The effect of recentness of consumer-grade wearable training data on the ability of a DNN to identify users

Niels van der Voort

Supervisor(s): David Tax, Arman Naseri Jahfari, Ramin Ghorbani

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Niels van der Voort
Final project course: CSE3000 Research Project
Thesis committee: David Tax, Arman Naseri Jahfari, Ramin Ghorbani, Guohao Lan

Abstract

Heart rate data and other data collected by consumer-grade wearable devices can give away quite useful information about the user. It can for example be used by machine learning algorithms such as Deep Neural Networks (DNN) to learn patterns about cardiovascular disease and fitness, or be used for identification. Heart rate patterns can also change quickly within the span of several months, which could make older heart rate data less useful when training a DNN. This paper shows that the DNN did indeed perform significantly worse when trying to identify people on older data compared to recent data. The accuracy calculated from the test set was 63.64% when trained on the most recently available training data, in comparison to 33.88% when trained on the least recent data which was more than 200 days older. When changing the recentness of training data only for a single user, there was also always an improvement in the accuracy of the model to identify that particular person. The accuracy to identify all users however did not necessarily increase, and sometimes even decreased. Using more data for training still outperforms using a smaller amount of samples of more recent data by slight margins, showing the trade-off between the recentness of data and the amount of data used for training. However, if fast training times are required, taking the most recent data windows can still lead to a similar performance as when training on all available data.

1 Introduction

Wearable devices such as smart watches and heart rate monitors are becoming increasingly popular, accessible and accurate when collecting data. The data collected by these consumer-grade wearables, including heart rate and step count, is seemingly innocent. However, from past experiments it has become clear that this data is enough to identify and derive information about wearable device users [3, 6, 8]. This derived information, such as general fitness and behaviour patterns, can be useful for medical practitioners to help patients, sports teams for improving training and elderly care for monitoring.

Research on this topic has been conducted for some years already, mostly by computer scientists and medical researchers. For example, the research by Retsinas et al. [8] explored the accuracy of a deep neural network with the task of identifying users with data collected by simple consumer-grade wearables. The findings of this paper show that the identification accuracy changes severely when a user was sleeping, walking or doing other activities. The paper also shows how the use of various sensors affects this accuracy, stating that when solely using a heart rate monitor, the model had the highest identification accuracy when using data from when the user was sleeping. However, this paper uses accelerometer data, which tracks changes in speed. It does not explore the use of step count data. A question that is raised by the paper is how selecting different subsets of training data can potentially affect the accuracy of the model, and if so by how much.

Different studies have also been done showing how a person's resting heart rate changes as a result of exercise and age [7, 9]. This means that a part of the heart rate pattern is always changing, which would need to be considered for identification algorithms.

The gaps in current research shown by these papers together lead to the following research question for this paper: How does the recentness of consumer-grade wearable data used for training a Deep Neural Network (DNN), impact the ability of the model to identify users? The hypothesis was that the performance of the model when trained on recent data would be better than the performance when trained on older data. This hypothesis was derived by the idea that heart rate patterns from more recent training data will better represent heart rate patterns in future unseen test data. Furthermore, it would be interesting to see if and by how much the performance of the model decreases when the data is not up-to-date, which could show how important it is to continuously retrain the model on newly available data.

This research would also show if data might become outdated or if the performance might even improve if some data is not included when training the model, because it does not represent future data of the user by which they would be identified. This would be an improvement to existing DNN models which use consumer-grade wearable device data. Finally, it could also lead to estimates of how much heart rate and step count patterns have changed for each user, and in what period of time these changes occurred.

The question that is tackled in this paper aims to improve DNNs which use consumer-grade wearable data such as heart rate and step count data as input. Many DNN models such as CNNs, LSTMs or CNN-LSTM hybrids based on the wearable data of users already exist such as the one described by Khatun [3]. Since heart rate is a feature that changes over time, there are improvements and requirements for the data which could improve the performance of the model and minimize the decrease in performance that might become visible when the data gets older. Therefore the outcome might also be that the model needs to be retrained or new data needs to be added in order for proposed methods to work properly in a real world scenario.

This paper is structured as follows: Chapter 2 shows the methodology, Chapter 3 highlights the contribution of the research to science, in Chapter 4 the results are presented, and Chapter 5 addresses the importance of conducting research responsibly. In Chapter 6, the results are discussed, and finally, Chapter 7 concludes the paper while also showing potential areas for future research.

2 Methodology

To address the research question, first a reliable DNN model which is able to identify users by wearable data is described. Then, the experiments which are run using this DNN model and their setups are explained.

2.1 Deep Neural Network Model

Multiple papers using wearable sensor data for machine learning models propose to use either a Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) or mixed CNN-LSTM model [5, 6, 8]. The CNN is often used for processing images, but can also be applied successfully with time-series data due to its ability to find relationships between different data points on smaller and larger scales [1]. The LSTM is a subtype of the Recurrent Neural Network which works well on time-series data due to its ability to memorize longer lasting patterns in data [2].

The model used for this research is a CNN-LSTM hybrid. This hybrid was chosen because the CNN-LSTM performed with the highest average identification accuracy when comparing models, after optimizing the hyper-parameters. This DNN model outperformed the CNN in terms of accuracy with a slight margin of approximately 2%. The CNN-LSTM is structured by connecting the output of the CNN layers to some LSTM layers to make a more complex model.

The model architecture includes two one-dimensional convolutional layers, followed by max pooling layers and dropout layers to improve the generalization ability of the model and prevent it from overfitting. After the convolutional layers, the data is flattened to a one-dimensional list of features which is then passed to the LSTM layer. After the LSTM layer there is a final linear layer with a Softmax activation function. The Softmax function is useful when tackling multi-class identification problems, because the output of this function can be interpreted as a list of probabilities. To identify from which of the 11 users the data was collected, 11 probabilities are returned. The index of the largest probability is chosen as the ID of the predicted user. Figure 1 shows the steps and shapes of the data as it passes through the layers of the model. The input represents two arrays; one for the step count data, and another for the heart rate data. The output is a one dimensional array with 11 numbers representing the output of the Softmax function. There are many hyperparameters which were optimized such as kernel sizes, strides, in- / output channels in CNN layers, dropout probabilities and the number of hidden layers between the LSTM and linear layer. The full schematic architecture including the choice for these hyperparameters can be seen below in figure 2. Cross-entropy loss was used as the loss function since it works well together with a Softmax activation function. Adam optimization was used as this converged quicker than SGD, which was beneficial since training the model on 80% of the available data took 20 minutes. This specific model specializes in identifying users by windows of approximately one day.

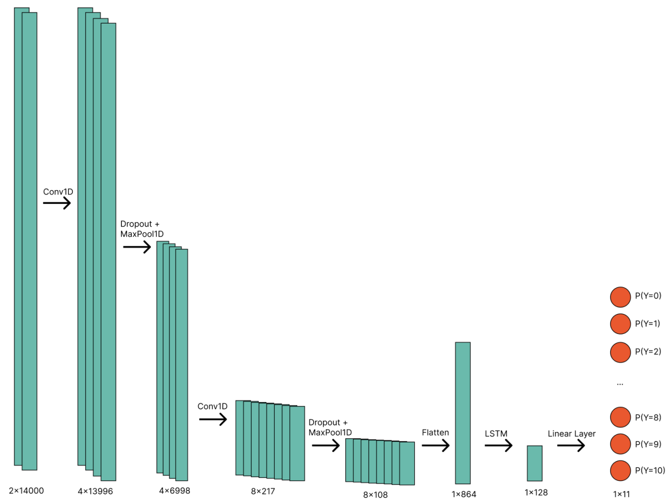


Figure 1: Visualization of CNN-LSTM model layers.

Layer Name	Hyperparameter settings
Conv1D	Kernel Size = 5, Stride = 1, Output Channels = 4, Activation = ReLU
Dropout	Dropout Rate = 0.1
MaxPooling1D	Pool Size = 2
Conv1D	Kernel Size = 64, Stride = 32, Output Channels = 8, Activation = ReLU
Dropout	Dropout Rate = 0.1
MaxPooling1D	Pool Size = 2
Flatten	-
LSTM	Output Channels = 128
Linear	Output Channels = 11, Activation = SoftMax

Figure 2: The detailed CNN-LSTM model architecture including hyperparameters.

2.2 Data

The dataset used for this study is from a clinical study, ME-TIME (registered at ClinicalTrials.gov with ID: NCT05802563) [4]. The time that users participated and thus the length of available data differs per user. Most users have approximately one and a half years of data, while other users have as little as 2 months. To prevent the model from becoming biased towards certain users, the same amount of data windows were taken to train and test for every user. To maximize the amount of data taken for each user, users that do not have at least a year of data points are left out from the experiments.

The collected data has information about heart rate, which is sampled every 5 seconds, and step count, which is sampled every minute.

Since the data was collected from consumer-grade watches, which are Fitbit watches in the case of the ME-TIME study, there are gaps in time, and also missing step values.

2.3 Experiments

In order to measure the effect of recentness of training data on the performance of the CNN-LSTM model, three experiments were conducted. Each of these experiments were aimed at a different sub-questions. Reimers et al. [7] concluded that changes in heart rate patterns can be found after on average three months of exercise. This was taken into account in the setup of the experiments. For this reason, the jumps in time between different training windows used in the experiments were approximately 1 to 2 months.

The model and hyperparameters stayed the same for every experiment to make results more comparable and controlled. Due to the best performance of the CNN-LSTM model varying when running the algorithm multiple times, the algorithm was run 5 times for every trial. After this, the average performance, best performance and standard deviation was reported. The average performance was plotted, which can be seen in the Results section. The performance was measured by calculating the identification accuracy on the test set, the f1-score and identification accuracy per person. The identification accuracy was calculated by comparing the predicted users the model returned, and the actual users to which the data belonged.

The first experiment, referred to as the sliding window experiment, had the goal of measuring how much the performance of the model is affected by recentness of the training data. To ensure that the change in performance was indeed caused by the recentness, the test set stayed the same among trials and the size of the training set also remained constant among trials. The training data was selected as shown in figure 3, where the training set is created as a sliding subset of the possible available training data.

Figure 3: Simplified version of the sliding window experiment.

The aim of the second experiment was to measure the trade-off between recentness of training data and the amount of training data used. Instead of a sliding window approach, there was no gap between the training and test sets in this experiment as shown in figure 4. This can be seen as an expanding window. The amount of training data used was gradually decreased by leaving out the oldest data and the performance was measured as in the first experiment. If the model trained only on the most recent data performs better than all others, it could be better to leave out older data when training the model. In this case, training data might differ too much from future testing data. This training data could then be regarded as outdated.

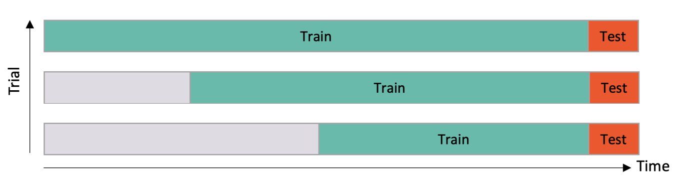


Figure 4: Simplified version of the expanding window experiment.

The sliding window and rolling window experiments both used approximately a year of data due to the requirement that each user must have the same amount of data. After filtering out users that did not have more than 330 windows of available data, which was set as a minimum, 11 users were left. The third and final experiment had a similar setup as the sliding window experiment. However, the sliding window was only applied to the training data of a single user. Therefore the recentness of training windows was also only changed for a single user in this experiment as shown in figure 5. For all other users the training and test sets were kept constant with approximately a year of data to train and test on. To create the train and test split in this setup, 330 windows are taken for each user, except for the user that is experimented on.

For this user, all their available data is used. Therefore, there will be more data available in both the train and test sets for this user. The sliding window approach is then applied on this available training data to create subsets which are as long as the data of other users, which is used in training. The test set for this user consists of the first 66 windows of their available test set to ensure the same number of test windows are taken for every user.

The individual users who were experimented on were chosen by selecting the four people with the most available data. The experiments for these users were run separately. To measure the impact of changing recentness of training data for a single user, two measures were taken. First the identification accuracy of the test set. Then the identification accuracy of the test data belonging to the user that is experimented on. This is referred to as the individual accuracy for a user.

To create a hypothesis of which users were expected to have changed the most, the minimum, maximum and mean values were calculated from their heart rates. This was done for the oldest training set used in the experiment, and the test set of each user. This showed approximately how much their heart rate patterns have changed over the time period in which the data was collected. Step count patterns were not accounted for in this hypothesis, because the features were affected too much by missing values.

The goal of this experiment was to look at the difference in heart rate patterns over time. Therefore for each individual experiment, the entire data set of the user that is experimented on is used to capture larger changes in heart rate patterns. However, this would cause the oldest data to be potentially more outdated for some experiments compared to others.

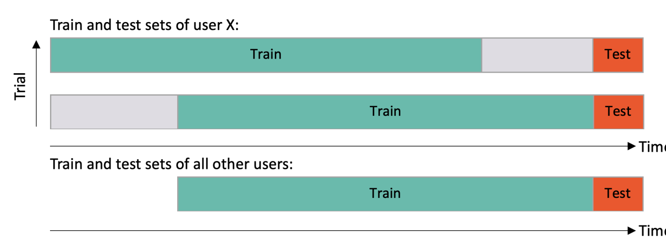


Figure 5: Setup of the per-person sliding window experiment. User X refers to the user that is experimented on.

3 Experimental Setup and Results

In order for the model and experiments to be considered reliable and reproducible, the preprocessing steps are described below. Then, the results of running the experiments are shown and discussed.

3.1 Preprocessing

The data used for the DNN is split into windows of 1 day. This is done by first splitting the data by date, and then only using windows which do not have too many missing values. More specifically, windows are filtered by the criteria of having at least 20 hours worth of data. As a result of this, there are missing days. Therefore a certain amount of windows does not necessarily represent the same amount of days. This causes the recentness of a data window to differ per user, where the oldest data window for one user might be a year old, and one and a half years old for another user. To accommodate for this, the age of the windows is calculated for the indices which are used in the experiments. This age is the gap in days between the window and the start of the test set. Since the amount of missing days differs per person, this age is then averaged for all users to get an average age of the data. This average age is supplied in the form of conversion tables in each results figure. The temporal order of the data is still maintained, implying that data windows with a lower index are less recent than data windows with a larger index in the experiments.

After converting the data to windows and filtering, normalization is applied independently on both step count and heart rate data, based on the standard deviation and mean from the training data. Because step count data is collected at a different frequency than heart rate data, there are many NaN values which have to be dealt with. These values are encoded after normalization, to an unrealistic value, -1. This does not break the DNN network when learning unlike NaN values.

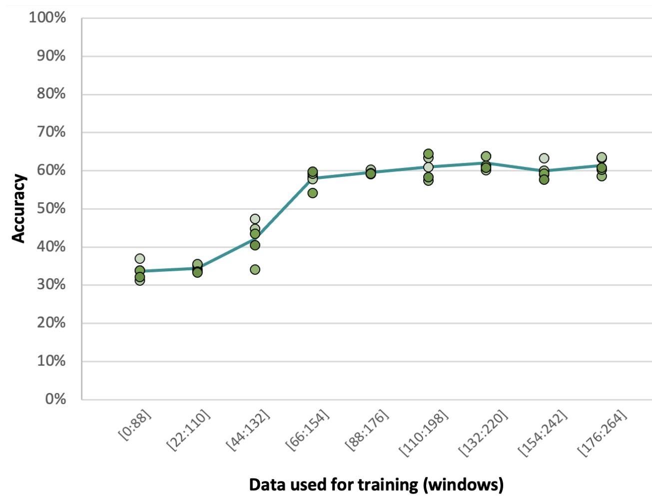
The training and test set are usually not random in time-series classification problems. This is because you want to predict to which user future data belongs. Thus, time series data that was taken after the testing data cannot be used for training. As a result of this, the test set is always the final portion of the data available. Furthermore to prevent the model from becoming biased towards certain users, the same amount of samples for every user is fed into the DNN per batch, while also alternating windows of each user. So the first training window of every user is used to train the DNN before moving onto the next training window for every user.

PyTorch, a machine learning framework, was used to implement this CNN-LSTM model due to its built-in support for many machine learning models. This includes the CNN and LSTM. There is also built in support for loss functions and optimization functions, including cross-entropy loss and the Adam optimization algorithm.

3.2 Effect of recentness on performance

The results of the first experiment are shown in figure 6. Each circle in the figure corresponds to running the model once. Therefore the five circles represent the five times the model was retrained for each trial with the exact same setup, and the trend line goes through the mean values of these five results. This was done because the model randomly initializes before every run, which causes it to converge to different losses and perform differently when the code is executed multiple times.

In the sliding window experiment, the test set contains the data from windows 264 until 330 for every user. These windows will be referred to as windows [264:330], where 264 is included in the range and 330 is not. The CNN-LSTM model performed better when training on data that was closer in time to the test set. However, the model still performed quite well on data with a gap of 110 windows between training and test sets. This corresponds to an average gap of 133 days, where the training data is between 133 days and 232 days old. When training on older data than this, the performance of the model quickly dropped.

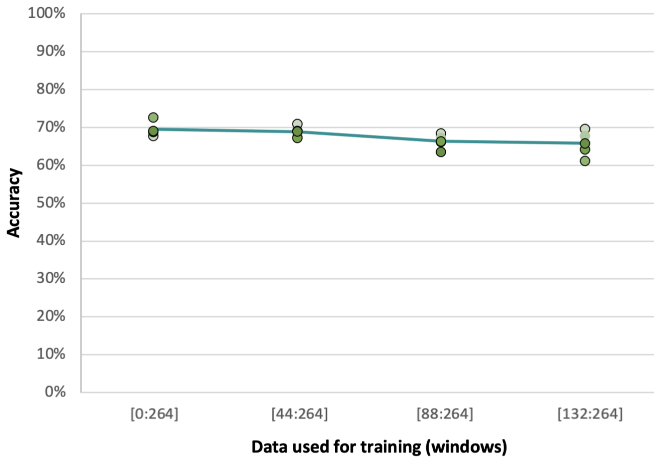


Windows	[0:88]	[22:110]	[44:132]	[66:154]	[88:176]	[110:198]	[132:220]	[154:242]	[176:264]
Age (days)	[206:304]	[181:280]	[157:256]	[133:232]	[107:206]	[78:181]	[50:157]	[26:133]	[0:107]

Figure 6: Results of the sliding window experiment with a conversion table converting the window numbers to the age of the data, which is the gap to the start of the test set. The test set used for calculating the accuracy was the data from [264:330].

3.3 Recentness vs amount of training data

The outcome of the second experiment is shown below in figure 7. The test set contains the same data as in experiment 1. In this experiment, the model performed best when the training set was the largest. The reduction in performance remained quite small however, showing the trade-off between the amount of training data and the recentness of the training data. The trial trained on only half the amount of training data performed with approximately 5% less accuracy.



Windows	[0:264]	[44:264]	[88:264]	[132:264]
Age (days)	[0:304]	[0:256]	[0:206]	[0:157]

Figure 7: Results of the expanding window experiment with a conversion table converting the window numbers to the age of the data, which is the gap in days to the start of the test set. The test set used for calculating the accuracy was the data from [264:330].

3.4 Effect of changing recentness per person

Figure 8 shows the results of the per person sliding window experiment. In this experiment, the recentness of training data was only changed for a single user. The four graphs correspond to four experiments which were run independently of each other. The test sets which were used for calculating the accuracy in the experiments regarding user 0, 1, 2 and 3 respectively are [439:505], [443:509], [356:422] and [373:439], and the training sets where changed using a sliding window approach as seen in figure 8. In each experiment, for every user except the one that was experimented on, the training set was [0:264] and the test set was [264:330]. Therefore the model was still trained on data from all users, but the only changes within an experiment was the recentness of the training data of a single user.

The method to predict which users were expected to change and by how much, as explained in the methodology was calculated by taking the handcrafted features of min, max and mean heart rates. These features were calculated for the oldest used training set and the test set. The differences were calculated subsequently to get a score of how much each user was expected to change. The score of user 0 was 27.56, meaning that user was expected to have changed the most. User 2 got a score of 6.59 which meant that he was expected to have changed the least. Users 1 and 3 had scores of 10.69 and 16.13 respectively, ranking within the range of the others.

An unexpected finding from the results of the per person sliding window experiments is that the overall performance of the model did not always improve.

From the mean individual accuracies, it is visible that the model was able to distinguish the users that were experimented on with a higher identification accuracy when trained on more recent data in each of the experiments.

4 Responsible Research

Responsible research is a crucial part of any research. In order for the study to have a positive impact and be considered trustworthy, there are some potential issues which have to be addressed. In this section the ethical considerations, potential biases and reproducibility of the research will be discussed.

4.1 Ethics

Heart rate and step count data can give away quite some information about a person according to Xu et al. [10], such as fitness, illnesses and sleeping patterns. This private information should not be accessible to anyone. Although no assumptions about illnesses and fitness are made in this study, it does try to identify a change in fitness patterns, which is sensitive data. To mitigate privacy concerns, users are not referred to by name in any part of the data or system, and any unnecessary metadata is unused.

Although wearables are becoming more accessible and cheaper, inequitable access is still a point of discussion. Accurate watches can still be quite expensive which could disadvantage people with less money available. This is because the model might not work as well with less accurate sensors.

There are false identifications which could lead to adverse consequences. Therefore the accuracy, reliability and limitations of the model need to be carefully evaluated and addressed. Also, thresholds for these values should be set before using this model to identify users in the real world.

Finally, the performance of the model differs for unique users, which could cause discrimination of individuals or groups that are very similar. To reduce this effect, the biases of the model are taken into consideration and minimized as explained in the next section to ensure that the model is trained on diverse groups and this performance change is minimized.

4.2 Biases

Several types of biases are important to consider for this research. Sampling bias could be present as the amount of data considered is only from 54 users. These users are quite diverse in fitness, gender, age, weight and height. However, due to the data being collected by a study into cardiovascular disease, a large proportion of the users have some form of cardiovascular disease, more than in society [4].

Another form of data bias present is watch inaccuracy. This is minimized by using data from the same brand of watches, namely Fitbit. The models worn by the users were either the inspire 2, charge 2 or charge 5. Therefore each model also uses a comparable heart rate monitor, although the age of these watch models might cause some versions of these watches to outperform others. Some users also have some gaps in step count and heart rate data, which might potentially cause the model to learn from other patterns such as when the watch is not being worn and sensor error.

4.3 Reproducibility

In order to reproduce the study, wearable data spanning several years needs to be collected or accessed to run through the model. The data used for this study is not publicly accessible, which somewhat limits the reproducibility of the study.

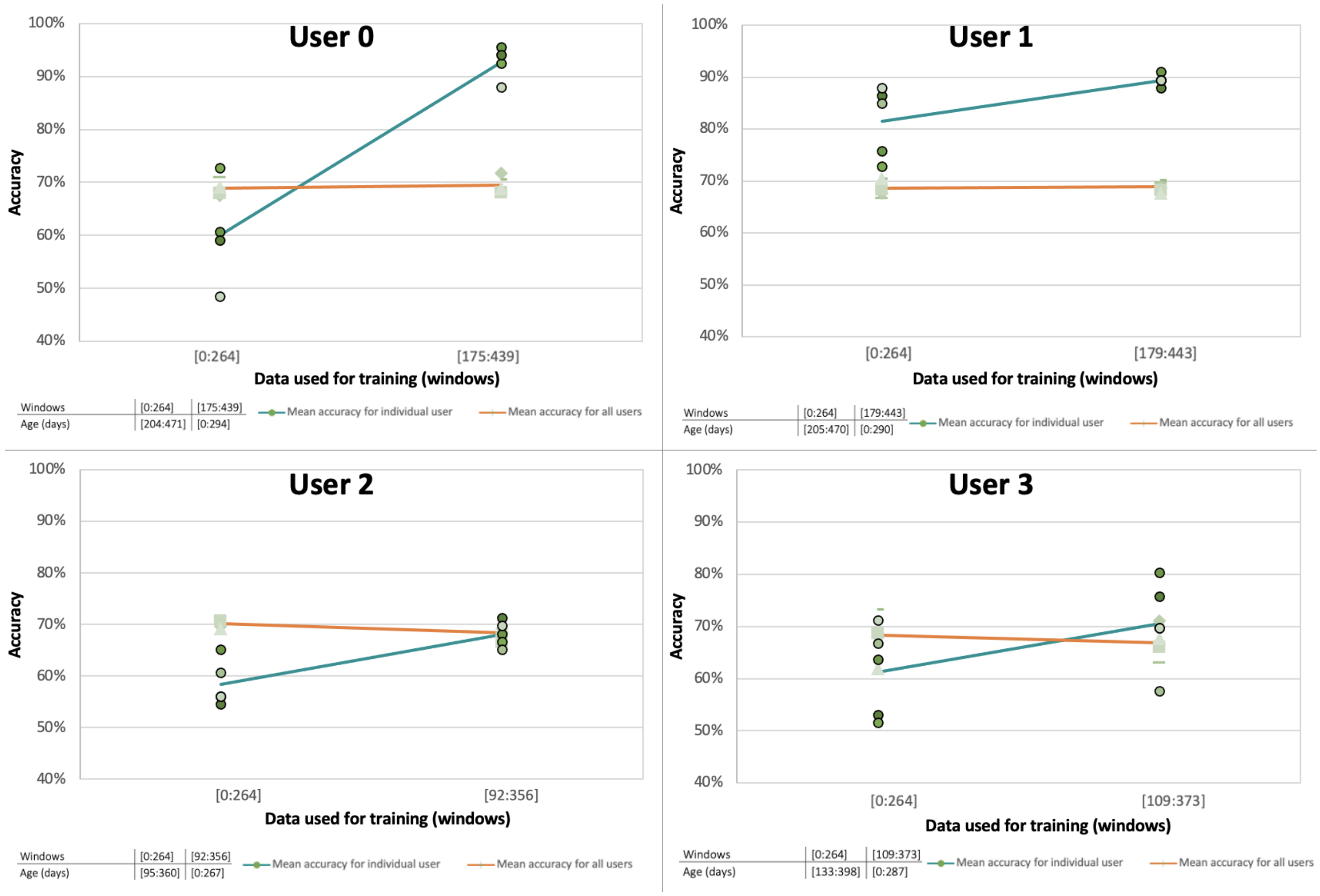


Figure 8: Results of per person sliding window experiments. Each graph represents an experiment. The blue line corresponds to the accuracy when identifying the user that is experimented on, while the orange line corresponds to the accuracy when identifying all users included in training.

The pre-processing steps and model are reproducible using the hyperparameters and layers specified in the methodology section. The repository with the implementation is also publicly available on the TU Delft Repository.

5 Discussion

The paper written by Reimers et al. [7] about resting heart changes seems to be in line with the findings of the first experiment, namely that the performance decreased after a gap of approximately three to four months. While the study found that the resting heart rate can change within several months, this alone should not lead to such a large decrease in performance. Therefore other heart rate metrics might also be changing within this time period, such as max heart rate and heart rate variability. However, this could also be caused by changes in other patterns which are potentially used by the DNN to identify users, such as sleeping patterns.

The results of the sliding window experiment found that the model decreased in performance especially when trained on data that was on average 256 to 157 days old. This is after approximately 5 months, which is in line with the findings of Reimers et al. [7]. It seems that after 5 months, the data becomes outdated enough to affect the performance of the CNN-LSTM model when using 11 subjects for identification.

5.1 Limitations

First, the availability of data is a limitation of this study. Santos et al. [9] wrote about the change in heart rate patterns over time due to age. Researching the effect of this on a DNN model was not feasible since there was approximately one and a half year of data available, which would not capture the expected change. The impact of recentness could also change if the data is more outdated than it was in the setup for the first two experiments. In these experiments approximately one year of data was taken. If more data were to be available, training on a more recent subset of data might be better than training on all available data.

Additionally, the results could be quite dependent on the data and implementation used for the DNN. For example, when data of different users is very similar, the model will have more difficulty identifying these users. In this case, recentness of training data might be more important than in the setup that is currently used. This could lead to results where the performance of the model might decrease with data that is more recent than 5 months. Ideally in this scenario, it would be good to train the model with data from more than 11 users, with measures to show how similar users are. This was not feasible for this research due to limited data availability.

There are also other factors than just recentness of training data that can impact the accuracy of the model. An example of this is data reliability, such as missing step count values, which are found in data from consumer-grade wearables. When there is less information available, the model might have more difficulty identifying the user. This might negatively affect the results.

6 Conclusion and Future Work

The main question of this paper was: How does the recentness of consumer-grade wearable data used for training a DNN, impact the ability of the model to identify users?

The results show that the model as hypothesized performed better on more recent data. The performance when trying to identify to whom future data belongs decreased, especially after approximately 132 windows. This corresponds to training on data that was, on average for all users, 157 to 256 days older than the start of the testing data. What especially caused the performance to decrease was that the ability of the model to correctly identify 4 users became 0.00. This could be explained by their testing data being entirely within the decision boundaries of other users.

In both the rolling- and sliding window experiments, no subset of data has been found to outperform the model that is trained on all available training data. This implies that the amount of training data has a slightly greater impact on identification accuracy than the recentness of the data. In the rolling window experiment, the models that were trained on less data still performed quite similarly to the model that was trained on all data. Therefore if the model needs to be trained quickly, training on less data is a good alternative.

In the results of the sliding window experiment, the accuracy remained within a margin of 3% for each model up to and including the model trained on data that was on average 232 - 133 days older than the start of the test set. This shows that recentness does not affect the accuracy to a large extent, until it suddenly affects the performance significantly. In this experiment, with approximately a 15% drop. This drop is quite sudden, which might show that the effect of recentness on performance is also quite abrupt. Furthermore, the individual accuracy for some users suddenly dropped close to 0. This is potentially due to testing data of users more closely representing another users training data, which would cause a lot of wrong classifications.

The results of the per person sliding window experiment showed that the recentness of training data for a single user does not necessarily positively impact the performance of the model. For each experiment on individual users, the ability of the CNN-LSTM to identify this particular user always increased when training on more recent data. The hypothesis that user 0 was expected to have changed the most showed positive results as seen in figure 5, since the individual accuracy increased the most for this user. The change in heart rate patterns for the other users did not seem to follow the order of the expected change which was described in the Experimental Setup and Results section. Each of the improvements in individual accuracies for these users was quite similar at approximately 10%. This could be due to the fact that the hypothesis was flawed by not including step count patterns due to too many missing values, or because the individual accuracies produced by the model were too unpredictable.

An unexpected result is that in the per person sliding window experiments of user 2 and user 3, the identification accuracy of the entire test set decreased by 1.84% and 1.46% respectively. It seems that the improvement in individual accuracy for the user that is experimented on might be paired with the decrease in accuracy of identifying other users that have similar patterns to the user that is experimented on.

Additionally there is a large variance in the individual accuracies when re-running the model with the exact same setup five times. This variance is especially present when training on the less recent training set. The CNN-LSTM has random initialization, meaning that it does not always provide the same results after training on the same data. The dropout layers also add some randomization into the DNN. This accounts for some variance between the runs of the model with the same setup. The quality and complexity of the training data can also affect the performance, so there will also be some change in performance which cannot be accounted for by the recentness of the data when running the model with different training sets.

Individual accuracy is related to the accuracy on the entire test set, however there are many potential combinations of individual accuracy that lead to the same full test set accuracy. This could explain the larger variance in the individual accuracy in comparison to the accuracy calculated by the entire test set. Additionally, individual accuracy is calculated on 11 times as little test samples.

6.1 Future Work

This paper focuses on changing heart rate and step count patterns across several months. However, as described by the research of Santos et al. [9], heart rate patterns also change across multiple years, which might have an even larger effect on the performance of the DNN due to the aging process. In this case, the trade-off between the amount of training data used and the recentness of training data might be very different since data might be even more outdated. Therefore it would be interesting to research the effect of recentness with data spanning a time period such as 5 or 10 years. Also, many affordable consumer-grade wearables have more sensors than just step count and heart rate sensors, which should positively impact the performance of the DNN model. This would also be interesting to investigate further.

Another possible area of future research is to design an optimizer which aims to find the ideal subset of training data. This could use a similar setup as the expanding window experiment shown in this paper. All time series problems where data patterns change over time might potentially be improved by this. This improvement could be through speeding up training times or through improving the performance of machine learning models on future testing data. This would also be beneficial in the domain of person identification using consumer-grade wearables.

References

[1] Aksan, F., Li, Y., Suresh, V., Janik, P. (2023). CNN-LSTM vs. LSTM-CNN to Predict Power Flow Direction: A Case Study of the High-Voltage Subnet of Northeast Germany. Sensors, 23(2), 901. https://doi.org/10.3390/s23020901

[2] Keith, M. (2023, March 30). Exploring the LSTM Neural Network Model for Time Series. Medium. https://towardsdatascience.com/exploring-the-lstm-neural-network-model-for-time-series-8b7685aa8cf

[3] Khatun, M. A., Yousuf, M. A., Ahmed, S., Uddin, M. Z., Alyami, S. A., Al-Ashhab, S., Akhdar, H. F., Khan, A., Azad, A., Moni, M. A. (2022). Deep CNN-LSTM With Self-Attention Model for Human Activity Recognition Using Wearable Sensor. IEEE Journal of Translational Engineering in Health and Medicine, 10, 1-16. https://doi.org/10.1109/JTEHM.2022.3177710

[4] ClinicalTrials.gov. (2023). Me-time: A randomized controlled trial on the effects of self-care interventions [Data set]. https://clinicaltrials.gov/ct2/show/NCT05802563?term=Me-timedraw=2rank=1

[5] Luwe, Y. J., Lee, C. P., Lim, K. M. (2022). Wearable Sensor-Based Human Activity Recognition with Hybrid Deep Learning Model. Informatics, 9(3), 56. MDPI AG. http://dx.doi.org/10.3390/informatics9030056

[6] Mekruksavanich, S., Jitpattanakul, A. (2021). Biometric User Identification Based on Human Activity Recognition Using Wearable Sensors: An Experiment Using Deep Learning Models. Electronics, 10(3), 308. MDPI AG. http://dx.doi.org/10.3390/electronics10030308

[7] Reimers, A. K., Knapp, G., Reimers, C. D. (2018). Effects of Exercise on the Resting Heart Rate: A Systematic Review and Meta-Analysis of Interventional Studies. Journal of clinical medicine, 7(12), 503. https://doi.org/10.3390/jcm7120503

[8] Retsinas, G., Filntisis, P. P., Efthymiou, N., Theodosis, E., Zlatintsi, A., & Maragos, P. (2020). Person identification using deep convolutional neural networks on short-term signals from wearable sensors. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (pp. 3657-3661). https://doi.org/10.1109/ICASSP40776.2020.9053910

[9] Santos, M. A., Sousa, A. C., Reis, F. P., Santos, T. R., Lima, S. O., & Barreto-Filho, J. A. (2013). Does the aging process significantly modify the Mean Heart Rate?. Arquivos brasileiros de cardiologia, 101(5), 388–398. https://doi.org/10.5935/abc.20130188

[10] Xu, Z., Zahradka, N., Ip, S., Koneshloo, A., Roemmich, R. T., Sehgal, S., Highland, K. B., & Searson, P. C. (2022). Evaluation of physical health status beyond daily step count using a wearable activity sensor. *npj Digital Medicine*, 5, 164. https://doi.org/10.1038/s41746-022-00696-5