,2192**

|5| CNN
- featureInputLayer
- fullyConnectedLayer
- batchNormalizationLayer
- reluLayer
- fullyConnectedLayer
- softmaxLayer
- classificationLayer
- **Valid-Acc.   M = 89,71 %**
- **Valid-Loss. M = 0,2227**

|6| LSCN
- featureInputLayer
- fullyConnectedLayer
- batchNormalizationLayer
- reluLayer
- lstmLayer
- fullyConnectedLayer
- softmaxLayer
- classificationLayer
- **Valid-Acc.   M = 89,54 %**
- **Valid-Loss. M = 0,2325**

|7| LSTM_Sig
- featureInputLayer
- lstmLayer
- fullyConnectedLayer
- sigmoidLayer
- softmaxLayer
- classificationLayer
- **Valid-Acc.   M = 86,74 %**
- **Valid-Loss. M = 0,8345**

Legend:
- *Input-Layer*
- *Convolution Layer*
- *Sequence Layer*
- *Activation Layer*
- *Normalization Layer*
- *Output ayer*

**Fig. 7. Validation accuracy and loss for seven different network structures
(mean values for miniBatchSize = [15, 50, 200])**

**Tab. 2. Comparison of mean validation accuracies obtained with classifiers and ANN for SSD-
and MSD-datasets**

Mean validation accuracies in %	Classifier			
	boosted trees	cosine knn	gaussian naive bayes	medium tree
SSD	94.5	94.2	**96.4**	88.9
MSD	**92.9**	92.0	89.4	89.8
Difference: MSD-SSD	-1.6	-2.2	-7	+0.9

Mean validation accuracies in %	ANN						
	LSTM_Sig	LSTM_2	LSTM	LSCN	CNN_3	CNN	BILSTM
SSD	86.7	**93.4**	92.6	89.5	90.8	89.7	92.8
MSD	79.8	92.4	91.9	**93.7**	92.6	93.2	91.4
Difference: MSD-SSD	-6.9	-1	-0.7	+4.2	+1.8	-3.5	-1.4

In order to detect possible causes for incorrect model predictions. the performance graph like shown in Fig. 1 was utilized. This form of visualisation enabled a simple temporal localisation of deviations between ground truth and prediction. Different spatial orientation of the examined people was considered the dominant cause of the error, often with the addition of abnormal exercise performance. Similarly, very slow executions were considered inferior. Another problem in the classification was the transition areas between activity and pauses, as it was not possible to determine the exact start and end points of an activity. There is also reason to assume that variations in the signal characteristics due to the individual corporality of the subjects (condition, mobility, gender, body height) lead to person-related variations in the accuracy of the trained models. The comparatively small number of subjects whose data were used for the model training could have increased this effect.

4. CONCLUSION

The article describes the steps for implementing a HAR system, from data generation through their preprocessing up to training and optimisation of selected classification models. The model structures studied were classifiers and ANNs, which both successfully performed sport exercise recognition based on inertial data. First, the measurement system and the generation of two differently scaled data sets (SSD, MSD) were described. SSD served as a smaller data set for the iterative optimisation of model and training parameters. Once the parameterisation for achieving the best possible validation accuracies was completed with SSD, the models were trained and validated again with the larger MSD set. The resulting accuracies were presented in a comparable form. In the end, occurring false assignments were localised in the course of time and potential causes were elaborated. The paper thus provides an insight into the available tools of Matlab® for processing supervised learning tasks. It is shown that motion classification can be performed well with simple AI methods without extensive expert knowledge. Differences in the use of classifiers and ANNs are also elaborated and model-specific advantages and disadvantages in application for HAR are described.

From the presented state of the art, further research tasks can be defined. For example, the obtained results should be verified by a more comprehensive database with more exercise classes and participants. An increase in the number of classes necessitates the use of additional performance indices, which describe the model accuracy on a class-specific basis. Another research focus could be the reduction of input variables by feature selection methods. This could allow the application on mobile platforms (smartphones, tablets) by reducing computational costs as well as increasing the robustness of the classification by discarding irrelevant signals. Finally, the extension of the classification functions should be mentioned. In addition to the pure recognition of the exercise execution, counting the completed repetitions would be another important implementation step. Functions for training monitoring, which provides feedback on the correct execution of exercises, would also significantly increase the potential of a HAR system in practice. To this end, the investigations described in this paper provide the basis for further developments.

REFERENCES

Brühl, V. (2019). Künstliche Intelligenz, Maschinelles Lernen und Big Data—Grundlagen, Marktpotenziale und wirtschaftspolitische Relevanz. *WiSt - Wirtschaftswissenschaftliches Studium*, *48*(11), 34–41. https://doi.org/10.15358/0340-1650-2019-11-34

Chakraborty, A., & Mukherjee, N. (2022). A deep-CNN based low-cost, multi-modal sensing system for efficient walking activity identification. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-022-13990-x

Helten, T. (2013). *Processing and tracking human motions using optical, inertial, and depth sensors*. Universität des Saarlandes. https://publikationen.sulb.uni-saarland.de/handle/20.500.11880/26607

Hussain, A., Zafar, K., Baig, A. R., Almakki, R., AlSuwaidan, L., & Khan, S. (2022). Sensor-Based Gym Physical Exercise Recognition: Data Acquisition and Experiments. *Sensors, 22*(7), 2489. https://doi.org/10.3390/s22072489

Javed, A. R., Sarwar, M. U., Khan, S., Iwendi, C., Mittal, M., & Kumar, N. (2020). Analyzing the Effectiveness and Contribution of Each Axis of Tri-Axial Accelerometer Sensor for Accurate Activity Recognition. *Sensors, 20*(8), 2216. https://doi.org/10.3390/s20082216

Kautz, T. (2017). *Acquisition, Filtering and Analysis of Positional and Inertial Data in Sports*. FAU University Press. https://doi.org/10.25593/978-3-96147-065-5

Polo-Rodriguez, A., Montoro-Lendinez, A., Espinilla, M., & Medina-Quero, J. (2022). Classifying Sport-Related Human Activity from Thermal Vision Sensors Using CNN and LSTM. In P. L. Mazzeo, E. Frontoni, S. Sclaroff, & C. Distante (Eds.), *Image Analysis and Processing. ICIAP 2022 Workshops* (pp. 38–48). Springer International Publishing. https://doi.org/10.1007/978-3-031-13321-3_4

Schuldhaus, D. (2019). *Human Activity Recognition in Daily Life and Sports Using Inertial Sensors*. FAU University Press. https://doi.org/10.25593/978-3-96147-226-0

Sequence-to-Sequence Classification Using Deep Learning (n.d.). *Mathworks*. Retrieved February 21, 2023, from https://de.mathworks.com/help/deeplearning/ug/sequence-to-sequence-classification-using-deep-learning.html

Soro, A., Brunner, G., Tanner, S., & Wattenhofer, R. (2019). Recognition and Repetition Counting for Complex Physical Exercises with Deep Learning. *Sensors, 19*(3), 714. https://doi.org/10.3390/s19030714

Steels, T., Van Herbruggen, B., Fontaine, J., De Pessemier, T., Plets, D., & De Poorter, E. (2020). Badminton Activity Recognition Using Accelerometer Data. *Sensors, 20*(17), 4685. https://doi.org/10.3390/s20174685

Train Network with Numeric Features (n.d.). *Mathworks*. Retrieved February 21, 2023, from https://de.mathworks.com/help/deeplearning/ug/train-network-on-data-set-of-numeric-features.html