Mood State Classification Using Time Series Data From Wearable Health Devices

Stefan Lewis

Master of Science
School of Informatics
University of Edinburgh
2023

Abstract

Mood state disorders such as depression and bipolar disorder significantly impact an individuals ability to function and are prevalent across the world with depression alone impacting 264 million people globally. Diagnosis of these disorders using the typical patient interview process has substantial challenges such as long wait times to see appropriate healthcare professionals. Identifying mood state using patient physiological data collected from wearable health devices and machine learning provides an opportunity to reduce these challenges by aiding diagnosis in a cheap and scalable way. Most research in this area mainly focuses on creating high performance machine learning models capable of accurately identifying mood state model performance but often neglect to focus on explainability of model prediction, particularly generating explanations for specific patients. For mood state detection via machine learning to be widely adopted, healthcare professionals must be able to trust the predictions being made. This project puts model explainability at the forefront and presents the first (of our knowledge) machine learning model capable of identifying mood state using data from wearable health devices with capability to explain how model inputs contribute to individual patient predictions.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Stefan Lewis*)

Acknowledgements

I would like to express my deepest appreciation to Antonio Vergari and Filippo Corponi for their support and guidance throughout this project.

I would also like to give a special thanks to my fiancée Lucinda, who has continuously supported me throughout this project and MSc. I could not have done this without you.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

[i] Mood disorders are a group of medical diagnoses that impact general emotional state or mood in a way which is inconsistent with the person's circumstances and often causes individuals to have an impaired ability to function [3]. These diagnoses include depression and bipolar disorder and are prevalent across the globe, with depression alone impacting 264 million people globally [41]. These mood disorders heavily impact patients' quality of life and are ranked as one of the top disease burdens worldwide [11].

The typical process of patient diagnosis for mood disorder is through a mental health professional conducting an interview, asking questions about symptoms, habits and behaviours [1]. Psychometric scales are often used to help assess the severity of mental health symptoms patients are experiencing. Two examples of these scales are the Young Mania Rating Scale (YMRS) [42] which is a questionnaire used to measure the presence and severity of mania symptom, and the Hamilton Depression Rating Scale (HDRS) [19], another questionnaire to identify symptoms of depression. Each of these ask patients a number of questions with responses given by patients on an ordinal scale. These responses can then be used to measure the presence and severity of symptoms.

While the standard process of patient interview has been effective at diagnosing mood disorders, the process of patients' self-description of symptoms during a short interview period can be sub-optimal [18]. Patients' self-description of symptoms have been shown to be inconsistent depending on factors such as mood, weather, and time of day.

[i] Section 1.1 is heavily influenced by the Informatics Project Proposal[23]

Furthermore, the mental health professional conducting the interview is only observing the patient for a short period of time, and waiting times to see specialists are often long, leading to equally long wait times before diagnosis [30].

Machine learning models using patient physiological and behavioural data has shown to be a potential alternative to the traditional interview-based diagnosis process [17][10]. The arise of easily accessible consumer grade wearable health devices such as ”fitbits” allows for numerous physiological and behavioural data channels to be collected and used as the basis for machine learning diagnosis [37][14]. Health devices and machine learning models have the potential to be deployed at scale to aid diagnosis in a cheap and effective manner and overcome many of the aforementioned issues facing the traditional diagnosis process. Previous works in literature have shown the efficacy of mood state diagnosis via machine learning models [32][2], however there are still a number of challenges before their use can be widely deployed in a clinical environment.

While mood disorder diagnosis by machine learning provides a number of opportunities, for it to be widely adopted models are often required to be “explainable” [43]. Machine learning model explainability refers to the concept where humans are capable of retaining intellectual insight over the model. That is, to take a machine learning model and explain it in human terms. Many machine learning models such as neural networks are very complex and struggle to allow the user to understand how a decision was made by the model [44]. In sensitive industries such as healthcare, where an incorrect decision can have severe and life changing consequences, healthcare professionals often require insight into how a model makes its predictions, in order to build trust and ensure the model is accurate, fair and transparent [43].

Many previous works within mood state classification have focused on creating high performing machine learning models [17][32][2]. Model performance is an important factor in the applicability of a machine learning model predicting mood state. However, explainable machine learning models with a similar or even lower model performance to that seen in literature could be more valuable in a clinical setting due to the need for trust to be built by healthcare professionals for widespread adoption [43].

Few papers have focused on making mood state diagnosis models more explainable. Whilst some have provided deeper insights into their models [10][34], none of these works have specifically focused on generating a model that is designed to be explainable from the start with a constant focus on explainability throughout development. This important unexplored area of research is the focus of this project.

1.2 Project Objective and Contributions

It is the opportunities and challenges outlined in section 1.1 that drives the main research question of this project: can mood state be accurately identified with explainable machine learning models using data from consumer-grade wearable health devices? This project aims to answer the core research question by building machine learning models using data collected from wearable health devices.

The objectives of this project can be summarised into the following:

- Extracted features from the patient time series data.
- Develop baseline models to benchmark and evaluate project developments.
- Develop explainable machine learning models to identify mania and depression mood state.
- Identify and implement steps to improve model performance whilst retaining model explainability.
- Apply techniques to the top performing models to further improve model explainability.

Each of the project objectives were successfully completed within the project with the best performing model generating an f1 score of 0.52, twice that of the baseline model (see section 3.8.2 for f1 score definition). This project is also the first known mood state classification model to generate explanations for individual patient predictions.

1.3 Structure of the Report

The remainder of the document is split into 4 sections. Section 2 discusses the background surrounding this project, where a literature review critically reviews works that are relevant to this project. Section 3 covers the methods undertaken to meet the objectives of this project. This includes the conceptual design of the experiments, the methods carried out and the challenges faced during their implementation. Section 4 critically discusses the results, their performance in context to relevant literature, and their contributions to the research area. Finally section 5 will conclude the research project summarising the work completed, the contributions made, and the potential directions for future work.

Chapter 2

Background

To ensure the research discussed in this literature review is relevant to this project, the research must meet three criteria. First, the data used to identify mood state or disorders must be time series data collected from wearable health devices. It is the low cost and easy use of these devices that enables the potential solution of large-scale mood disorder diagnosis to be scaled to match the number of individuals impacted. Second, mood state or mood disorder symptoms should be evaluated using the YMRS and HDRS scales [42][19]. This criteria is set to be consistent with the data used in this project and make more relevant comparisons. Thirdly, the types of models used to aid mood disorder diagnosis should be from machine learning, again to provide direct comparison to the work carried out in this project.

The relevant literature can be divided into two key categories, classification and regression tasks. Regression tasks (i.e. predicting numerical scores) are limited within literature, with only three known papers focusing on this area. The typical motivation behind the regression-based tasks is to identify the level of symptomology experienced by a patient.

2.1 Regression works

In 2017 a regression task was set up to predict the overall scores of the HDRS [17]. Data was collected using E4 wristbands to obtain physiological signals. However, additional data was collected on how the participant was using their phone, including meta-data of calls, text messages, location and app usage. 700 features were crafted and used across 6 different machine learning models, including Random Forrest, Gaussian Processes and a customized ensemble method that uses all 4 other methods to generate a more robust prediction. The best performing model was the ensemble approach with an average Root Mean Square Error (RMSE) of 4.5 on the test set.[i]

More recently in 2020, researchers were able to identify the HDRS total score with an average RMSE of 5.35 when using physiological data collected from an E4 wristband [29]. Additional data collected from smartphones such as number of calls, text and activity patterns were used alongside the physiological data to train alternative models. However, the performance of these models decreased generating an RMSE of 5.43.

The most recent regression-based work predicted both the overall and individual item scores of the YMRS and HDRS scales [10]. The researchers were able to predict the overall YMRS and HDRS using the same dataset used within this project. Using an artificial neural network the overall YMRS and HDRS scores were predicted with an average RMSE of 5.6 and 4.4 respectively. The authors also predicted the individual scores for each item in the YMRS and HDRS questionnaire. Providing the item level prediction gives a deeper level of understanding of the diagnosis as two identical overall YMRS or HDRS scores can have very different symptomology.[ii]

2.2 Classification works

Patient mood state labels can be generated by converting the sum of the YMRS and HDRS scales scores to a binary state (0 or 1), with scores crossing thresholds defining an acute mood state. Across the literature different thresholds were used to identify mood state.

In 2022 research was conducted to try and distinguish between two groups of patients: those who are experiencing euthymic or mania mood states [2]. To achieve this, patients wore wristbands on their dominant wrist for 24 hours to measure data including 3-axis acceleration, EDA sensor data, skin temperature and photoplethysmography (a method to detect blood volume changes) data to derive heart rate. To obtain labels required for the machine learning approach, patient mood states were established at regular intervals using the YMRS, with total scores ranging from 0 to 60 and scores below 10 being classified as euthymic. Data processing and feature extraction are key aspects to the success of the research with features such as heart rate

[i] Paragraph is taken verbatim from the Informatics Project Proposal[23]

[ii] Paragraph is heavily influenced by the Informatics Project Proposal[23]

variability, bipolar complexity-variability (BCV), and mean amplitude of the Skin Conductance Response (SCR) used as inputs for the machine learning model. These example features and others were used to develop a deep-learning approach with a long-short ensemble network that achieved a classification accuracy of 91.59% for euthymic/manic mood states [2].

A second paper attempted to use machine learning to identify patients with depression symptoms. Data was collected from an "Silmee W20", which is a wearable health device that captures similar data to the E4 Empatic wristband, but with additional channels including sleep time and ultraviolet light exposure [35]. The paper achieves a 0.76 accuracy of identifying symptomatic patients. Researchers found that skin temperature and patient time spent asleep were important in identifying patient mood state. Although positive results were achieved, there were several limitations with this approach. There were 86 patients used in the study which may have weakened the statistical significance of the results. Patients with illnesses other than depression were not considered which simplified the task compared to real world scenarios. Patients with multiple illnesses including depression could show additional symptoms which the model would likely struggle to accurately model. The model accuracy would therefore likely decrease in real world use.

Obtaining sufficient data with mood state labels is a significant challenge within the research area. It should be noted across all works described in this literature review that limited data and model overfitting is a common challenge in this field.

2.3 Model explainability

There are numerous techniques available to improve model explainability such as LIME and counterfactuals [27][31]. However, the most commonly used is SHAP [24]. SHAP stands for SHapley Additive exPlanation and combines shapley values from game theory with local explanations to explain the output of any machine learning model. There are a number of SHAP visualisations which can give insight into feature importance, feature relationships, and much more [25].

Throughout all the works discussed in this literature review, none have put model explainbility as a core focus throughout. Many works have discussed feature importance, and many have explained in depth the characteristics of their models

[10][17]. One tangentially related paper (but was not part of the core literature review as it did not use HDRS or YMRS scores) used SHAP to identify which model features most impacted the model outcomes [34]. These are all examples of global explainability, which aid the understanding of the data. However, none of these models have generated local model explanations, that is, to explain individual decisions made by a model.

2.4 Dynamic Time Warp

Dynamic Time Warp (DTW) is an algorithm used to identify similarity between two temporal sequences. The approach works by minimizing the Euclidian distance between aligned time series data. The efficacy of DTW comes from its ability to deal with time series data that have different velocities or are shifted from one another [40]. This approach is not implemented in this project but is considered as one of the potential solutions to the project objectives.

Chapter 3

Methods

3.1 Methods Introduction

The possible approaches to solving the key project objectives can be summarised into three broad categories:

1. Train a deep learning model with the unprocessed time series data as input.
2. Compare similarity to other example data using dynamic time warping and apply a similarity model to determine mood state.
3. Train a machine learning classification model using summary and descriptive statistics extracted from the time series data.

Approach 1 is the most common approach seen in literature. Whilst this approach yields good results due to the powerful deep learning models available, it also has poor explainablility due to its complexity. Model explainablity techniques such as SHAP can still be applied to these models to inform the user which input has the strongest influence on the outcome. However, as the inputs to the model are unprocessed time series data, it would not be interpretable to users, rendering this approach inappropriate for this project.

DTW has not been used in the context of identifying mood state using wearable health device data. This method has the potential to outperform other methods due to its high performance on other time series classification tasks [26]. Using this method for this project would consist of dynamic time warping comparing similarity between other data points then using a model such as K Nearest Neighbours (KNN) [15] to classify mood state. This approach would not be appropiate for this project as the features generated from dynamic time warp are not easily human interpretable and so again would limit the explainability of the model.

Approach 3 uses a combination of feature extraction and a machine learning classifier to identify mood state. The first step in this approach processes the time series data into summary and descriptive statistics that ideally capture key information. The second step requires a machine learning model to be trained on this processed data. To ensure this approach generates an explainable model the features extracted need to interpretable by non-technical users and the machine learning model needs to be explainable. For example, taking the mean value of the acceleration data channel is interpretable by non-technical users as it can be understood as an estimate of how much the patient moved during that given recording segment. Using this data in a simple model such as logistic regression can tell us how important the feature was in the outcome. We require both steps to prioritise explainability to ensure the outcome is explainable.

Approach 3 is the most appropriate of the possible methods and will be used for this project.

3.2 Task Summary

The key focus of the project is to build machine learning models capable of identifying mood state whilst retaining model explainability. This project does so by training models on a dataset containing over 7000 hours of patient physiological data with labels identifying patients depression and mania mood state. Before machine learning models are trained features need to be extracted from the time series data, data needs to be segmented and cleaned, evaluation methods and metrics need to be chosen, and baseline models need to be implemented to benchmark performance. Each of these steps are discussed in detail from section 3.3 to 3.12.

3.3 Data

In these experiments we used patient physiological and behavioural data recorded using Empatica E4 devices worn on the patients' non-dominant wrists. This data was collected from 267 recording sessions from 140 individuals generating over 7000 hours

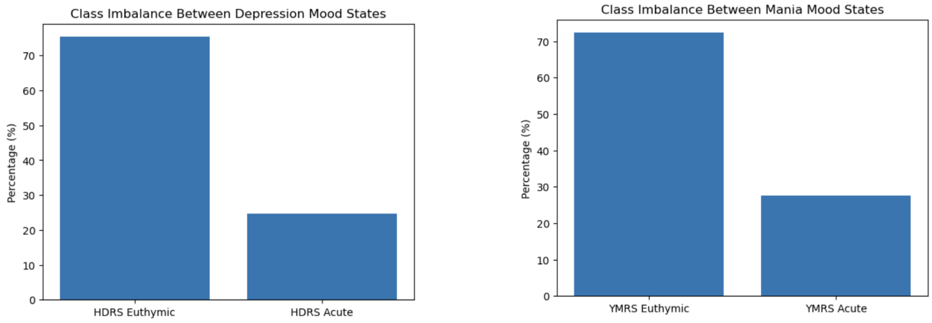


Figure 3.1: Bar chart visualising the class imbalance of HDRS and YMRS mood states.

of times series data. Each session recording typically lasted 48 hours until the E4 device ran out of battery, with some subjects undertaking multiple monitoring sessions. The wristbands collected 6 channels of information: 3D acceleration (ACC), blood volume pressure (BVP), electrodermal activity (EDA), heart rate (HR), skin temperature (TEMP), and interbeat intervals (IBI). These data channels were sampled at a rate of 32, 64, 4, 1, and 1 Hz respectively with IBI sampled at non-regular intervals. A subset of this data has been used in previous publication [10]. However, more data has been collected since these previous experiments were carried out.

To generate labels, subjects were assessed using the HDRS and YMRS questionnaires to understand the depression and mania symptoms present at the time. The questionnaires generate a score, with values greater than 6 representing an acute mood state and below or equal to this threshold showing a euthymic mood state. This threshold was selected to be consistent with previous research undertaken on this dataset [10]. Subjects that were recruited during an acute episode were assessed up to four times, whilst subjects who were clinically stable were interviewed once.

There is a significant class imbalance between the euthymic and acute mood states for both HDRS and YMRS scores with 75% and 71% of the datapoints having an euthymic mood state label respectively. Figure 3.1 shows the class imbalance.

3.4 Classification Subtasks

The classification of mood state can be broken down into 4 subtasks. Firstly, there are two different types of mood state (mania and depression) we are identifying. Secondly, patient data behaves very different when awake or asleep. For instance, heart rate variability is not available when the patient is awake and acceleration is near zero when patients are asleep. To manage these challenges data was segmented to identify if the patient was awake or asleep (sleep state 0 and 1 respectively) using the van hees algorithm [39]. As the data differed significantly, separate models were used for each sleep state. This gives a total of 4 subtasks, YMRS classification with sleep state 0, YMRS classification with sleep state 1, HDRS classification with sleep state 0, and finally HDRS classification with sleep state 1.

It should be noted that a third possible patient state occured when the health device was off body (sleep state 2). However, there were no instances of this occuring in our dataset.

3.5 Segmentation

Recording sessions were split into non-overlapping segments with segment length a variable parameter. Selecting segment length was a trade-off between maximising segment length and maximising the number of datapoints available for models to learn on. Starting segment length was set to 256 (seconds) as a reasonable midpoint. This generated a total of 104,931 datapoints between all 4 subtasks where segmentation is carried out on each sleep state independently.

3.6 Core features

The approach taken to solve the task of identifying mood state using wearable health data begins by generating features from the time series data. The core features used in these experiments were generated from the FLIRT library [16]. This python package is capable of taking time series data such as IBI, HRV and ACC and generating 'meaningful features' which can be passed to machine learning models. The focus of FLIRT is on processing and feature generation of data from wearable health devices such as smart watches rather than medical-grade data recording devices. A total of 184 features were generated for each data segment from the FLIRT library. Table 3.1 shows a handful of the features generated from the FLIRT library with the full list shown in appendix A.1. The vast majority of these features are summary statistics of the time series data with much of the functionality acting as a wrapper function around existing numpy or scipy functions.

Feature Name	Description
acc_x_mean	average acceleration in the x axis
acc_y_pct_95	95th percentile of the y axis acceleration
TEMP_min	minimum skin temperature
hrv_peaks	number of peaks identified in heart rate variability

Table 3.1: Example FLIRT features and their description.

3.7 Baseline Data Cleaning

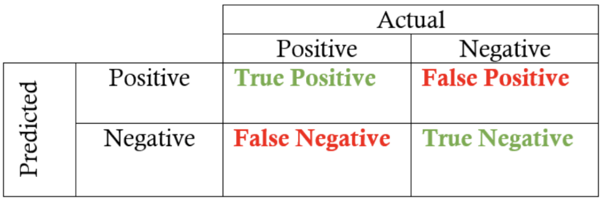
Many of the models used within this project required removing or imputing invalid values within the data. Many of the FLIRT outputs generate non-numeric values due to missing data or invalid transformation (e.g. taking logarithm of negative number). To overcome this issue, a data cleaning function was created to remove all columns with more than 20% null values. Any remaining rows with null values would then be removed. The column null threshold of 20% was selected after experimenting with the threshold to maximise the number of datapoints kept whilst retaining sufficient columns.

3.8 Task Evaluation

3.8.1 Model Evaluation

To evaluate the performance of our models, data for each of the 4 subtasks were randomly split into train, validation and test sets according to a 70%, 15%, 15% split. The random split and given ratios were chosen to ensure consistency with previous research undertaken on this dataset [10].

The training set is used for the model to learn the relationship between the input features and the labels. Model performance is evaluated on the validation set where inputs are passed to the model but labels withheld. The predictions made by the model on the validation set are then compared to the true values to evaluate its performance. The test set is used at the very end when models have been developed to evaluate the final generalisation performance of the model. This methodology is used to prevent “sub-conscious” overfitting to the test set [7]. This phenomenon occurs when the model



		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 3.2: Figure to explain the relationship between predicted and true values. Original image from [12].

developer frequently checks model performance on the test and can lead to the developer subconsciously choosing model design decisions and parameter selection that best fits the test set. This can give an overestimate of the generalisation performance and mislead future expected performance.

3.8.2 Evaluation Metrics

When evaluating the performance of models on each data subset we can use a multifaceted approach where multiple metrics and visualisations are used to gain a more holistic understanding of model performance. As binary classification is the key task, we can define many of our metrics through the number of true positives, false positives, true negatives, and false negatives generated for each model. See Figure 3.2 for definitions of each of these terms. To summarise aspects of the model performance, metrics including accuracy, precision, recall, and f1 score are used. See equation 3.1 through 3.4 for their formal definitions. Confusion matrices are an additional method used in this project to visualise the number of true positives, false positives, true negatives, and false negatives predicted from a model.

$$precision = \frac{TP}{TP+FP} \tag{3.1}$$

$$recall = \frac{TP}{TP+FN} \tag{3.2}$$

$$f1\ Score = \frac{2 \times precision \times recall}{precision + recall} \tag{3.3}$$

$$accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{3.4}$$

$$Beta\ Score = (1+\beta^2)\frac{precision \times recall}{(\beta^2 \times precision) + recall} \tag{3.5}$$

The key metric used when evaluating the model performance will be f1 score (see equation 3.3). f1 score is a metric where precision and recall are weighted equally and combined into a single metric. Whilst this is a reasonable metric to use in an academic setting, in real world environments beta score (see equation 3.5) may be more likely to be used as this metric has a variable parameter to weight the importance of precision against recall. For example, it may be more important to reduce false negatives in a real world environment and this has a much less severe consequence than a false positive as the first may result in a patient not receiving the care they need, whilst the latter would only result in an unnecessary specialist appointment.

3.9 Baseline Models

To generate a performance benchmark and put future performance developments into context, 2 baseline models were implemented. These are the naïve mode baseline and naïve binomial model.

The purpose of the two naïve baseline models is to show expected performance from a "random guess" model. Establishing these baselines allows to check two criteria. First, that our models are "learning" from our data and not just making a "random guess". Second, we have established a lowerbound for our expected performance of our models. For machine learning models to be applicable in a real world environment they must be better than simple statistical models such as the two naïve baseline models.

The first naïve baseline model (naïve mode model) simply predicts the class most seen in the training set. The model only predicts the negative class and from the definitions, precision, recall and f1 score will be zero. However, as we have a class imbalance the model will maximise its accuracy score given it is a random guess. When evaluating future models we can use the naïve mode model as a baseline reference when evaluating model accuracy.

The second naïve baseline model (naïve binomial model) samples from a binomial distribution using the counts of euthymic and acute mood states to calculate sample probabilities. Given the model is a "random guess" it maximises its precision, recall, and f1 score but has a worse accuracy compared to the naïve mode model. When evaluating future models we can use the naïve binomial model as a baseline reference when evaluating model precision, recall and f1 score.

3.10 Machine Learning Models

Once baseline models were implemented and appropriate evaluation methods are put in place the next stage of experimentation can be undertaken. Machine learning models can now be trained and evaluated.

When considering machine learning models there is typically a trade off between interpretability and performance. Model interpretability refers to how transparent the inner workings of a model are. Combining model interpretability with explanations of model decision in human terms creates explainability. Typically, the higher performing a model is the less interpretable it is.

Logistic regression and decision tree classifier models have been selected as the first two machine learning models to be implemented on the FLIRT processed data. These models were selected due to their natural explainability.

Logistic regression is a naturally interpretable model as all inputs are linearly weighted and passed through a sigmoid function to generate the output. Equation 3.6 defines the sigmoid function where x is the sum of the linearly weighted inputs. We can therefore view the model prediction and understand which of the inputs contributed most to the output. This natural interpretability can also help us to further develop our models by understanding which features are most valuable to predict the mood state.

$$S(x) = \frac{1}{1+e^{-x}} \tag{3.6}$$

The decision tree classifier was also selected as one of the first machine learning models as its output can be clearly visualized and also understood by non-technical individuals [6]. We can also use feature importance functionality to again see which inputs are most important to predict the outputs, which also aids the development of future models.

K Nearest Neighbours was selected as a midpoint between interpretability and performance. The algorithm uses proximity to other datapoints to make predictions. If we want to explain why a decision was made we can refer to the datapoints nearby that influenced the prediction. This is useful if we have a good way to explain a single point, which we do, as a core focus of this project is to generate interpretable features [15].

The final model selected was the XGBoost Classifier [9]. This is an ensemble technique where multiple decision trees are trained sequentially, with data points weighted to account for previous decision tree errors. XGBoost was selected as it has shown to be very high performing in real world datasets [5]. XGBoost is not as explainable as the three other models and provides a good comparison when evaluating the performance/explainability trade off.

All of the machine learning models will be run with default hyperparameters to reduce variability and establish the first machine models on the FLIRT processed data. The default hyperparmeters are those set by sklearn [13].

3.11 Steps to Improve Model Performance

A key point in this project was the evaluation after the first machine learning models were implemented. After this point, the machine learning models performance was evaluated by comparing to baseline models and relevant results seen in literature. The following steps were then identified and implemented with an aim to improve model performance whilst retaining model explainability.

3.11.1 Hyperparameter Tuning

Typically machine learning models use a hyperparameter tuning step where models are training multiple times with a range of model parameters. By selecting appropriate hyperparameters, models typically generate a better loss function score and generalise better [20].

The approach taken in this project was hyperparameter tuning via gaussian process optimization. To complete hyperparameter tuning via gaussian process, possible hyperparameter ranges are defined by the user along with a scoring metric (f1 score). This method then trains a model, evaluates its performance and then uses past evaluations to guide the next hyperparameter choice, providing a much more efficient

Metric Name	Feature Derivation
Patient Motion	Average, Standard Deviation, and Fraction of Time in Motion
Number of SCR peaks	N/A
Heart Rate	Mean, Min, Max, Standard Deviation, Coefficient of Variance

Table 3.2: Summary of new metrics and features generated. Multiple features can be generated by calculating statistics from metrics. e.g. standard deviation of heart rate

search per iteration compared to grid search or random search. During the project a maximum of 20 iterations were chosen for the gaussian process optimization due to computation restrictions.

3.11.2 Additional Feature Engineering

Additional features were generated with the objective of improving model performance by capturing the most relevant information from the data. This was done by reviewing previous literature that had used consumer-grade wearable health devices to predict mood state classification from HDRS and YMRS questionnaires. A total of 22 features were considered with potential features ranked based on their efficacy in previous works and a qualitative review of how similar these features are to existing features used in baseline models. 3 new metrics were introduced (patient motion, number of SCR peaks, and heart rate) and a total of 9 features were derived from these metrics. The final features developed can be seen in Table 3.2.

3.11.3 Imputing Missing Data

The baseline data cleaning process (as described in section 3.7) uses a process of dropping columns and rows to remove non-numeric values which the model can't process. This causes a loss of data and a potential loss of performance. To overcome this, we can impute the missing data to maximise the amount of data available to train models. A basic approach to this would to be impute column mean, or mode if the variable is categorical. However a more advanced technique uses an iterative imputer, as was applied in this project [38]. The iterative imputer treats missing values as a function of other features. For example, to impute values for a given column, a regressor is fitted using all other columns as inputs, and the labels are the non-missing rows for our given column. This trained model then can be used to impute the missing values.

This process was applied to impute missing data with a Bayesian Ridge model used as the regressor [36]. This model was selected as it was computationally cheap compared to other probabilistic models such as a gaussian process. It should be noted that columns with more than 20% nulls were still dropped otherwise there was not sufficient data to learn an appropriate imputation.

3.11.4 Feature Transformations

Exploratory data analysis was carried out to help understand the underlying data and aid model development. One of the key aspects of the exploratory data analysis was visualizing the distribution of the input variables. This was achieved through using a kernel density estimate function which plots the estimated probability density function for a given input and was applied to all features This visualization showed that a number of features had very high skewness with many of the features having long right-hand tails. This high skewness can make modelling the data more difficult (particularly for the logistic regression model). To overcome these issues all input features with a skewness metric (see Equation 3.7) greater than 10 and where all input values are positive had a log transformation applied. These criteria were set to transform high skew inputs and ensure that the transformation generated numeric outputs (as taking logs of negative numbers is invalid). The skewness threshold of 10 was selected empirically after visualising the distribution of features after their log transformation.

$$Skewness = \frac{1}{N}\sum_{i=1}^{N}\frac{(x_i - \bar{x})^3}{\sigma^3} \tag{3.7}$$

Figure 3.3 shows two example changes in probability distribution function once features were log transformed.

3.11.5 Removing Redundant Variables

A total of 193 features have been generated for machine learning models. Many of these will be considered to have poor correlation to the target variable and generate

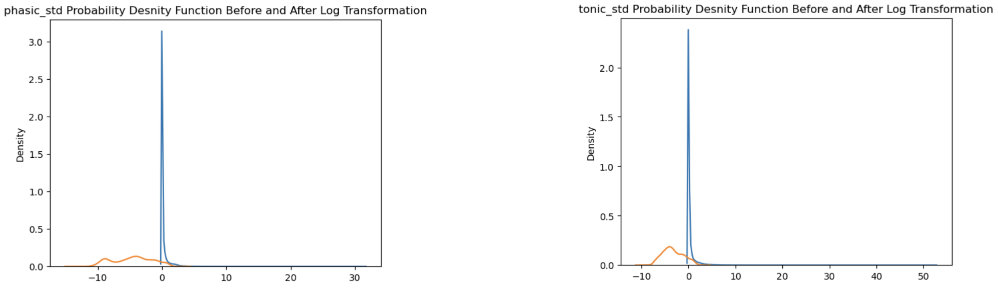


Figure 3.3: Probability density functions of features before and after a log transformation (blue and orange respectively)

more ‘noise’ to the model rather than ‘signal’. By removing redundant variables that do not benefit the model there is potential to improve model performance. A number of approaches were considered including Principal Component Analysis (PCA) and Kernel Principal Component Analysis (KPCA) [28][33]. PCA and KPCA were not selected as the output features from these processes are not as interpretable as the original features which is in conflict with our objective of generating explainable machine learning models. Instead, Boruta was selected as the feature selection method [22]. Boruta is described an “all relevant” feature selection method that tries to find all features carrying useable information. Boruta works by copying and shuffling the columns of the dataset, training random forest classifiers, and then using feature importance scores, and z scores to find the important features. This process is done iteratively to robustly find the most important feature. Boruta has also shown to be highly effective in real world data problems which drove most of the motivation for this selection [21]. Boruta was applied to each of the 4 subtasks to ensure only the most relevant features to be used for each task.

3.11.6 Class Imbalance

As shown in section 3.3, there is a strong class imbalance between segment labels. This can cause bias toward to the majority class which is unfavourable. There are a number of approaches to reduce this problem including undersampling, oversampling, and class weighting. Undersampling of the majority class was not considered as it causes loss of data. Typical oversampling can cause overfitting as we are duplicating datapoints. Class weighting can also cause overfitting as well as create model instability if used with a stochastic solver as the optimizer can sometimes struggles to converge. The chosen approach was Synthetic Minority Oversampling Technique (SMOTE) which is a form of oversampling [8]. However, it attempts to lessen the impact of overfitting seen in normal oversampling by generating plausible new minority class datapoints.

SMOTE was applied to training sets to help the model manage the class imbalance while training. However, SMOTE was not applied to the validation or test set and this would distort the dataset used for evaluation and would not give an accurate measure of generalisation performance.

3.12 Final Model Evaluation & Shapley Values

Due to computational limitations the final model evaluation can not be applied to all models and will be restricted to the two best performing models. Final results are generated by evaluating the two best models on the test set. First, the two final models will be trained using the default method but with 50 iterations of the gaussian process hyperparmeter step. This will allow more of the parameter search space to be evaluated, potentially finding a better performing model. Second, the best set of hyperparameters will be used to train the model on a training and validation set combination to maximise the data available. Model evaluation will then take place on the test set.

As an additional measure, SHAP will be applied to each of the final models to improve their explainability [24]. The key visualisations used will be Force Plots. This was motivated by the benefits it would provide in a real world environment. Healthcare professionals would potentially gain trust in machine learning models if they can see what has driven individual predictions. From current knowledge this is the first time SHAP explanations at the datapoint level have been made for mania and depression mood state classification using data collected from wearable health devices.

Chapter 4

Results

To set benchmarks and establish proper context for future model development, first baseline results are introduced and discussed in context to the relevant literature. This allows for proper context to be set for further model development and robustly evaluate these developments.

As outlined in section 3.4 the original mood state classification task is divided into 4 sub tasks (YMRS with sleep state 0, YMRS with sleep state 1, HDRS with sleep state 0, HDRS with sleep state 1). Therefore, each modelling approach developed will be evaluated on each of the 4 tasks.

4.1 Comparative Models From Literature

The main comparative result from literature for the HDRS task is from 2020 [35]. This was selected due to the fact similar data channels were used to create input features for the model. It should be noted that the literature results are based on a different dataset and different data channels and so cannot be directly used as a target performance to improve on. Rather, the results are to build context to the results generated from this project. The paper presents mood state classification f1 score of 0.75 with an accuracy of 0.76.

Literature on binary classification of acute vs euthymic mania mood state is very limited and so direct comparison to results in literature cannot be made. The most relevant research score with an accuracy of 91.29% was generated when predicting mania (YMRS) mood state binary classification [2]. Precision, recall and f1 score were not detailed in this paper. It should also be noted that the publication used leave-one-out

Naive **Mode Model**	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
f1 Score	0.00	0.00	0.00	0.00	0.00
Accuracy	0.78	0.68	0.71	0.72	0.72

Table 4.1: Naive mode baseline model performance across each of the subtasks.

Naive **Binomial Model**	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
f1 Score	0.28	0.28	0.29	0.29	0.27
Accuracy	0.65	0.57	0.59	0.60	0.60

Table 4.2: Naive binomial baseline model performance across each of the subtasks.

validation which will also generate a higher score than the train/validation/test split used in this project as leave-one-out provides more data for the model to train on.

The results of both the mode and binomial naïve baseline models can be seen in Table 4.1 and 4.2 respectively. For all tasks, the naïve mode baseline model has an f1 score of zero which is expected as the model only predicts the negative class. The accuracy across all 4 subtasks initially appears quite high, but this is a reflection the class imbalance within the data.

The naïve binomial model demonstrates baseline f1 scores across all the 4 subtasks. The model generates similar f1 scores of around 0.28 with the exception of HDRS with sleep state 0, where performance significantly drops. This drop in performance is likely due to the stronger class imbalance within this specific subtask. Other subtasks have a majority class $72\% \pm 2\%$ while HDRS sleep state zero has a majority class 77% of the time.

4.2 Machine Learning Models with Default Hyperparameters

The predictions made by the linear regression model are identical to the naïve mode baseline model. This suggests the linear regression model is severely underfitting and is unable to learn the underlying trend of the data as it has only predicted the “0” class for

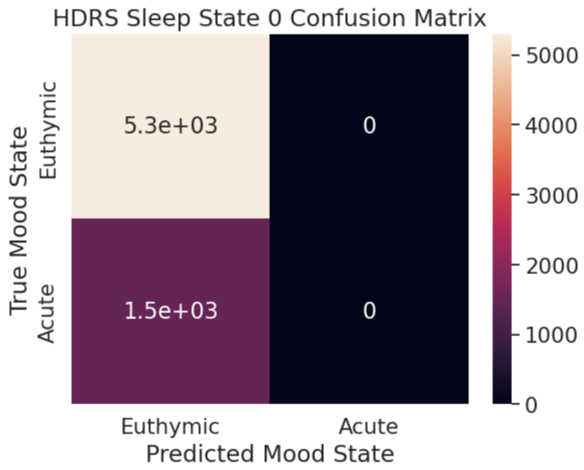


Figure 4.1: Confusion matrix comparing the logistic regression model predictions to their true values.

all datapoints in the validation set. Figure 4.1 shows the confusion matrix for HDRS sleep state 0, with similar results seen across all subtasks. This result also suggests that there are non-linear relationships between the input and the target variables, and that more complex models may perform better.

The decision tree model with default parameters performed much better than linear regression, with f1 score outperforming the naïve binomial model, suggesting that the decision tree model is capable of learning the underlying trend of the data. Although the decision tree performance is above the baseline models, it is still far below what has previously seen in literature.

Both the XGBoost and KNN models are an improvement over the naïve baseline models with f1 scores that exceed these models in all tasks. This suggests that the models are somewhat capable of learning the underlying trend of our data and could be viable models if further performance gains are found using other techniques used in this project. Although the KNN results are a 0.09 improvement over the naïve baseline f1 score, in a clinical setting, the performance gain may not be significant enough to justify its use over simple statistical models such as the naïve baseline models.

4.3 Hyperparameter Tuning

When applying hyperparameter tuning, all models except logistic regression improve their performance on all tasks (Tables 4.5 and 4.6) compared to their default

f1 Score	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.00	0.00	0.00	0.00	0.00
Decision Tree	0.42	0.41	0.44	0.37	0.41
KNN	0.35	0.35	0.39	0.35	0.36
XGBoost Classifier	0.40	0.44	0.43	0.38	0.41

Table 4.3: Model f1 score of the 4 machine learning models using default hyperparameters.

Accuracy	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.78	0.71	0.71	0.74	0.74
Decision Tree	0.73	0.66	0.67	0.67	0.68
KNN	0.78	0.71	0.71	0.73	0.73
XGBoost Classifier	0.81	0.77	0.76	0.78	0.78

Table 4.4: Model accuracy of the 4 machine learning models using default hyperparameters.

f1 Score	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.00	0.00	0.00	0.00	0.00
Decision Tree	0.42	0.43	0.44	0.41	0.43
KNN	0.41	0.42	0.44	0.43	0.43
XGBoost Classifier	0.40	0.44	0.43	0.40	0.42

Table 4.5: Model f1 score of the 4 machine learning models using tuned hyperparameters.

Accuracy	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.77	0.71	0.71	0.74	0.73
Decision Tree	0.36	0.41	0.41	0.41	0.39
KNN	0.70	0.66	0.65	0.68	0.67
XGBoost Classifier	0.76	0.71	0.70	0.73	0.73

Table 4.6: Model accuracy of the 4 machine learning models using tuned hyperparameters.

hyperparameter counterparts (Tables 4.3 and 4.4). This is expected as the hyperparameters allow the models to fit the training better. The logistic regression model did not improve compared to its non-tuned counterpart, which supports the original evaluation that the model does not have sufficient complexity to model the task.

The model with greatest improvement was KNN with an average f1 score of 0.08 above the default counterpart. This is likely due to the ”weights” hyperparameter selected to be ”distance” for all tasks. This hyperparameter allows for the k nearest neighbours to be inversely weighted according to their distance.

f1 Score	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0	0	0	0	0
Decision Tree	0.40	0.46	0.44	0.41	0.43
KNN	0.40	0.42	0.44	0.40	0.42
XGBoost Classifier	0.48	0.51	0.51	0.47	0.49

Table 4.7: Model f1 score of the 4 machine learning models using tuned hyperparameters and additional features.

Accuracy	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.78	0.72	0.71	0.75	0.74
Decision Tree	0.74	0.69	0.67	0.70	0.70
KNN	0.78	0.71	0.71	0.73	0.73
XGBoost Classifier	0.81	0.75	0.75	0.75	0.77

Table 4.8: Model accuracy of the 4 machine learning models using tuned hyperparameters and additional features.

4.4 Additional Columns

All experiments up to this point have used the core features generated from FLIRT which were a total of 184 features. An additional 9 features were generated by comparing high importance features from relevant literature and the core FLIRT features. For a more detailed motivation and methodology see section 3.11.2.

The most significant change comes from the XGBoost classifiers where on average the model f1 improves by 0.08. This f1 score improvement demonstrates that the additional features provide useful information to the model (Tables 4.7 and 4.8). A strong improvement from the decision tree model is also observed with an average f1 score improvement of 0.07 across all tasks.

f1 Score	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.00	0.00	0.00	0.04	0.01
Decision Tree	0.41	0.43	0.42	0.44	0.43
KNN	0.39	0.41	0.41	0.42	0.41
XGBoost Classifier	0.46	0.49	0.49	0.48	0.48

Table 4.9: Model f1 score of the 4 machine learning models using tuned hyperparameters, additional features, and imputing missing data.

KNN, the logistic regression model and the decision tree have a near zero change in model f1 score. This suggests that the new features do not simplify the tasks, rather provide additional information to the models. A possible intuition to these performance changes are that the features provide more information to the models but at the cost of noise.

4.5 Imputing Missing Data

The original hypothesis around imputing null values was that it would cause less rows to be dropped therefore giving more data to the model to learn on. However, it is almost always the case that model performance dropped across all tasks (Tables 4.9 and 4.10). This is likely because although the iterative imputer is an advanced approach to impute missing data, we have used a linear model to impute the data and as shown by the logistic regression model the relationship between variables is likely to be non-linear. We therefore have likely imputed unrealistic values and added noise to the data. This hypothesis is supported by the results as all model performances have either a near equivalent or reduced performance with the imputation of missing data.

It should be noted that the logistic regression model has slightly improved in f1 score but dropped in accuracy. The change in performance is negligible and conclusions cannot be drawn with confidence.

Accuracy	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.77	0.72	0.71	0.72	0.73
Decision Tree	0.72	0.70	0.67	0.69	0.70
KNN	0.76	0.71	0.70	0.70	0.72
XGBoost Classifier	0.82	0.78	0.74	0.75	0.77

Table 4.10: Model accuracy of the 4 machine learning models using tuned hyperparameters, additional features, and imputing missing data.

4.6 Feature Transformations

Logistic regression was the model which was expected to benefit from this data transformation. By taking the log transform of highly skewed variables the hypothesis was that some of the non-linear relationships that need to be modelled would be reduced to linear relationships. However, the results showed very little/no improvement (Tables 4.11 and 4.12). The only task with improvement was the HDRS tasks with sleep state 1 (awake). The improvement is so small that similarly to previous steps, we cannot robustly make conclusions from the change seen in model performance.

All other models on average across the 4 tasks had a decrease in validation set performance. However, their change in performance is very small and not significant enough to clearly identify that it worsens the model performance, as there are a number of stochastic processes within the evaluation performance including the gaussian processes optimization of the model hyperparameters.

4.7 Dropped Columns

In this step, Boruta is applied to identify the columns that should be dropped. Boruta was applied individually to each of the 4 tasks which gives a total of 4 exclusion lists. Table 4.13 shows the columns dropped consistently across all 4 tasks and Table 4.14 shows the number of columns dropped for each tasks. Appendix A.2 contains the full list of features dropped.

f1 Score	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.00	0.02	0.00	0.04	0.02
Decision Tree	0.40	0.41	0.43	0.42	0.42
KNN	0.38	0.41	0.41	0.40	0.40
XGBoost Classifier	0.45	0.49	0.49	0.47	0.48

Table 4.11: Model f1 score of the 4 machine learning models using tuned hyperparameters, additional features, imputing missing data, and log transformation of highly skewed variables.

Accuracy	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.78	0.73	0.71	0.72	0.74
Decision Tree	0.72	0.70	0.67	0.68	0.69
KNN	0.76	0.71	0.70	0.70	0.72
XGBoost Classifier	0.79	0.77	0.77	0.75	0.77

Table 4.12: Model accuracy of the 4 machine learning models using tuned hyperparameters, additional features, imputing missing data, and log transformation of highly skewed variables.

Dropped Columns for All Tasks		
tonic_perm_entropy	tonic_skewness	acc_y_iqr_5_95
phasic_min	tonic_kurtosis	l2_n_sign_changes
phasic_n_sign_changes	tonic_peaks	BVP_cv
acc_x_n_above_mean	tonic_n_above_mean	BVP_cv
acc_x_n_below_mean	tonic_n_below_mean	
acc_y_n_above_mean	tonic_n_sign_changes	

Table 4.13: Names of the columns consistently dropped across all subtasks when using Boruta.

Number of Columns Dropped			
HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1
23	28	29	39

Table 4.14: The number of columns dropped for each subtask when using Boruta.

As can be seen in Table 4.15 and 4.16, with features dropped both the tree based models improve. The intuition behind this result is that by removing redundant columns, noise is also removed from the data, making the tasks easier for the models. However, the logistic regression and KNN approaches have no improvement in model performance.

4.8 Class Imbalance

SMOTE was applied to all models to manage the class imbalance seen in the training set. Using this technique along with all previous improvement steps generated the best performing model with XGBoost averaging an f1 score of 0.51 (Table 4.17) . This improvement also holds true when evaluating accuracy (Table 4.18). XGBoost averages an accuracy of 0.78 across the 4 tasks which is the best performing model.

KNN had a significant improvement and outperformed the decision tree. An average f1 score of 0.45 improves the model by 0.05 compared to the previous improvement step. This is the second best model seen (after XGBoost) and is a considerable improvement over the baseline models.

Logistic regression generated an average f1 score of 0.36 which appears to be a significant improvement. However, this change is not an increase in performance but rather a change from predicting a single class (similar to the naive model model) to now randomly selecting a class (similar to our binomial model). When comparing the logistic regression to our naive baseline models we can see there has been no significant change in performance.

The decision tree model performance dropped in comparison to previous improvement steps. This may be due to noise generated from the additional datapoints from SMOTE. Overall the best performing decision tree model was when additional columns and hyperparameter tuning was applied.

f1 Score	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.00	0.02	0.00	0.04	0.02
Decision Tree	0.41	0.43	0.45	0.43	0.43
KNN	0.38	0.41	0.41	0.40	0.40
XGBoost Classifier	0.48	0.51	0.52	0.50	0.50

Table 4.15: Model f1 score of the 4 machine learning models using tuned hyperparameters, additional features, imputing missing data, log transformations, and dropped columns.

Accuracy	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.77	0.73	0.71	0.72	0.73
Decision Tree	0.73	0.70	0.68	0.68	0.70
KNN	0.76	0.71	0.70	0.71	0.72
XGBoost Classifier	0.80	0.78	0.78	0.78	0.79

Table 4.16: Model accuracy of the 4 machine learning models using tuned hyperparameters, additional features, imputing missing data, log transformations, and dropped columns.

f1 Score	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.35	0.39	0.35	0.37	0.37
Decision Tree	0.37	0.38	0.40	0.36	0.38
KNN	0.42	0.45	0.46	0.46	0.45
XGBoost Classifier	0.48	0.51	0.52	0.53	0.51

Table 4.17: Model f1 score of the 4 machine learning models using tuned hyperparameters, additional features, imputing missing data, log transformations, dropped columns, and class balancing using SMOTE.

Accuracy	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
Logistic Regression	0.55	0.52	0.55	0.55	0.54
Decision Tree	0.70	0.50	0.48	0.54	0.56
KNN	0.68	0.61	0.62	0.61	0.63
XGBoost Classifier	0.78	0.74	0.75	0.78	0.76

Table 4.18: Model accuracy of the 4 machine learning models using tuned hyperparameters, additional features, imputing missing data, log transformations, dropped columns, and class balancing using SMOTE.

f1 Score	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
KNN	0.41	0.46	0.46	0.44	0.44
XGBoost Classifier	0.47	0.56	0.53	0.52	0.52

Table 4.19: Model f1 score of the two final machine learning models on the test set with all previous improvements steps applied. Additional training data was provided via the validation set and 50 iterations of the gaussian process hyperparameter optimization process was applied.

4.9 Final Results

Final results are generated by evaluating the final models on the test set (Tables 4.19 and 4.20). The two final models are the best performing models (KNN and XGBoost from section 4.8). See section 3.12 for a detailed methodology of how these final models were trained to maximise performance.

The best performing model is the XGBoost classifier with an average f1 score of 0.52. This is an improvement over any other previous score and could be expected due to the addition data used during training alongside extra tuning iterations. The mania classification performance is very consistent across sleep states. However the depression mood state performance significantly drops between sleep states, which has been consistent with all other results. The suspected intuition behind this is that in sleep state 0 there is near zero acceleration data to be used and literature had previously shown the importance of patient movement when identifying depression [4].

The XGBoost classifier has shown an f1 score twice that of the naive binomial model with many of the improvement steps contributing to this increase in performance. In comparison to the literature, this performance is still below state of the art performance. However as the dataset is different the results can not be directly compared.

KNN also generated the best performing model of its type with an average f1 score of 0.44. f1 scores across each subtasks followed similar trends to the XGBoost classifier where HDRS sleep state 0 was lower than all other subtasks.

Accuracy	HDRS Sleep State 0	HDRS Sleep State 1	YMRS Sleep State 0	YMRS Sleep State 1	Average
KNN	0.67	0.62	0.62	0.62	0.63
XGBoost Classifier	0.78	0.79	0.76	0.78	0.78

Table 4.20: Model accuracy of the two final machine learning models on the test set with all previous improvements steps applied. Additional training data was provided via the validation set and 50 iterations of the gaussian process hyperparameter optimization process was applied.

4.10 SHAP Visualisations

To aid explainability of the top 2 performing models, SHAP was applied at the individual patient prediction level. Force plots have been generated for the XGBoost and KNN models and can be seen in Figure 4.3 and 4.2.

Figure 4.2 shows an example force plot on the XGBoost model for patient 1424331 (patient number is anonymized) with the red and blue segments indicating the impact each feature had on the model. Firstly, this plot shows us that the sum of the contributing factors (labeled as f(x)) is below the base value and so will predict an euthymic mood state. We can also see that a large number of features are considered when generating the prediction. Each red segment represents a feature that provides a positive influence on the outcome (i.e. influences the predicted mood state to be acute) and the blue segments influence the prediction negatively. We can see the biggest influence on the outcome is from the feature ”TEMP_median”. This feature is interpretable to non-technical users and easily provides insight into what is driving the model output. Some of the more obscure feature can be clarified by a simple data dictionary with explanations of what the features refer to.

Figure 4.3 shows a force plot for patient 1656030 (patient number is anonymized) using the KNN model. We can immediately see that the model is much simpler with many fewer features influencing the outcome. The feature ”acc_y_energy” is the largest contributor to the outcome, and driving an euthymic state.

Applying force plots to explain model outcomes significantly improves model explainability. The force plot is just one of the visualisations available from SHAP,

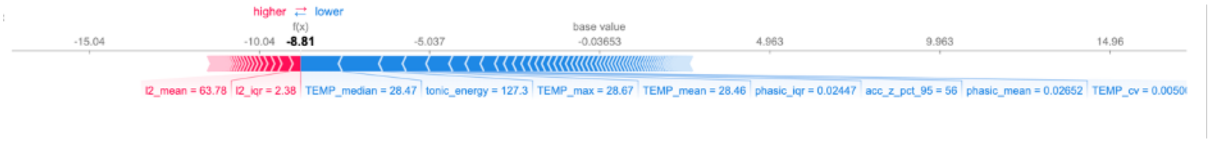


Figure 4.2: Example force plot generated using SHAP and an XGBoost Classifier.

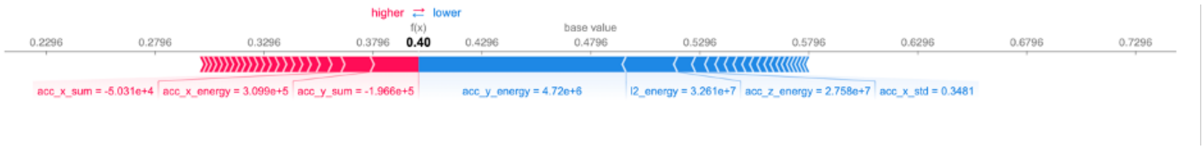


Figure 4.3: Example force plot generated using SHAP and an KNN Classifier.

with others including summary and dependence plots that summarise different aspects of the model. However, this is outside the scope of this project and should be considered in future work.

Chapter 5

Conclusions

The aim of this project was to answer the question: can mood state be accurately identified with explainable machine learning models using data from consumer-grade wearable health devices?

A clear methodology was outlined identifying the steps that needed to be undertaken to answer the key project question. During the implementation of this methodology, 4 types of machine learning models were trained using data collected from wearable health devices. The dataset was cleaned, processed, segmented and summarised into interpretable features. Model training and evaluation steps were put in place with appropriate evaluation metrics and throughout the project model explainability was kept as a key consideration.

The key results generated from this project include:

- 184 features extracted on the time series data using the FLIRT python package.
- Developed baseline models to benchmark model performances.
- Developed appropriate training and evaluation steps with suitable metrics.
- Developed 4 types of machine learning models to identify mania and depression mood state.
- Identified and implemented 6 steps to improve model performance whilst retaining model explainability.
- Generated SHAP force plots to explain how model inputs contributing to an individual prediction.

The most significant result from this project is the high performing XGBoost classifier that had good levels of model explainability. The XGBoost classifier scored an average f1 score of 0.52 across the 4 subtasks, which is twice that of the naïve binomial baseline model. From our knowledge this is the first time mood state classification has been undertaken with a focus on model explainability throughout. The results show that mood state can be accurately identified with machine learning models using data from consumer-grade wearable health devices.

Very good results were generated from this project, however there is still scope for future work in this area. Firstly, SHAP provides a number of visualisations to inspect different aspects of the model. Future work could continue investigating how to make models more explainable by leveraging this functionality. Secondly, many of the features used as model inputs were easily interpretable. However, some of the features are not as easily understandable (e.g. "I2_IQR"). Further work could be taken to make some of the more complex features more accessible to a non technical audience. Thirdly, the approach used to impute missing data did not improve model performance. Researching more effective ways to impute the missing data may have a positive impact on classification of mood state via machine learning.

Bibliography

[1] Daniel A. Adler, Fei Wang, David C. Mohr, and Tanzeem Choudhury. Machine learning for passive mental health symptom prediction: Generalization across different longitudinal mobile sensing studies. *PLOS ONE*, 17(4):e0266516, 2022.

[2] Ulysse Côté Allard, Petter Jakobsen, Andrea Stautland, Tine Nordgreen, Ole Bernt Fasmer, Ketil Joachim Oedegaard, and Jim Tørresen. Long-short ensemble network for bipolar manic-euthymic state recognition based on wrist-worn sensors. *CoRR*, 2021.

[3] DS American Psychiatric Association, American Psychiatric Association, et al. *Diagnostic and statistical manual of mental disorders: DSM-5*, volume 5. American psychiatric association Washington, DC, 2013.

[4] Ruth M. Benca. Sleep and psychiatric disorders. *Archives of General Psychiatry*, 49(8):651, August 1992.

[5] Candice Bentéjac, Anna Csörgo, and Gonzalo Martínez-Muñoz. A comparative analysis of xgboost. *CoRR*, 2019.

[6] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, 1984.

[7] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 1721–1730, New York, NY, USA, 2015. Association for Computing Machinery.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, 2016.

[10] Filippo Corponi, Bryan M. Li, Gerard Anmella, Ariadna Mas, Miriam Sanabra, Eduard Vieta, INTREPIBD Group, Stephen M. Lawrie, Heather C. Whalley, Diego Hidalgo-Mazzei, and Antonio Vergari. Automated mood disorder symptoms monitoring from multivariate time-series sensory data: Getting the full picture beyond a single number. *medrxiv*, 2023. preprint.

[11] Jamileh Shadid Damian F Santomauro, Ana M Mantilla Herrera. Global prevalence and burden of depressive and anxiety disorders in 204 countries and territories in 2020 due to the COVID-19 pandemic. *The Lancet*, 398(10312):1700–1712, 2021.

[12] Cem Dilmegani. Machine learning accuracy: True-false positive/negative [2023].

[13] Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] Lynne M Feehan, Jasmina Geldman, Eric C Sayre, Chance Park, Allison M Ezzat, Ju Young Yoo, Clayon B Hamilton, and Linda C Li. Accuracy of fitbit devices: Systematic review and narrative syntheses of quantitative data. *JMIR mHealth and uHealth*, 6(8):e10527, 2018.

[15] Evelyn Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):238, 1989.

[16] Simon Föll, Martin Maritsch, Federica Spinola, Varun Mishra, Filipe Barata, Tobias Kowatsch, Elgar Fleisch, and Felix Wortmann. FLIRT: A feature generation toolkit for wearable data. *Computer Methods and Programs in Biomedicine*, 212:106461, 2021.

[17] Asma Ghandeharioun, Szymon Fedor, Lisa Sangermano, Dawn Ionescu, Jonathan Alpert, Chelsea Dale, David Sontag, and Rosalind Picard. Objective assessment of depressive symptoms with machine learning and wearable sensors data. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 325–332, 2017.

[18] Iria Grande, Michael Berk, Boris Birmaher, and Eduard Vieta. Bipolar disorder. *The Lancet*, 387(10027):1561–1572, 2016.

[19] M. Hamilton. A RATING SCALE FOR DEPRESSION. *Journal of Neurology, Neurosurgery & amp Psychiatry*, 23(1):56–62, 1960.

[20] Kazi Ekramul Hoque and Hamoud Aljamaan. Impact of hyperparameter tuning on machine learning models in stock price forecasting. *IEEE Access*, 9:163815–163830, 2021.

[21] Smitha S Kumar and Talal Shaikh. Empirical evaluation of the performance of feature selection approaches on random forest. In *2017 International Conference on Computer and Applications (ICCA)*, pages 227–231, 2017.

[22] Miron B. Kursa and Witold R. Rudnicki. Feature selection with the boruta package. *Journal of Statistical Software*, 36(11):1–13, 2010.

[23] Stefan Lewis. Informatics project proposal, 2023.

[24] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[25] Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10):749, 2018.

[26] Theophano Mitsa. *Temporal Data Mining*. Chapman and Hall/CRC, 2010.

[27] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.

[28] Karl Pearson. LIII. ion lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[29] Paola Pedrelli, Szymon Fedor, Asma Ghandeharioun, Esther Howe, Dawn F. Ionescu, Darian Bhathena, Lauren B. Fisher, Cristina Cusin, Maren Nyer, Albert

Yeung, Lisa Sangermano, David Mischoulon, Johnathan E. Alpert, and Rosalind W. Picard. Monitoring changes in depression severity using wearable and mobile sensors. *Frontiers in Psychiatry*, 11, 2020.

[30] Anika Reichert and Rowena Jacobs. The impact of waiting time on patient outcomes: Evidence from early intervention in psychosis services in england. *Health Economics*, 27(11):1772–1787, 2018.

[31] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[32] Yuri Rykov, Thuan-Quoc Thach, Iva Bojic, George Christopoulos, and Josip Car. Digital biomarkers for depression screening with wearable devices: Cross-sectional study with machine learning modeling. *JMIR mHealth and uHealth*, 9(10):e24872, October 2021.

[33] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Lecture Notes in Computer Science*, pages 583–588. Springer Berlin Heidelberg, 1997.

[34] Rutvik Shah, Gillian Grennan, Mariam Zafar-Khan, Fahad Alim, Sujit Dey, Dhakshin Ramanathan, and Jyoti Mishra. Personalized machine learning of depressed mood using wearables. *Translational Psychiatry*, 11:338, 2021.

[35] Yuuki Tazawa, Kuo ching Liang, Michitaka Yoshimura, Momoko Kitazawa, Yuriko Kaise, Akihiro Takamiya, Aiko Kishi, Toshiro Horigome, Yasue Mitsukura, Masaru Mimura, and Taishiro Kishimoto. Evaluating depression with multimodal wristband-type wearable device: screening and assessing patient severity utilizing machine-learning. *Heliyon*, 6(2):e03274, February 2020.

[36] Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.

[37] Mark A Tully, Cairmeal McBride, Leonnie Heron, and Ruth F Hunter. The validation of fitbit zip™ physical activity monitor as a measure of free-living physical activity. *BMC Research Notes*, 7(1):952, 2014.

[38] Stef van Buuren and Karin Groothuis-Oudshoorn. bmice/b: Multivariate imputation by chained equations. *Journal of Statistical Software*, 45(3), 2011.

[39] Vincent Theodoor van Hees, S. Sabia, S. E. Jones, A. R. Wood, K. N. Anderson, M. Kivimäki, T. M. Frayling, A. I. Pack, M. Bucan, M. I. Trenell, Diego R. Mazzotti, P. R. Gehrman, B. A. Singh-Manoux, and M. N. Weedon. Estimating sleep parameters using an accelerometer without sleep diary. *Scientific Reports*, 8(1), 2018.

[40] T. K. Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics*, 4:52–57, 1968.

[41] Theo Vos, Christine Allen, and Megha Arrora et al. Asystematic analysis for the global burden of disease study 2015. *The Lancet*, 388(10053):1545–1602, 2016.

[42] R. C. Young, J. T. Biggs, V. E. Ziegler, and D. A. Meyer. A rating scale for mania: Reliability, validity and sensitivity. *British Journal of Psychiatry*, 133(5):429–435, November 1978.

[43] Yiming Zhang, Ying Weng, and Jonathan Lund. Applications of explainable artificial intelligence in diagnosis and surgery. *Diagnostics*, 12(2):237, 2022.

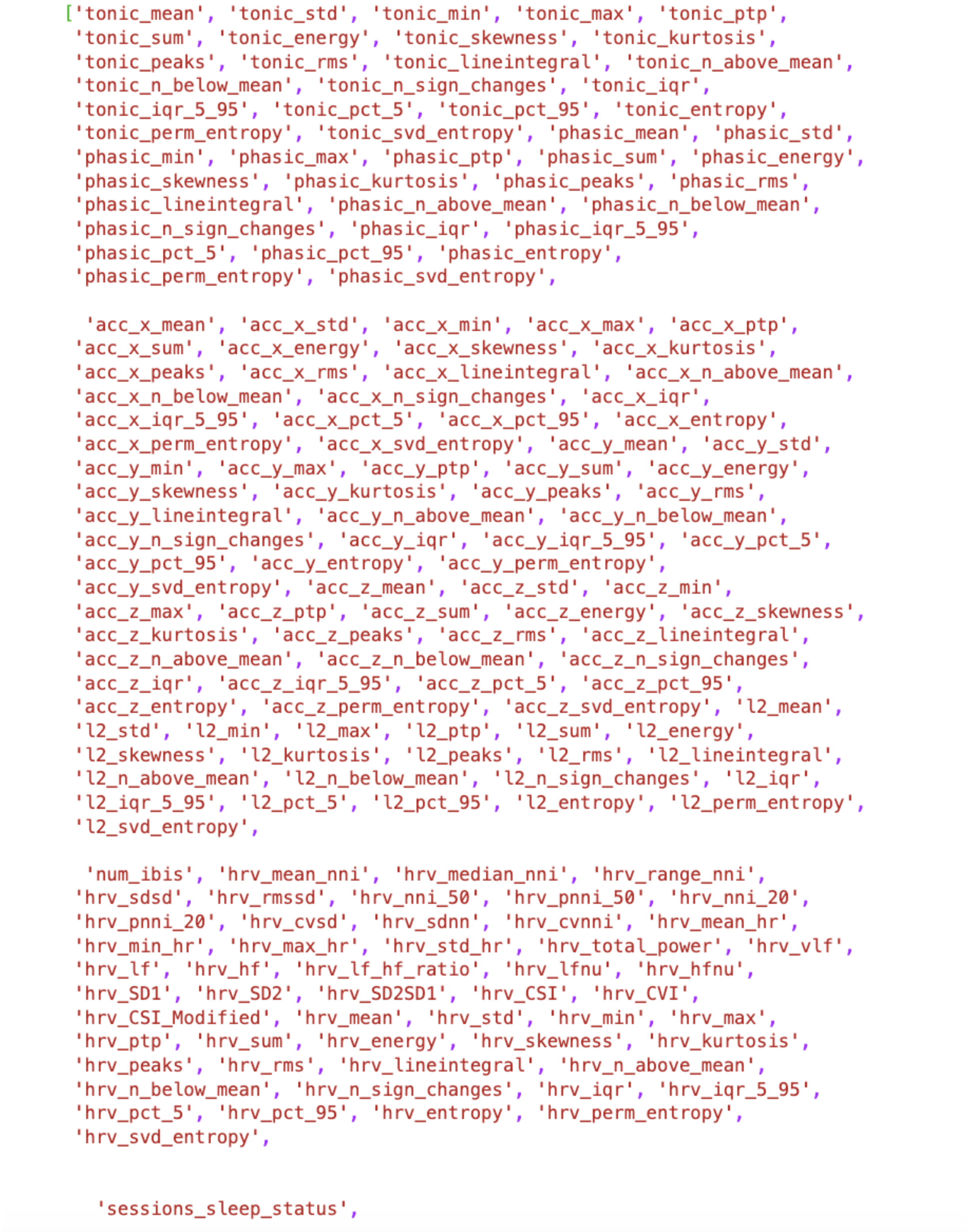
[44] Yu Zhang, Peter Tiño, Ales Leonardis, and Ke Tang. A survey on neural network interpretability. *CoRR*, 2020.

Appendix A

First appendix

A.1 Baseline Features

A.2 Column Dropped Using Boruta



```
['tonic_mean', 'tonic_std', 'tonic_min', 'tonic_max', 'tonic_ptp',
 'tonic_sum', 'tonic_energy', 'tonic_skewness', 'tonic_kurtosis',
 'tonic_peaks', 'tonic_rms', 'tonic_lineintegral', 'tonic_n_above_mean',
 'tonic_n_below_mean', 'tonic_n_sign_changes', 'tonic_iqr',
 'tonic_iqr_5_95', 'tonic_pct_5', 'tonic_pct_95', 'tonic_entropy',
 'tonic_perm_entropy', 'tonic_svd_entropy', 'phasic_mean', 'phasic_std',
 'phasic_min', 'phasic_max', 'phasic_ptp', 'phasic_sum', 'phasic_energy',
 'phasic_skewness', 'phasic_kurtosis', 'phasic_peaks', 'phasic_rms',
 'phasic_lineintegral', 'phasic_n_above_mean', 'phasic_n_below_mean',
 'phasic_n_sign_changes', 'phasic_iqr', 'phasic_iqr_5_95',
 'phasic_pct_5', 'phasic_pct_95', 'phasic_entropy',
 'phasic_perm_entropy', 'phasic_svd_entropy',

  'acc_x_mean', 'acc_x_std', 'acc_x_min', 'acc_x_max', 'acc_x_ptp',
 'acc_x_sum', 'acc_x_energy', 'acc_x_skewness', 'acc_x_kurtosis',
 'acc_x_peaks', 'acc_x_rms', 'acc_x_lineintegral', 'acc_x_n_above_mean',
 'acc_x_n_below_mean', 'acc_x_n_sign_changes', 'acc_x_iqr',
 'acc_x_iqr_5_95', 'acc_x_pct_5', 'acc_x_pct_95', 'acc_x_entropy',
 'acc_x_perm_entropy', 'acc_x_svd_entropy', 'acc_y_mean', 'acc_y_std',
 'acc_y_min', 'acc_y_max', 'acc_y_ptp', 'acc_y_sum', 'acc_y_energy',
 'acc_y_skewness', 'acc_y_kurtosis', 'acc_y_peaks', 'acc_y_rms',
 'acc_y_lineintegral', 'acc_y_n_above_mean', 'acc_y_n_below_mean',
 'acc_y_n_sign_changes', 'acc_y_iqr', 'acc_y_iqr_5_95', 'acc_y_pct_5',
 'acc_y_pct_95', 'acc_y_entropy', 'acc_y_perm_entropy',
 'acc_y_svd_entropy', 'acc_z_mean', 'acc_z_std', 'acc_z_min',
 'acc_z_max', 'acc_z_ptp', 'acc_z_sum', 'acc_z_energy', 'acc_z_skewness',
 'acc_z_kurtosis', 'acc_z_peaks', 'acc_z_rms', 'acc_z_lineintegral',
 'acc_z_n_above_mean', 'acc_z_n_below_mean', 'acc_z_n_sign_changes',
 'acc_z_iqr', 'acc_z_iqr_5_95', 'acc_z_pct_5', 'acc_z_pct_95',
 'acc_z_entropy', 'acc_z_perm_entropy', 'acc_z_svd_entropy', 'l2_mean',
 'l2_std', 'l2_min', 'l2_max', 'l2_ptp', 'l2_sum', 'l2_energy',
 'l2_skewness', 'l2_kurtosis', 'l2_peaks', 'l2_rms', 'l2_lineintegral',
 'l2_n_above_mean', 'l2_n_below_mean', 'l2_n_sign_changes', 'l2_iqr',
 'l2_iqr_5_95', 'l2_pct_5', 'l2_pct_95', 'l2_entropy', 'l2_perm_entropy',
 'l2_svd_entropy',

  'num_ibis', 'hrv_mean_nni', 'hrv_median_nni', 'hrv_range_nni',
 'hrv_sdsd', 'hrv_rmssd', 'hrv_nni_50', 'hrv_pnni_50', 'hrv_nni_20',
 'hrv_pnni_20', 'hrv_cvsd', 'hrv_sdnn', 'hrv_cvnni', 'hrv_mean_hr',
 'hrv_min_hr', 'hrv_max_hr', 'hrv_std_hr', 'hrv_total_power', 'hrv_vlf',
 'hrv_lf', 'hrv_hf', 'hrv_lf_hf_ratio', 'hrv_lfnu', 'hrv_hfnu',
 'hrv_SD1', 'hrv_SD2', 'hrv_SD2SD1', 'hrv_CSI', 'hrv_CVI',
 'hrv_CSI_Modified', 'hrv_mean', 'hrv_std', 'hrv_min', 'hrv_max',
 'hrv_ptp', 'hrv_sum', 'hrv_energy', 'hrv_skewness', 'hrv_kurtosis',
 'hrv_peaks', 'hrv_rms', 'hrv_lineintegral', 'hrv_n_above_mean',
 'hrv_n_below_mean', 'hrv_n_sign_changes', 'hrv_iqr', 'hrv_iqr_5_95',
 'hrv_pct_5', 'hrv_pct_95', 'hrv_entropy', 'hrv_perm_entropy',
 'hrv_svd_entropy',


   'sessions_sleep_status',
```

Figure A.1: List of all features used in baseline models. For full description and definitions see FLIRT python package [16].

Column Names Dropped	
HDRS Sleep State 0	HDRS Sleep State 1
tonic_skewness	tonic_skewness
tonic_kurtosis	tonic_kurtosis
tonic_peaks	tonic_peaks
tonic_n_above_mean	tonic_n_above_mean
tonic_n_below_mean	tonic_n_below_mean
tonic_n_sign_changes	tonic_n_sign_changes
tonic_perm_entropy	tonic_perm_entropy
phasic_min	phasic_min
phasic_skewness	phasic_peaks
phasic_n_sign_changes	phasic_n_above_mean
acc_x_n_above_mean	phasic_n_sign_changes
acc_x_n_below_mean	acc_x_skewness
acc_x_iqr	acc_x_kurtosis
acc_y_ptp	acc_x_n_above_mean
acc_y_n_above_mean	acc_x_n_below_mean
acc_y_n_below_mean	acc_y_std
acc_y_iqr	acc_y_ptp
acc_y_iqr_5_95	acc_y_skewness
acc_z_n_sign_changes	acc_y_kurtosis
acc_z_iqr	acc_y_n_above_mean
l2_n_sign_changes	acc_y_n_below_mean
BVP_mean	acc_y_iqr
BVP_cv	acc_y_iqr_5_95
	acc_z_n_sign_changes
	acc_z_iqr
	l2_peaks
	l2_n_sign_changes
	BVP_cv

Table A.1: Features dropped using the Boruta package for depression mood state tasks.

Column Names Dropped	
YMRS Sleep State 0	YMRS Sleep State 1
tonic_skewness	tonic_skewness
tonic_kurtosis	tonic_kurtosis
tonic_peaks	tonic_peaks
tonic_n_above_mean	tonic_n_above_mean
tonic_n_below_mean	tonic_n_below_mean
tonic_n_sign_changes	tonic_n_sign_changes
tonic_perm_entropy	tonic_perm_entropy
phasic_min	phasic_min
phasic_skewness	phasic_skewness
phasic_peaks	phasic_peaks
phasic_n_above_mean	phasic_n_above_mean
phasic_n_below_mean	phasic_n_sign_changes
phasic_n_sign_changes	phasic_perm_entropy
acc_x_skewness	acc_x_ptp
acc_x_kurtosis	acc_x_skewness
acc_x_n_above_mean	acc_x_kurtosis
acc_x_n_below_mean	acc_x_n_above_mean
acc_x_n_sign_changes	acc_x_n_below_mean
acc_x_iqr	acc_x_n_sign_changes
acc_y_skewness	acc_x_iqr
acc_y_n_above_mean	acc_x_iqr_5_95
acc_y_n_below_mean	acc_y_std
acc_y_iqr_5_95	acc_y_skewness
acc_z_skewness	acc_y_kurtosis
acc_z_n_sign_changes	acc_y_n_above_mean
l2_n_above_mean	acc_y_n_below_mean
l2_n_below_mean	acc_y_iqr
l2_n_sign_changes	acc_y_iqr_5_95
BVP_cv	acc_z_skewness
	acc_z_n_above_mean
	acc_z_n_below_mean
	acc_z_iqr
	l2_peaks
	l2_n_above_mean
	l2_n_below_mean
	l2_n_sign_changes

Table A.2: Features dropped using the Boruta package for mania mood state tasks.