Applied Machine Learning Project Report

Kristian Andersen, Jeppe Dreyer Matzen, Oliver Kew, Rasmus Lynggaard

Abstract

Activity recognition with the purpose of minimizing human mistakes have been an important part of recent research and could be a beneficial contributor in many fields of work. This report carries out an analysis of features extracted from Electrocardiogram (ECG), Thoracic Electrical Bioimpedance (TEB), and Electrodermal activity (EDA) signals to recognize activity in four categories: Physical, Emotional, Mental and Neutral activity. A total number of 529 features are tested for the activity recognition. Others have used the dataset for classification of the same activities which has been an inspiration for the methods in this report. The classification is evaluated on the total recall for three different varieties of the dataset, using all original features, 40 feature-selected features with ANOVA, and 64 feature-extracted Principal Components. This work determines that the best classification occurs when using all the original features for classification reaching an average recall-score of 94%. Next, with the feature extracted data an average recall-score of 90% is obtained. Finally, with the feature selected data with 40 features reaches an average recall-score of 80%. When comparing to related work there is still issues regarding misclassification of mental and emotional activity which should lead to further research trying to address this issue.

1. Introduction

Activity can be defined as the state or quality of being active, which implies that the activity can be emotional, intellectual, physical, etc. When thinking of activity you would often think of physical activity such as running, training or cooking, mental activity like solving cross-word puzzles or emotional activities like being in love. There are many scientist who have tried to recognize human activity with different variations of machine learning through e.g. smartphones [1], video surveillance [2] or electrical signals from the body [3].

Activity recognition have many benefits. When used properly it can help prevent errors in human activity perhaps before the humans themselves notice they are making mistakes. By further developing of existing technologies this can be used by many professions around the globe. This report will investigate how different signals from the body can be used to recognize specific activities performed by people participating in the experiments.

In order to elicit the different activities, the following was used: A documentary to induce Neutral Activity. In order to elicit emotional activity, a set of segments extracted from 4 validated movies. The mental activity was elicited using a set of games based on mental arithmetic and playing the game Tetris. The physical activity is induced by making participants walking up and down stairs.

This report aims at utilizing the available data to classify different kinds of activity: Physical, Emotional, Mental and Neutral activity. The data is from the article [3] which consists of features from Electrocardiogram (ECG), Thoracic Electrical Bioimpedance (TEB) and Electrodermal Activity (EDA). The focus of the paper will be the classification of the activity while also diving into the importance of feature selection and Feature extraction in the evaluation of an accuracy score.

Email addresses: krand17@student.sdu.dk (Kristian Andersen), jemat17@student.sdu.dk (Jeppe Dreyer Matzen), olkew16@student.sdu.dk (Oliver Kew), ralyn17@student.sdu.dk (Rasmus Lynggaard)

The paper is structured as follows: Section 1 introduces the problem at hand; Section 2 is a presentation of the results of the available literature that examines the same data as used in this report, which is followed by a discussion of their methods in comparisons to the ones used in this report; Section 3 will present the proposed methods for this report; Section 4 will introduce the experiments conducted on the data to classify the four different activities; Section 5 will present the results of the experiments; Section 6 will discuss these results; Section 7 presents the main conclusions.

2. Related Work

This section will introduce related work to the dataset used for this report. The findings of two articles will be analyzed and discussed [3] [4]. The content of this section will primarily cover what has been in the course of Applied Machine Learning. If anything beyond the scope of the course is referenced in the articles it might not be covered in this report.
This section will serve as a starting point for this reports' proposed method as it will be an inspiration of improvements and other methods to conduct on the dataset.

2.1. ***Article 1***

The article "Activity Recognition Using Wearable Physiological Measurements: Selection of Features from a Comprehensive Literature Study" is using the same dataset as used in this report [4]. The purpose of the study was to recognize one of the four activities being done by the subjects: Emotional, mental, physical or neutral activity.
The authors of the article chose to find the best feature selection method for classifying the activity. They extracted all the features described in the literature and applied it in different experiments and the applied feature selection to determine the most suitable feature set and number of features. To classify the activity they have used a variety of well-known classifiers: Least Squares Linear Classifier (LSLC), Least Squares Quadratic Classifier (LSQC), Support Vector Machines (SVMs), Multi-Layer Perceptrons (MLPs), k-nearest Neighbor (kNN), Centroid Displacement-based k-Nearest Neighbor (CDNN) and Random Forests (RF).
The feature selection method used in this article builds on a version of Genetic Algorithm (GA). The algorithm takes all possible features and selects a combination of features that can solve the problem optimally. This was used with k-fold cross-validation where k = 40, which is the number of subjects. They use different combinations of the features and different number of features which then gives an error probability for each of the classifiers using the combination of signals. While doing feature selection the LSLC is used because of its fast learning process.
The result of the article shows that a window length of 40s for TEB signals, 60s for the ECG and 10s for the EDA are the best with error probability being the indicator. When combinations of the signals are used, a window of 40s was best. Therefore, they chose a fixed window length of 40s. Classification selection happens by comparing the classifiers with different combinations of features and combinations of signals. The best combination of signals for classification is all the physiological signals with 40 features, obtaining a 22,5% error rate. The second best is the combination of ECG and TEB with 60 features with an error rate of 24,5%. Furthermore, the authors examine the individual activities and how the classifiers performs on these. For neutral activity the best classification happens at a minimum error probability of 19,11% with the combination of all signals. In emotional activity the the least error probability is 27,14% with TBA and EDA signals. In mental activity the minimum error probability is 41,07% using a combination of all signals. The minimum error probability for physical activity ranges from 2,95% only using EDA-signal to 5,45% with ECG. The error probability with all the signals is 4,20%.
A confusion matrix in the article shows that the LSLC - with 40 features and a combination of all three signals - most commonly misclassifies emotional and mental activity. Furthermore, experiments show that when using all the signals the linear classifiers (LSLC and LINSVM) performs best. When using the ECG and TEB feature set the best results is again present with the linear classifier LINSVM (24,5% error probability). When only using the TEB signal the Random Forest classifier performs best with and error probability of 28,9%.

2.2. ***Article 2***

The dataset in the article "A Wrapper Feature Selection Algorithm: An Emotional Assessment Using Physiological Recordings from Wearable Sensors" is the same as the same used for this report [3]. The purpose of the article is to determine the appropriate feature selection to distinguish between three emotional states (sadness, disgust and neutral) measured with features from ECG and TEB signals.
A main focus of the article is that the algorithm that is chosen can run efficiently on a smartphone which is being considered throughout the entire article.
For feature selection they are using a GA like in the previous article. In this case the goal was to have the minimal classification error with a restraint on number of operations to cut the computational cost. The choice of feature selection algorithm was of great importance because of the small amount of available data in the dataset which could lead to generalization issues. To avoid this problem the authors decided to compare a standard-mean squared error (MSE) based function, called SDEO, and a novel k-fold-based fitting function, called KFBEO, aiming to find the better one in order to increase generalization capabilities in the feature selection process. The experiments conducted in the paper is done to evaluate the feature selection with four evaluation parameters: Computational cost, Performance of the classifier with MSE, ranking of features and power consumption in a specific smartphone.
On the measure of classification performance, the error probability was plotted as a function of the maximum number of operations per second. The SDEO performed a minimum error probability of appr. 26,5% for 50.000 operations per second, using 35 features. The minimum error probability in the KFBEO method was appr. 22,5% for 50.000 operations per second, using 10 features.
In the ranking of features in the two variations of feature selection the algorithms classification performance were presented as means of 100 estimations, all being 40-fold. Here the SDEO performed best with 35 features and the KFBEO performed best with 10 features. The sections with computational cost and power consumption will be left out in the reporting of results from the article since these are both results that primarily relates to the fact, that this study wishes to examine methods that can be used for smartphones. This has no relevance for this report and will not be discussed further.

2.3. ***Comparison and discussion***

The purpose of this report will only have some similarities with the articles described above. Both articles goes into depth with feature selection, which is not directly transferable to the experiments of this report. This is because the Genetic Algorithms used in the articles is not a part of the curriculum of this course and will therefore not be used for feature selection in the following experiments. Other types of feature selection - and Feature extraction - will be tested in the experiments to see how they perform in comparison to this method.
The article from [3] has few points to consider in the following report. This is due to the purpose of this report in comparison to the purpose of that article. The purpose of their article is to find the best feature selection method for classifying emotions. This is also a classification task, but the parameters for evaluation are different from each other. Since this report is not reliant on doing the classification on a smartphone the power consumption will not be of concern. Furthermore, the part with computational cost is always of some relevance, but this report will mainly focus on the number of features as a measure of computational cost of the classifier. Because of the focus on computational cost the error probability might be higher than if this was not a parameter for evaluation since there could be a correlation between the computational cost and error probability.
The two articles have an approach that is more general and therefor not very specific in the classification. The overall classification error probability ranges from 20-30% with some better in specific classification of activities like physical activity and worse in mental activity. What is sought in this report is to achieve a higher accuracy in classification of the four activities than in the article from [3]. The method used in [3] is a very extensive search for the best feature selection method for classification of the activity. Does feature selection actually increase the accuracy or does it only have an impact on the computational cost? And is this a trade-off that is reasonable? This will be examined in the experiments of this report to see if feature selection has a positive or negative influence on the accuracy. Furthermore, Feature extraction will be used to see how it influences the accuracy of the classifiers.

3. Proposed Method

This section will present the methods used for the experiments throughout this report. Furthermore, the use of different machine learning techniques for classification will be examined. First feature selection and Feature extraction will be explained. Next, three types of classifiers - Logistic regression, Random Forests and Neural Network - will be explained since these will be used to classify the activities.
A variety of experiments will be conducted with these methods. The goal of the following sections will be to find the simplest and most accurate method to classify activity in humans. The goal will be obtained by checking the importance of feature selection and Feature extraction on accuracy. Furthermore, to check an assumption that a more complex classification method is better than a simpler and less computational costly method. This will unfold in the following three sections.

3.1. ***Feature extraction***

Principal Component Analysis or PCA is a way to reduce the dimension of the feature space. This is also called Feature extraction.
PCA is often used if a dataset consists of many variables. When having a large amount of variables it is difficult to understand the relationship between each of the variables. Furthermore, having too many variables increases the risk of overfitting a model. [5]
There are two methods for dimensionality reduction: feature selecting and feature extraction.
In feature selection the feature space is reduced by selecting the features that contributes to explain the largest amount of variance.
In feature extraction new independent variables are created, where each new variable is a combination of all of the original variables. The new variables are ordered by how much they predict of the dependent variable. The variables that explain the least amount of variance, are eliminated. The constructed variables are independent of one another, but they are less interpretative [6].
A PCA works by first fitting a random line that goes through the data. The line is rotated to find the best fitted line, the best fit is found by determining the line that minimizes the distance from the data points to the line or by finding the line that maximizes the distances from the projected points on the line to the origin, the second method is called the sum of squared distances or the eigenvalue. The line with the largest sum of squared distances/eigenvalue is the best fitted line and this line will be the first PC (principal component) [6].

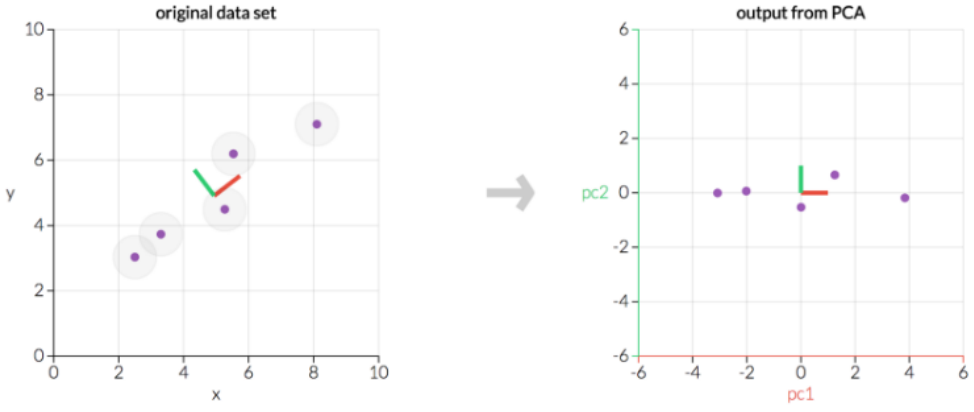

Figure 1: Transformation to align new directions

The next PC is found by drawing a perpendicular line to PC1. Next the orientation of the PC1 and PC2 is rotated so PC1 is horizontal, as shown in figure 1.

All the eigenvectors have eigenvalues. The eigenvectors of the covariance matrix are the directions of the axes where the most variance is found. The eigenvalues are the coefficients attached to eigenvectors and explain the amount of variance in each eigenvector [6].
The variation for each PC is found by dividing the eigenvalue by $n-1$, where n is the number of variables. There is as many PCs as there are variables in the dataset. The optimal number of PCs can be chosen with different methods. Method 1 where a number of PCs are chosen, often two if a graphical representation is desired. Method 2 where a threshold of the cumulative explained variance is set. Method 3 with a scree plot where the elbow point is found [6]. Method 3 is used in this report.

3.2. ***Feature selection***

Feature selection is the technique of reducing the number of input variables in model development. By reducing the number of features, the computational cost can be reduced and perhaps even better performance can be achieved. This is achieved by excluding possible features that are not relevant to the target variable [7].
In general there are three different supervised feature selection methods: Univariate, Model-based and Iterative. Iterative methods creates many models with different subsets of features and evaluates the models afterwards. Model-based models perform feature selection automatically as part of learning the model. An example of an model-based feature selection model is a decision tree. The last method is univariate. The univariate method uses statistical techniques to evaluate the relationship between each input feature and the target variable [7]. These statistical scores are used to evaluate each feature and select the best features for predicting the target.
The univariate method was chosen to perform the feature selection. There are different statistical measures to select from, but the choice is depending on the variable data types in the dataset. The dataset contains a numerical input variable and a categorical output variable and therefore the ANOVA feature selection method was chosen.
The ANOVA method is a univariate statistical measure. This means that the score is calculated with one feature at a time. Hereby ignoring any possible interaction effect between input variables. This can potentially lead to worse performance as information is lost in the feature selection. By applying ANOVA, the correlation score quantifies the association between any of the input features and the target variable. The correlation coefficient is a measure of the linear relationship. The logic behind by applying the coefficient is that a good variable will have high correlation with the output variable.

3.3. ***Logistic regression***

This purpose of this section is to present the relevant theory regarding logistic regression. Logistic regression is a technique for classification. Logistic regression is similar to linear regression, but instead of predicting some continuous value it predicts whether something is true or false. Logistic regression fits an S shaped curve ranging from 0 to 1, which indicates the probability [7].
This can be expressed as:

$$1/(1 + exp(-value))$$

Where value is the numerical value of the input variable. The logistic regression is used to predict binary classification, but can also be used for multi-class. This can be achieved in several ways. One way is to convert the classification from multi-class into binary by either doing one vs. rest classification, or one vs. one. Another approach involves changing the logistic regression to support the prediction of multiple class labels. This is a logistic regression that is adapted to learn multinomial probability [7].

3.4. ***Random forest***

In this section theory about random forests will be presented. In order to understand the random forest, the theory of the decision tree is presented. Random forest is an ensemble learning method used for either regression, or in this case classification. Ensemble learning means that the model builds on many smaller

models. In the case of a random forests, the smaller models are decision trees.
The entire random forest and decision tree-subsection is based on the theory presented in [6].

3.4.1. Decision tree

The decision tree is, like the Random forest, an algorithm that can be utilized in a regression or a classification problem. A decision tree can be thought of as a model that is continuously trying to divide the data into two. This is achieved by the two components of the decision tree: The leaf node and the decision node. The leaf node is the final outcome, and the decision node is where the data is divided. This division of the data relies on specific parameters, that can be tuned to fit a specific dataset. A decision tree for classification could look like the tree in figure 2.

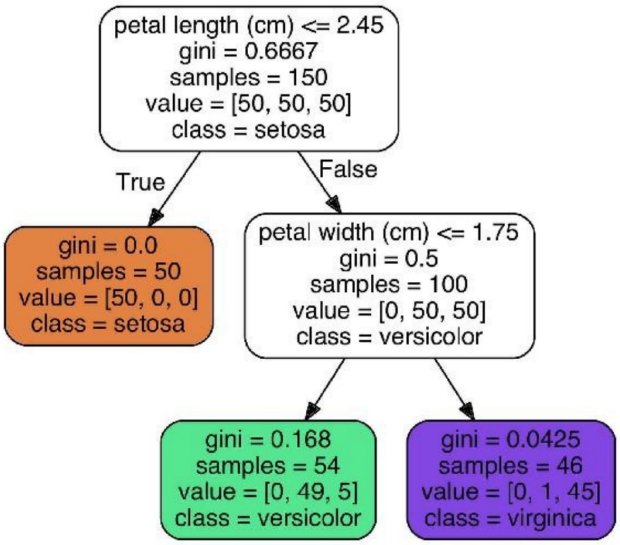

Figure 2: Decision tree example using the iris dataset

The tree exemplifies how the tree classifies the iris dataset into three classes based on petal length and petal width. There is a condition (petal size), that decides whether the instance is true or false, hence which leaf node the instance ends up at. To determine how to order the decision node a gini impurity score is calculated. The score is a measure for the likelihood of misclassifying a new instance, hence a low gini impurity score is desired. The exact formula can be seen below.

$$G = \sum_{i=1}^{c} p(i) * (1 - p(i))$$

G is the gini impurity score, c is the total amount of classes and p(i) is the probability of picking data from class i.
The node having the highest impurity is placed as the top node, and then the following nodes are placed by descending gini impurity scores. Looking at figure 2 it is seen that petal length have a gini impurity score of 0.6667 and petal width a score of 0.5 hence petal length is the first node in the tree.
The working of the decision tree makes it very prone to overfitting, this can be mitigated in a random forest.

3.4.2. Random forest

As the name suggest a random forest is a collection of decision trees. The random forests avoids the tendency of overfitting of decision tress by using many unique decision trees. The trees are randomly

constructed in two parts, first of a bootstrapped set of data, and a random subset of features. This means that two trees can consist of entirely different features, and classifying alike or consist of similar features, but classify different due to the different dataset. This constructs many different decision trees that together makes the forest better performing than decision trees.
The parameters for tuning a random forest algorithm tunes the single trees in the model, as well as the forest. For instance the depth of each tree is tuneable, as is the number of trees within the forest. The way the random forest works in a classification problem is that it collects the individual trees classification and the class with most votes wins, and the new data point is classified as said class.

3.5. ***Neural Network***

This section will present the Neural Network(NN) theory utilized in this report. A NN consists of three types of layers, an input, hidden and an output layer. Each layer consists of neurons. An example of a simple and deep Neural Network is shown in figure 3 [8].

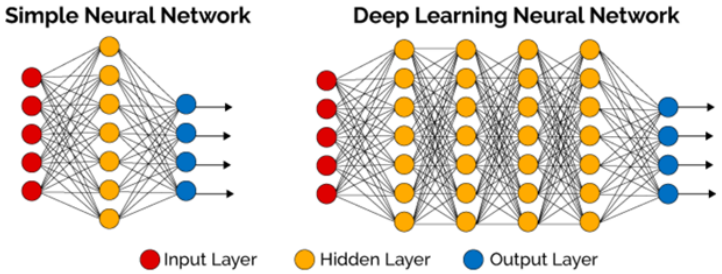

Figure 3: Example illustrating Neural Network

The dimensions of each layer, or neurons and the number of hidden layers is specific to each NN [7]. The architecture is based on a linear function, but these are separated by a non-linear function, the activation function. This functions' purpose is adding complexity to the model, which makes the model better at detecting non-linearities in the dataset. One of the most used activation functions are the ReLu activation function, this function allows all positive values, while all negative values are set to 0. The equation for this function is:

$$f(x) = max(0, x)$$

At the output layer the Softmax activation function is used for multi-class predictions. The Softmax function takes a vector as input and normalizes it into a probability in range of 0 to 1. The function is [9]:

$$probability = exp(value)/(sum(v) * exp(v)$$

Where value is the input and v is the list of values. The network consists of weights, which are updated by backpropagation during the training of the model. Backpropagation is performed by applying the chain rule and starting with the last parameter and working backwards to estimate all the other parameters [6]. The unknown parameters are then optimized with a specified optimizer algorithm after each batch of data. When a NN is being trained, a bunch of hyperparameters can be optimized to fit as good as possible e.g. number of epochs and batch size [7]. The number of epochs is the number of times the whole training data is shown to the network during training. Batch size is the number of data samples given to the network between updates of the weights. Another parameter that can be tuned is the regularization technique, dropout. This technique is very often used to avoid overfitting the model to data. The dropout rate indicates the percentage of neurons that should be ignored at each hidden layer.

4. Experiments

The following section will introduce the experiments conducted with the data to classify activity. First, an introduction to the dataset and the software used for the experiments will be presented. Next, the experiments will be described; logistic regression and Random Forests followed by the Neural Network-classifier followed by the results of the hyperparameter-tuning with the original data, feature extraction and feature selection. In each sub-section a description of the implementation of feature selection and feature extraction is presented as these plays an important part in the results of the report.

4.1. ***Dataset***

The following is described in the information-sheet on the dataset in [10]. In the present analysis, we have used four different activities:
Neutral activity, registered during the last 140 s of the first movie (the documentary). As each individual watched each movie twice, there are 280 s for each individual in the database
Emotional activity, registered during the viewing of the last 70 s of the second and third movies (140 s); therefore, we obtained a total of 280 s per individual.
Mental activity, registered during the last 140 s of both games, producing 280 s in total.
Physical activity registered during the last 280 s of the physical activity stage. To elicit physical load the participant had to go up and down the stairs for five minutes.
Each attribute was determined using a 40 s window. Measurements were collected from 40 subjects. The first column correspond to the index of the subject. The next 174 attributes are statistics extracted from the ECG signal. The next 151 attributes are features extracted from the TEB signal. The next 104 attributes come from the EDA measured in the arm, and the next 104 ones from the EDA in the hand. The last attribute is the pattern class, that is, the corresponding activity: 1-neutral, 2-emotional, 3-mental and 4-physical.
The designed activity recognition system had to take a decision every 10 s, and each individual generated 28 time slots of each activity (the database is balanced). Thus, the total number of patterns (decisions) for this analysis was 4480, and each class is composed of 1120 different patterns.

4.2. ***Software***

For this report different kinds of software has been used. Jupyter Notebook and Google Colab have been used for implementing the experiments. Both have been used since e.g. a Neural Network is better to run in Google Colab while using the free Google GPU. This report have been written in OverLeaf.

4.3. ***Data Pre-Processing***

Before conducting the experiments described in 3, data pre-processing is necessary. In this section the conducted prepossessing is explained.
The first pre-processing step was to make all labels unique. Issues arose when trying to do commands with the data. Therefore a uniquify-function was implemented to change any duplicates in the labels. Then the dataset was read again without the duplicates.
Next, the class-labels was changed from being an object to being an integer. This makes is easier when doing the experiments when looking at the output of the classification. This is not a necessary step, but it makes it easier to work with the data. Finally we check for NA-values of any sort. This is done by checking all available data to see if any values are missing or NA. There were no missing data in the data which is good since we do not have to make any assumptions of data values when doing the experiments.
Afterwards the data was scaled with the $MinMaxScaler()$ [11] function. The PCA and ANOVA was applied prior to splitting the data with the $train\_test\_split()$ [12] function of 0,8/0,2.
The next step in the preprocessing was to remove a couple of features. The subject index feature was removed and a $VarianceThreshold()$ [13] function was used to remove constant features from the dataset.

4.4. ***Feature extraction - PCA***

The first step was to investigate the optimum number of principal components. The PCA components was created with $PCA(n\_components = none)$ to see all the 529 possible components. The explained variance was plotted as a scree plot to find the elbow point [14]. The figure on the left in figure 4 shows the elbow point at 2 PCs. With 2 PCs 62% of the variance of the results is explained.

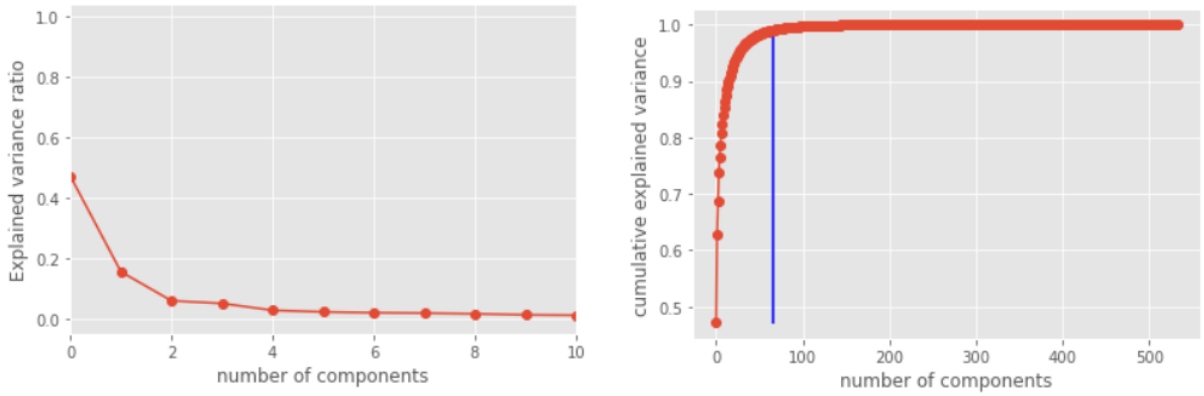

Figure 4: From left: Scree plot of PCA, Cumulative explained variance.

The 2 PCs resulted in a very low accuracy. The testing of the 2 PC was done with a GridSearch for logistic regression, the accuracy was 50,2%. The parameters for the GridSearch will be explained in detail in section 4.6. Investigating the figure on the right in figure 4, the cumulative explained variance hits 99% at 64 PCs and does not increase to 100% before all 529 PC is in use. With this in mind the number of PCs used for classification further on is 64. This is done by setting $n\_components = 0.99$ in the PCA. The next step is to fit and transform the data, this is done with $pca.fit\_transform(X\_scaled)$ [15], the function will conduct the PCA.

4.5. ***Feature selection - ANOVA***

To apply ANOVA as feature selection, functionality from scikit-learn has been applied. The functions $SelectKBest()$ [16] and $f\_classif()$ [17] is applied. The $f\_classif()$ function calculates the ANOVA correlation coefficients for each of the features. The $SelectKBest()$ function is applied afterwards to select a specific amount of features with the greatest correlation with the output variable. In this project a value of 40 features have been used, this has been selected due to the original article [3] uses the same amount of features in their work. This parameter has not been investigated further.
In figure 5 the calculated correlation coefficients for all features are seen. From the plot it is clear that the greatest coefficients are grouped together. From investigating the 40 greatest features it has been seen that many of these are derived from the ECG signal. 30 of the 40 features that have been selected on the basis of the ANOVA feature selection are from the ECG signals. 10 of the signals are form TEB-signals and none of the EDA-signals have been selected.

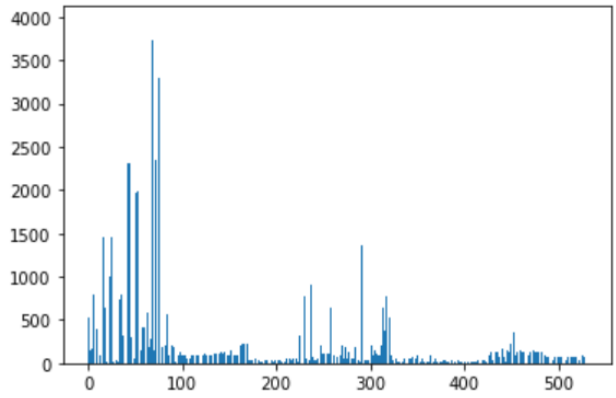

Figure 5: ANOVA correlation coefficients

4.6. ***Logistic regression***

For the development of a logistic regression model a grid search developed with the $GridSearchCV()$ [18] function, with a 3-fold cross validation and 0.2 validation split. including three hyperparameters have been made. The parameters that are tuned are the C parameter, that is the inverse of regularization strength. The penalty parameter is used to specify the norm used in the penalization. The third trained hyperparameter is the solver, which is the algorithm to use in the optimization problem.
The $LogisticRegression()$ [19] is able to conduct multi-class classification, by setting the parameter $multi\_class$ as $multinomial$. 3-fold cross validation as applied. For the parameter grid, the following values have been applied. $C = [1, 10, 20, 50]$, $solver = ['newton-cg', 'lbfgs', 'sag', 'saga']$, $penalty = ['l1', 'l2']$. Some of these combinations are not allowed as saga is the only solver that can handle l1 penalty term.

4.6.1. Original data

From the grid search the best mean cross validation score 0.865 was achieved with parameters found to be: C=50, penalty=l2 and optimizer=sag.

4.6.2. Feature extraction - PCA

From the grid search the best mean cross validation score of 0.837 was achieved with hyperparameters found to be: C=50, penalty=l2 and solver=newton-cg.

4.6.3. Feature selection - ANOVA

The grid search had the best cross validation scores of 0.73 with the parameters: 50 as inverse regularization strength, l1 as the penalty term and saga as the optimizer for the model.

4.7. ***Random forest***

In this section the experimenting, namely the tuning, of the random forest will be described. The tuning was performed in sections, using $for - loops$ and each iteration will be described starting with the initialization of the classifier.

For developing and tuning the random forest classifier, a random forest classifier were created using the $RandomForestClassifier$ [20] command. The classifier have several hyperparameters, but in order to keep the tuning time within reason, a few hyperparameters have been selected, these are, $n\_estimators$, $max\_features$ and max_depth.

The first hyperparameter tuned were the number of estimators, meaning the amount of decision trees created in training. The tuning were conducted in a $for - loop$ where the mean accuracy in a five fold cross validation were calculated for a specified amount of estimators and appended to a list to be plotted. The range of estimators went from 10 to 300 in increments of 10, the graph from the tuning can be seen in figure 6.

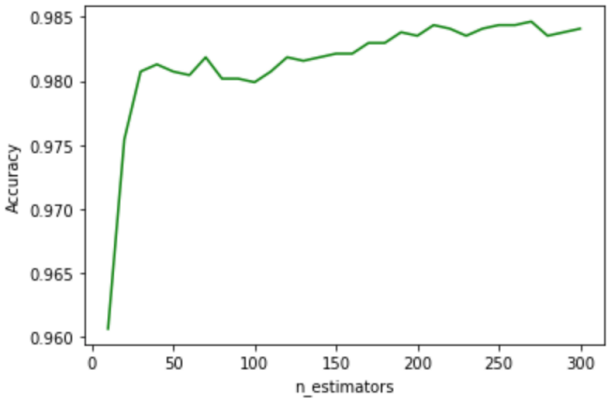

Figure 6: Accuracy of n_estimator tuning

As seen in figure 6, the optimal number of estimators is 270, resulting in the highest accuracy. The accuracy is above 98% for all estimators above 30, and for 270 estimators the accuracy is only marginally higher at 98.46%.

Using the same approach, as for the number of estimators the maximal number of features were tuned. The maximal estimator is the maximal amount of features each individual decision tree are allow to contain. The feature range went from 1 to 50 with an increment of 1.

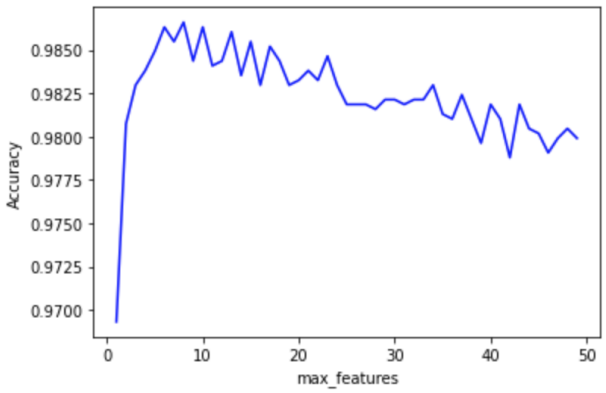

Figure 7: Accuracy of max_feature tuning

As seen in figure 7 the maximal amount of features that result in the highest accuracy is 8, and hereafter the accuracy decreases, the accuracy is fairly high for all the different numbers of $max\_features$.
The final parameter for tuning is the maximum depth of each tree, the approach for the tuning is the same as for the earlier tuning. The range of the depths tested goes from 1 to 50 with an increment of 1. The graph can be seen in figure 8.

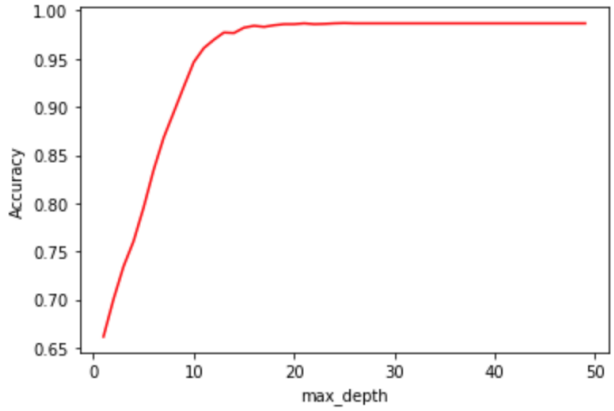

Figure 8: Accuracy of max depth tuning

Figure 8 indicates that the accuracy is approximately the same when the depth of the trees is above 15, reading the numbers it is found that the highest accuracy is found at a max_depth of 25.
This process was duplicated using the different dimensionality reduction methods on the dataset beforehand. The following sections will describe the model constructed using the different constructed dataset.

4.7.1. Original data

The parameters for the original data has been described above. A summary is presented here.
n_estimators = 270, max_feature = 8, max_depth = 25.

4.7.2. Feature extraction - PCA

The parameters found in the model using the components of the PCA, where much alike the parameters found using the original data.
n_estimators = 230, max_feature = 7, max_depth = 27.

4.7.3. Feature selection - ANOVA

The parameters found in the model using the features from the ANOVA filtered data, is the odd one out for the three models.
n_estimators = 260, max_feature = 33, max_depth = 30.

4.8. ***Neural Network***

This section will present the creation and tuning of the Neural Networks used in this report. To create a Neural Network for activity classification, the networks architecture was decided thereafter trained by the gridsearch function, to tune different tuneable hyperparameters. From the grid search the best set of hyperparameters was selected and used to test the model on the test dataset.
In the following section the methodology of the code for the NN will be explained. First the labels is one-hot encoded, this is done with the OneHotEncoder [21] function. Next, the KerasClassifier [22] wrapper for a classification in Keras is used. This is accomplished by creating a function, called $create\_function()$, which contains the model, a compilation and a return statement and is called by the wrapper function. The wrapper is then used as the estimator in the $GridSearchCV()$ function, with a 3-fold cross validation and 0.2 validation split. Finally the model is trained using the fit() function.
In the $GridSearchCV()$ function the parameter $param\_grid$ defines the different parameters that should be used in the grid search. The model is chosen as a simple Neural Network created with Keras as a Sequential model [23]. The model consists of an input layer followed by a repeated sequence consisting of a fully connected layer followed by a activation and dropout layer. This sequence is repeated twice. Following the last dropout layer is two fully connected layers, where the last layer has the same number of neurons as there are classes in the dataset, in this case 4 classes.
The hyperparameters chosen to tune in this network are: Number of epochs, batch size, activation function, optimizer, dropout rate and number of neurons in hidden layers. The approach to the tuning was to try a broad range of values for two hyperparameters, with all other parameters set at a specific value. For the activation function "ReLu" was set; for the optimizer "Adam" was chosen; the number of neurons in each hidden layer was 40; Dropout was set to 0,2. Epochs and batch size was not set as they were the first hyperparameters to be tuned.
For the tuning of epochs and batch size a list of values were created. The values of number of epochs were [25, 50, 100, 250, 400] and batch size [8, 16, 32, 64]. From this grid search, looking at learning curves to see when the training stabilizes, number of epochs was chosen to be 250, and batch size set to 32.
The next grid search tuned the activation function and optimizer. The different activation functions were: $['softmax','softplus','relu','tanh','sigmoid','linear','elu']$ and optimizers:
$['SGD','RMSprop','Adagrad','Adadelta','Adam','Adamax','Nadam']$. From the results it was found that especially three activation functions and three optimizers resulted in better validation accuracies. These were: $['relu','tanh','elu']$ and $['Adam','RMSprop','Nadam']$. These will be used in a final tuning in a grid search when the rest of the parameters have been tuned. For the broad search of number of neurons and the dropout rate, the activation function was chosen to be 'relu' and the 'adam' optimizer. The different values for the dropout rate were $[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$ and for number of neurons $[10, 20, 40, 80, 120, 160]$. When looking at the results it was found by inspection that lower values of dropout gave better results, so values of $[0.1, 0.2, 0.3]$ was chosen. For the number of neurons 80 and 120 neurons gave the best result.
After the initial grid searches, the ambiguous hyperparameter values were combined in a single grid search.

4.8.1. Original data

From the final grid search the best mean cross validation score of 0.911 and a train score of 0.972 was achieved. This result was achieved with the following hyperparameter: Epoch=250, batch size=32, $neurons = 120$, $dropoutrate = 0,1$, $activationfunction = tanh$, $optimizer = Nadam$.

The best activation function is the Tanh function. This function is similar to the sigmoid function with the typical S shape. It takes any value and outputs it in a range of -1 to 1. The function [9] is:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

The grid search found that the Nesterov-accelerated Adaptive Moment Estimation (Nadam) optimizer resulted in the best results. The Nadam optimizer is a combination of the Adam optimizer and Nesterov accelerated gradient. The learning process is accelerated by summing the decay for the previous and current gradient. [9]
In figure 9 an overview of train (noted as train) and validation (noted as test) results are shown for all different combinations of hyperparameters. From the graph the greatest value on the blue graph is not corresponding with the highest value in training graph. Further it can be seen that some overfitting is occurring due to greater values for the train data.

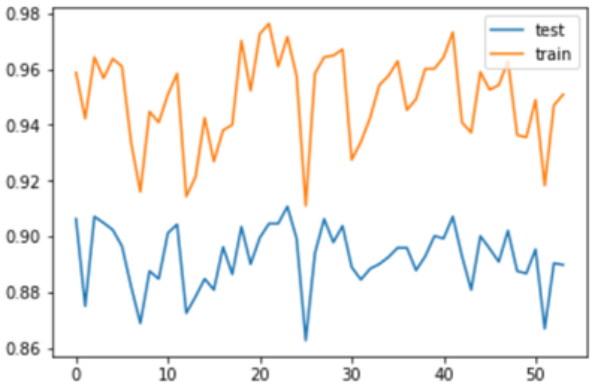

Figure 9: Train/validation score for all combinations of hyperparameters

4.8.2. Feature extraction - PCA

The same procedure was applied to the model working on the data with reduction in dimensionality. From the final grid search the best mean cross validation score of 0.905 was achieved. This result was achieved with the following hyperparameter: Epoch=250, batch size=32, neurons=120, dropout rate=0.3, activation function=tanh, optimizer=Adam.

4.8.3. Feature selection - ANOVA

The same procedure was applied to the model working on the data with reduction in dimensionality. From the final grid search the best mean cross validation score of 0.751 was achieved. This result was achieved with the following hyperparameter: Epoch=250, batch size=32, neurons=120, dropout rate=0,1, activation function=tanh, optimizer=Adam.

5. Results

In this section the results of the classifiers will be presented. A confusion matrix for each classification model fitted to the original, PCA or ANOVA filtered dataset, will be the focal point, combined with a summed up classification report, for each fitted model.
The matrices show the predicted classifications along the y-axis and true classifications along the x-axis. If the model makes perfect predictions, the confusion matrix should have 100% of the predictions matching the ground truth, meaning a diagonal line of 100%.

*5.1. **Logistic regression***

In this section the results from the classification with the logistic regression model will be presented. In figure 10 the confusion matrices from the three different datasets are shown. In the figure it can be seen that the model with the original and PCA data performs better, when compared to the ANOVA feature selected data. It can be seen that all classes perform good at classifying the physical class with recall values of 96%, 100% and 96%. The greatest misclassifications occur between the classes emotional and mental. The ANOVA filtered model has difficulties classifying the neutral class, with a recall value of 66%.

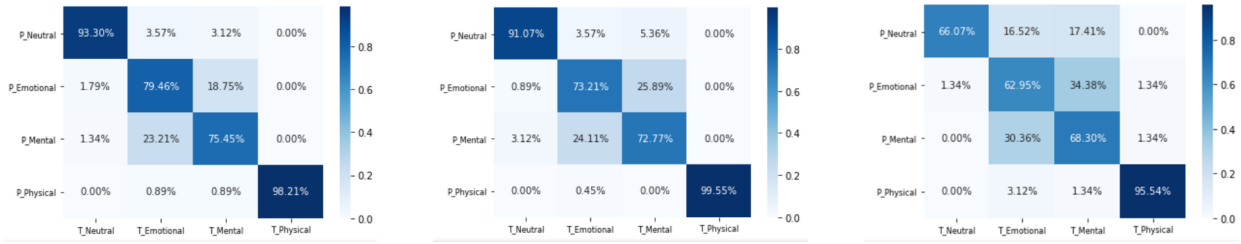

Figure 10: Confusion matrix for logistic regression. From left: Original data, PCA data, ANOVA filtered data.

A classification report for the logistic regression model has been made for each of the three datasets. In table 1 the overall precision and recall is presented. From the table it can be seen that the models with original and PCA gets 0.87 and 0.84 overall recall, while the model using the ANOVA feature selected data has a lower value of 0.73.

Classification report	Original	PCA	ANOVA
Precision	0.87	0.84	0.73
Recall	0.87	0.84	0.73

Table 1: Classification report summary for Logistic regression

5.2. *Random Forest*

In this section the results of the random forest classifier will be presented. Figure 11 shows that the classifier gives a relatively high percentage of correct classifications, the lowest being 83.93% correct classifications of emotional activity, after the features have been selected using the ANOVA method. The model that performs best is the model that is fitted using the original dataset, here there is a near perfect result, the model misclassifies 2.68% of the neutral activity, as being mental activity, the rest of the classification is at 100%

Figure 11: Confusion matrix for random forest. From left: Original data, PCA data, ANOVA filtered data.

Looking at the model fitted to the PCA, the ability to classify mental activity decreases, and in 11.16% are mistaken for emotional activity. The tendency is seen the other way around as well, where emotional activity is misclassified as mental activity in 8,48% of the cases. The recall values for the original data is 0,99, for feature extracted data using PCA the recall is 0,92 and for feature selected data using ANOVA the recall is 0,91.

Classification report	Original	PCA	ANOVA
Precision	0.99	0.92	0.91
Recall	0.99	0.92	0.91

Table 2: Classification report summary for Random forest

Looking at the classification report resume, seen in table 2, the overall best model is the model fitted to the original data, PCA and ANOVA is more equal, and peharps another random state in the split of the data could lead to ANOVA performing better that PCA.

5.3. *Neural Network*

In this section the results from the classification with the Neural Network model will be presented. In figure 12 the confusion matrices from the three NNs are shown. The matrices show the predicted and true classifications. From the matrices it can be seen that the model with the original data and the PCA data generally performs better for the classification of all classes. It can be seen from the ANOVA filtered data confusion matrix, that it predicts as good as the others on class physical. In all the NNs the greatest misclassification is between the emotional and mental classes.

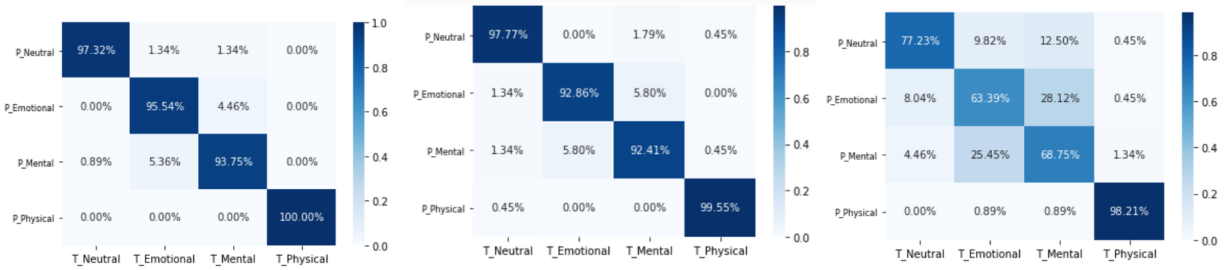

Figure 12: Confusion matrix for Neural Network. From left: Original data, PCA data, ANOVA filtered data.

A classification report for the Neural Network model has been made for each of the three NNs. In table 3 the overall precision and recall is presented. From the table it can be seen that the models with original and PCA gets 0.97 and 0.96 overall recall, while the model using the ANOVA feature selected data only has a value of 0.77.

Classification report	Original	PCA	ANOVA
Precision	0.97	0.96	0.77
Recall	0.97	0.96	0.77

Table 3: Classification report summary for NN

5.4. *Summary of results*

In this section a brief summary of the results from the different models is conducted. In table 4 a comparison on the obtained recall scores are seen. From the table it can be seen that all models perform best on the original data, worse on the components of the PCA and worst on the ANOVA filtered data. It can further be seen that the Random forest and Neural Network performs better than the logistic regression on both the original and PCA data. On the ANOVA filtered data the random forest performs better classification, when compared to the other models.

Precision	Original	PCA	ANOVA
Logistic regression	0.87	0.84	0.73
Random forest	0.99	0.92	0.91
Neural Network	0.97	0.96	0.77

Table 4: Summary of precision scores for all models.

In table 5, 6 and 7 the percentages for recall from the confusion matrices are used to compare the models, and how they classify each of the four classes in the dataset.
In table 5 it can be seen that on the original dataset all models perform quite well to classify classes neutral and physical. From the table it can also be seen that the random forest classifies better than the Neural Network when classifying the emotional and mental classes.

Recall for original	Neutral	Emotional	Mental	Physical
Logistic regression	93.3%	79.5%	75.5%	98.2%
Random forest	97.3%	100%	100%	100%
Neural Network	97.3%	95.5%	93.8%	100%

Table 5: Summary of recall for original data.

In table 6 some of the same pattern as in table 5 can be seen. A difference is that on the PCA data the Neural Network performs better than the random forest. It can also be observed that the logistic regression almost performs as well as the random forest for classifying the neutral and physical classes.

Recall for PCA	Neutral	Emotional	Mental	Physical
Logistic regression	91.1%	73.2%	72.8%	99.6%
Random forest	92.9%	89.7%	86.6%	100%
Neural Network	97.8%	92.9%	92.4%	99.6%

Table 6: Summary of recall for PCA data

In table 7 it is clear that all models performs best at predicting the physical class. Both the logistic regression and Neural Network is not scoring high percentages in any other classes. The random forest performs relatively well on all four classes.

Recall for ANOVA	Neutral	Emotional	Mental	Physical
Logistic regression	66.1%	63.0%	68.3%	95.5%
Random forest	88.0%	83.9%	91.1%	99.1%
Neural Network	77.2%	63.4%	68.8%	98.2%

Table 7: Summary of recall for ANOVA data

6. Discussion

In this section a discussion of the methods, experiments and results of this report will be presented. This will be done in the same order as the previous sections starting with Feature Extraction and Feature Selection, and the classifiers afterwards. Finally, a more overall discussion of the results and their implications will be presented.

6.1. Feature extraction & Feature selection

The goal with the PCA was to try to reduce the number of features without losing too much accuracy in the later classification. The first approach was to use the scree plot to find the elbow-point and use that amount of PCs. The scree plot showed that 2 PCs was the optimum number. The testing of the 2 features was done in logistic regression but the accuracy was very low (50,2%).
The next approach was to look at the cumulative explained variance. The plot showed that 64 PCs explained 99% of the variance. 100% was not reached before all 529 possible components was in use. The 64 PC's gave good accuracy and was therefore chosen.
Looking back at the approach in PCA both random forest and the Neural Network should also have been tested with 2 PC. This might have lead to a different amount of PCs in the following experiments.
A grid search on the optimum number of PCs for the best accuracy could have been done and might have shown that a lower explained variance was just as good as the 99% found in this report.
A different PCA method e.g. kernel PCA, could also have been tested to see if it resulted in better accuracy.
It could also have been interesting to analyse the loadings of the different variables in the PC, to see the composition of each PC, to see if any of the variables could be removed to make the PCs more interpretable.
The ANOVA feature selection method was chosen due to the numerical input and categorical output data.
It was also found that the original article [3] applied iterative feature selection and it was therefore sought to apply a different approach and see how applying a different feature selection affected the results.
In this project the ANOVA feature selection has been applied on both the logistic regression, random forest and Neural Network. By applying the feature selection method on all models it has been possible to compare how feature selection has affected the results on all models. This has been a good approach to be able to see the overall effect of applying feature selection.
A downside of the ANOVA is that it is a univariate statistical feature selection method. If two features are correlated, the model only really needs one of them, as the second one does not add additional information. This is not taken into account in this project. The ANOVA feature selection selects the 40 best features for classifying the activity, but as previously written 30 of the 40 features was from the ECG-signal and the rest from the TEB-signal. The fact that the majority is concerning the heart and changes in the rhythm could imply that the ANOVA would be best at predicting the class physical, where the subject walked up staircases. This has been confirmed by analyzing the results. Furthermore, it could be interesting to look into other feature selection methods that includes EDA-signals. With ANOVA feature selection method this type of signal does not seem important when classifying, but maybe another feature selection method would show a different result.
To improve the feature selection it would be interesting to look at correlation between features and exclude features that contributes with the same information. Further it would be interesting to analyse the effect of changing the amount of included features. By doing this an improvement in the classes emotional and mental could be observed.

6.2. Classifiers

6.2.1. Logistic Regression

The logistic regression was applied in this project to try a linear model for classification and see whether it resulted in good classifications.
Linear models are very fast to train, and also to predict. This was also observed during training of the model.
From the results it has been seen that the model generally achieves great scores for neutral and physical with

the original and PCA data, but has trouble classifying the emotional and mental classes. When applying the ANOVA filter, the model has further trouble classifying the neutral class. These results implies that the data is non-linear, and therefore it is difficult for the linear model to classify non-linear data.
An interesting subject for further investigation would be to apply other methods of classification. The methods one vs. one or one vs. rest could be implemented and tested to see whether this could improve results.

6.2.2. Random Forest

The random forest algorithm performs very well on the data, especially when trained with the original dataset. The accuracy for the algorithm when trained on the altered datasets performs worse, but the parameter tuning time is reduced significantly. The training time for the three models remains equal, which suggest the best model is the one trained on the original dataset. The parameter tuning is a one time procedure, and when said parameters is found the model can be trained and used for classification.
Regarding the number of estimators and max depth the two are not very different in the models trained using the original data and PCA. The max features is considerably higher than within the other models. This could be due to the fact that ANOVA selects the features that explains the most variance of the dataset, hence they are all relatively important in order to classify.
The parameters selected for tuning, where selected randomly, and only three of the many possible hyperparameters have been altered from default. It is possible that tuning more parameters could lead to a even better model. The reason for not tuning the model further where the substantial training time during tuning and the high accuracy achieved with the current model. If the tuning were to happen again a more strict scientific approach would have been of choice. The selection of parameters to tune, should have been chosen with starting point in empirical knowledge, giving a better reason for choosing said parameters.
Another approach for tuning the parameters could be to use the GridSearch function. This would allow to test the model with every constellations of parameters, instead of testing parameters individually. The GridSearch could lead to a new and more accurate model.
An interesting thing to test would be the tuning time of the model, varying the amount of estimators, to see how it would affect the computational cost. In the experiment section of the random forest, it was found that 30 estimators resulted in 98% accuracy and 260 estimators resulted in a 98,48% accuracy. Sticking to 30 estimators would affect the model accuracy minimally but perhaps reduce the tuning time in an amount that would allow for tuning more hyperparameters.
This could be due to the components in the PCA regarding emotional and mental activity are not separated into separate components. The same reason could explain the misclassifications of the emotional activity that are classified as mental activity.
The model fitted to the ANOVA filtered dataset, shares the same problems in misclassifing between emotional and mental activity, this could be because of the ANOVA filter favoring the data regarding heart monitoring through the ECG. This ensures that the model is good at differentiating between physical and neutral activity, but not between mental and emotional.
It could have been interesting to try other tree classifiers, for example try to make a decision tree, in order to compare the results, and verify that the random forest actually is superior, and if that was not the case then the decision tree would have been a better algorithm regarding the computational requirements. In either case is the tree classifier an easy model to comprehend, the "black box" term is totally absent when working with trees, hence the workings of the model is easier to pass on to people who do not know machine learning which could be of interest especially in healthcare.
It would have been interesting to see how the random forest compared to other ensemble models as well. It could be AdaBoost build on decision trees. It work by training the tree, find the pitfalls in the trees classification, and then train a new tree adjusted to focuses more on the hard cases, leading to a potentially better model.

6.2.3. Neural Network

The developed Neural Network performs well on the original and PCA data, but has difficulties with the ANOVA data. This is probably due to the data lacking information on the different classes, and not due to the network.

The hyperparameter optimization was conducted over several steps, due to the large computational costs of the tuning process. This can have led to a lack of possible great combinations of hyperparameters. Another approach would be to implement the random search instead of the grid search. This would decrease the computational cost, but might also decrease the accuracy of the classification.

The Neural Network performs almost as good on the PCA(0.96) as the original(0.97) data, and the PCA is a simpler model and hereby having less computational costs. The PCA is therefore the preferred data for the Neural Network. Another possible improvement that has not been investigated is the number of hidden layers in the model. By adding hidden layers the model will gain more complexity, but might also improve the classification results.

In comparison to the results of the original article [3], the results obtained in this research performs better, especially for classifying the emotional and mental classes. As described earlier the article [3] uses feature selection prior to all models. In this study it has been found that by using the original dataset or applying dimensionaly reduction by PCA performs better then the ANOVA feature selection applied in this study. The best performance in classification comes with the original data containg all the features. This comes with a higher computational cost which is something to consider when figuring out how to use the data. If the classification should be done real-time you would have to consider if the high performance of a classifier is better than fast classification and lower computational cost.

The accuracy achieved by the random forest and Neural Network seems too high when compared to the accuracy achieved in article [3]. When using all features the article achieves a error rate probability of 26% using random forest, which approximately translates to an accuracy of 74%. It could seem that there is some problems in either model, a test accuracy of 99% as in the random forest or 97% as in the Neural Network should be subject of scepticism. It seems "too good". It could be because the model is overfitting and the test data is too similar to the training set.

7. Conclusions

In conclusion on this report the results show that there is a difference in the performance of a classifier when comparing the use of all features, 40 features selected by ANOVA-feature selection and Feature extraction with PCA. In the three classifiers the best classification occurs when using all the original features (averaging 94%). When using the PCA reduced dataset the classification is lower (averaging 90%) than the original data. Worst of the three is the classification when using the feature selected data with ANOVA (averaging 0,80%). This means that in this case, with these classifiers and the parameters used, it would be better to keep all the original data when doing the activity recognition. Overall there is still issues regarding misclassification of emotional activity and mental activity as in the primary article [3]. The classification using original data and PCA outperforms the classification in the article when comparing the results of emotional and mental activity-scores. The feature selected data with ANOVA did on the other hand perform worse than the primary article in both Logistic regression and Neural Network. The Random Forest performs better.

Summing up, a lot could have been done differently, the models produced in this report was produced within the time constraint and current knowledge of machine learning. More time for research could have led to more complex and potentially better model, this was an natural boundary for the report.

8. Future Work

In future work it would be beneficial to look into other methods to compensate for still relatively poor classification of emotional and mental activity. This could be done by trying other combinations of feature selection, Feature extraction or classifiers and with other parameters which potentially could lead to a better classification. Furthermore, it would be interesting to go into the details of the feature selection with ANOVA. Here there are several things to investigate. In the PCA feature extraction it could be interesting to look into the correlation between features, and analyze the effect of changing the amount of features. In logistic regression there are still ways of classifying that has not been investigated in this report e.g. one vs. one or one vs. rest. For the random forest it would be of interest to tune more hyperparameters, and different constellations of these. For the neural network it would be interesting to work on the depth of the network.

9. Acknowledgements

We would like to thank Inma Mohino Herranz, Roberto Gil Pita, Manuel Rosa-Zurera and Fernando Seoane for the collection of the data, that has been applied in this report.

10. Contributions

We (Oliver Kew, Jeppe Dreyer Matzen, Rasmus Bank Lynggaard and Kristian Egstrand Andersen) hereby declare that all authors of this report have contributed equally to the project. All sections of this report have been approved by all members of the group.

References

[1] S. K. Bashar, A. Al Fahim, K. H. Chon, Smartphone based human activity recognition with feature selection and dense neural network (2020) 5888–5891doi:10.1109/EMBC44109.2020.9176239.

[2] K. K. Htike, O. O. Khalifa, H. A. Mohd Ramli, M. A. M. Abushariah, Human activity recognition for video surveillance using sequences of postures (2014) 79–82doi:10.1109/ICeND.2014.6991357.

[3] M. R.-Z. F. S. Inma MohinoHerranz, Roberto GilPita, Activity recognition using wearable physiological measurements: Selection of features from a comprehensive literature study, Sensors (2019) 1–27doi:10.3390/s19245524.

[4] J. G.-G. M. R.-Z. F. S. Inma Mohino-Herranz, Roberto Gil-Pita, A wrapper feature selection algorithm: An emotional assessment using physiological recordings from wearable sensors, Sensors (2020) 1–15doi:10.3390/s20010309.

[5] scikit learn.org, Decomposing signals in components: Principal component analysis, latest visit 12.may, 2021 (2020).
URL https://scikit-learn.org/stable/modules/decomposition.html#pca

[6] A. Géron, Hands-On Machine Learning with Scikit-Learn & Tensonflow, O'Reilly Media, 2017.

[7] A. C. M. . S. Guido, Introduction to Machine Learning with Python, O'Reilly Media, 2017.

[8] masteriesc angers.com, Deep learning, latest visit 12.may, 2021 (2018).
URL https://master-iesc-angers.com/artificial-intelligence-machine-learning-and-deep-learning-same-context-different-co

[9] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[10] mdpi.com, Activity recognition using wearable physiological measurements: Selection of features from a comprehensive literature study, latest visit 11.may, 2021 (2019).
URL https://www.mdpi.com/1424-8220/19/24/5524

[11] scikit learn.org, class sklearn.preprocessing.minmaxscaler, latest visit 14.may, 2021 (2020).
URL https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[12] scikit learn.org, sklearn.model$_s election.train_t est_s plit$, latest visit 14.may, 2021 (2020).
URL https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[13] scikit learn.org, sklearn.feature$_s election.variancethreshold$, latest visit 14.may, 2021 (2020).
URL https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html

[14] S. Galli, Python Feature Engineering Cookbook, Packt Publishing Ltd, 2020.

[15] scikit learn.org, sklearn.decomposition.pca, latest visit 14.may, 2021 (2021).
URL https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html#sklearn.decomposition.PCA.fit_transform

[16] scikit learn.org, sklearn.feature_selection.selectkbest, latest visit 12.may, 2021 (2021).
URL https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

[17] scikit learn.org, sklearn.feature$_s election.f_classif$, latest visit 12.may, 2021 (2021).
URL https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html

[18] scikit learn.org, sklearn.model$_s election.gridsearchcv$, latest visit 14.may, 2021 (2021).
URL https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[19] scikit learn.org, sklearn.linear$_m odel.logisticregression$, latest visit 14.may, 2021 (2021).
URL https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[20] scikit learn.org, sklearn.ensemble.randomforestclassifier, latest visit 14.may, 2021 (2021).
URL https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[21] scikit learn.org, sklearn.preprocessing.onehotencoder, latest visit 14.may, 2021 (2021).
URL https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

[22] tensorflow.org, tf.keras.wrappers.scikit$_l earn.kerasclassifier$, latest visit 14.may, 2021 (2021).
URL https://www.tensorflow.org/api_docs/python/tf/keras/wrappers/scikit_learn/KerasClassifier

[23] keras.io, The sequential model, latest visit 14.may, 2021 (2021).
URL https://keras.io/guides/sequential_model/