Feature extraction and classification on heart rate time series for cardiovascular diseases

Michael Beekhuizen

TU Delft

Feature extraction and classification on heart rate time series for cardiovascular diseases

by

Michael Beekhuizen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday June 23, 2023 at 10:00 AM.

Bioinformatics specialization, Master Computer Science, EEMCS

Student number:	4895258	
Project duration:	October 28, 2022 – June 23, 2023	
Thesis committee:	Prof. dr. ir. M. J. T. Reinders,	TU Delft, supervisor
	dr. J. A. Martinez Castaneda,	TU Delft
	dr. D. M. J. Tax,	TU Delft
	ir. A. Naseri Jahfari,	TU Delft, daily supervisor
	ir. R. Ghorbani,	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Image cover source:
https://yqfile.alicdn.com/b5b533179374f3f2f592c862dfdc895569c5edc1.jpeg

TU Delft

Preface

Eight months ago I started on a journey to analyse and classify heart rate time series. During this time I learned and experimented a lot. Within this project, I mainly used two datasets: the BigIdeasLab_-STEP and the ME-TIME dataset. The first one consisted of heart rate time series of subjects performing different activities whereas the second dataset contained heart rate and step time series of subjects with or without a heart disease. The first dataset showed how important it was to transform the data in such a way to minimize the intra-subject variability. The ME-TIME dataset was used to determine if long-term heart rate series could be used for the detection of heart diseases. As it stands now, differences can be seen between subjects with or without heart diseases. However, it is not distinctive enough to use it as an automatic detection process in clinical practice. I liked experimenting with this data and finding new potential ways to find differences between the groups.

I would like to thank Marcel Reinders and Arman Naseri Jahfari for all their time and energy to guide me through this project and discuss work and new research directions. Moreover, I would like to thank my family and friends who listened to my ideas and comments.

Michael Beekhuizen
Delft, June 2023

Structure

This thesis report is split into two separate papers. Over the whole time span of the thesis project, we looked at the possibility of using heart rate time series for classification tasks like activity prediction and using long-term Fitbit data (heart rate and step time series) to classify and differentiate different heart diseases. To investigate the activity classification using heart rate time series we used the BigIdeasLab_STEP dataset. This dataset contains heart rate time series data which is annotated with different activities. For the classification and differentiation of heart diseases, we used the Fitbit data from the ME-TIME study. This dataset contains heart rate and step time series with a label that indicates if someone is healthy or has specific heart disease. The idea was to use the results from these first experiments as the basis for the other dataset with the long-term Fitbit data. Due to the nature of the Fitbit data (small number of samples and only one label per subject instead of per timeframe), the results were difficult to transfer to this new dataset and therefore new techniques have been used. To make all the results and conclusions as clear as possible for the reader it is split up into two papers. These papers can be found after this chapter and we will start with the paper discussing the classification of heart rate time series data using the BigIdeasLab_STEP dataset. Finally, the paper that utilises the Fitbit data and investigates the classification of heart diseases can be found.

Improving performance of heart rate time series classification by grouping subjects

Michael Beekhuizen (4895258)
Bioinformatics specialization
MSc Computer Science
EEMCS
Delft University of Technology
Graduation date: 23-06-2023

Abstract

This paper investigates the use of heart rate time series to perform activity classification. To test this, the BigIdeasLab_STEP dataset was used which includes heart rate time series with annotation of a specific task an individual performs. This was used to investigate if classification was possible in general.

The analysis showed a correlation between the window/stride size and the accuracy when performing classification on the BigIdeasLab_STEP dataset. Moreover, there was variability found between subjects due to differences in the physical structure of their hearts. Various techniques were used to minimize this variability. First of all, normalization proved to be a crucial step and significantly improved the performance. Secondly, grouping subjects and performing classification inside a group helped to improve performance and decrease inter-subject variability. Finally, handcrafted features in deep learning (DL) networks can improve the classification performance.

These findings indicate that heart rate time series can be utilized for classification tasks like predicting activity. Normalization or grouping techniques need to be chosen carefully to minimize the issue of subject variability.

1 Introduction

In recent years, wearable devices and smartwatches have been equipped with more sensors, including electrocardiogram (ECG) and photoplethysmography (PPG), for the detection of heart rate and heart rhythm [1]. These devices enable us to collect long-term heart rate time series data of a subject's heart rate in beats per minute (BPM). In this paper, we will look into the classification of heart rate time series data to predict different activities a subject is doing. This is performed to evaluate how well heart rate time series can be used for classification in general. In the research community, there are many papers that attempt to perform classification using ECG or PPG data. The time series in these datasets are typically short, ranging from one beat to a few seconds. Moreover, the signals from ECG or PPG sensors represent separate beats. Heart rate time series only have one value (heart rate in beats per minute) every x seconds. To research if this kind of data can be used for classification tasks, we will make use of the BigIdeasLab_STEP dataset which contains annotated heart rate time series of subjects performing different activities.

The paper is structured as follows. Section 2 discusses previous work on the classification of heart rate data. Next, section 3 show the results of all the different experiments and 4 interprets the results and states the limitations of the research. Section 5 gives a conclusion and section 6 goes deeper into detail about the dataset and models used in the experiments.

2 Related works

The use of wearables for long-term data acquisition is a relatively new field, and there has been limited research on predicting and analyzing this data. A paper written by Ballinger et al. [2], describes a model that predicts according to weekly data if a person has high cholesterol, hypertension, sleep apnea or diabetes. This data is generated by users that use an Apple Watch and are categorized according to a health survey in conjunction with a hospital. Another recent work investigated if wearable data could be used for sleep analysis [3]. They found that with heart rate time series in beats per minute, raw acceleration data of an Apple Watch and an estimated clock proxy, they could predict with 90% accuracy if someone was awake or sleeping. Moreover, an accuracy of around 72% was achieved when differentiating between awake, NREM, and REM sleep. In our paper, we will only use heart rate time series to perform classification. This is an important difference as in [3] they showed for the awake prediction the estimated clock proxy and acceleration data were more important. Dahalan et al. [4], showed how to do classification via a rule-based system. The input to this system was heart rate, age and fitness level. This method only used a single number as input and not a time series. A paper written by Maguire and Frisby [5], showed that activity classification with raw accelerometer data and heart rate data is possible. The classification was mostly possible due to the fact that the accelerometer was placed at strategic places to identify specific movements and the subjects were all the same age and fitness level. Our paper uses data from subjects that differ in age (from 18-54). The next section describes the experiment

and the results we achieved on the classification of the heart rate times series.

3 Results

For all the upcoming experiments we will make use of the BigIdeasLab_STEP dataset and will look into the classification of the heart rate time series. In short, the dataset contains around 13 minutes of heart rate time series data per subject. This dataset is annotated with the activity a subject is performing. The activities were: resting, breathing, performing an activity, resting after the activity and typing. The data is split up into windows and a specific stride is used between each window. A window size refers to the number of consecutive samples one takes from a certain start point. The stride indicates the number of samples the start point is shifted for the next window. A more detailed description of the dataset can be found in the methods, section 6.1. We start the results section by stating the influence of different window and stride sizes on the classification performance and in addition grouping the subjects before performing classification. Next, we analyze the performance of various deep learning models and investigate the addition of handcrafted features to the DL networks.

3.1 Comparison of different window and stride sizes

The first result that we can draw from the experiments is the change in performance when varying the window and stride sizes. This experiment was conducted by running a Support Vector Machine (SVM) model with data of different window and stride sizes. A short explanation of the SVM can be found in the methods, section 6.2. The window size varied between 50, 80, 100 and 120. The stride size varied between 10, 25, 40, 50, 80, 100 and 120. We performed the experiment twice. During the first time, we used a train and test set where some windows of a person were in the train set and some were in the test set. The second time we only used a train and test set where all the windows of a person were either in the train or the test set (leaving whole subjects out). The results for the first two experiments can be found in Figures 1 and 2.

As we can see in both figures, for every coloured line and thus every window size, the accuracy increases as the stride size decreases. Moreover, as the window sizes get larger, the accuracy also gets higher. However, there is a difference between the two figures. In the distinct train/test set case the accuracies seem to converge to one point or at least stabilise, whereas in the overlapping case, the lines show an overall increasing trend.

3.2 The effect of clustering subjects

In the previous section, we showed that classification with high accuracy is possible. However, a problem could arise because of the individual physical differences between subjects [6]. This could lead to a performance decrease and worse generalizability. To mitigate this, we conducted several experiments which looked at grouping the subjects in order to eventually gain performance during classification. The first experiment investigated the possibility to cluster the subjects

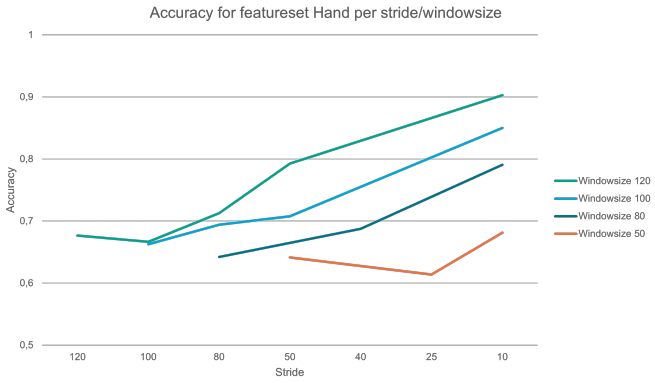


Figure 1: Achieved accuracy when training an SVM with overlapping train/test set with different window and/or stride size. The accuracy increases as the window size increase and stride size decrease. The achieved accuracies are plotted on the y-axis and the stride sizes are on the x-axis. The different window sizes are represented by different coloured lines.

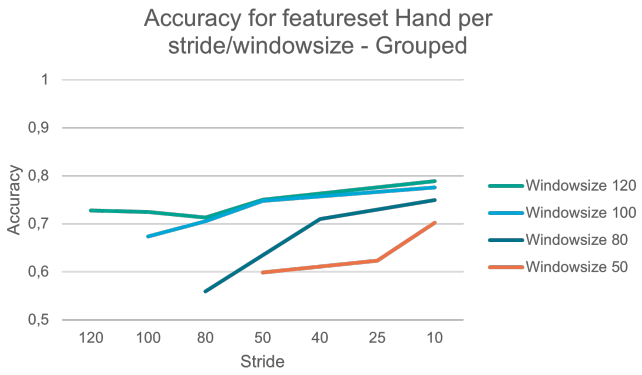


Figure 2: Achieved accuracy when training an SVM with distinct train/test set with different window and/or stride size. The accuracy seems to converge to one point. The achieved accuracies are plotted on the y-axis and the stride sizes are on the x-axis. The different window sizes are represented by different coloured lines.

based on their time series data. This was performed by calculating the average BPM of every activity and this resulted in five values per person. These five values represented a time series of five points in the exact same order as the activities performed: rest, breath, activity, rest, and type. When we clustered these time series per person with different resulting numbers of clusters, a cluster assignment as in Figure 3 was achieved.

To determine whether there were differences between the cluster groups, we trained an SVM on one cluster while another cluster was used as a testing set. The combinations and the corresponding scores achieved are represented in Table 1.

We can observe in this table that the clusters that look similar (eg. 1 and 2) achieve a better performance than clusters that look more dissimilar (eg. 1 and 5). This suggests that there exists inter-subject variability in this dataset.

To investigate the existence of variability within a clus-

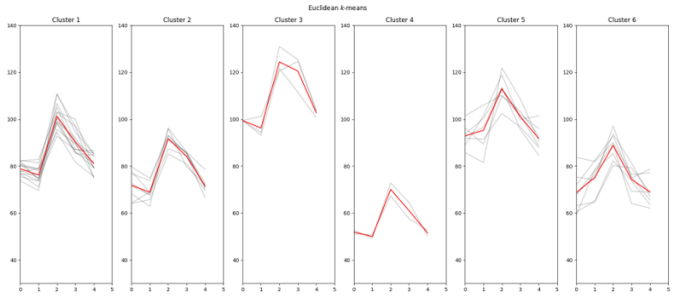


Figure 3: A cluster assignment of TimeSeriesKmeans clustering with the number of clusters equal to 6 using a time series of a subject's mean BPM per activity. Subplots from left to right represent the six different clusters and the subjects included. Grey lines represent the individual time series and thus represent a single subject. Red lines are the averages of the time series in the cluster. The x-axis shows the different activities numbered from 0 to 4 and the y-axis shows the heart rate in BPM.

Train x / Test y	Averaged balanced accuracy
Train 1 / Test 2	0.7261
Train 1 / Test 5	0.4035
Train 1 / Test 6	0.5724
Train 5 / Test 6	0.3572
Train 5 / Test 3	0.4464

Table 1: Accuracies of training an SVM and using one cluster as training set and another cluster as testing set. The numbers indicate the clusters in Figure 3 counted from left to right. Similar clusters achieve higher accuracy than more dissimilar ones.

ter/group, we trained an SVM on all the data in a cluster except for one subject, which was used for testing. We performed this for every cluster and for every combination inside a cluster. We considered two different standardization methods namely 'Feature' and 'Data' standardization. In Feature standardization, the features are calculated and then standardized on the training data. The parameters used for standardising the features of the training data are also used for the standardization of the features in the testing data. In Data standardization, the original time series is standardized per person and features are then calculated on this data. After the calculation of the features, no standardization is performed. The results of both methods can be found in Figures 4 and 5.

These figures show us that the Feature standardization case is performing better. In three out of four (larger) clusters, the average accuracy within a cluster is higher than the trained SVM on the 'full' train/test set. Conversely, only two out of four clusters perform better/above average in the Data standardization case. Clusters 3 and 4 contain an insufficient number of samples to provide an accurate representation

Next, we conducted an additional experiment to investigate if the clustering could be improved by using multiple features instead of only the mean heart rate per activity. To test this, we evaluated the within-cluster accuracies using different methods of clustering. The two different methods we investigated were the use of temporal features and the use of

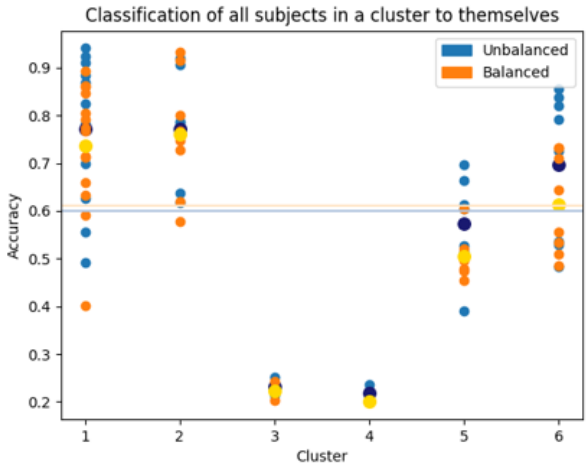


Figure 4: Results of accuracies within a cluster for the Feature standardization method when training an SVM with leave-one-subject out testing. Larger yellow and dark blue points represent the mean per cluster and horizontal lines represent the accuracy of the SVM when trained on a distinct train/test set. Light/dark blue represents unbalanced and yellow/orange represents balanced. In three out of the four larger clusters, the mean accuracy within a cluster is higher than an SVM trained on all the data.

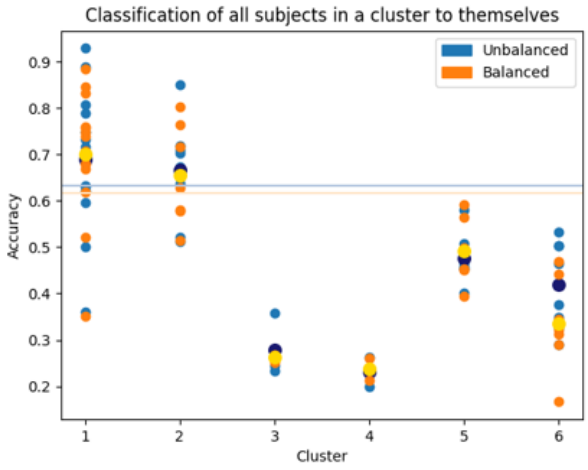


Figure 5: Results of accuracies within a cluster for the Data standardization method when training an SVM with leave-one-subject out testing. Larger yellow and dark blue points represent the mean per cluster and horizontal lines represent the accuracy of the SVM when trained on a distinct train/test set. Light/dark blue represents unbalanced and yellow/orange represents balanced. In two out of the four larger clusters, the mean accuracy within a cluster is higher than an SVM trained on all the data.

statistical features instead of mean heart rate. The results can be seen in Figures 6 and 7.

These figures show that the statistical features are better for clustering than the temporal features. In all large clusters, it achieves better performance than the SVM trained on the

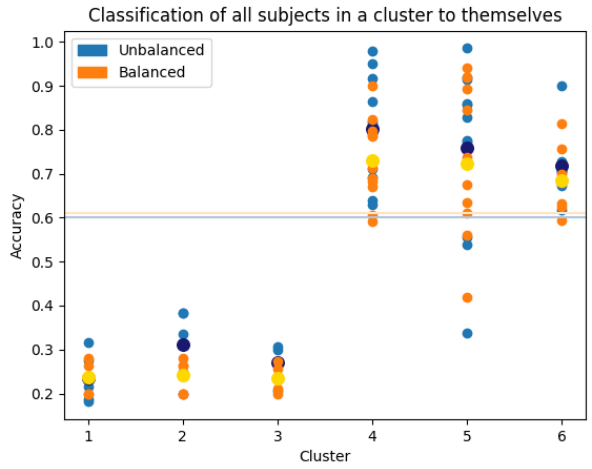


Figure 6: Results of accuracies within a cluster for the Feature standardization method when training an SVM with leave-one-subject out testing and temporal features for clustering. Larger yellow and dark blue points represent the mean per cluster and horizontal lines represent the accuracy of the SVM when trained on a distinct train/test set. In three out of the four larger clusters, the mean accuracy within a cluster is higher than an SVM trained on all the data.

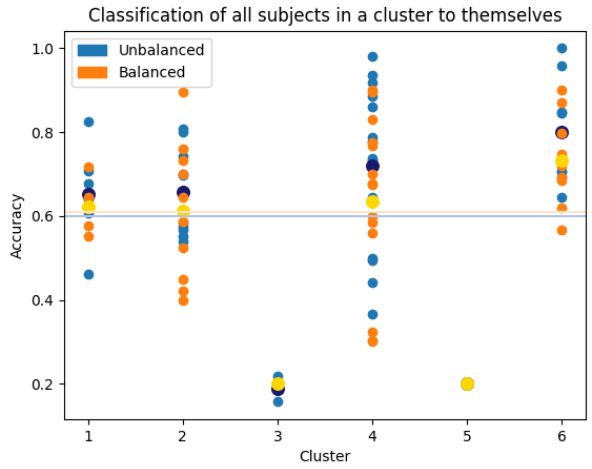


Figure 7: Results of accuracies within a cluster for the Feature standardization method when training an SVM with leave-one-subject out testing and statistical features for clustering. Larger yellow and dark blue points represent the mean per cluster and horizontal lines represent the accuracy of the SVM when trained on a distinct train/test set. In all of the four larger clusters, the mean accuracy within a cluster is higher than an SVM trained on all the data.

'full' train/test set. In the temporal case, this is only 3 out of 4 just like with the mean BPM clustering method.

To demonstrate that it can also help with previously unseen samples, we conducted several additional experiments. These experiments all make use of a distinct train and test set. We used the training set for generating the clustering model and cluster assignment, as well as to train a model for each cluster.

	Distinct Train Test	Distinct Train Test Majority vote
6 clusters	0.4633	0.7404
5 clusters	0.5038	0.6260
4 clusters	0.5582	0.7491
3 clusters	0.6837	0.7241

Table 2: Achieved accuracies when using different numbers of resulting clusters and clustering techniques to find a cluster model for activity prediction. Grouping the subjects in 4 clusters and using the Majority vote method achieves the highest accuracy.

The test set was used in two different ways. The first approach was per-window classification. With this approach, a window of a test subject was obtained, the corresponding cluster was determined, and the model associated with that cluster was used to classify the window. The results of this approach can be seen in the first column of Table 2.

The second approach was to group windows of one test subject. In this setting, the subject of origin for each window is known. First, we used the clustering model to determine to which cluster a single window belongs. We repeated this process for every window of a test subject. After this, the cluster with the highest number of assigned windows was used to obtain the model for classifying all windows of a specific subject. We called this the majority vote method. The result of this experiment is presented in the second column of Table 2. The majority vote method achieves higher accuracies than the per-window classification method. Comparing this to the SVM when using the exact same train and test set it achieves an accuracy of 0.7143 while the majority vote method with 4 clusters achieves 0.7491.

We demonstrated that using the majority vote method with subject grouping worked better than training an SVM with the identical train and test set. In the next paragraph, we delve more into the differences in prediction between both methods, rather than solely examining the achieved accuracies.

The confusion matrices of the two methods can be found in Figure 8. From this figure, we can make several observations. First of all, we can find the most prominent difference within the two largest classes (Rest and Activity). In the right figure, which represents the grouping case, we can see that the Rest class is misclassified as the Breathe or RestAC class. Conversely, in the left figure, which represents the SVM without grouping, the Rest class is not only misclassified as Breathe or RestAC but also as Activity. It is preferable for the Rest class to be only misclassified as Breathe or RestAC rather than Activity due to their closer proximity and the higher likelihood of confusion.

The second point that we can notice is the difference in the classification of the Activity class. In both cases, the Activity class is misclassified as Rest, Breathe or RestAC. However, the clustering model performs the classification better by having fewer mispredictions in general and less misprediction in Rest, which is the least probable among the three classes that are occasionally predicted instead of Activity. In

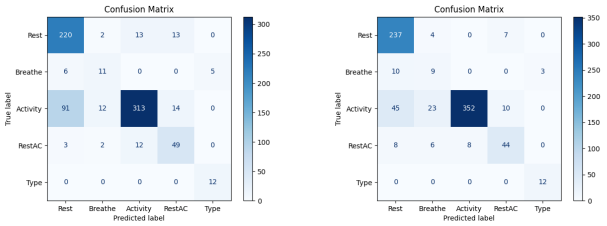


Figure 8: Confusion matrices of the SVM trained on the distinct train and test set (Left) and the SVM trained with grouping subjects and majority vote (Right). The true/actual labels are shown on the vertical axis and the predicted labels are on the horizontal axis. The biggest difference can be seen in the predictions of the Rest and Activity class.

general, we find that clustering primarily helps with reducing the misclassification of the Rest and Activity classes.

3.3 Deep learning with handcrafted features

Current research mostly focuses on deep-learning networks for feature extraction and classification. Especially in the field of heart rate variability analysis, there exist some standard features for measuring the variability. This inspired us to see if we could integrate handcrafted (HC) features within deep learning networks to see if it benefits from them. To this extent, we conducted several experiments to see if incorporating HC features in a DL model has any advantages.

First of all, we performed an experiment where the performance of the SVM has been compared against the deep learning models with and without HC features. The results can be found in Figure 9. The blue line is the baseline DL network that will get as input the standardized data. The red lines are the runs with an SVM and the green lines are the runs with the proposed DL networks. Models 1, 2, and 3 represent the three different DL models, which are explained in the method section 6.2. Models 1 and 3 make use of late integration and model 2 of early integration of the DL and HC features. The parameters column represents which feature set is used. The base feature set represents the basic HC features like max, min, mean, std and means of different (first and second-order) derivatives. Base and MFCC [7] represent the feature set where there are all the base parameters plus MFCC features. Statistical and temporal features are the features generated by TSFEL [8]. The column standardized indicates if the HC features are calculated on the standardized input or not. The raw heart rate time series data is always standardized. When we talk about standardized or non-standardized HC features in the next sections, we mean the features calculated on a standardized or non-standardized input.

This figure illustrates that the addition of HC features results in an increase in balanced accuracies in comparison to the DL baseline model in certain instances. Additionally, the top four accuracies are achieved without standardizing the HC features. Furthermore, every DL model outperformed the SVM.

Subsequent to this experiment, we investigated the usage of temporal and statistical features as an alternative to the base

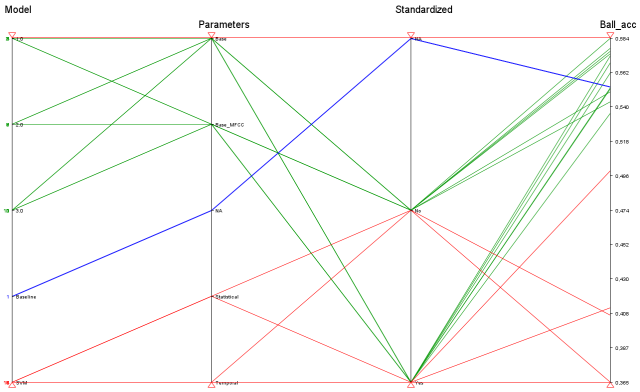


Figure 9: Achieved accuracies of different DL models with base HC features compared to DL baseline and SVM. The DL baseline is shown in blue, the SVM in red and the DL models with HC features in green. It can be noticed that all the DL models outperform the SVM and some DL models with HC features outperform the baseline.

set of HC features. Each DL model was trained with temporal or statistical features and with or without standardization. The outcomes can be found in Figure 10. What we can notice is that there were combinations that achieved higher performance than the DL baseline. A distinction was that among the top eight accuracies, five configurations employed standardized input. While in the previous experiment with the base features, the non-standardized HC features performed better. In addition to this, we combined both the statistical and temporal features into a single feature set, resulting in a slight improvement in performance to 58,84 % accuracy. Similarly, in this experiment, the standardized HC feature set worked better than the non-standardized one.

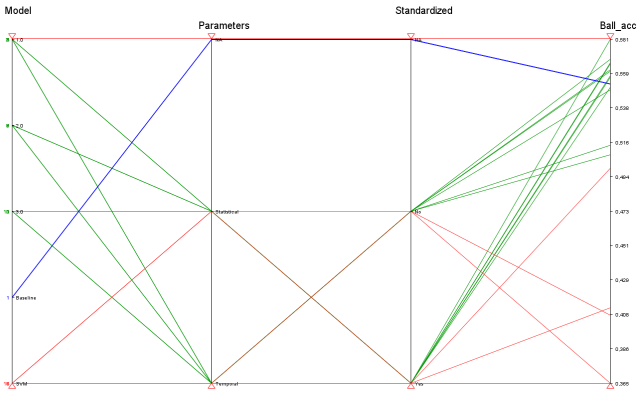


Figure 10: Achieved accuracies of different DL models with temporal and statistical features compared to DL baseline and SVM. The DL baseline is shown in blue, the SVM in red and the DL models with HC features in green. Again, it can be noticed that all the DL models outperform the SVM and some DL models with HC features outperform the baseline.

Besides solely examining the accuracies, it is relevant to investigate whether the HC features were indeed utilized by the DL model. To this extent, we used SHAP values to see how important the HC features are in addition to the raw input data. Figure 11 depicts the top 20 SHAP values with the highest importance. As we can see, the highest SHAP values correspond to an HC feature. Another observation is that primarily the beginning or end of the raw input is important (utilizing a window size of 50).

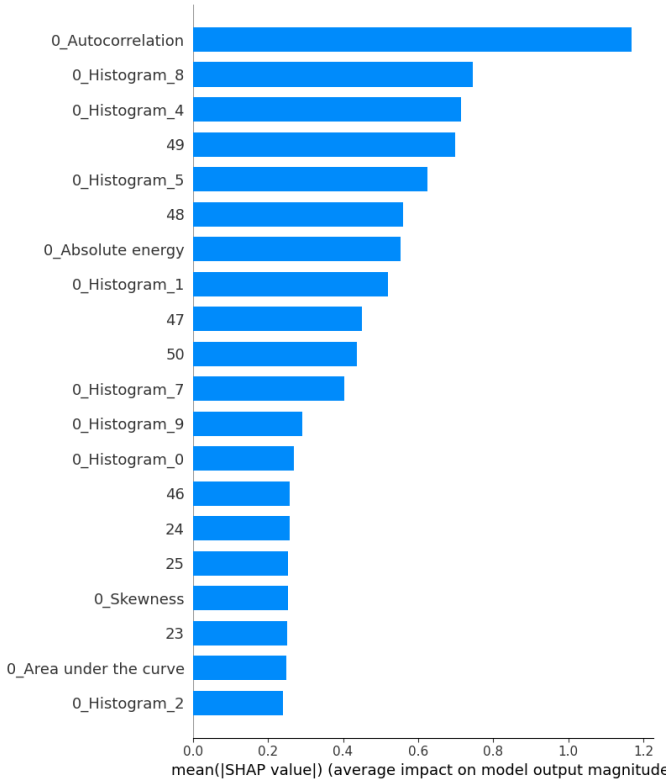


Figure 11: Resulting top 20 SHAP values of 3rd DL model with the addition of temporal + statistical features. On the x-axis the SHAP value is shown and on the y-axis the feature. Features starting with '0_' indicate HC features, and numbers indicate the point in the raw time series input.

Furthermore, we investigated the importance of the standardization technique. We already observed that standardizing has an influence when using temporal and/or statistical features. The standardization we used in the previous experiments transformed the time series per person in such a way that the mean is zero and the variance one. Next to this technique, we evaluated two other standardization techniques in this paper. The first one was performing Z-standardization (mean zero, variance one), but then on the concatenated time series of all the training persons, and subsequently utilising these parameters to transform the testing set. The second standardization method involved selecting the 10% quantile per person and employing this value instead of the mean value in the standardization formula. In addition, the Mean/Median absolute difference between every other value in the time series and the 10% quantile value per person was calculated and used as the variance in the standardization formula. Results can be found in Tables 3 and 4. We can observe that the first alternative standardization technique is ineffective, resulting in a decrease in performance of approximately 10%. The second alternative method works better and achieves a performance increase of around 1-2%.

Model	Parameters	Standardized	Ball_acc
Baseline	NA	NA	.5532
2	Statistical	Yes	.5809
2	Stat + Temp	Yes	.5844
1	Stat + Temp	Yes(New)	.4782
2	Stat + Temp	Yes(New)	.4786
3	Stat + Temp	Yes(New)	.4624

Table 3: Resulting accuracies of the first alternative standardization technique compared to the Z-standardization method. The new method is evaluated on all three DL models with statistical + temporal features. As can be seen in the table, the accuracy does not increase.

Model	Parameters	Standardized	Ball_acc
Baseline	NA	NA	.5532
2	Statistical	Yes	.5809
2	Stat + Temp	Yes	.5844
1	Stat + Temp	Yes(New/Mean)	.5775
1	Stat + Temp	Yes(New/Median)	.5471
2	Stat + Temp	Yes(New/Mean)	.5897
2	Stat + Temp	Yes(New/Median)	.5623
3	Stat + Temp	Yes(New/Mean)	.5918
3	Stat + Temp	Yes(New/Median)	.5675

Table 4: Resulting accuracies of the second alternative standardization technique compared to the Z-standardization method. The new method is evaluated on all three DL models with statistical + temporal features. For the second alternative standardization technique, both the Mean and Median variant are examined. This second alternative standardization technique seems to achieve higher accuracies.

Finally, we examined how different window and stride sizes affect the performance of the DL models. We evaluated the performance with a window size of 120 and stride 10, window 100 stride 10, window 80 stride 10 and the original window 50 stride 25. A problem that is arising is that the BigIdeasLab_STEP database does not have enough data for the Type class to make this work. For this experiment, we decided that it was best to merge the Type class with the RestAC class given that they are one after another in the time series and look similar. So now the dataset has four different classes remaining instead of five. To make a fair comparison

we evaluated every configuration of window and stride size on all the three different DL models that use HC features, the baseline DL model and an SVM. The results can be found back in Table 5. What we can observe from these results is that the performance of the models increases when going from the original 50_25 to the larger 80_10. After this, the performance stays around the same or decreases slightly for the DL models that make use of HC features. Furthermore, the DL models all achieve better accuracy than the SVM. Lastly, the second DL model which makes use of HC features and a window size of 80 and stride of 10, performs the best compared to all the models and configurations.

	50_25	80_10	100_10	120_10
1st model	.7013	.7577	.7592	.7170
2nd model	.7072	.7711	.7697	.7408
3rd model	.6918	.7571	.7519	.7423
SVM	.6216	.6877	.7102	.7090
DL Baseline	.7079	.7463	.7594	.7567

Table 5: Resulting accuracies of the three different DL models, DL baseline, and SVM when using the BigIdeasLab_STEP dataset with 4 classes (combining RestAC and Type class) and varying the window/stride size. The DL networks which uses HC features perform better with a window size of 80 and stride 10.

3.4 Misclassification with DL models

In order to evaluate the performance of the DL model and identify misclassified segments of the signal, we generated plots to visualize the misclassified parts and their predicted classes. An example of such a plot can be found in Figure 12. In this figure, parts that are correctly classified are depicted in black and the parts that have been misclassified are depicted in the colour of the predicted class. The grey vertical lines denote the change between one activity/class with the next one.

What we can see in this figure is that most of the misclassifications happen at the border (around the grey vertical lines). Moreover, most of the mispredicted parts have the predicted class label of their direct neighbouring class. This could mean that there is a mislabelling in the dataset itself or that the distinction between multiple classes is overlapping and therefore difficult to predict accurately.

4 Discussion

We looked into the classification of activity with the use of solely heart rate time series. The result section showed some interesting points. First of all, there seems to be a relation between the window and stride size when performing classification on time series data. The higher the window size and smaller the stride, the higher the accuracy. This could be explained by the information that is encapsulated when changing these parameters. A bigger window size means more information in one piece. The smaller stride sizes ensure that

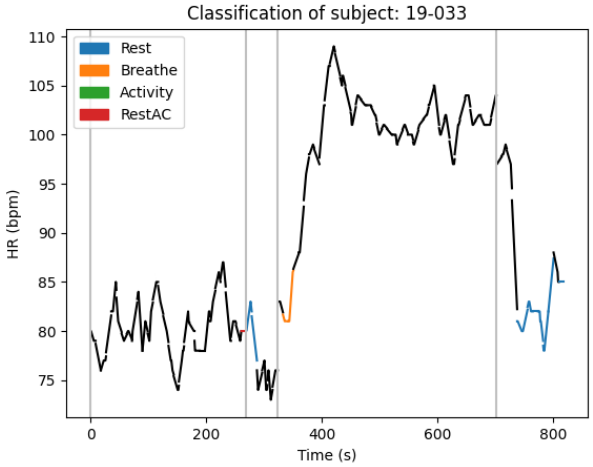


Figure 12: Plot which visualizes the misclassification of the DL model. Black lines correspond to rightly predicted classes and a coloured section indicates a misprediction of a specific class. The colour corresponds to the class that is mispredicted. The grey vertical bars correspond to a change in activity/class.

there is more overlap between the frames. These two parameters could then help the classifier to perform better. Secondly, grouping subjects also seems to improve the performance in the context of heart rate time series classification. It improves performance when determining with multiple windows of a subject to which group a subject needs to be assigned. When determining this with only a single window the performance decreased. So reducing the subject variability by grouping subjects together and performing classification within these groups will be a good idea when dealing with similar data. In a different dataset where the variability could be even bigger, this method could improve the performance significantly. Thirdly, deep learning combined with handcrafted features improves performance when classifying heart rate time series. By adding features manually, the network learns to find patterns in the time series itself but also uses some of the given HC features to make a decision. This result supports the finding in [9]. However, in that paper, they use a DL network for feature extraction and afterwards a feature selection and aggregation method to concatenate features. In this paper, the DL network will integrate the HC features with self-learned features. In addition, we saw when performing classification, normalization is an important step. Especially in the case of heart rate time series where the morphology differs between people. During classification with a DL network, the mean quantile normalization worked better than the standard normalization or no normalization. This could be explained by the fact that for every person we take into account the base heart rate and the deviation in the time series with this specific value.

All these points have a common limitation and that is the data that was available. This is particularly true for the grouping finding. Due to the limited number of subjects available, we did not have enough subjects in some situations to do a proper train/test split within a cluster. However, in the groups

where there were enough subjects, a positive increase in performance was shown. Moreover, the experiments have only been conducted with one dataset. In the future when other larger heart rate time series datasets are available, the experiments could be performed again to see if the same conclusions hold. Furthermore, different normalization techniques can be designed to even further reduce the variability among the subjects. This could involve data like age or fitness to create a potentially stronger normalization method.

5 Conclusion

The aim was to find out if the heart rate time series could be used for the classification of different activities using the BigIdeasLab_STEP dataset. First of all windowed data were used to predict which activity a person was performing. The results showed that there were significant differences between the heart rate time series of different subjects due to morphological differences. This resulted in a better performance when creating classifiers for groups of subjects with similar characteristics. Furthermore, when using deep learning networks an improvement was seen when adding handcrafted features internally instead of only giving the heart rate time series to the network. Moreover, the network's performance increased more by using another standardization method which took into account the resting heart rate per person. All in all, heart rate time series can be used for the classification task of predicting a specific activity, but the subject variability should be taken into account by utilizing techniques like grouping or normalizing.

6 Methods

6.1 BigIdeasLab_STEP

In this paper we used the BigIdeasLab_STEP dataset from PhysioNet [10]. This dataset includes data from 53 participants and was recorded in July-August 2019. The age of the participants ranged from 18 to 54. Each person needed to perform three study protocol rounds with different types of wearables. One study protocol round consisted of five activities in the following order:

1. Seated rest (4 min)
2. Paced deep breathing (1 min)
3. Physical activity (5 min)
4. Seated rest (2 min)
5. Typing (1 min)

In the experiment, every person wore all the available devices spread over multiple rounds: Empatica E4, Apple Watch 4, Fitbit Charge 2, Garmin Vivosmart 3, Xiaomi Miband and Biovotion Everion. During the whole experiment, the participant always wore an ECG device (Bittium Faros 180) as a reference.

The dataset consists of a synchronised heart rate value in bpm between the smartwatch and the ECG device. Moreover, it is annotated with one of the five activities the person is performing. In the dataset, this is denoted by the labels Rest, Breathe, Activity, Rest after Activity (RestAC) and Type. In the experiments, only the heart rate data is used of the Apple Watch because of its strong correlation with the heart rate time series of the ECG ground truth in comparison with the other wearables.

6.2 Classification models

In several experiments, we used a support vector machine (SVM). An SVM tries to maximize the margin between two classes. The SVM maximizes the generalization of a model [11]. For multiclass classification one can use multiple binary SVMs. Two of the methods used for this are One-against-all and one-against-one [12]. For the experiments, we used the implementation provided by scikit-learn, which uses the one-against-one method [13].

In addition to the other experiments, we researched the influence of the addition of handcrafted (HC) features with deep learning models. To investigate this, three different DL models with the addition of HC features were used alongside a DL baseline. All of the DL models started with a 1-D convolution and had three or four fully connected layers.

The baseline model starts with a 1-D convolution where the raw sequence input will be processed. Next, it goes through a ReLU, Dropout and Max pooling layer and finally, a flatten layer. After the flattening, it is processed by a fully connected layer, followed by a ReLU and a last fully connected layer to bring the output dimension to the required number of classes. A high-level graphical overview can be seen in Figure 13a.

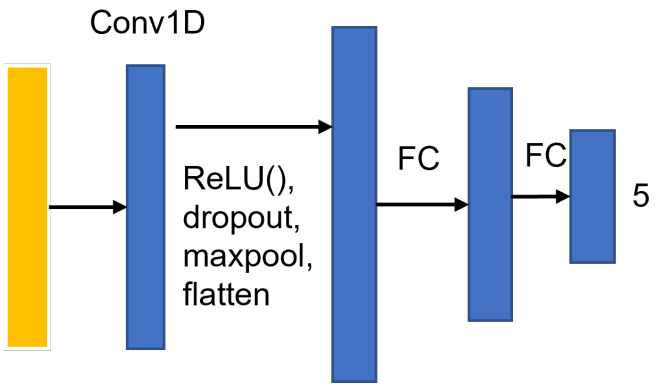
The first model is highly similar to the baseline model but it adds an extra layer between the last two fully connected layers. So after the first fully connected layer after flattening, the model adds the HC features to the output of this layer. Next, it processes through another fully connected layer and thereafter it goes through the last fully connected layer. This layer ensures that it ends with the correct dimension. A simple graphical representation can be found in Figure 13b.

The second model is integrating the HC features directly at the beginning of the DL model. This is achieved by concatenating the HC features with the raw time series input. This results in a larger input vector than with the previous model. The third model is very similar to the first one but with one addition. Instead of adding the HC features directly to the output of the fully connected layer, the HC features first go through a fully connected layer and this output is connected to the output of the first fully connected layer of the model. A graphical representation of both models can be found in Figure 14.

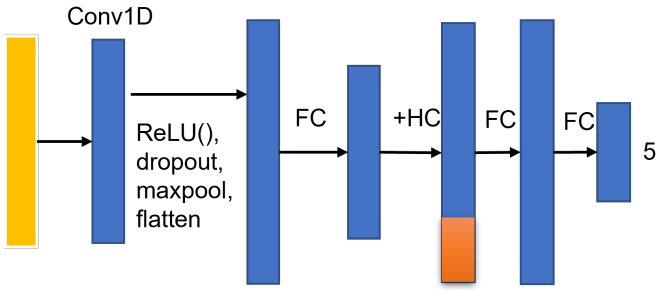
References

[1] J. Torres-Soto and E. A. Ashley, "Multi-task deep learning for cardiac rhythm detection in wearable devices," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–8, 2020.

[2] B. Ballinger, J. Hsieh, A. Singh, N. Sohoni, J. Wang, G. H. Tison, G. M. Marcus, J. M. Sanchez, C. Maguire, and J. E. Olgin, "Deepheart: semi-supervised sequence learning for cardiovascular risk prediction," in *Thirty-Second AAAI Conference on Artificial Intelligence*, Conference Proceedings.

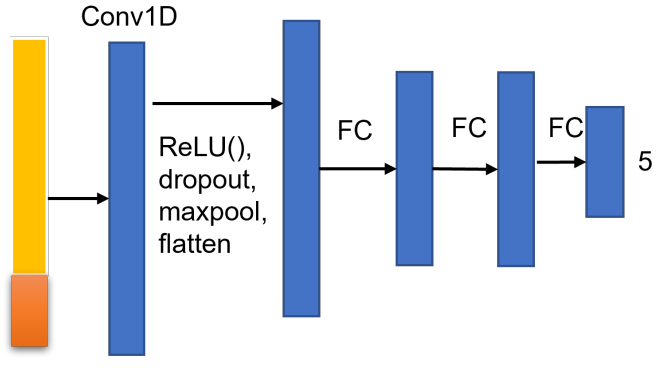


(a) baseline network

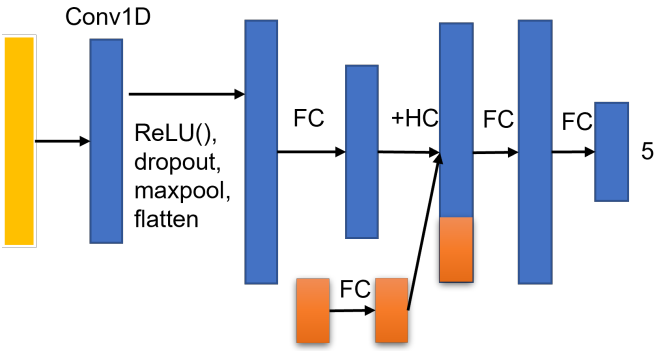


(b) First network

Figure 13: High-level overview of the internal working of the baseline DL model(a) and the first DL model (b) that make use of HC features. Yellow represents the raw input sequence and orange represents the HC features.



(a) Second network



(b) Third network

Figure 14: High-level overview of the internal working of the second(a) and third(b) DL model that makes use of HC features. Yellow represents the raw input sequence and orange represents the HC features.

[3] O. Walch, Y. Huang, D. Forger, and C. Goldstein, "Sleep stage prediction with raw acceleration and photoplethysmography heart rate data derived from a consumer wearable device," *Sleep*, vol. 42, no. 12, p. zsz180, 2019.

[4] A. J. Dahalan, T. R. Razak, M. H. Ismail, S. S. M. Fauzi, and R. A. J. Gining, "Heart rate events classification via explainable fuzzy logic systems," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 4, p. 1036, 2021.

[5] D. Maguire and R. Frisby, "Comparison of feature classification algorithm for activity recognition based on accelerometer and heart rate data," in *9th. IT T Conference*, Conference Proceedings, p. 11.

[6] J. Niu, Y. Tang, Z. Sun, and W. Zhang, "Inter-patient ecg classification with symbolic representations and multi-perspective convolutional neural networks," *IEEE journal of biomedical and health informatics*, vol. 24, no. 5, pp. 1321–1332, 2019.

[7] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C. Xu, and Q. Tian, "Hmm-based audio keyword generation," in *Advances in Multimedia Information Processing-PCM 2004: 5th Pacific Rim Conference on Multimedia, Tokyo, Japan, November 30-December 3, 2004. Proceedings, Part III 5*. Springer, Conference Proceedings, pp. 566–574.

[8] M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Liu, T. Schultz, and H. Gamboa, "Tsfel: Time series feature extraction library," *SoftwareX*, vol. 11, p. 100456, 2020.

[9] A. S. Eltrass, M. B. Tayel, and A. I. Ammar, "Automated ecg multi-class classification system based on combining deep learning features with hrv and ecg measures," *Neural Computing and Applications*, vol. 34, no. 11, pp. 8755–8775, 2022.

[10] B. Bent and J. Dunn, ""bigideaslab_step": Heart rate measurements captured by smartwatches for differing skin tones" (version 1.0)," https://doi.org/10.13026/cqfy-d860.

[11] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[12] E. Mayoraz and E. Alpaydin, "Support vector machines for multi-class classification," in *Engineering Applications of Bio-Inspired Artificial Neural Networks: International Work-Conference on Artificial and Natural Neural Networks, IWANN'99 Alicante, Spain, June 2–4, 1999 Proceedings, Volume II*. Springer, 2006, pp. 833–842.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Acceleration-deacceleration curves on heart rate time series for arrhythmia detection

Michael Beekhuizen (4895258)
Bioinformatics specialization
MSc Computer Science
EEMCS
Delft University of Technology
Graduation date: 23-06-2023

Abstract

In this paper, we investigated the utilization of long-term heart rate time series to perform classification and distinguish individuals with or without a heart disease like atrial fibrillation. To test this we made use of the long-term Fitbit data from the ME-TIME study. This dataset includes heart rate and step time series from 22 subjects with or without heart disease. The length of the data ranged from several weeks to two years.

Analysis of the long-term Fitbit data showed that there is a difference between individuals based on their health condition. However, this difference was only apparent when specific manipulations to the data were performed such as Onset-Recovery alignment of peaks and mean inactive peak normalization. In addition, the classification of individual peaks was possible and worked best when utilizing a time series-specific support vector machine and grouping peaks together. Grouping peaks per week from a person and calculating a percentage of heart disease-predicted peaks also worked relatively well to distinguish between heart disease and reference subjects. An issue with classification and finding differences between subjects was the subject variability due to the physhical differences between hearts. Normalization proved to be a crucial step to minimize this issue and moreover improved the performance. The best normalization method for this data was the mean inactive peak normalization.

These findings indicate that long-term heart Fitbit data can be utilized for the detection and classification of heart diseases. To make it work, normalization techniques need to be chosen carefully to minimize the issue of subject variability. However, due to the small dataset size and classification results achieved, the method is not ready for clinical practice but a proof of concept is shown.

1 Introduction

Cardiovascular diseases (CVD) are one of the primary causes of mortality worldwide [1]. To find such diseases, doctors make use of an electrocardiogram (ECG). This tool is used to find cardiac anomalies, functional disorders, and cardiac arrhythmias [2]. One specific type of CVD is atrial fibrillation (AF). A problem with AF is that it is difficult to detect and diagnose because it could be the case that AF is only 'active' during a specific period. Even with a 72-hour ECG monitor, the discovery rate is around 6.1% [3]. To overcome this issue of a limited time frame, a device that is easy and non-disturbing to wear should be used. In the last couple of years, wearable devices and smartwatches have been getting more sensors, including ECG and photoplethysmography (PPG) for the detection of heart rate and heart rhythm [4]. By using the data from these sensors, the goal is to predict if and preferably when a person has an abnormal rhythm and of which type. In the research community, there are a lot of papers written that try to predict with ECG or PPG data which type of arrhythmia a person has. The time series in these datasets are usually short, from one beat to a couple of seconds. In this paper, we will make use of long-term Fitbit data to investigate if different types of arrhythmias can be detected. This is different from other research where they use signals from ECG or PPG sensors which represent separate beats. Heart rate time series only have one value (heart rate in beats per minute) every x seconds. It is important to research this because as earlier stated AF for example is only 'active' during a specific period. Having data over a longer time span can help increase the detection rate.

To answer the questions of if and how long-term Fitbit data could be used for arrhythmia detection, we conducted several experiments. First of all, analysis is conducted on the data to find interesting patterns and afterwards examined if these are useful for prediction tasks. This analysis consists of finding acceleration-deacceleration curves and performing manipulations on them. Next, different methods are evaluated to classify if a specific time series is from a healthy person or not. For all these experiments, we made use of the Fitbit data from the ME-TIME study. This dataset consists of long-term heart rate and step time series data gathered from different persons where the time series ranged from weeks up to two years. The subjects belonged either to the reference or heart disease group.

The paper is structured as follows. Section 2 discusses previous work on the classification of data in relation to heart diseases. Section 3 show the results of all the different exper-

iments and 4 interprets the results and states the limitations of the research. Next, section 5 gives a conclusion and finally section 6 discusses the dataset and techniques used in the paper.

2 Related works

ECG recordings are used to diagnose heart diseases both by cardiologists and algorithms [5]. A frequently used dataset is the MIT-BIH dataset [6]. This dataset consists of 48 half-hour excerpts of two-channel ECG recordings and describes five different classes. A limitation of the algorithms trained on this dataset is that they only work with ECG samples of short duration. For example, the algorithm developed in [7], makes use of segments of data that are only around 175 ms long. With wearables, one is able to gather long-term data on the heart rate of a person together with other data such as activity or steps. The use of wearables for long-term data acquisition is a relatively new field with limited research on predicting and analyzing this data. Ballinger et al. [8] proposed a model that predicts high cholesterol, hypertension, sleep apnea or diabetes based on weekly data generated by users who use an Apple Watch. The users are categorized according to a health survey in conjunction with a hospital. Another recent work investigated the use of wearable data for sleep analysis [9]. The study found that with heart rate time series in beats per minute (BPM) and raw acceleration data of an Apple Watch, they could predict with 90% accuracy if someone was awake or sleeping. Moreover, the study achieved an accuracy of around 72% when differentiating between awake, NREM, and REM sleep.

There has been little research into using long-term heart rate signals in BPM to perform heart disease classification. Moreover, what is missing from the previous works is interpreting the data they use by identifying interesting patterns. A paper written by Al-Makhadmeh et al. [10] is performing classification to see if a patient is healthy or not but is not solely using heart rate in BPM but also several other features like ECG segments. Another paper, [11], is explaining that there is a need for more long-term analysis/classification methods. It concludes that when using heart rate features of the instantaneous heart rate derived from ECG data, an support vector machine (SVM) can still perform well for the prediction of AF episodes. This is an indication that heart rate values for long-term data gathered by wearables could help with the prediction of heart diseases. Hochstadt et al. [12], created an algorithm that detects AF using continuous heart rate monitoring. This algorithm uses peak-to-peak interval sequences derived from PPG or ECG wearable data. Another paper written by Ford et al. [13] is reviewing two algorithms for wearables that predict AF. The algorithms used in the paper make use of 30-second ECG recording from wearables and also not long-term heart rate data. The same holds for [14] where the authors make use of 1-minute PPG data to predict AF. They speculate that this will open doors for long-term AF monitoring/prediction. That is why in this paper, we analyze long-term heart rate data to find out if classification is possible.

3 Results

In this section, we will focus on the long-term Fitbit dataset from the ME-TIME study. The dataset contains the heart rate (in beats per minute every 5 seconds) and step data of 22 subjects belonging to either the reference or heart disease group. The data ranged from a couple of weeks to two years. A more detailed description of the dataset can be found in the method section 6.1. First of all, we performed an analysis on the data and determined features for it. Secondly, we used these features to perform classification and investigate if it is possible to distinguish between a reference and a disease group. Finally, we performed outlier detection and grouping on the data to see if this can increase the detection rate of heart diseases. We start by describing the results of examining the characteristics of the data and determining which parts/features were practical to utilise.

3.1 Peak alignments and analysis

In most of the following experiments, we made use of data from a cardiac event where the heart rate increases and after peaking, recovers back to some baseline. In this paper, we characterised these curves by three fiducial points, namely the onset, peak and recovery point. We refer to these as acceleration-deacceleration curves. We made use of a function that detects the peaks and then we performed several steps to find the corresponding onset and recovery points. This process can be found in the methods section 6.2. An example signal where a peak is detected together with an onset and recovery point can be seen in Figure 1.

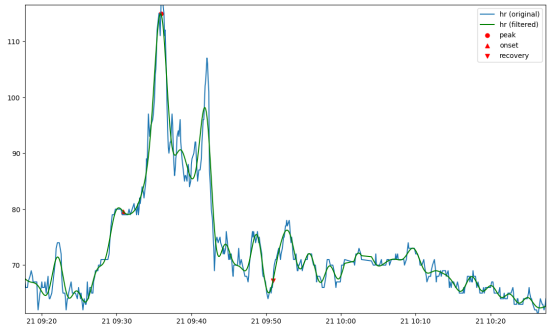


Figure 1: Plot that shows the onset, peak and recovery point for an example peak.

All these peaks or points can be aligned with each other in order to find patterns and differences. In the experiments, we made use of two different types of alignments. The first type aligns all the peaks according to their peak time. The second alignment method is the Onset-Recovery alignment, which aligns all the onset points and recovery points with each other. Besides using different alignment methods we also used two different normalization methods. The first method was the peak normalization and the second one was the mean inactive peak normalization. Further explanations and visualization about these alignment and normalization methods can be

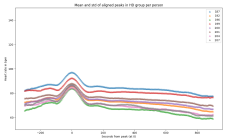
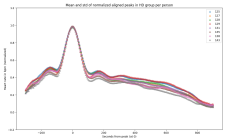
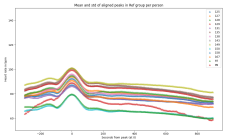
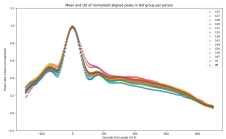
Group	Raw	Peak normalized
HD		
Ref		

Table 1: Plots of the aligned peaks for all the subjects in the HD group (Top) and the Ref group (Bottom) while using raw or peak normalized data. The x-axis shows the time points in seconds before and after the peak and the y-axis the (normalized) heart rate in BPM. Mean values are taken per time point for every subject.

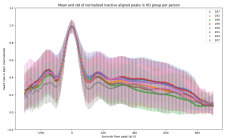
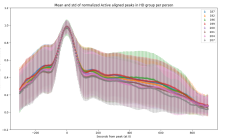
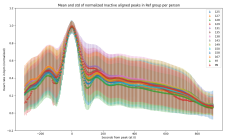
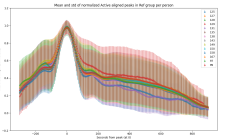
Group	Inactive	Active
HD		
Ref		

Table 2: Plots of the peak aligned and peak value normalized peaks for all the subjects in the HD group (Top) and the Ref group (Bottom) during activity and inactivity. The x-axis shows the time points in seconds before and after the peak and the y-axis the normalized heart rate. The mean and standard deviation are visualized per time point for every subject.

found in the method section 6.2.

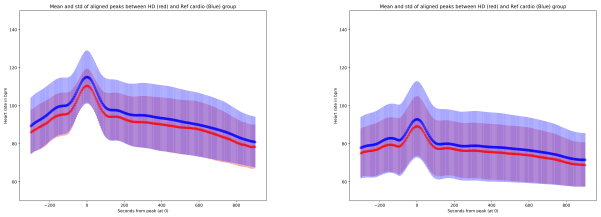


Figure 2: Plot that shows the differences between the peak aligned HD (Red) and Ref (Blue) peaks with a peak value greater than 100 BPM (Left) and 40 BPM (Right). The x-axis shows the time points in seconds before and after the peak and the y-axis the heart rate in BPM. For both groups, the mean and standard deviation is shown at every time point.

In Figure 2 left, we can see the resulting plot of performing the peak alignment method when plotting the mean values at every timestamp for the HD (Red) and Ref (Blue) group where the peaks have a value of 100 BPM or higher. What can be seen from this figure is that the mean heart rate of the HD class is always underneath the Ref class, but it follows roughly the same pattern. Moreover, the variance is higher within the Ref group than the HD group. Examining the right plot in Figure 2, in which all the peaks are taken into account (higher than 40 BPM), we can see a similar pattern to that observed in the previous figure. One notable difference is that for both groups the overall mean heart rate drops due to the inclusion of all the peaks.

Instead of only analysing the two groups as a whole, we examined also individual subjects within each group. To assess variations among the subjects, we generated plots which visualized all the subjects in the two groups using both raw time series data and peak normalized data. These plots are presented in Table 1.

We can see from the first column of Table 1 that the mean peak heart rate value for the reference group is indeed higher (around 90 BPM) compared to the HD group (around 80 BPM). Moreover, we can observe that there is some intra-group variability. Within the HD group, two subjects have a higher mean heart rate value/pattern when looking at most subjects. The same can be said for the reference group. In this group, two persons have a considerably lower mean heart rate value compared to the other subjects in the group. Therefore, in both groups there exists some variability. The second column of Table 1 shows the plots with peak normalized data. We performed this normalization to minimize the variability among the subjects which should help to find potential patterns related to heart diseases in the data. Examining the two plots show that there is not that much difference between the two. A difference that we can notice is with the Ref case, there are two subjects in the recovery phase that follow a different pattern.

A limitation of Table 1 is that it incorporates all the different peaks when a subject is active or inactive. To mitigate this we created Table 2 which presents plots where there is a distinguishing made between active and inactive while still utilising the peak normalization.

We can see several differences between these plots. First of all, a difference in the onset phase is apparent when we compare the inactive and active peak alignment plots. The pattern of the active peak alignment rises, flattens around -200, and then rises again towards the peak. In contrast, the pattern of the inactive peak alignment rises, decreases around -200, and then rises again towards the peak. Another difference can be seen in the recovery phase. Approximately 100 seconds after the peak in the inactive case, the signal is having a local minimum. Furthermore, the variation between subjects in the Ref group is larger than between the subjects in the HD group. Overall, it is hard to detect substantial differences between the HD and Ref group.

Table 3 shows the differences between active and inactive peaks when using the Onset-Recovery alignment method with peak normalization. We can see a subtle difference in

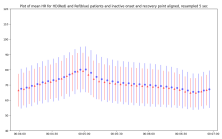
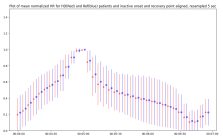
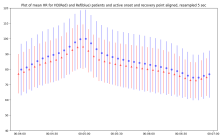
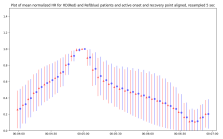
Activity	Raw	Peak normalized
Inactive		
Active		

Table 3: Plots of all the onset-recovery aligned peaks in the HD and Ref group during inactivity and activity for raw data (first column) or peak normalized data (second column). The x-axis shows the time in minutes and seconds and the y-axis the heart rate in BPM. The mean and standard deviation are visualized per time point for both groups.

Constraints		**Value**
Minimum peak height		100 bpm
# steps 7 minutes after peak		$<$10
# steps 4 minutes before peak	Inactivity	0
	Light activity	1-20
	Higher activity	$>$20

Table 4: The constraints used for selecting peaks to generate figures in Table 5.

the second column during the 1 minute after the peak where the HD line is marginally below the Ref line for both the inactive and the active case. Another point that we can observe is that the mean values in the active plot are higher than in the inactive plot. Examining the first column, where the raw data is used, several observations can be made. First of all, a similar pattern is visible in both cases. During the inactive case, the difference between the HD and Ref group is small, but the same 1-minute decrease pattern in the recovery phase is still visible. The most considerable difference is observed in the active case, where there is a larger gap between the HD and Ref line. Moreover, the HD line is fully under the Ref line.

Cardiologists often use a stress test or treadmill exercise to perform diagnosis [15; 16]. To mimic this behaviour, we divided the data into three different groups (0, 1, and 2) and selected only peaks if there were no more than 10 steps in the 7 minutes after the peak. This relates to inactive behaviour after a peak/activity. Group zero described peaks where there was no activity in the four minutes before the peak, group one described peaks with 1 to 20 steps in the four minutes before the peak and group two described peaks with more than 20 steps before the peak. We can see this as peaks with no, light or higher activity. Information about selecting this threshold can be found in the Supplementary, section A. The constraints we used can be found back in Table 4. The following plots all employ the Onset-Recovery alignment method and the mean inactive peak normalization method to better mitigate the subject variability. To even further disentangle the variability we divided the HD group into three groups, Paroxysmal atrial fibrillation (PAF), persistent atrial fibrillation (PerAF) and heart failure (HF). Table 5 presents all plots for the various activity levels and subject groups.

In these plots, we can find more interesting patterns. Starting with the first row, in the light activity case, we can observe that the HD group is most of the time above the Ref group and has a higher peak relative to its baseline. In the recovery phase, the HD group is slower during the recovery which can be noticed from around 5:30 onwards, where the distance between the HD and Ref group increases. Examining the plot with the higher activity peak, we can observe a similar pattern in the recovery phase, but slightly tuned down. Furthermore, there is only a small difference during the beginning of the onset phase where the HD group is a little higher than the Ref group, but overall they follow the same pattern. Lastly, examining the inactive case, we see that the HD line is during the onset phase crossing the reference group by starting lower, but ending up slightly higher during the peak. The most significant difference can be observed in the recovery phase where the HD group is fully under the Ref group with a distinct pattern. We can see that the HD group initially decreases during recovery and then stabilizes before decreasing again. This is in contrast to the Ref group, which goes down more gradually.

In the next sections, we examine the three different groups within the HD group. We compare the new situation against the full HD group. These figures can be found in the first row of Table 5. Starting by looking at the PAF subjects during higher activity reveals that there is a difference around the peak value. In the PAF case, the mean values around the peak are lower compared to the Ref group and the full HD group. The rest of the time it follows a similar pattern as the full HD group. Examining the low activity and inactivity peaks, we can observe that the PAF group follows a similar pattern as in the full HD case, but the distance between the Ref group becomes smaller.

We observe more substantial differences in the subjects with persistent atrial fibrillation, visualized in Table 5 third row. Most of the observed differences we see during inactivity in the onset and recovery phases. In the first 30 seconds in the onset phase, the PerAF subject's values are significantly lower than the reference group. Around the peak value they are around the same but in the recovery phase, the PerAF subjects show interesting behaviour. It first goes steeply down, stabilizes/goes up and then decreases again. This behaviour is different compared to the reference or full HD group. In the case of higher activity peaks, the difference is minimal as opposed to the light activity peaks. In the light activity plot, there is a major difference visible in the recovery phase. At the beginning of the recovery phase, it follows the same pattern as the reference group, but after timestamp 5:30, the PerAF group stabilized again and even goes up before going down again after 6:10. This ensures that there is a significant difference between the Ref and PerAF group during recovery.

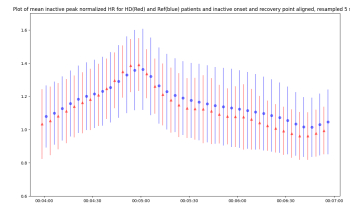
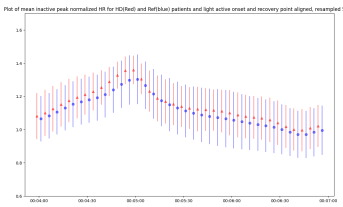
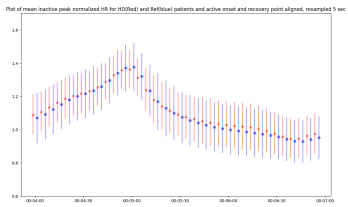
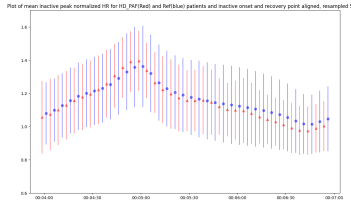
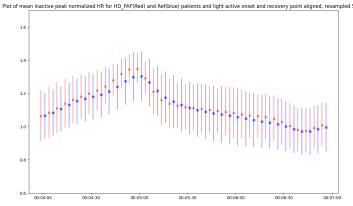
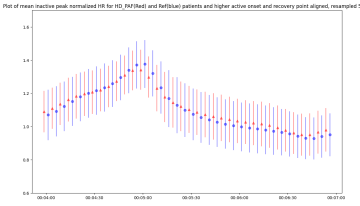
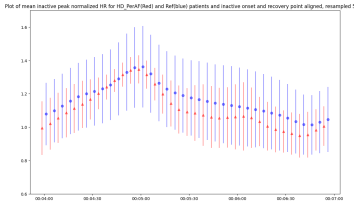
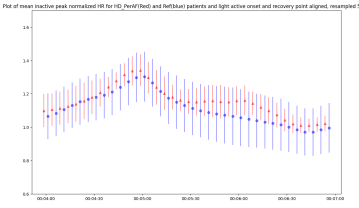
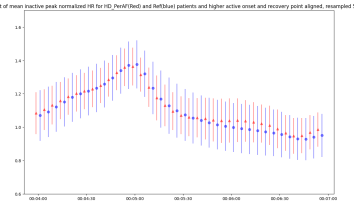
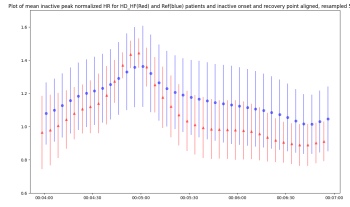
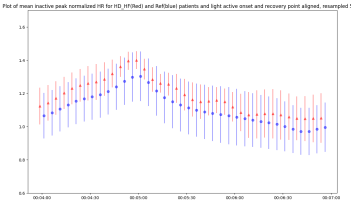
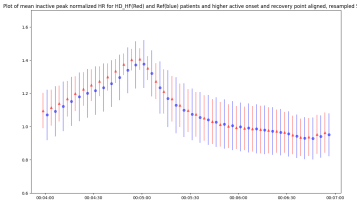
Group	Inactivity	Light activity	Higher activity
HD			
PAF			
PerAF			
HF			

Table 5: Plots of all the mean inactive peak normalized and onset-recovery aligned peaks in the HD, Ref, PAF, PerAF and HF group during inactivity (First column) and light activity (Second column) and higher activity (Third column). The x-axis shows the time in minutes and seconds and the y-axis the normalized heart rate. The mean and standard deviation are visualized per time point for every group.

Lastly, we will compare the HF and the reference subjects. Again with the higher activity peaks, we observed no difference between the full HD group and the reference group. This is not the case when examining the case of light activity. In this case, we observe a lot of differences. We can see that the HF group is fully above the Ref group. Another significant difference is the pattern of the HF group during recovery. The pattern is not going down slowly, but fluctuating between descending and stabilizing/rising. The largest difference is visible within the inactive peaks. Just like with the PerAF case, the HF group is below the Ref group in both the onset and the recovery phase. During the peak, we see the largest difference. Instead of being equal or just under the Ref group, the HF group is above the Ref group by a large margin. Finally, we spotted a difference in the recovery phase. Instead of descending and rising like the PerAF group, the HF group descends more gradually and has a large separation from the Ref group. This was not the case for the PerAF group or the full HD group.

3.2 Heart rate during inactivity

Despite the fact that people most of the time look at heart rate when a person is active, we conducted an experiment to determine if long-term inactivity data could be utilized to find differences between the classes. To accomplish this, we filtered the heart rate values to exclude instances where the number of steps was greater than zero. Subsequently, we removed additional heart rate values that were correlated with previous activity. To this extent, we assumed that activity can influence heart rate 4 minutes after activity. So we discarded all the heart rate values between an activity timestamp and 4 minutes after that timestamp. As said earlier, the goal is to find differences between groups. In this case between the HD (PAF, PerAF and HF) and Ref group. To search for this difference, we split the inactivity data up into weeks. Afterwards, we used different metrics to visualize it into a 2d plot

and coloured the samples by class.

When working out the previously described methodology, we obtained a total of 58 weeks of data for the HD group and 742 weeks of data for the Ref group. As we described earlier in the previous paragraph the HD group will be split into three smaller pieces, PAF, PerAF and HF. The metrics that we will be using first are mean heart rate, RMSSD, and SD1 and SD2 from Poincaré charts. These metrics are chosen because it has been shown earlier in literature with other types of data like ECG recordings that these metrics could be useful to detect (paroxysmal) atrial fibrillation [17; 16]. We found the previous metrics to be ineffective when applied to the data. The results of these experiments can be found in the Supplementary section B.

To find other metrics, we used a library called TSFEL. This library calculates statistical or temporal features for time series data. A disadvantage of these kinds of features is that it returns a large number of features and it becomes impossible to visualize them all. We solved this by performing a PCA on the resulting features. The PCA returns 2 components which can be used to create a 2d plot again. First of all, we calculated the features (statistical or temporal) for every week's time series. Afterwards, we standardized the data (zero mean, unit variance) and performed a PCA. The PCA transformed the data into 2d. Lastly, we created a plot where every sample gets coloured. In this case, we also gave every subclass of the HD group a distinctive colour. The plot uses thus four different colours (Blue: Ref, Green: PAF, Yellow: PerAF, Red: HF). The plots of both the statistical and temporal features can be seen in Figure 3.

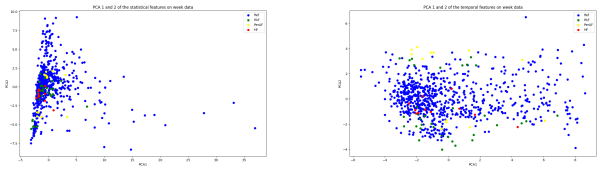


Figure 3: PCA plots of the inactive weekly data based on statistical features (Left) and temporal features (Right). The x and y-axis represent the first and second PCA components. Weeks are colour-coded by group.

We can note that with the statistical features on the left, there is again overlap and it is difficult to distinguish the classes. Looking at the right-hand side, with the temporal features, we see less overlap. Several yellow points at the top and green at the bottom seem to be forming groups. It seems promising that there is a difference between some classes when using a dimensionality reduction method. To test it even further, we used TSNE instead of PCA.

In Figure 4, we can see the TSNE plot for the temporal features. A cluster of yellow and green dots can be observed at the top, potentially indicating the presence of AF during those weeks. Another such region can be observed at the bottom of the figure, where several green and yellow dots appear to form a group despite being surrounded by numerous blue points. When we investigated the origin of these blue points we found that they all belonged to the same individual, identified as person 97. This is illustrated in Figure 5. The green points represent the samples belonging to person 97 and the blue points represent samples from other individuals in the reference group. Note that the samples from the HD group are not included in the figure. Moreover, we can observe that the green points of person 97 are overlapping with the same green points in Figure 4, which represent PAF samples. When having more data, it could be the case that this person does not belong to the reference group but actually has some variant of AF.

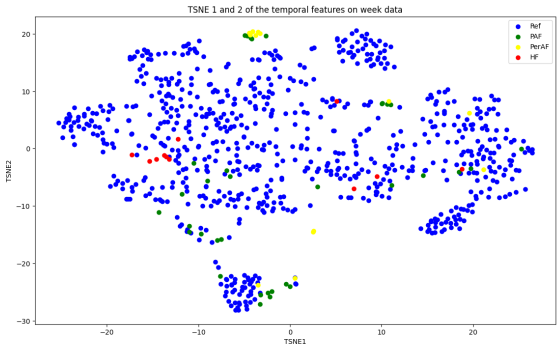


Figure 4: TSNE plot of the inactive weekly data based on temporal features. The first and second TSNE components are represented by the x and y-axis. Weeks are colour-coded by group.

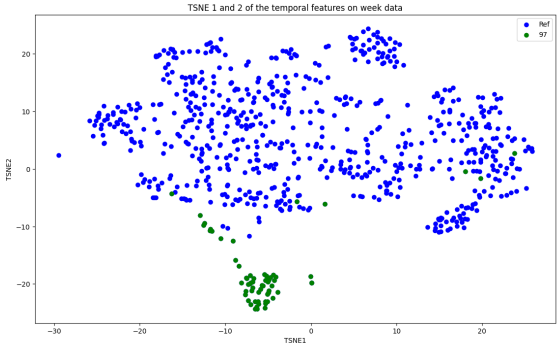


Figure 5: TSNE plot of the inactive weekly data based on temporal features, only reference samples. The first and second TSNE components are represented by the x and y-axis. Green points represent weeks of subject 97, and blue points represent other reference subjects.

3.3 Classification

Although we have seen some differences and potential groupings in the last two subsections, no classification has been performed yet. In this section, we explain which techniques have been used and which worked well in practice.

The classification algorithms used the onset-recovery alignment method with mean inactive peak normalization data. We started by training an SVM to distinguish between

the HD and Ref group. Unfortunately, it did not work at all. Almost every time all samples from the HD group were misclassified. We tried several combinations: normalizing the data, non-normalizing the data, different kinds of features, and using a random forest instead of an SVM. Unfortunately, all these variations did not improve the accuracy significantly. We did observe an improvement when dividing the HD group into smaller sub-groups. As described earlier these groups correspond to the patients having PAF, PerAF or HF.

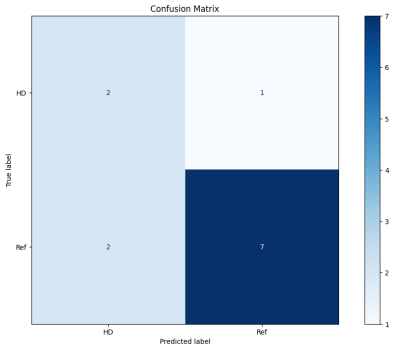


Figure 6: Confusion matrix of the time series SVM to classify PerAF and Ref peaks for light activity. The true/actual labels are shown on the vertical axis and the predicted labels are on the horizontal axis.

When we trained an SVM or RF with data from only PerAF subjects during inactivity, still no acceptable results were obtained. The improvement came when we utilised a Time series-specific SVM (TS SVM). This is essentially a regular SVM but with a specific kernel that can deal with time series input. The kernel that is used is a Fast global alignment kernel and is implemented by the tslearn framework [18; 19]. The TS SVM takes a 3D input in a format like: (# samples, # time points, # features per time point). The resulting confusion matrix can be seen in Figure 6. Examining the figures in the previous section regarding the various alignments, some differences were observed. One of these differences we observed was the variance between the HD and Ref group at a specific timestamp. To include this into the time series SVM, we grouped peaks from a person so the mean and std for every time point could be calculated. We divided the data of every person into roughly three equal parts, depending on the amount of data available. When inputting this time series into the TS SVM it worked better than in the previous case with only one feature per time point. This resulted in an accuracy of 0.8. The confusion matrix can be found in Figure 7. We also performed the same classification with the TS SVM with subjects of the other two diseases: PAF and HF. Classification with HF subjects against reference subjects performed similarly to the PerAF subjects. The performance went down when trying to distinguish between the PAF and reference group. This could be explained by examining the alignment plots in the previous section. In these plots, we can see that there are fewer (significant) differences between PAF and Ref subjects than PerAF or HF and Ref.

Finally, we trained a regular SVM on the same grouped 3D

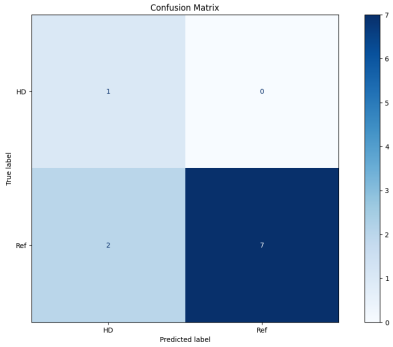


Figure 7: Confusion matrix of the time series SVM to classify PerAF and Ref peaks for light activity by grouping peaks and using mean and std at every timestamp. The vertical axis displays the true labels while the horizontal axis shows the predicted ones.

data but then flattened it so it suits the classifier. The results of this can be found in Figure 8. As we can notice, the HD group is still correctly classified, however, the accuracy of correctly predicting the Ref group went down. When we retrained the SVM a couple of times, the accuracy did not improve and was still worse than the TS SVM.

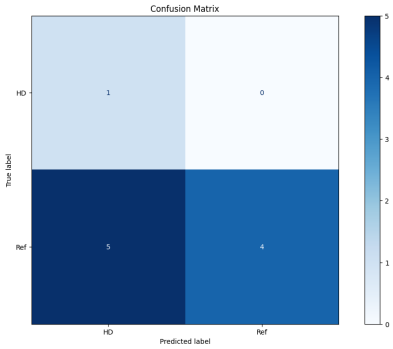


Figure 8: Confusion matrix of the SVM to classify PerAF and Ref peaks for light activity by grouping peaks and using mean and std at every timestamp. The true/actual labels are shown on the vertical axis and the predicted labels are on the horizontal axis.

Besides classifying the data with SVMs we also used DL models. Unfortunately, when we divided the data based on disease group there was too little data to properly train the DL models. The variations we examined can be found in the Supplementary section C. We solved this issue by relaxing some constraints and generated a new dataset by retrieving every peak above 80 BPM instead of 100 BPM. We saved the Onset-Recovery heart rate aligned time series together with the steps time series around the peak (+- 4 minutes). Next, we resampled the number of samples per reference person so it had an equal amount of samples based on the average samples per HD person. This resulted in a total of 7385 peaks for the HD group and 11092 peaks for the Ref group. Now the DL

network itself is responsible for determining the relationship between HD and Ref samples when given the heart rate and steps time series.

During the experiments, we made use of two different deep-learning networks. The first network has a convolution part for both the heart rate and steps time series. The second network is first concatenating the two different time series and then starts with the convolution. Visualizations of both networks can be found in Figure 9. The two variations can be seen as early and late integration of feature sets.

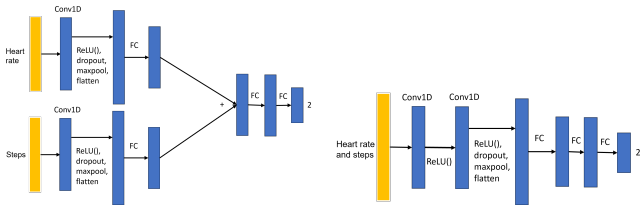


Figure 9: Visualization of first (Left) and second (Right) deep learning network used to classify peaks as HD or Ref class. The first network can be seen as late integration while the second network uses early integration.

We trained both networks with the newly generated dataset. When generating the dataset we used both the peak normalization and the mean inactive peak normalization. These combinations resulted in four different configurations and outcomes. These outcomes can be seen in the four confusion matrices in Table 6.

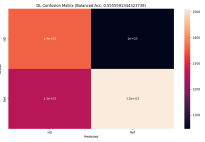
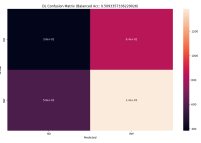
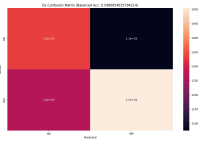
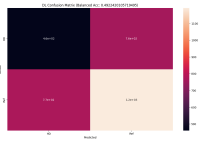
Network	Mean inactive peak norm	Peak norm
First		
Second		

Table 6: Confusion matrices for both DL networks and normalization methods using the Onset-Recovery time series and step data 4 minutes around a peak if a peak above 80 BPM. The true/actual labels are shown on the vertical axis and the predicted labels are on the horizontal axis.

It can be noticed that the mean inactive peak normalization achieves the best prediction performance in both networks compared to the peak normalization method. However, this does not mean that the peak normalization method is performing poorly. When we look at the Reference group we can see that a larger percentage of peaks are classified correctly as reference when using the peak normalization method. It is performing worse when examining the HD class. Despite this, we could argue that the peak normalization method achieves the preferable outcome. Not all the peaks in the HD group would correspond to a heart disease. So it is preferable to have a high percentage of ref peaks and some HD peaks to classify correctly. In the supplementary section D, visualizations are shown of the predicted samples with the first network and peak normalization.

With the previously used data for training the deep learning networks, we used the Onset-Recovery alignments. This could be seen as an already-engineered feature. To mitigate this influence we created a new dataset which first selects all peaks above the 80 BPM. Next, we extracted time series from both the heart rate and the steps between 2 minutes before the peak and 5 minutes after the peak. This resulted in time series which are peak aligned and contain the same length time series for both the heart rate and steps. Moreover, we used mean inactive peak normalization and resampling of the Ref group. This dataset is referred to as the new dataset in later paragraphs.

The two previously used DL networks, which are visualized in Figure 9, are again utilized and trained on the new dataset. For both networks, we examined three variations of weights in the loss function. These variations were no weights, balanced weights and weights which prioritize not misclassifying the Ref class.

With the first network, we observed that the results when training without any weights in the loss function seems best. In this case, 75% of the Ref samples were classified correctly and 37% of the HD samples. This is achieving better performance than the classifiers tested previously in Table 6 When we prioritized the classification of the Ref class, only 1.8% of the Ref samples were misclassified but also only 3.5% of the HD samples were classified correctly.

The results of the second network were interesting. There was not much difference between the outcome when training without weights or with prioritizing the Ref class. In both cases, a minimal amount of Ref samples were misclassified (6.3%) but also only 9.8% of the HD group was classified correctly. Still, these results are not performing well enough to be used as a classification for heart diseases.

3.4 Outlier detection

As explained earlier we assumed that not all samples/peaks within the HD group show signs of a heart disease. This could be a reason why some or most of the samples may be classified as reference. Instead of performing classification, we examined outlier/novelty detection methods to investigate differences between the HD and Ref group.

We started the experiment by investigating the use of the Scikitlearn LocalOutlierFactor for novelty detection. This method is based on the algorithm explained by Breunig et al. [20]. First of all, we only used the Onset-Recovery heart rate time series as input for the algorithm. The Ref samples were used for training and the HD samples for testing. Of all the 7385 HD samples only 87 were predicted as new and thus not belonging to the Ref class. An interesting observation was that all the HD subjects were represented in these 87 samples. Next, besides only using the heart rate time series, the step time series was also concatenated. Now the algorithm

returned 558 samples of the HD group that were found to be an outlier. Adding the steps as input to the algorithm seems like an improvement. In addition to only classifying the HD samples, we also held a subset of the Ref class apart. When we tested the trained algorithm on this set it returned 1105 samples as novelty instances. This was 4.5% of all the testing samples. For the HD group, it classified 7.5% as a novelty. Moreover, all individuals of the Ref test set were represented in the novelty examples, just like the case with the HD group. This is not something that is preferable and this means that as it stands now, it is not useful for outlier/novelty detection with this data and algorithm.

We conducted another experiment for outlier detection but now used PCA to detect outliers. It works by first fitting a PCA on the Reference data. Next, it calculates for every sample the reconstruction error. If the outlier detection would work, we expect the method to give a higher reconstruction error with samples of the HD class than with the Ref class. We performed the experiment with different configurations. These results can be found in the Supplementary section E.2. When we performed PCA on Onset-Recovery heart rate time series and steps time series data without standardization and minimum peak height of 100 BPM, some differences could be seen. There was a difference in reconstruction error when looking at the samples with PAF and PerAF. The mean error in both groups, 0.82 and 0.84 respectively, was higher than in the Ref group (mean error 0.78). The table with the exact numbers can be found in Table 10. Figure 10 shows the zoomed-in error distribution of both the HD and Ref samples on the right-hand side of Figure 23, which can be found in the Supplementary material. In this figure, we can see that there were some HD samples with a higher reconstruction error compared to the Ref group.

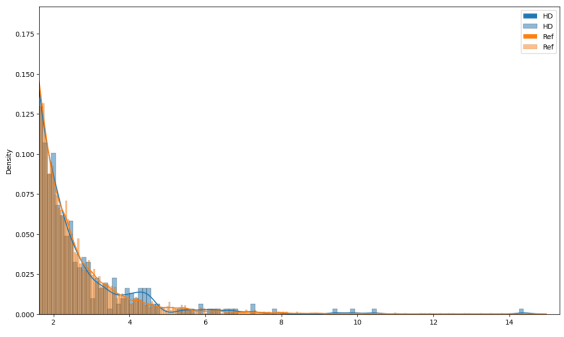


Figure 10: Zoomed-in on the right-hand side of Figure 23 which visualizes the error distribution of both the HD and Ref group after performing PCA. The x-axis shows the error and the y-axis the density.

In addition to performing outlier detection using PCA, we also performed outlier detection with an AutoEncoder. The AutoEncoder we used is a slight modification of [21], so it can make use of batches. The AutoEncoder is designed for time series and uses two LSTM layers in the encoder and two LSTM layers in the decoder. When we visualized the reconstruction losses for the Ref and HD group we could see that the losses of the HD group are not exceeding the Ref ones. This can be examined in Figure 11. Next, we used the new dataset to train the AutoEncoder and investigated different configurations. These variations can be found in the Supplementary section E.1. Overall none of these variations worked well on the inputted data.

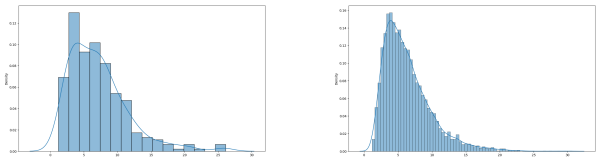


Figure 11: Visualization of the reconstruction losses during training with Ref data (Left) and testing with HD data (Right). The x-axis shows the error and the y-axis the density.

3.5 Grouping peaks per week

Similar to the grouping of peaks in section 3.3, we will now group the peaks of a person per week but only during training. As stated multiple times in this paper, some HD samples can be observed, but the HD group also shows many Ref-alike peaks. To this extent, we propose another way of testing. During testing, we use a DL model to make predictions for all peaks within a week of a single subject. Next to this, we calculate a percentage of predicted HD peaks in that week. This is done for every available week of a subject and for all subjects. Finally, we create a distribution representing the percentage of predicted HD peaks per week for all weeks in both the Ref and HD group. We hypothesized that in a week from an HD subject, more peaks are classified as HD than in a week from a Ref subject. To solve the class imbalance, we resampled the Ref group by taking randomly 8 weeks of data and sampling 115 peaks per week if there were more than 115 peaks available in that week. These values are chosen because it is the average amount of weeks and peaks per week for the HD group. The dataset that we used for these experiments was the new dataset. We performed this method by examining two DL networks and three different variations for the loss function. The two DL networks are the same ones used in the classification section, see Figure 9. The loss function variations consist of a custom loss function, Cross Entropy loss without weights and with more weight on the Ref class. Figure 12 shows the outcome when training the second model without weights. It can be seen that there is some distance between the peak for the Ref class (blue) and the HD class (red). This could indicate that in a week of an HD subject on average, there are more predicted HD samples present. This configuration also gave us the largest separation between the classes. The results of all the other examined variations and an explanation of the custom loss function can be found in the Supplementary section F.

A disadvantage of the previous experiments is that the resulting outcome is not optimized directly. Only the classification is optimized within the DL network. In additional experiments, we first optimized the classification loss by back-

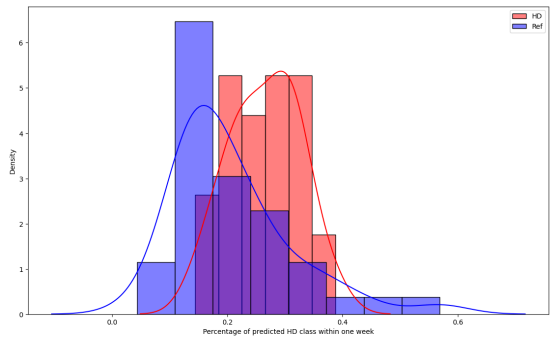


Figure 12: Distribution of the weekly predicted HD peaks percentages for both HD (Red) and Ref (Blue) group. Created by training the 2nd DL network without any weights in the loss function. The x-axis shows the percentage of HD peaks per week and the y-axis the density.

propagation for every batch. After this optimization for the classification loss, we created the histogram/distribution and calculated a KL divergence loss. We optimized the model for that loss and a new epoch was started. A graphical overview can be found in Figure 13. Several variations were investigated. The variations we tried used different networks, different classification losses and different KL losses. The experiments made again use of the two previously described models and used the cross-entropy loss or the custom loss function for the classification loss. For the determination of the KL loss, we first calculated the KL divergence. The KL divergence is zero when two distributions are similar. The loss should be the exact opposite of this since the distributions should be as dissimilar as possible. To this extent, we calculated the KL loss as $1/KL\ divergence$. Other variations we investigated were using $500/KL\ divergence$ for the KL loss or switching the order of the distributions around when calculating the KL divergence. Next to this other different calculations of the KL loss are examined like negative KL divergence and negative log KL divergence. These are examined because the loss needs to be minimized and negative values are potentially working better than the previous $1/KL\ divergence$ calculation which will converge to zero when having a large KL divergence. Finally, we changed the way the KL loss is optimized. Right now, the KL loss is optimized once. We can optimize this KL loss multiple times within a single epoch to have a potentially higher influence on the outcome.

From all the configurations, we found that the two that used the second network with single optimized $1/KL\ divergence$ and the cross-entropy or custom loss function worked satisfactorily. The two distributions can be found in Figure 14.

We can see that especially in the left figure the distance between the distributions is greater than in the case without using the KL loss (Figure 12). The goal is to investigate whether we can improve the classification performance by using this grouping. We examined this by setting a threshold and used this to predict whether a week belongs to the Ref or HD group. The results were visualized using a confusion

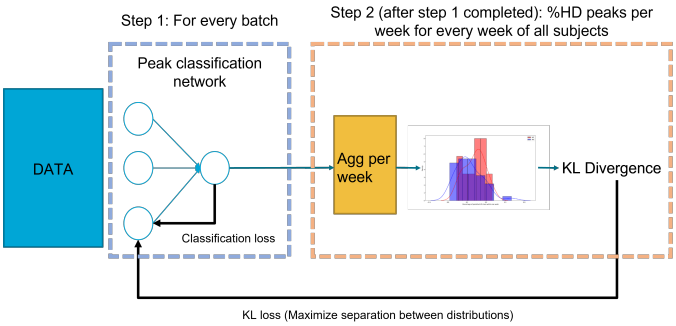


Figure 13: Graphical overview of the training of the DL network with extra KL loss. Black lines represent the backpropagation of the loss into the peak classification network.

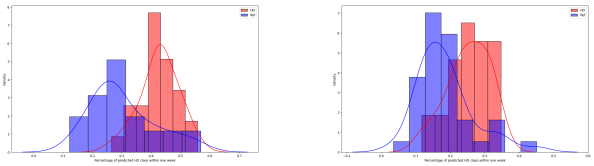


Figure 14: Distributions of the weekly predicted HD peaks percentages for the second network with cross-entropy loss (Left) and the custom loss function (Right). The x-axis shows the percentage of HD peaks per week and the y-axis the density.

matrix.

We achieved the best performance by using the distribution achieved with the second network and custom loss function and setting the threshold to 0.25. This confusion matrix can be found in Figure 15 Right.

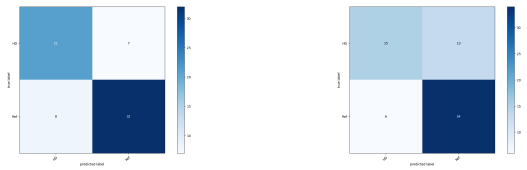


Figure 15: Confusion matrices of the weekly predicted peaks for the second network with KL loss and cross-entropy loss (Left) or the custom loss function (Right). The true/actual labels are shown on the vertical axis and the predicted labels are on the horizontal axis.

It minimizes the misprediction of the Ref class by mispredicting 6 Ref samples as HD. On the other hand, it is predicting just more than half of the HD samples correctly. When we relax the minimization of the misprediction for the Ref group slightly, an improvement can be seen when using the distribution achieved with the second network and the CE loss and setting the threshold to 0.40. Figure 15 left, shows the resulting confusion matrix. The error in the Ref class is slightly increased to 8 samples, but the HD group is classified drastically more correctly.

To determine how the KL loss improved the performance, Figure 16 shows the confusion matrix for the 2nd network

without KL loss. While we have the same prediction for the Ref class as the second network with the custom loss function, it only predicts 9 samples of the HD group correctly.

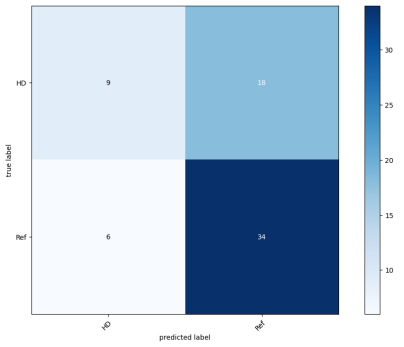


Figure 16: Confusion matrix of the weekly predicted peaks by using the second model without using the KL loss. The vertical axis represents true labels while the horizontal axis represents predicted labels.

Although the previous configurations with KL loss improved the performance, another configuration worked best The best variation was the negative log KL divergence loss which was calculated 10 times in a row in each epoch with the custom loss function for the classification loss. This variation could make seven mispredictions in the Ref class and correctly classify 22 HD weeks. This is one more HD week with one less misprediction in comparison with the previous best variant. We could change the threshold in such a way that it could even have 4 or 5 mispredictions in the Ref class and still have 18 or 19 correct HD weeks respectively. Comparing the AUC of the two variations, negative log KL divergence optimized multiple times and $1/KL\ divergence$, the first variant achieves a score of 0.889 and the second variant a score of 0.768. The ROC plots can be found in the Supplementary, section F.

Related to the topic of grouping peaks per week and outlier detection, is the use of positive unlabeled (PU) learning and multiple instance learning (MIL). We conducted additional experiments to see how these algorithms performed.

Starting with PU learning, we used a library that implements the algorithm developed by Elkan and Noto and the Bagging-based algorithm developed by Mordelet and Vert [22; 23]. This library accepts scikit-learn classifiers as input like an SVM. When we ran these two algorithms with an SVM, both resulted in an unfavourable result. All the HD samples were classified as Ref. In the classification section, we demonstrated that the TS SVM performed better than the SVM. That is why we investigated what happens when performing PU learning with a TS SVM. Unfortunately, the algorithm did not end and thus no results were obtained.

Better results were obtained by making use of multiple-instance learning. For these experiments, we used the mil library for Python [24]. With MIL the data is required to be split up in bags or sets. To this extent, we split the data up into weeks per person. We started the experiments by using only the heart rate time series and a simple SVM to classify the bags. This resulted in the algorithm almost classifying every sample as HD. Changing some parameters resulted in the exact opposite scenario where almost every sample was classified as Ref. When we changed other parameters like the addition of steps time series or changing bag embedding methods, the performance did not improve. A positive result was achieved when we changed the model to MILES (Multiple-Instance Learning via Embedded Instance Selection) [25]. We trained the model with and without the addition of the step time series. The corresponding confusion matrices can be seen in Figure 17.

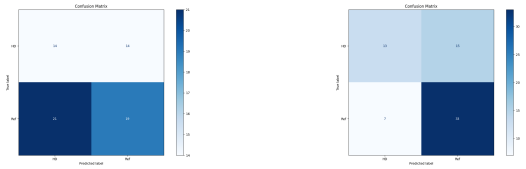


Figure 17: Confusion matrices of prediction per week of the MILES model with only heart rate data (Left) and with steps included (Right). The true/actual labels are shown on the vertical axis and the predicted labels are on the horizontal axis.

We can observe that the addition of the step time series helps to reduce the misclassification of the Ref group. Comparing this to the results of the DL with KL loss we can see it performs worse. The prediction in the HD group is flipped in relation to the confusion matrix in figure 15 right and the MILES model made one more misprediction of the Ref class.

4 Discussion

We looked into feature extraction and classification of heart rate and step time series in the context of cardiovascular diseases. When performing feature extraction on heart rate time series for cardiovascular diseases it appears that the Onset-Recovery alignment with mean inactive peak normalization worked well. However, this is only the case when dividing the HD group based on the occurring heart diseases. Using this method, differences in patterns between the disease group and reference group can be seen and used for classification. In addition to this, inactive (week) data could also be used to discriminate between classes, as shown with the TSNE plots. A limitation of these findings is the number of subjects in the dataset. The experiment should be performed again in the future when more data is available.

Classification of the data to distinguish between the disease or reference group is possible. When performing peak classification, we achieved the best performance by using a time series-specific SVM and the Onset-Recovery alignment data. The data was aggregated into smaller groups per person and was used as input for the TS SVM. The classification worked best for the HF and PerAF group. This could be explained by the fact that persons with PAF only suffer from the disease temporarily and thus most of the time could look similar to a reference subject. Future works should look into the classification performance when using deep learning frameworks

to classify the separate disease groups. Due to the amount of data available at this time, it is not feasible.

To perform classification and feature extraction, normalization is an important step. Especially in the case of heart rate time series where the morphology differs between people. When performing feature extraction on the data, the mean inactive peak normalization, performed well. We assume this method works because it normalizes all the heart rate time series by the base heart rate and thus minimizes this difference between subjects.

In addition to classification, we also performed outlier detection. This was done to identify 'real' HD samples because the HD class consist logically of samples that look like or are Ref due to the nature of the diseases. Unfortunately, LocalOutlierFactor, PCA and AutoEncoder models were not able to distinguish between Ref and HD samples. The models used were found to be effective in for example breast cancer detection. Therefore the problem could be the kind of data that is given to the models. An improvement was seen when performing testing on weekly data with DL models and determining the percentage of HD samples per week. It seems that indeed HD subjects have more HD-like peaks per week than Ref persons. Using KL as an extra loss ensured that the accuracy increased and good separation between HD and Ref class could be observed. However, we cannot detect perfectly if a week belongs to an HD or Ref subject and therefore we need to make a trade-off of how many mispredictions in the Ref or HD class we allow. For now, we can allow a low amount of Ref misprediction (10%) and have still 64% of the HD group correctly classified. Employing a multiple instance learning model like MILES also performed well but less than the DL models. Surprisingly, positive unlabeled learning models did not perform at all. These models are designed to work with datasets where the test set can contain Ref samples in this case. Future works could look into the reasons why the previously described models did not perform well and see in what way the data or model should be transformed to make it work. Moreover, future works should look into how the grouping peaks per week classification can be improved to make it useful for clinical practice.

5 Conclusion

The aim was to find out if and how acceleration-deacceleration curves on long-term heart rate time series could be used for the prediction of different kinds of arrhythmias. We used the ME-TIME dataset for the prediction/detection of arrhythmias in subjects. To achieve this, peaks were found in the time series and different manipulations of the resulting peak time series were performed. The results show that when using Onset-Recovery peak alignment with mean inactive peak normalization there is a difference between the reference and the three disease groups (PAF, PerAF, HF). However, this is only noticeable when splitting the hospital group based on the disease. When we perform peak classification with this data, the best performance is achieved when using a time series-specific SVM. Moreover, grouping peaks help to increase performance. Next to this, the results show that weekly inactive heart rate time series data can also be helpful in distinguishing subjects based on disease. However, due to the small sample size of the current dataset, it is difficult to say now which region belongs exactly to which disease group. In the future when more data is available such a plot should be created again to give an answer to this question. In addition, when we split up the peaks into weeks per person, MILES and specific DL models can be used to a certain extent to predict if a week belongs to an HD or Ref subject. Overall, we have shown that classification with heart rate time series is possible. Furthermore, analysis of the data showed that there are differences between the reference and disease groups in the heart rate time series. A remark is that it is important to use proper standardization techniques, which helps to reduce inter-subject variability. For the classification task, it is important to group peaks in some way to improve performance. In the future, this could help doctors to detect such arrhythmia earlier or easier over a longer time span instead of only a maximum of 72 hours.

6 Method/experiment

6.1 ME-TIME dataset

The ME-TIME dataset (registered at clinicaltrials.gov with ID NCT05802563) consists of two long-term Fitbit groups. The Ref (reference) group consists of 14 reference subjects without cardiovascular disease where the cardiovascular annotation is done via self-reporting. The heart disease (HD) group consists of eight subjects with either atrial fibrillation (AF) or heart failure (HF). The atrial fibrillation group is split up into paroxysmal atrial fibrillation (PAF) and persistent atrial fibrillation (PerAF). The data available per person can range from a couple of weeks up to 2 years. Every person has a separate pickle file containing heart rate data, step (activity) data and a dictionary with metadata. The heart rate and step data are synchronised and resampled to 0.2 Hz (one sample every 5 seconds).

6.2 Peak detection and alignments

As we explained in the main text, most experiments make use of the three fiducial points and perform different alignments on them. The algorithm we used for the peak point detection is the scipy find_peaks method. The peaks found by the algorithm are processed as follows. First of all, the onset point is detected. This point is defined as the minimum value before the peak in a window of 300 seconds. Afterwards, the offset/recovery point is determined. For this, the heart rate time series was considered 900 seconds after the peak. First, the signal is again smoothed with a window size of 120 seconds to get rid of some spikes, whereafter the minimum value is taken, similar to the onset detection.

After determining these points different alignments can be performed. The first type aligns all the peaks according to their peak time. The data is resampled every 5 seconds, so every peak with a corresponding onset and recovery point consists of 1205 seconds (300 (onset) + 900 (recovery) + 5 (peak value)). When a peak has an onset smaller than 300 seconds and/or recovery before 900 seconds, the missing values were filled with NaNs. All the peaks were aligned on zero when looking at a time interval ranging from -300 to 900 seconds.

The implementation, therefore, works with a vector of size 1205 / 5 = 241.

The second method is the Onset-Recovery alignment. As the name suggests all the onset and recovery points will be aligned with each other. Still, this is performed in such a way that all the peaks are aligned as well. The data is resampled so that every alignment has the same length. Figure 18 visualizes how these two alignments work. On the left side, the peak align method is visualized and on the right side the Onset-Recovery alignment.

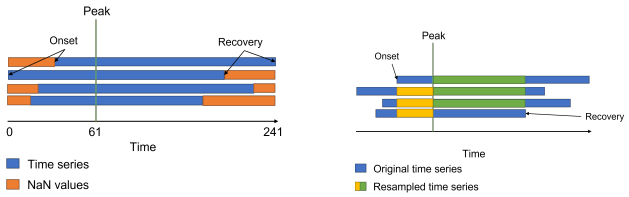


Figure 18: Visualization of the working of both the peak and Onset-Recovery alignment methods.

Next, we used two methods for the normalization. The first approach is the peak normalization method, which transforms every time series in such a way that the lowest value in the time series is now zero and the peak value is one. The data is thus rescaled according to the following formula: $rescaled = (data - data_{min})/(val_{peak} - data_{min})$, where $data$ represents one time series of a single peak and $data_{min}$ represents the lowest value in the time series.

The second way is the mean inactive peak normalization. First, for every person, all the peaks are calculated where the person is inactive. From all the resulting peaks the mean peak height (heart rate) is calculated. This value is used as a normalization constant for all the time series of that specific person.

References

[1] A. Shi, Z. Tao, P. Wei, and J. Zhao, "Epidemiological aspects of heart diseases," *Experimental and therapeutic medicine*, vol. 12, no. 3, pp. 1645–1650, 2016.

[2] T. Golany and K. Radinsky, "Pgans: Personalized generative adversarial networks for ecg synthesis to improve patient-specific deep ecg classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, Conference Proceedings, pp. 557–564.

[3] D. Jabaudon, J. Sztajzel, K. Sievert, T. Landis, and R. Sztajzel, "Usefulness of ambulatory 7-day ecg monitoring for the detection of atrial fibrillation and flutter after acute stroke and transient ischemic attack," *Stroke*, vol. 35, no. 7, pp. 1647–1651, 2004.

[4] J. Torres-Soto and E. A. Ashley, "Multi-task deep learning for cardiac rhythm detection in wearable devices," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–8, 2020.

[5] J. Zhang, A. Liu, M. Gao, X. Chen, X. Zhang, and X. Chen, "Ecg-based multi-class arrhythmia detection using spatio-temporal attention-based convolutional recurrent neural network," *Artificial Intelligence in Medicine*, vol. 106, p. 101856, 2020.

[6] Z. F. M. Apandi, R. Ikeura, and S. Hayakawa, "Arrhythmia detection using mit-bih dataset: A review," in *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*. IEEE, Conference Proceedings, pp. 1–5.

[7] P. Kanani and M. Padole, "Ecg heartbeat arrhythmia classification using time-series augmented signals and deep learning approach," *Procedia Computer Science*, vol. 171, pp. 524–531, 2020.

[8] B. Ballinger, J. Hsieh, A. Singh, N. Sohoni, J. Wang, G. H. Tison, G. M. Marcus, J. M. Sanchez, C. Maguire, and J. E. Olgin, "Deepheart: semi-supervised sequence learning for cardiovascular risk prediction," in *Thirty-Second AAAI Conference on Artificial Intelligence*, Conference Proceedings.

[9] O. Walch, Y. Huang, D. Forger, and C. Goldstein, "Sleep stage prediction with raw acceleration and photoplethysmography heart rate data derived from a consumer wearable device," *Sleep*, vol. 42, no. 12, p. zsz180, 2019.

[10] Z. Al-Makhadmeh and A. Tolba, "Utilizing iot wearable medical device for heart disease prediction using higher order boltzmann model: A classification approach," *Measurement*, vol. 147, p. 106815, 2019.

[11] R. Czabanski, K. Horoba, J. Wrobel, A. Matonia, R. Martinek, T. Kupka, M. Jezewski, R. Kahankova, J. Jezewski, and J. M. Leski, "Detection of atrial fibrillation episodes in long-term heart rhythm signals using a support vector machine," *Sensors*, vol. 20, no. 3, p. 765, 2020.

[12] A. Hochstadt, E. Chorin, S. Viskin, A. L. Schwartz, N. Lubman, and R. Rosso, "Continuous heart rate monitoring for automatic detection of atrial fibrillation with novel bio-sensing technology," *Journal of electrocardiology*, vol. 52, pp. 23–27, 2019.

[13] C. Ford, C. X. Xie, A. Low, K. Rajakariar, A. N. Koshy, J. K. Sajeev, L. Roberts, B. Pathik, and A. W. Teh, "Comparison of 2 smart watch algorithms for detection of atrial fibrillation and the benefit of clinician interpretation: Smart wars study," *JACC: Clinical Electrophysiology*, vol. 8, no. 6, pp. 782–791, 2022.

[14] J. Selder, T. Proesmans, L. Breukel, O. Dur, W. Gielen, A. C. van Rossum, and C. Allaart, "Assessment of a standalone photoplethysmography (ppg) algorithm for detection of atrial fibrillation on wristband-derived data," *Computer Methods and Programs in Biomedicine*, vol. 197, p. 105753, 2020.

[15] T. Peçanha, N. D. Silva-Júnior, and C. L. d. M. Forjaz, "Heart rate recovery: autonomic determinants, methods of assessment and association with mortality and cardiovascular diseases," *Clinical physiology and functional imaging*, vol. 34, no. 5, pp. 327–339, 2014.

[16] D. Blanco-Almazan, D. Romero, W. Groenendaal, L. Lijnen, C. Smeets, D. Ruttens, F. Catthoor, and R. Jané, "Relationship between heart rate recovery and disease severity in chronic obstructive pulmonary disease patients," in *2020 Computing in Cardiology*. IEEE, Conference Proceedings, pp. 1–4.

[17] B. Broux, D. De Clercq, L. Vera, S. Ven, P. Deprez, A. Decloedt, and G. van Loon, "Can heart rate variability parameters derived by a heart rate monitor differentiate between atrial fibrillation and sinus rhythm?" *BMC veterinary research*, vol. 14, pp. 1–7, 2018.

[18] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, and K. Kolar, "Tslearn, a machine learning toolkit for time series data," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 4686–4691, 2020.

[19] M. Cuturi, "Fast global alignment kernels," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, Conference Proceedings, pp. 929–936.

[20] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, Conference Proceedings, pp. 93–104.

[21] V. Valkov, "Time series anomaly detection using lstm autoencoders with pytorch in python," Mar 2020. [Online]. Available: https://curiousily.com/posts/time-series-anomaly-detection-using-lstm-autoencoder-with-pytorch-in-python/

[22] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 213–220.

[23] F. Mordelet and J.-P. Vert, "A bagging svm to learn from positive and unlabeled examples," *Pattern Recognition Letters*, vol. 37, pp. 201–209, 2014.

[24] A. Rosas Garcia, "Mil/mil at master · rosasalberto/mil," 2021. [Online]. Available: https://github.com/rosasalberto/mil/tree/master/mil

[25] Y. Chen, J. Bi, and J. Z. Wang, "Miles: Multiple-instance learning via embedded instance selection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 12, pp. 1931–1947, 2006.

Supplementary

A Number of steps before peak

It may be a surprise why the threshold between light and higher activity lies at 20 steps. We conducted an experiment to determine the number of steps to use. In this experiment, we created similar figures as in Table 5. We set the threshold at 100 steps. The results show that there were in this case minimal differences between the Ref and HD group as opposed to the figures shown in Table 5.

In addition to this, we investigated if narrowing the intervals would lead to a bigger distinction between the HD and Ref classes. We used two intervals in this experiment, 15 to 20 steps and 20 to 25 steps. The data we used ranged from peaks with a minimum height of 60, 80 or 100 BPM. When examining the figures in Figure 19 where the minimum peak height is set to 60 (Left) or 80 (Right) BPM no significant differences can be observed between the HD and Ref group.

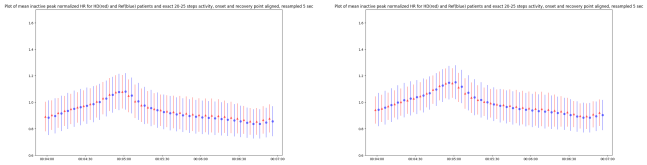


Figure 19: Plots of Onset-Recovery alignment with mean inactive peak normalization for a fixed amount of steps for peaks higher than 60 BPM (Left) or 80 BPM (Right). The mean and standard deviation are visualized per time point for both the HD and Ref group.

A difference could be observed in both step intervals when setting the maximum peak height to 100 BPM. This difference is visualized in Figure 20. The problem is that the amount of samples with the 100 BPM threshold is low for the HD group. In the 15 to 20 steps only four samples were available and in the 20 to 25 only 25 samples were available. So fixing the step interval did not help with the current amount of data.

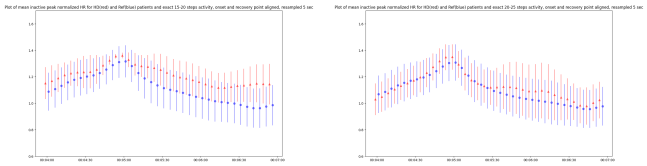


Figure 20: Plots of Onset-Recovery alignment with mean inactive peak normalization for 15 to 20 steps (Left) and 20 to 25 steps (Right) for peaks higher than 100 BPM. The mean and standard deviation are visualized per time point for both the HD and Ref group.

B Figures metrics week inactivity data

This section describes the results of utilizing the mean heart rate, RMSSD, and SD1 and SD2 from Poincaré charts metrics on the inactive week data. In Figure 21, we can see the result where the points are plotted based on their RMSSD and mean HR value. On the left, the plot only shows the HD points coloured on their sub-group. The right plot shows all the Ref and HD points in blue and red respectively. In the HD plot, we could argue that there is a distinction between the groups. Especially when reasoning that, logically, there are some PAF points close to PerAF points. This is logical because the patients with a blue dot represent the PAF group which only once in a while shows signs of atrial fibrillation, as opposed to PerAF, which shows signs of AF more frequently. However, if we look at the bigger picture on the right-hand side, there are a lot of reference points overlapping with the HD group. So it is very hard to differentiate these groups according to this plot. There are, however, a couple of red points at the bottom of the figure that are distanced from some blue points, but clear separation is hard to find.

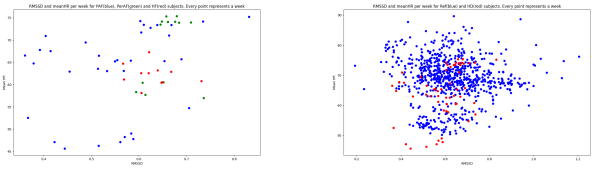


Figure 21: Plots of the inactive heart rate data per week based on the mean HR and RMSSD for the PAF, PerAF and HF group (Left) and Ref and HD group (Right). The x-axis represents the RMSSD values and the y-axis the mean HR values. Points are coloured by class. In the left figure blue is for PAF, green is for PerAF and red is for HF. In the right figure blue represents the Ref weeks and red the HD weeks.

We created similar plots for the data but then with SD1 and SD2 on the x and y-axis, as illustrated in Figure 22. Unfortunately, we can draw the same conclusion as with the previous figures. Although there is significant overlap between the Ref and HD group, some separation appears to exist within the HD group.

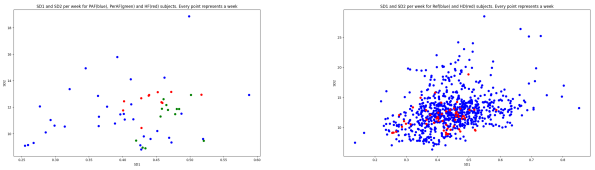


Figure 22: Plots of the inactive heart rate data per week based on the SD1 and SD2 for the PAF, PerAF and HF group (Left) and Ref and HD group (Right). The x-axis represents the SD1 values and the y-axis the SD2 values. Points are coloured by class. In the left figure blue is for PAF, green is for PerAF and red is for HF. In the right figure blue represents the Ref weeks and red the HD weeks.

C DL prediction with Fitbit Onset-Recovery data

As we already explained in the main text, classification by using DL frameworks for the Onset-Recovery heart rate Fitbit data was no success. For the classification, the DL base-

line model was utilized from the BigIdeasLab_STEP experiments. Different configurations of the model and data were investigated like using normalized and non-normalized data and with or without weights in the loss function. All these variations did not improve the performance. Moreover, resampling of the Reference dataset was investigated to balance the dataset even more. This was tried with the Persistent AF group but unfortunately did not improve performance. All HD samples were misclassified as Ref.

D Prediction visualization first DL network with Fitbit data

In this section, we show the visualizations of the predicted samples with the first DL network and peak normalization by using the Fitbit data described in 3.3. We generated these visualizations to examine if there are differences between the predicted samples and the Ref and HD class according to the DL network. Table 7 shows these plots.

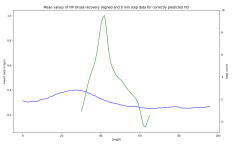
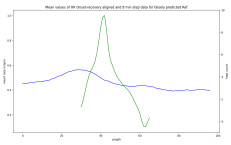
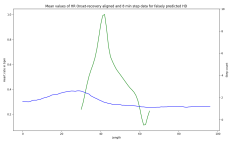
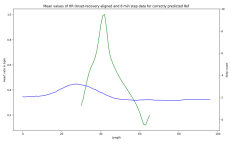
True/Pred	HD	Ref
HD		
Ref		

Table 7: Mean heart rate and steps of predicted test samples with the first DL network and peak normalized data. The true/actual labels are shown on the vertical axis and the predicted labels are on the horizontal axis.

When examining these plots we can see a subtle difference during the recovery phase for the samples predicted as HD. During the recovery phase, there are two points where there is a change in how fast the signal decreases. Comparing this to the samples predicted as references we see that this pattern is less. Furthermore, there is a change in the number of steps. The HD samples that have been predicted as Reference have a higher step size overall than the correctly predicted HD samples.

E Outlier dection

E.1 AutoEncoder

In this section, we discuss the investigated variations tried for outlier detection with an AutoEncoder. One specific adjustment was to resample the step time series. The step time series is just like the heart rate time series sampled every 5 seconds. Due to this sampling frequency, a lot of values are zero. In two of the experiments, we resampled the step time series to once per minute. Next, we utilized different ways of standardizing across experiments. The standardization has been performed over the heart rate and steps simultaneously, independently and not at all. The various experiment configurations can be found in Table 8.

Heart rate	Steps	Standardization
Normal	Per minute	Concatenated
Normal	Per minute	Independent
Per timestamp	Per timestamp	Independent
Normal	Normal	Independent
Normal	Normal	None

Table 8: The different configurations used for the Outlier detection experiments with an AutoEncoder.

When plotting the distribution of the losses, all configurations were not able to distinguish the HD and Ref group. When setting a threshold on the losses of the Ref group we get a correct prediction rate of around 97.9%. Using this threshold then for the HD group only around 1% is predicted as HD. This does not seem to work for the detection of outliers.

E.2 PCA

This section shows the results of performing outlier detection with PCA. Figure 23, shows the error distribution between the Ref and HD group. Tables 9, 10, and 11 present the numeric results of performing the PCA outlier detection under different configurations.

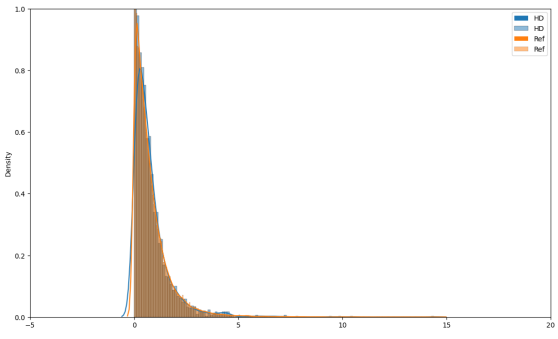


Figure 23: Error distribution for the HD and Ref group when performing PCA. The x-axis represents the error and the y-axis the density.

F Classification per week

Here we show the results of the distribution gathered by training the 1st and 2nd deep learning networks with different configurations. These distributions can be seen in Table 12.

The first two rows are using the standard cross-entropy loss without weights and with weights that minimize the misprediction of the Ref class. The last two rows make use of a

	Mean	Std
Ref	5.62	7.59
Full HD	4.83	6.09
PAF	5.47	6.86
PerAF	4.29	4.919
HF	2.80	3.386

Table 9: Mean and Std for the errors achieved by performing PCA with peaks higher than 80 BPM.

	Mean	Std
Ref	0.78	0.933
Full HD	0.806	0.9619
PAF	0.8244	1.0188
PerAF	0.8476	0.9761
HF	0.6315	0.6732

Table 10: Mean and Std for the errors achieved by performing PCA with peaks higher than 100 BPM and no usage of z-standardization.

	Mean	Std
Ref	10.498	11.289
Full HD	9.584	9.100
PAF	10.186	9.961
PerAF	8.483	6.907
HF	6.562	6.73

Table 11: Mean and Std for the errors achieved by performing PCA with peaks higher than 100bpm and the of z-standardization.

custom loss function. The idea behind this loss function is that it is asymmetric and that it sets weights in such a way that it penalizes Ref samples mispredicted as HD but has a lower penalty on HD samples classified as Ref. Internally the loss function first calculates the cross-entropy loss and thereafter multiplies this with the according weights. The initial custom loss function used the following weights: $\begin{bmatrix} -10 & 0 \\ 10 & -5 \end{bmatrix}$. As can be seen in the Table the distributions did not look promising and the wanted effect was not achieved. A reason for this could be the negative weights used. To fix this the weights for the custom loss function were changed accordingly: $\begin{bmatrix} 3 & 5 \\ 10 & 3 \end{bmatrix}$. This configuration worked better but still not better than the configuration shown in the first row.

After introducing the extra KL loss, the best network was the multiple optimized negative log KL loss with the custom loss function. The ROC curve for this variant and the previous best variant with $1/KL\ divergence$ and custom loss function can be found in Figure 24.

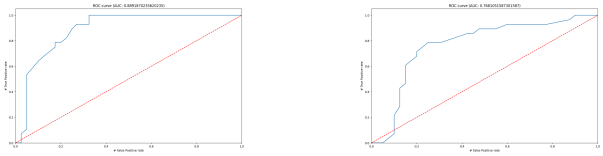


Figure 24: ROC curves for the multiple optimized negative log KL loss (Left) and single optimized $1/KL\ divergence$ (Right) with both utilizing the custom loss function.

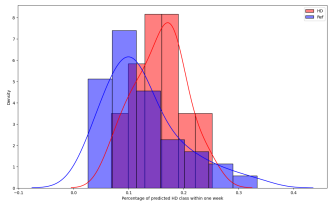
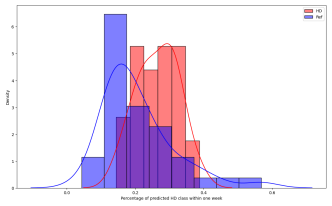
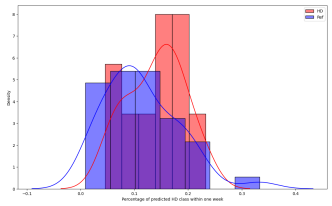
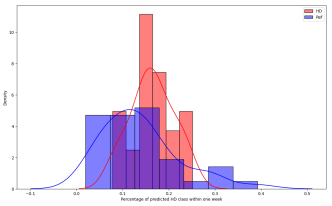
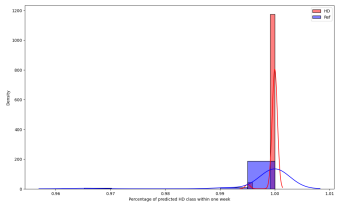
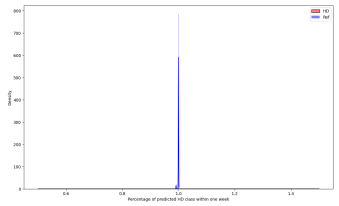
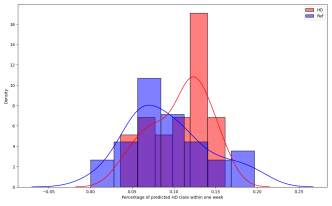
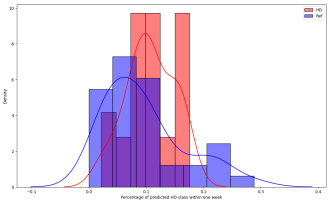
Configuration	1st network	2nd network
CE No weights		
CE Weights on Ref		
Custom loss		
Custom loss improved		

Table 12: Distributions of the percentage of weekly predicted HD peaks for both HD (Red) and Ref (Blue) group. These are created for the 1st and 2nd DL networks and the investigated configurations.