Monty Matlab Group 8 Project Documentation

L.Freisinger, R.Jacumet, O.Inak, A.Misik
Chair for Data Processing, Technical University of Munich
<leonie.freisinger, adam.misik, onat.inak, r.jacumet>@tum.de

***Abstract*—The task of time series classification spans several domains, such as the analysis of sales records, the analysis of transportation data, or the analysis of EEG signals. In this project report, the task of Human Activity Recognition is investigated. More specifically, methods are proposed that discriminate between normal and unusual walks by utilizing acceleration data from an IMU smartphone sensor. While LSTM networks have been shown to be the most accurate with a $99.2\,\%$ percent classification accuracy, SVM has been shown to have the lowest computational complexity.**

***Keywords*—Time Series Classification, Human Activity Recognition, Machine Learning, LSTM, kNN**

I. Introduction

Human Activity Recognition, the research field of detecting and analyzing human movements, activities and gestures, is becoming increasingly popular in research [1]. Measurement data aquired by sensors as found in smartwatches, smartphones, or even toothbrushes, alows to analyze and monitor respective activities.
In this work, we develop models based on a long short-term memory neural network (LSTM) and feature-based classifiers that are used to classify human movements. Using acceleration measurements taken by Intertal Measurement Units (IMUs), such as those found in smartphones, the trained models determines whether the user carrying the mobile device has taken a "Normal Walk" or a "Silly Walk". The concept of a "Silly Walk" is adopted from the comedy group Monty Python, which made a famous skit in which actors moved in an unusual way that the classifier would robustly distinguish from normal walking [2].

II. Methodology

Subsection II-A describes the data collection and processing steps used for the project. Our best model, the LSTM network, will be discussed in more detail in Subsection II-B.

A. Data Collection and Pre-Processing

The goal of the data collection process is to maximize the walking scenarios (Normal/Silly) covered in the dataset, i.e., to maximize the variance of the dataset to obtain high model generalization. To this end, a data collection strategy is designed and implemented by each of the four team members. The initial part of the strategy is the definition of seven walk categories for both walk types. These categories can be thought of as surface and pace conditions under which the walks are recorded.

- Category A: Walk on even surface, normal pace.
- Category B: Walk on uneven surface, normal pace.
- Category C: Walk on even surface, fast pace.
- Category D: Walk on even surface, slow pace.
- Category E: Walk up a hill, normal pace.
- Category F: Walk down a hill, normal pace.
- Category G: Walk on even surface, HIIT pace (slow-fast-slow-fast).

Another important part of the collection strategy is the pocket position of the mobile device, which influences the direction of the acceleration data. A total of four positions are possible: device screen to the inside/outside of the pocket, device facing the top/bottom of the pocket. All four positions are varied during the recording of each walk category listed above. Also, the device is exchanged between the right and left front pockets for the recordings.

Furthermore, each team member utilizes a different sample rate and recording length for the recording. Each team members sample rate differed from $60\,\mathrm{Hz}$ to $90\,\mathrm{Hz}$, in $10\,\mathrm{Hz}$ intervals. Recording lengths vary from $40\,\mathrm{s}$ to $5\,\mathrm{min}$.

Team members use the Matlab Mobile App running either on iOS (iPhone 12) or Android (Huawei Mate Pro 20).

A script is written to record a walk, which is implemented using the Matlab mobile app. The team member can parameterize the sampling rate and recording length, along with the name of the walk (walk number, walk class, category of walk). The script gives the member time to place the device in the pocket and stopped recording after the specified recording length, reducing the steps to slice the signals. Nevertheless, each team member still needs to check the validity of the recorded walk and slice out inappropriate recording parts.

B. Model

Deciding on the most efficient model to achieve sufficiently good results is a challenge for the project. We mainly focus on two methodological clusters from the machine learning domain:

- Feature-based classifiers
- Neural networks

From a number of different feature-based classifiers and neural networks applicable to the classification task of Normal walks and Silly walks, the LSTM model is presented in more detail below.

The LSTM model is a recurrent neural network and can exhibit temporally dynamic behavior by using its memory to process variable length sequences of inputs. This model has the advantage that measured time sequences of acceleration data can be used directly to train the model without first extracting features. Unlike the feed-forward neural network, the LSTM model is able to process entire sequences of data by implementing feedback connections. Therefore, the pre-processing amounts to extract time sequences of a defined length from the measurement data. The detected correspondencies and similarities between signals of the same classes of walks are affected by the selected hyperparameters (discussed in Subsection III) which infers the performance of classification.

III. Evaluation

Several classification models from the two mentioned clusters are investigated, and the best results are obtained for the LSTM method. For this reason, the LSTM model is selected as our main classification model for this project.

Finding the optimal layers and options for a deep learning approach requires particular effort. Therefore, an iterative hyperparameter optimization algorithm is implemented to find the optimal hyperparameters for the LSTM model, i.e., the number of hidden layers, the mini batch size and the maximum number of epochs. At each iteration of the optimization, the training data is randomly split into train and validation data with a $90/10\,\%$ split, and the model is iteratively trained using the split train data with the subset of the possible hyperparameter combinations. Then, the accuracy of the trained model at the corresponding iteration is obtained by predicting the validation data. The model with the best accuracy is saved, with the corresponding model hyperparameters. Regarding our implemented optimization algorithm, we observed that:

- mini batch size = 6
- number of hidden layers = 60
- maximum number of epochs = 45

were optimal for our LSTM model. With our optimal hyperparameter selections, we have observed in the training of our LSTM model that the loss was exponentially decreasing towards zero, as seen in Fig. 1, which was a sign that we have chosen accurate hyperparameters for our model.

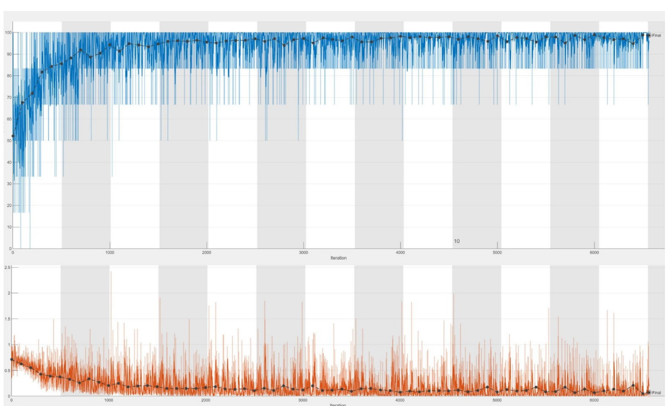

Fig. 1. **Training Results :** Accuracies in every iteration (blue), loss in every iteration (orange)

The performance of the optimal model with this hyperparameter choices is examined by predicting the class of the walks in the test data ($20\,\%$ of the processed data pool, while the rest is used as training data). In the end, we reached an accuracy of $99.2\,\%$ by predicting the classes of the different walks with the LSTM model.

IV. Discussion and Conclusion

The results show that the LSTM model predicts the walk classes with an accuracy of $99.2\,\%$ and therefore perfectly fulfills the stated classification task. Compared to two feature-based approaches studied, a kNN model and a SVM model, the LSTM model has the disadvantage of a comparably high training time of $456.1\,\mathrm{s}$. In contrast to the LSTM model, specific features have been extracted for the kNN and SVM approach instead of taking time sequences as input, inspired by the work of [3] and [4]. Therefore a significantly shorter training time of $10.11\,\mathrm{s}$ for the kNN and $6.96\,\mathrm{s}$ for the SVM approach were achieved while the accuracy of $91.6\,\%$ for the kNN and $90.4\,\%$ for the SVM approach are still sufficiently high.

Overall, the LSTM approach involves less preprocessing. On the other hand it requires a large training time, while the kNN and SVM approaches achieve a comparably short training time. Both approaches show sufficient results for accuracy and therefore the choice of the model depends on the given conditions of application.

Furthermore, the data that has been strategically acquired in terms of quantity and variance and the chosen train-/ test split also has proved to be expedient for the classification performance. An interesting finding in the model evaluation was that training data that includes Silly Walks of the Category D and Category G leads to a worse accuracy than training data without those categories. This result can be explained by the fact that both categories contain walks with slow movement and therefore overlap in their characteristic with Normal Walks. Taking this into account the accuracy of the LSTM model further was improved.

In this work an LSTM approach including its architecture, the data acquisition and data pre-processing was presented that achieves the task of classifying Silly and Normal walks. Overall this approach solves the classification sufficiently by reaching an accuracy up to $99.2\,\%$ and can be easily applied on data recorded by a smartphone‘s IMU.

References

[1] O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[2] Ministry of silly walks. Last view: 09.06.2021. [Online]. Available: https://www.youtube.com/watch?v=eCLp7zodUiI

[3] A. Mannini and A. M. Sabatini, “On-line classification of human activity and estimation of walk-run speed from acceleration data using support vector machines,” *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 15, no. 3, pp. 3302–3305, 2011.

[4] K. Rieke, “Human activity recognition using a smartphones inertial measurement unit,” 2017.