# Using motor activity recordings to infer mental disorders

1st David Sasu
*IT University of Copenhagen*
Copenhagen, Denmark

2nd Erik Konstenius
*IT University of Copenhagen*
Copenhagen, Denmark

3rd Anders Weile Larsen
*IT University of Copenhagen*
Copenhagen, Denmark

***Abstract*—The aim of this paper was to investigate machine learning models' ability to detect attention deficit hyperactivity disorder (ADHD), depression and schizophrenia using motor activity recordings. A random forest classifier that was trained on an upsampled dataset achieved the best performance with a weighted F1 score of 0.4241. However, the results also indicated that it is challenging to diagnose patients based solely on motor activity recordings and that larger datasets are needed before machine learning models can be reliably used in this field. This paper also asserts that to further enhance the trustworthiness and usefulness of these models in clinical practice, data scientists should incorporate explainability into the overall assessment process of the developed models.**

***Index Terms*—motor activity recordings, mental disorders, machine learning, deep learning**

## I. Introduction, motivations and novel research contributions

The diagnosis of mental disorders is a complex and multifaceted process that involves the integration of multiple sources of information, including a person's symptoms, medical history, and behaviour. Diagnostication is typically carried out by healthcare providers, such as psychiatrists and psychologists, who rely on their specialized training and expertise to evaluate a person's mental health status. Currently, motor activity recordings are being explored as a potentially valuable feature for developing more objective measures of mental health disorders. This is largely due to the fact that motor activity recordings are richly imbued with representations of physical states at different points in time and are also relatively easy to obtain through the use of portable devices such as smartwatches and mobile phones.

There has been a growing interest in the scientific literature to apply machine learning to aid in the mental disorder diagnostic process through the analysis of motor activity data. However, most of these efforts have largely focused on two-way classification tasks where researchers have tried to distinguish between a patient group and a control group. Although these kinds of two-way classification models can be useful in certain circumstances, they are not particularly helpful when applied in real-world settings where there are multiple other mental health disorders that a patient may be suffering from. In addition to this highlighted problem, few of the current classification models developed for the diagnosis of mental health disorders offer any explanations for their predictions. In the medical domain, explainable machine learning models are essential for building trust and confidence in the diagnostic process, as they allow healthcare providers and patients to understand and verify the reasoning behind a model's diagnostic decision. In light of these noted limitations, this work explores the explainability of machine learning models that can be applied to perform multi-class classification among different mental health disorders. In this work, the different mental health disorders focused on are ADHD, Depression and Schizophrenia.

## II. Previous literature

Previous research literature on the topic of using motor activity data to infer mental health disorders can be broadly categorized into two main groups.

The first group involves classifying mental disorders given actigraph recordings by manually extracting input features as detailed by [1], [2], [3], [4] and [5]. This process is usually done by decomposing the motor activity sequences into statistical features such as the mean, variance and kurtosis. With respect to the machine learning models used for the mental health disease classification task when the input features are manually extracted, [2], [3], [5], [6], [7] and [8] showcase that the most widely used algorithm is the random forest algorithm. Although the manual extraction of input features provides an intuitive sense of the information that is important in the detection of mental health diseases, the classification models that utilize this information usually fail to explain how it is uniquely considered and combined to arrive at a classification decision.

The second group entails applying models that automatically extract features from actigraph data. [5] states that deep learning models are usually applied in the automatic extraction of features during the task of mental health disease classification. To further solidify this claim, [10], [9], [3] and [11] all make mention of the various deep learning-based approaches, including convolutional neural networks and recurrent neural networks, that can be applied in the automatic feature extraction process. However, one major drawback of using deep learning models for automatic

feature extraction in the task of mental health disease classification, is that the classifications made by these models are often difficult to explain. This is especially true when the architectures of such deep learning models are large and intricate.

## III. Research Methodology

### A. Data

The multi-class data used for this paper was obtained by merging three different datasets, namely Hyperaktiv, Depresjon and Psykose. These datasets contained the motor activity data of patients suffering from ADHD, Depression and Schizophrenia. All three datasets were collected by the Simula Research Laboratory, using identical actigraph equipment and setups to record the acceleration of the wrist of each patient within a three-dimensional space for each minute for several days [12]. The Hyperaktiv dataset consists of recordings from 51 patients with ADHD and 52 clinical controls [12]. Whereas the Depresjon and Psykose datasets, written about in [13] and [14], consists of recordings of 32 depressed and 22 schizophrenic patients respectively, sharing a control group of 32 healthy controls.

### B. Models based on manually extracted features

The machine learning models that utilize manually extracted features for the multi-class classification task explored in this paper included Naïve Bayes, Logistic Regression, Support Vector Machines, Random Forest, XGBoost, CatBoost, and Multi-layer Perceptron. These models were chosen after taking into consideration their individual strengths and weaknesses with regards to the task of multi-class classification. The manually extracted features from the input dataset that were used to implement the chosen models included the mean activity, coefficient of variation, interquartile range, percentage of sequence with zero activity, kurtosis and variance. These features were chosen based on their stated effectiveness in previous literature [5].

The training of the selected models was performed on both the original train dataset and an upsampled train dataset. The upsampled dataset was generated by applying the Synthetic Minority Over-sampling Technique (SMOTE) to the original train dataset to deal with class imbalances. SMOTE creates synthetic data points for the minority classes within a given dataset. It does this for each minority class by iteratively selecting each data point within a selected minority class and then generating synthetic data points between the chosen point and its k nearest neighbours. SMOTE terminates when each minority class is balanced with the majority class in the given dataset. Tackling the issue of large class imbalances within training data is imperative since, in the occurrence of a large class imbalance, a large portion of the time allotted for training will be spent on samples belonging to the majority class that in extreme cases could completely ignore the minority class samples. This problem is especially prevalent in smaller datasets where the model does not have enough data points to learn the behaviours of the observations in the minority class [15].

The optimal hyperparameters of the two best-performing models that were trained on upsampled and non-upsampled data were chosen using Bayesian Optimization, since it explores the hyperparameter space more efficiently than traditional methods such as grid search and random search [21]. The complete modelling pipeline of the manual feature extraction models is shown in Figure 5 in the appendix. The performance of each model was evaluated by an overall assessment of its accuracy, precision, F1 score and five-fold cross-validated F1 score. Further analysis of the highest-performing models trained on upsampled and non-upsampled data respectively, was demonstrated through classification reports, graphical depictions of the distribution of the predictions made by both models and their corresponding confusion matrices.

### C. Models based on automatically extracted features

The automatic feature extraction machine learning models that were chosen and used in this work included different variants of recurrent neural networks (RNNs) and convolutional neural networks (CNNs), as these two models were suggested to have the best performance in binary classification tasks for mental health disease detection within the reviewed literature [5]. For these models, two main approaches for preprocessing the data samples within the motor activity dataset were considered.

In the first of these approaches, we had to make the decision of whether to divide the full dataset into subjects or subsequences. In the latter case, we considered 24-hour motor activity recordings to be a meaningful unit of analysis based on our exploratory data analysis. By splitting each sequence into 24-hour subsequences, we increased the effective sample size of our data, providing more samples for the models to train on. In some cases, splitting the sequences resulted in better model fitting since long and variant sequences impeded model fitting in some cases. However, the interpretation of results becomes less obvious since each patient may get assigned different diagnoses on different days. It should be noted that the data was still split into training, validation, and test sets by subject to avoid data leakage which inadvertently biases test scores.

In the second approach, we considered whether or not to use the average of the data points within the data sequence recorded for each subject. We finally decided to implement two simple preprocessing functions to deal with the data sequences. These two functions involved: (1) an unweighted moving average smoothing across each data sequence and (2)

a binned averaging across each data sequence (see Figure 3 in the appendix for an example). Through experimentation, we discovered that smoothing the data sequence aided in the stabilization of some of the simpler LSTM-based models during training while the CNN models were able to fit to the raw sequence data without any smoothing.

Optimal data preprocessing was, however, found to differ for the different variants of our chosen models. Throughout our experiments, we tried different learning rates, loss functions, and optimizers. In addition to this, we also experimented with weighting the loss function, scaling the loss by the inverse of the size of each class and biasing our data sampling procedure when splitting the dataset into training, validation, and test sets, to ensure that the issue of class imbalances was mitigated. The full modelling pipeline for the models based on the automatic extraction of features is shown in Figure 6 in the appendix.

*1) Recurrent neural networks:* A bidirectional LSTM model, inspired by [11], was the main recurrent neural network model architecture implemented in this work. As shown in Figure 1, the information flow in the model resembles that of a standard seq2vec model, but with an important modification. Instead of passing the output from the hidden units from the LSTM layers directly to the model's dense layers, we first scale the hidden units with attention weights. These attention weights are computed by passing the values from the hidden units through a dense layer and a softmax activation function. A context vector $c$ is then computed where an entry is computed as

$$c_i = a(t_i) \times h(t_i) \tag{1}$$

which is passed to a MLP with two hidden layers. The graphical representation of our recurrent neural network architecture can be seen in Figure 1.

*2) Convolutional neural networks:* The sequences can also be modelled using a CNN. A CNN consists of one or more convolutional layers, where each layer has kernels that convolve across some given input data to create feature maps that pick up different characteristics of the data. When using a CNN to make a prediction, these extracted feature maps are then passed through one or more dense layers to generate the prediction [22]. In our specific case, the convolutions of the CNN were executed through the implementation of equation 2 below, where $C$ denotes the result of a dot product convolution applied to each data sequence $X$ of length $T$ with a filter of length $l$, a bias parameter $b$ and a final non-linear function $f$, which in our chosen architecture was a Rectified Linear Unit (ReLU).

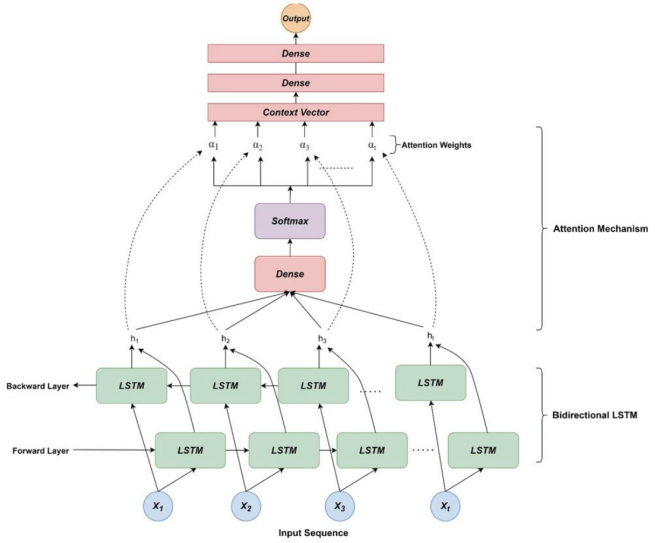


Fig. 1: Bi-LSTM neural network with attention (Figure taken from [11])

$$Ct = f(\omega * X_{t-l/2:t+l/2} + b) \mid \forall t \in [1, T] \tag{2}$$

## IV. Experiments

### A. Experiments performed with the manual feature extraction models

The results from testing the efficacy of the implemented manual feature extraction models are shown in table 1 below. The random forest model was the best-performing model trained on the upsampled dataset as demonstrated by its high average performance across all the noted performance metrics.

| Model | Up-sampled? | Accuracy | Precision | F1 | Mean CV (weighted F1) |
|---|---|---|---|---|---|
| CATB | Yes | 0.4286 | 0.4405 | 0.4214 | 0.72054 |
| RF | Yes | 0.4524 | 0.461 | 0.4426 | 0.66562 |
| XGB | Yes | 0.3333 | 0.3482 | 0.331 | 0.65146 |
| SVM | Yes | 0.5 | 0.4963 | 0.4888 | 0.57288 |
| LREG | Yes | 0.5476 | 0.5284 | 0.5101 | 0.54864 |
| NB | Yes | 0.4762 | 0.551 | 0.4682 | 0.48892 |
| MLP | Yes | 0.4524 | 0.4801 | 0.4452 | - |
| SVM | No | 0.3095 | 0.119 | 0.169 | 0.41154 |
| LREG | No | 0.3095 | 0.1508 | 0.1905 | 0.38014 |
| XGB | No | 0.3333 | 0.2836 | 0.2642 | 0.36846 |
| RF | No | 0.2857 | 0.3507 | 0.225 | 0.36642 |
| CATB | No | 0.3095 | 0.2525 | 0.2384 | 0.34038 |
| NB | No | 0.4524 | 0.6155 | 0.4211 | 0.32478 |
| MLP | No | 0.381 | 0.2774 | 0.316 | - |

Table 1: Results from baseline models with manually extracted features. Upsampling was carried out using the Synthetic Minority Over-sampling Technique (SMOTE) on the training set.

| Model | Up-sampled? | Accuracy | Precision | F1 | Mean CV (weighted F1) |
|---|---|---|---|---|---|
| SVM | No | 0.303 | 0.3593 | 0.221 | 0.4812 |
| RF | Yes | 0.4242 | 0.4848 | 0.4241 | 0.6697 |

Table 2: Results from tuned baseline models with manually extracted features. Hyperparameter tuning performed using Bayesian Optimization.

### *B. Experiments performed with the automatic feature extraction models*

*1) Recurrent neural networks:* In our initial experiments, we passed the average of all the generated hidden units from the bidirectional LSTM layer into the dense layers of the network. This resulted in very poor performance when the model was fitted to both full patient data sequences and 24-hour subsequence splits. We postulate that this poor performance obtained was because of the information loss incurred as a result of the implementation of the averaging procedure.

Therefore, in our second set of experiments, we fed all the generated hidden outputs of the bidirectional LSTM layer into the dense layers of the network without taking the average of these outputs. However, after varying the hidden unit sizes, the number of dense layers, the number of LSTM units per layer, the dropout rates between the dense layers, and the hyperparameters within the LSTM units themselves, we were still not able to fit models that generalized to our test set.

In the third batch of experiments, we decided to batch the data points within each input data sequence into four distinct vectors of equal length. Each vector consisted of motor activity data for a particular time frame of the day: 00:00-05:59, 06:00–11:59, 12:00–17:59, and 18:00–23:59. This batching process was implemented to closer model the dependencies within within each time frame. This transformed the shape of each input data sequence from a $1 \times N$ dimensional vector to a $4 \times n$ dimensional vector, where $N$ represented the total number of data points within the entire input sequence and $n = N/4$ represented the number of data points within each of the 4 vectors the input sequence was split into. For experiments regarding the full patient data sequences, we included an extra preprocessing step which involved encoding the input sequences before batching. This preprocessing step was necessary since the full length data sequences for the patients had varying lengths. During the encoding process, the sequence length of the longest full patient data sequence was first found and then every other patient data sequence length was made to equal this found longest sequence length through the repetition of data points. After this, each reconstructed patient data sequence was then passed through a Linear Layer or an LSTM Encoding Layer to generate a data sequence containing 1440 data points, which corresponds to the same number of points in a 24-hour subsequence split. The generated data points were then batched and fed into the remainder of the network to produce a prediction. However, even though batching the input data points seemed to have stabilized the model's training and improved its performance on the test set, the test performance obtained was not high enough. The results obtained from the best performing BiLSTM architecture can be seen in Table 3 below.

| Model | Accuracy | Precision | Weighted F1 |
|---|---|---|---|
| BiLSTM with attention | 0.38 | 0.3768 | 0.4 |

Table 3: Results from the BiLSTM with attention architecture

After these experiments, we finally concluded that the poor performance of our implemented recurrent neural network models was due to misalignment between models and data. Although LSTMs are specifically designed to handle long sequences, the applied LSTM models may not have had enough training data to properly learn the complex dependencies within the data.

*2) Convolutional neural networks:* In our initial set of experiments using a CNN, a CNN with one-dimensional convolutional layers was deployed. This model took in univariate data sequences that comprised of 2048 timesteps as input, with data sequences of shorter length being padded with zeros. This input was first passed through a batch normalization layer for normalization and then passed through two 1D convolutional layers, consisting of kernels with sizes of 2 and 3, respectively. These convolutional layers also had 16 and 32 kernels, respectively, with each kernel having a stride of 1. The activation function used in this CNN architecture was ReLU. After passing through the convolutional layers, the generated output was then passed through another batch normalization layer and then through a global max pooling layer. The function of the pooling layer was to reduce the number of parameters the CNN had to learn, as a large number of learnable parameters within a model could lead to overfitting. The output from the pooling layer was then flattened and passed through two dense layers with 12 units each and ReLU activation. Finally, the output from the two previous dense layers was passed through a dense layer with 4 units and a softmax activation function. The CNN was compiled with an SGD optimizer and the categorical cross-entropy loss function with specified weights for each class to partially deal with the class imbalances within the dataset. These weights tell the model to "pay more attention" to samples from an under-represented class [23]. The network is summarized in Figure 7 in the appendix.

For the second set of experiments, a CNN with two-dimensional convolutional layers was applied after the

one-dimensional data input sequences were converted into two-dimensional images using Gramian Angular Field imaging (GAF). GAF first rescales an input data sequence using min-max scaling, and then represents it in polar coordinates by using the equation below [24],

$$\phi = \arccos(\tilde{x}_i), -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} r = \frac{t_i}{N}, t_i \in \mathbb{N} \quad (3)$$

The polar coordinates are then represented in matrix form that can be interpreted as an image. Below are examples of sequences for six patients that have been converted using Gramian Angular Field imaging.

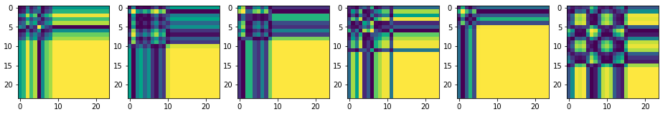

Fig. 2: Gramian Angular Field imaging of univariate time series of activity level where sequence length converted to 2,048 timesteps by truncation and padding.

The generated image representations is then fed through a CNN with a similar architecture to the CNN used in the first set of experiments, but engineered to handle two-dimensional images. The full architecture of this model can be seen in Figure 7 in the appendix.

In the last set of experiments, a CNN architecture was designed to train on the concatenation of vector representations of the two-dimensional images of the input data generated using GAF and feature vectors generated from a multilayer perceptron network that was fed with the manually extracted data features as input. The main idea behind this architecture was to ensure that the features of each image generated from the data would be augmented with the general features of the disease the image represented. Even though this architecture performed better than the two previous architectures, in terms of weighted f1 scores, its general results were still not very impressive. The full architecture can be seen in Figure 8 in the appendix.

It is however useful to note that the hyperparameters that defined our implemented CNN architectures required careful fine-tuning, as getting stable training results was difficult. Hence, the architectures presented here represent a small fraction of all architectures that were trained and only represent ones that provided somewhat stable performance.

The results from the CNNs are shown in table 4 below. Furthermore, the training process in terms of the models' loss, accuracy and F1 score can be seen in figures 13, 14 and 15 in the appendix. From table 4, it can be inferred that the 1D CNN architecture achieved the most stable training, however its major drawback is that it misclassifies all the test data points as belonging to the control group (see Figure 16 in the appendix). The 2D CNN with manual features produced similar predictions but with a much more unstable training process (see Figure 15 in the appendix).

| Model | Accuracy | Precision | Weighted F1 |
|---|---|---|---|
| CNN1d | 0.5172 | 0.2956 | 0.3762 |
| CNN2d with GAF | 0.2759 | 0.239 | 0.253 |
| CNN2d with GAF + manual features | 0.5517 | 0.3044 | 0.3923 |

Table 4: Results from convolutional neural networks

## V. Discussion

### A. Model performance

In conclusion, the random forest classifier trained on an upsampled dataset achieved the highest performance when taking the accuracy, precision, weighted F1 score, cross-validation score, type of predictions and the stability in the results into account. Models fit to upsampled data outperform their counterparts fit to non-upsampled data on all performance metrics. Models based on manually extracted features provided higher and more stable performance. The RNNs and CNNs did not generate stable results across training instances.

### B. Data limitations

In accounting for the generally poor performances of our models, two crucial factors are the size and the quality of the dataset. It is notable that even the best-performing deep learning model, acccording to the performance metric of accuracy, which uses both raw sequence data and manually extracted features, is only able to achieve a weighted f1 score shy of 0.4. However, the models that were fit only to features manually extracted from upsamled data are able to achieve significantly higher performance. We see that performance is generally higher when the same model is trained on upsamled data instead of non-upsamled data (see Table 1). This appears to indicate that data size and data balance are of central importance to our deep learning models' inability to achieve good predictive performances.

The issue with data quality shows up in different ways. As we can see in Figure 17, the same subject's activity levels can vary quite dramatically from day to day. Though apparently meaningful analytical units, 24-hour sequences may not be able to fully capture the complexity of a person's weekly or monthly rhythms. However, models fit to the full sequences were also not able to perform very well. This may be caused by high variance both within subjects suffering from the same condition and within each individual subject. Inherent noise puts an upper boundary on the performance that the model will be able to achieve. This would be true even if we had a larger dataset. Other sources of noise include potential differences in how the actigraph devices were calibrated.

Calendar effects may also be present in the data. Especially effects of the season are likely to be a source of potential bias. As can be observed in Figure 18, the groups differ on average with respect to what time of the year the recordings took place. One striking observation is that patients suffering from schizophrenia only had recorded data in September and October. Even in a binary classification, this could constitute a major bias source since data from the respective control group (labelled *DEPR_Control* in the plot) was mostly recorded in other months. To our knowledge, no paper working on these datasets has addressed issues surrounding potential seasonality effects. Controlling for seasonality is infeasible since it would require identifying the effect of the month or the season on the signal. This is not possible given the high levels of noise and the small size of the data.

### C. Model Explainability

Explainable AI is, as previously mentioned, important in the classification of psychological disorders. Although understanding how a random forest with 620 decision trees that are all trained on bootstrap datasets with just a subset of all features can become difficult, there are ways to still infer global and local interpretability. Figure 11 in the appendix shows that the percentage with zero activity, the coefficient of variance and the variance are the most important features of the model when looking at how well each feature helps the decision trees split the data set into nodes.

Furthermore, SHAP values are a model-agnostic method that can be applied to understand how the features affect each prediction. Figure 12 in the appendix show the three most important features for predicting ADHD. A higher percentage of zero activity, higher variance or higher coefficient of variation could be an indication for ADHD. This could be connected to restlessness, impulsivity or how easily the patient gets distracted - which are all common symptoms of ADHD.

Arguably, model explainability is only relevant when the model performs well at the task at hand. Due to the unsatisfactory performance of our neural network models, attempts at explaining the models' predictions would be moot. By including an attention mechanism in our LSTM model, we attempted to incorporate explainability into a model that would otherwise be far from inherently explainable. However, the claim that attention mechanisms provide faithful explanations of model outputs is controversial. We would need to further scrutinize our model to be able to confidently claim that we had properly explained how the model makes a certain prediction [17] [18]. Alternative frameworks for explaining model attributions include TimeSHAP: a modification of Kernel SHAP to recurrent models [19]. These methods are still in their infancy, and we expect to see developments in the area of neural network explainability.

## VI. Conclussion and suggestions for future work

In summary, motor activity recordings have the potential to be a useful feature for developing more objective measures of mental health disorders that could complement the numerous other factors healthcare providers normally incorporate in their assessments. This paper has trained several machine learning and deep learning models on a dataset of motor activity recordings to understand if a machine learning-based approach to diagnosing patients can be applied in a multiclass setting where multiple mental disorders are present. A random forest classifier that was trained on an upsampled dataset achieved the best performance with a weighted F1 score of 0.4241. However, the results also indicated that it is challenging to diagnose patients based solely on motor activity recordings and that larger datasets are needed before machine learning models can be reliably used in this field. To further enhance the trustworthiness of these models, data scientists could incorporate explainability elements into the overall assessment process.

## VII. References


[1] Zanella-Calzada, L. A., Galván-Tejada, C. E., Chávez-Lamas, N. M., Gracia-Cortés, M. D. C., Magallanes-Quintanar, R., Celaya-Padilla, J. M., Galván-Tejada, J. I., & Gamboa-Rosales, H. (2019). Feature Extraction in Motor Activity Signal: Towards a Depression Episodes Detection in Unipolar and Bipolar Patients. Diagnostics (Basel, Switzerland), 9(1), 8. https://doi.org/10.3390/diagnostics9010008

[2] Rodríguez-Ruiz, J. G., Galván-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., Luna-García, H., Magallanes-Quintanar, R., & Soto-Murillo, M. A. (2020). Comparison of Night, Day and 24 h Motor Activity Data for the Classification of Depressive Episodes. Diagnostics, 10(3), 162. https://doi.org/10.3390/diagnostics10030162

[3] Jakobsen, P., Garcia-Ceja, E., Riegler, M., Stabell, L. A., Nordgreen, T., Torresen, J., Fasmer, O. B., & Oedegaard, K. J. (2020). Applying machine learning in motor activity time series of depressed bipolar and unipolar patients compared to healthy controls. PLOS ONE, 15(8), e0231995. https://doi.org/10.1371/journal.pone.0231995

[4] Rodríguez-Ruiz, J. G., Galván-Tejada, C. E., Luna-García, H., Gamboa-Rosales, H., Celaya-Padilla, J. M., Arceo-Olague, J. G., & Galván Tejada, J. I. (2022). Classification of Depressive and Schizophrenic Episodes Using Night-Time Motor Activity Signal. Healthcare, 10(7), 1256. https://doi.org/10.3390/healthcare10071256

[5] Adamczyk, J., & Malawski, F. (2021). Comparison of Manual and Automated Feature Engineering for Daily Activity Classification in Mental Disorder Diagnosis. COMPUTING AND INFORMATICS, 40(4), 850–879. https://doi.org/10.31577/cai_2021_4_850

[6] Enrique Garcia-Ceja, Michael Riegler, Petter Jakobsen, Jim Tørresen, Tine Nordgreen, Ketil J. Oedegaard, and Ole Bernt Fasmer. 2018. Depresjon: a motor activity database of depression episodes in unipolar and bipolar patients. In Proceedings of the 9th ACM Multimedia Systems Conference (MMSys '18). Association for Computing Machinery, New York, NY, USA, 472–477. https://doi.org/10.1145/3204949.3208125

[7] Galván-Tejada, C. E., Zanella-Calzada, L. A., Gamboa-Rosales, H., Galván-Tejada, J. I., Chávez-Lamas, N. M., Gracia-Cortés, M. d. C., ... Celaya-Padilla, J. M. (2019). Depression Episodes Detection in Unipolar and Bipolar Patients: A Methodology with Feature Extraction and Feature Selection with Genetic Algorithms Using Activity Motion Signal as Information Source. Mobile Information Systems, 2019. https://doi.org/10.1155/2019/8269695

[8] Rodríguez-Ruiz, J. G., Galván-Tejada, C. E., Luna-García, H., Gamboa-Rosales, H., Celaya-Padilla, J. M., Arceo-Olague, J. G., & Galván Tejada, J. I. (2022). Classification of Depressive and Schizophrenic Episodes Using Night-Time Motor Activity Signal. Healthcare, 10(7), 1256. https://doi.org/10.3390/healthcare10071256

[9] Espino-Salinas, C. H., Galván-Tejada, C. E., Luna-García, H., Gamboa-Rosales, H., Celaya-Padilla, J. M., Zanella-Calzada, L. A., & Tejada, J. I. G. (2022). Two-Dimensional Convolutional Neural Network for Depression Episodes Detection in Real Time Using Motor Activity Time Series of Depresjon Dataset. Bioengineering (Basel, Switzerland), 9(9), 458. https://doi.org/10.3390/bioengineering9090458

[10] Frogner, J. I., Noori, F. M., Halvorsen, P., Hicks, S. A., Garcia-Ceja, E., Torresen, J., & Riegler, M. A. (2019). One-Dimensional Convolutional Neural Networks on Motor Activity Measurements in Detection of Depression. In Proceedings of the 4th International Workshop on Multimedia for Personal Health & Health Care (HealthMedia '19). Association for Computing Machinery, New York, NY, USA, 9-15. https://doi.org/10.1145/3347444.3356238

[11] Bondugula, R. K., Sivangi, K. B., & Udgata, S. K. (2022). Identification of Schizophrenic Individuals Using Activity Records Through Visualization of Recurrent Networks. In S. K. Udgata, S. Sethi, & X.-Z. Gao (Eds.), Intelligent Systems (pp. 653-664). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-0901-6_57

[12] Hicks, S., Stautland, A., Fasmer, O. B., Førland, W., Hammer, H. L., Halvorsen, P., Mjeldheim, K., Oedegaard, K. J., Osnes, B., Syrstad, V. E. G., Riegler, M., & Jakobsen, P. (2021). HYPERAKTIV: An Activity Dataset from Adult Patients with Attention-Deficit/Hyperactivity Disorder (ADHD). In Proceedings of the 12th ACM Multimedia Systems Conference. https://doi.org/10.1145/3458305.3478454

[13] Garcia-Ceja, E., Riegler, M., Jakobsen, P., Tørresen, J., Nordgreen, T., Oedegaard, K. J., & Fasmer, O. B. (2018). Depresjon: A Motor Activity Database of Depression Episodes in Unipolar and Bipolar Patients. In Proceedings of the 9th ACM on Multimedia Systems Conference (MMSys '18). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3204949.3208125

[14] Jakobsen, P., Garcia-Ceja, E., Stabell, L. A., Oedegaard, K. J., Berle, J. O., Thambawita, V., Hicks, S. A., Halvorsen, P., Fasmer, O. B., & Riegler, M. A. (2020). PSYKOSE: A Motor Activity Database of Patients

with Schizophrenia. In 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS). https://doi.org/10.1109/CBMS49503.2020.00064

[15] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357. https://doi.org/10.1613/jair.953

[16] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. arXiv.org. https://doi.org/10.48550/arxiv.1206.2944

[17] Jain, S., & Wallace, B. C. (2019). Attention is not explanation. arXiv preprint arXiv:1902.10186. https://doi.org/10.48550/arXiv.1902.10186

[18] Wiegreffe, S., & Pinter, Y. (2019). Attention is not not explanation. arXiv preprint arXiv:1908.04626. https://doi.org/10.48550/arXiv.1908.04626

[19] Bento, J., Saleiro, P., Cruz, A. F., Figueiredo, M. A., & Bizarro, P. (2021, August). TimeSHAP: Explaining recurrent models through sequence perturbations. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (pp. 2565-2573). https://doi.org/10.48550/arXiv.2012.00073

[20] O' Shea, K. & Nash, R. (2015). An Introduction to Convolutional Neural Networks. https://doi.org/10.48550/arXiv.1511.08458

[21] Agnew, A., & Batra, N. (2020, May 5). Exploring Bayesian optimization: Breaking Bayesian optimization into small, sizeable chunks. Distill. doi:10.23915/distill.00026

[22] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press. Chapter 9: "Convolutional Networks." (pp. 189-207). Retrieved from http://www.deeplearningbook.org.

[23] TensorFlow. "tf.keras.Model — TensorFlow," [Online]. Retrieved from: https://www.tensorflow.org/api_docs/python/tf/keras/Model.

[24] Z. Wang and T. Oates, "Imaging Time-Series to Improve Classification and Imputation," CoRR, vol. abs/1506.00327, 2015. [Online]. Available: http://arxiv.org/abs/1506.00327.

# VIII. Appendix

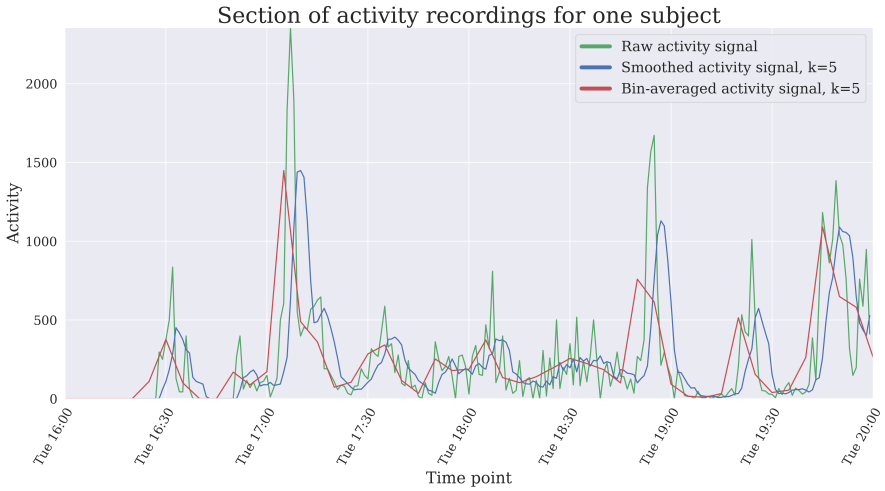


Fig. 3: Time series plot for one patient

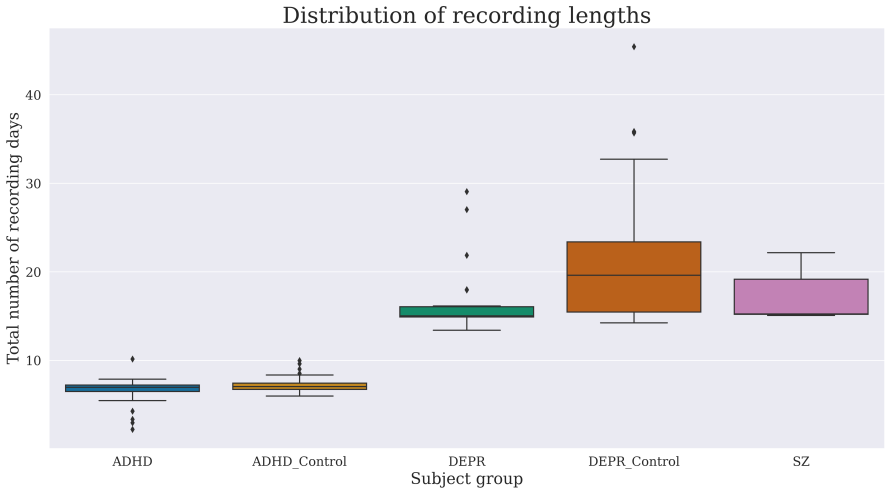


Fig. 4: Box plots of total number of recorded days per class

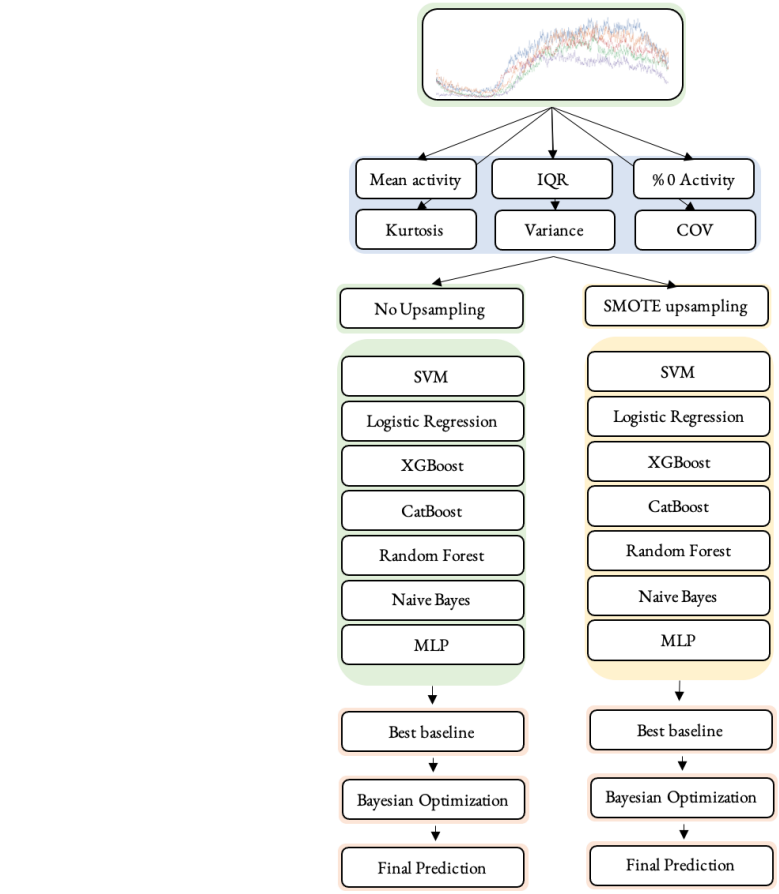


Fig. 5: Modelling pipeline for models using manually extracted features. "COV" = coefficient of variance, "IQR" = interquantile range Q3 minus Q1.

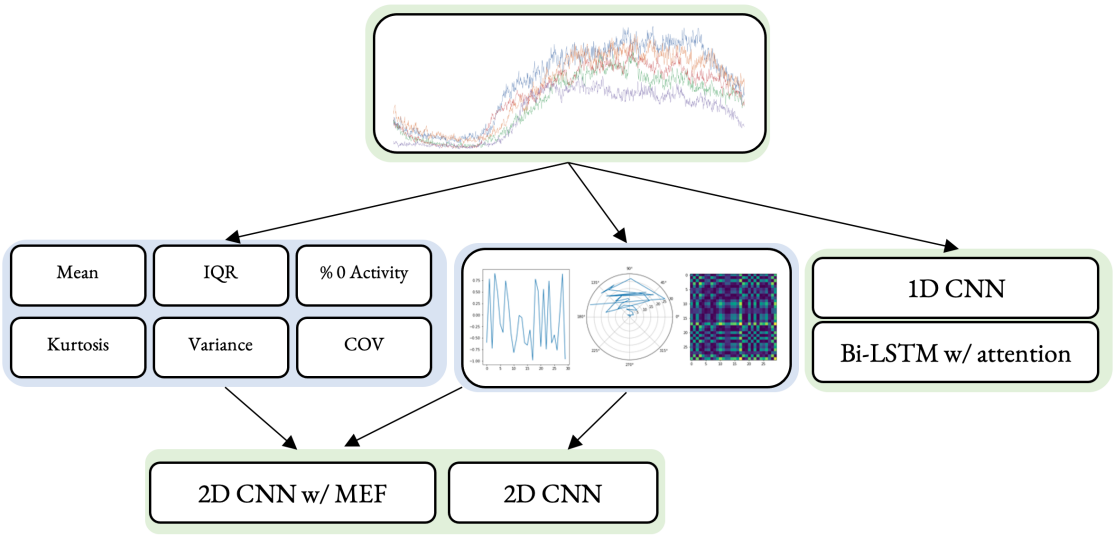


Fig. 6: Modelling pipeline for models using automatically extracted features (and manually extracted features). "MEF" = manually extracted features, "COV" = coefficient of variance, "IQR" = interquartile range Q3 minus Q1

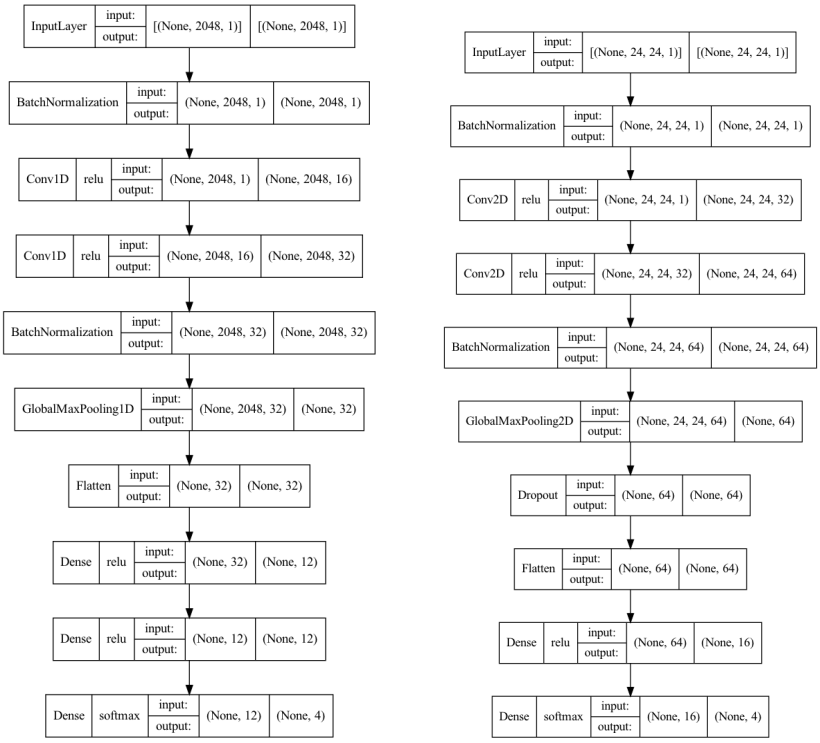


Fig. 7: Architectures of CNN (1D left, 2D right)

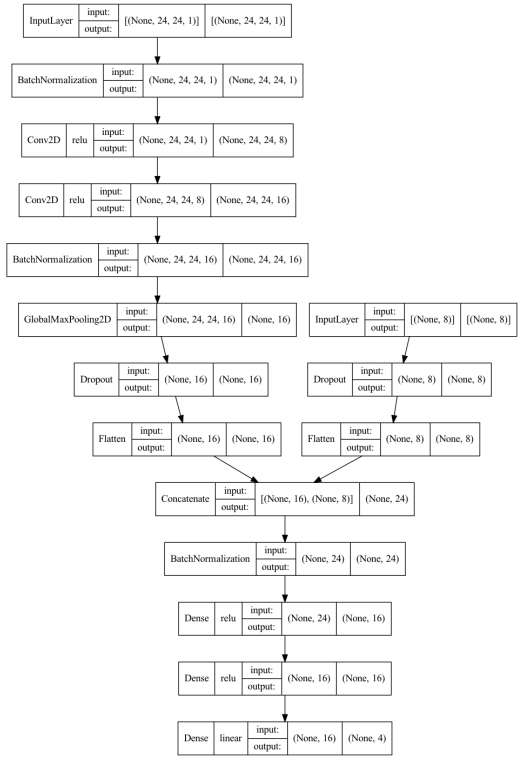


Fig. 8: Architecture 2D CNN with manually extracted features

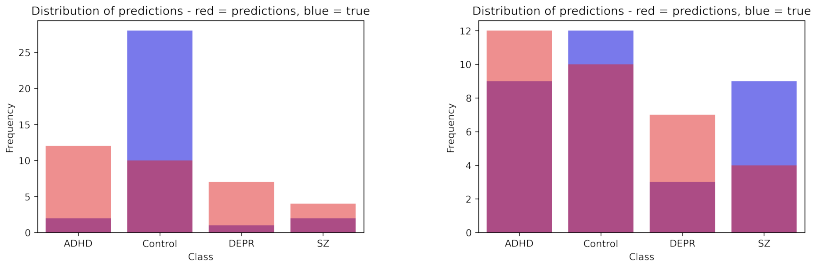


Fig. 9: Distribution of predictions (non-upsampled SVM left, upsampled RF right)

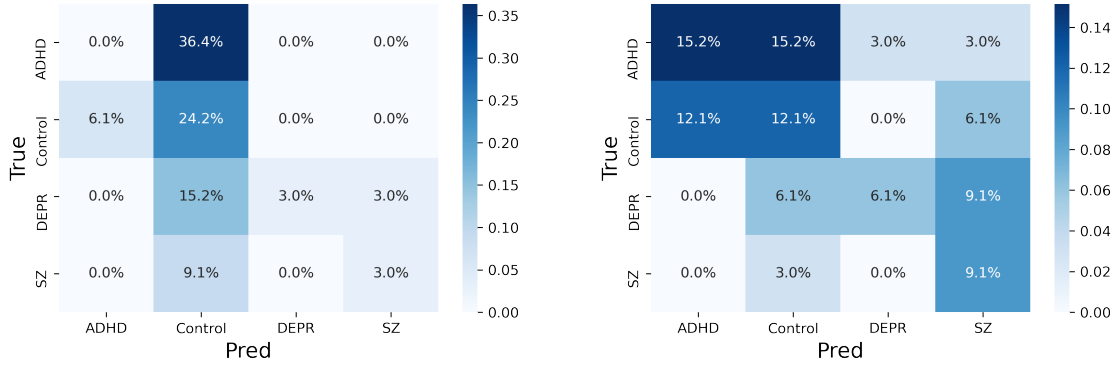


Fig. 10: Confusion matrices (non-upsampled SVM left, upsampled RF right)

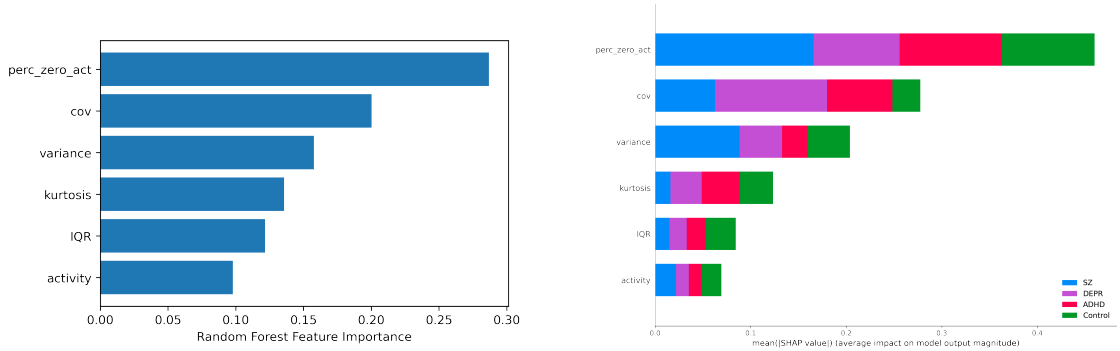


Fig. 11: Feature importance (left), SHAP values (right)

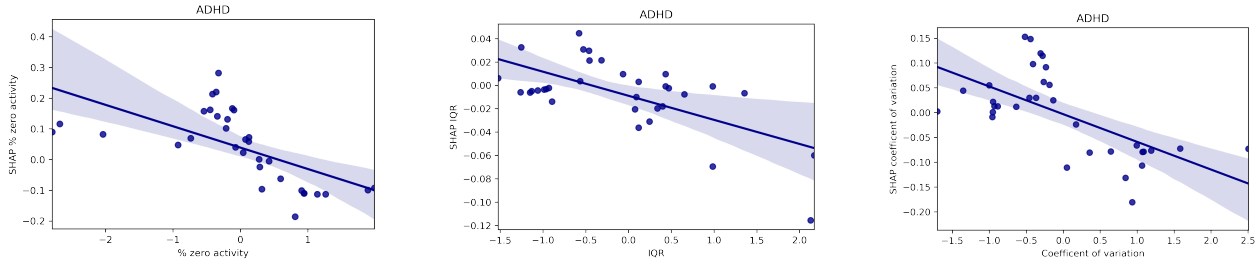


Fig. 12: SHAP Dependence Plot, ADHD, % zero activity (left), Shap Dependence Plot, ADHD, variance (middle), SHAP Dependence Plot ADHD, Coefficient of variation (right)

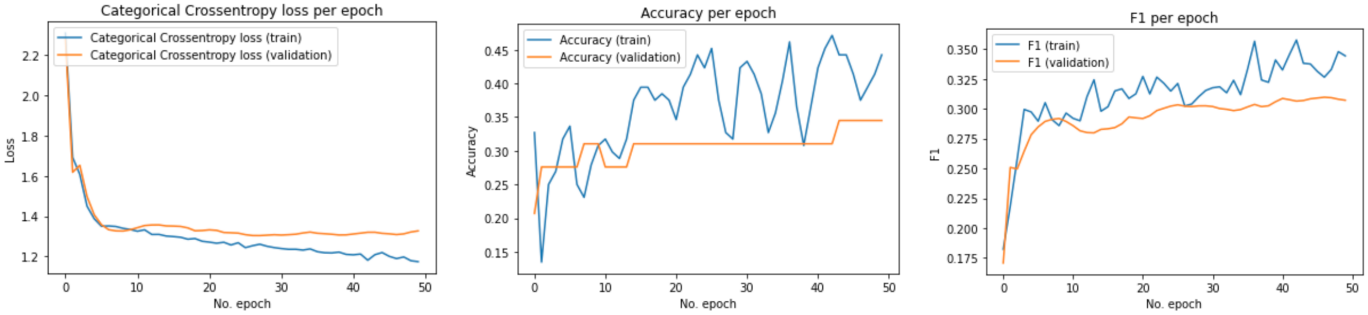


Fig. 13: 1D CNN, Training and validation loss per epoch (left), accuracy per epoch (middle), F1 per epoch (right)

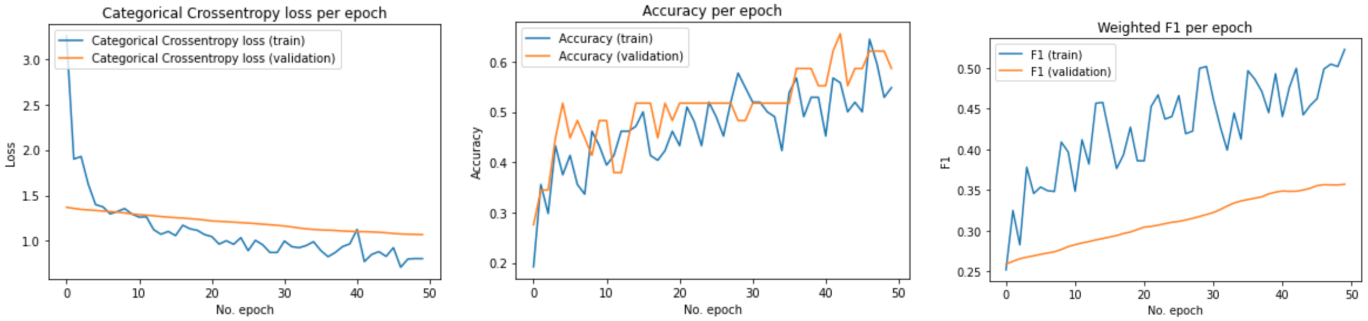


Fig. 14: 2D CNN, Training and validation loss per epoch (left), accuracy per epoch (middle), F1 per epoch (right)

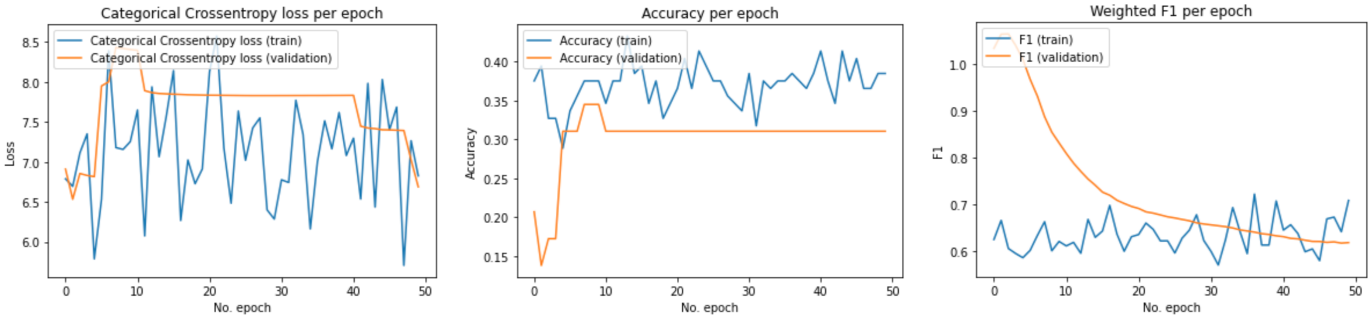


Fig. 15: 2D + manual features CNN, Training and validation loss per epoch (left), accuracy per epoch (middle), F1 per epoch (right)

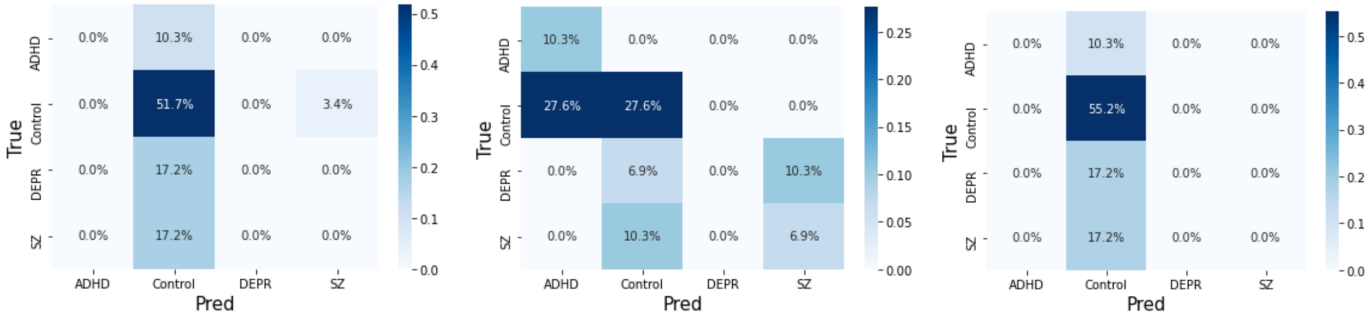


Fig. 16: 2D + manual features CNN, Training and validation loss per epoch (left), accuracy per epoch (middle), F1 per epoch (right)

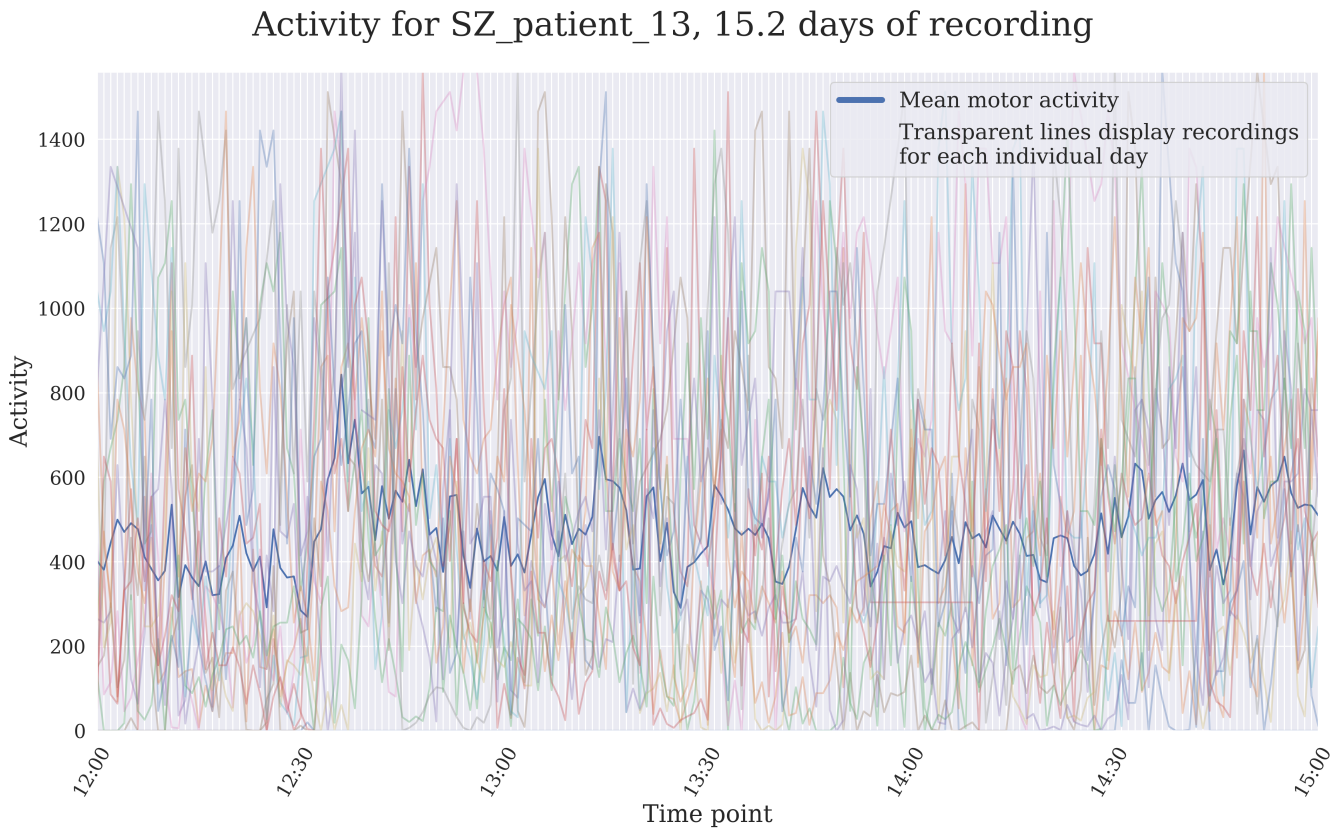


Fig. 17: Plot showing the variance in one subject's activity levels. Signals above the 99th percentile were removed.

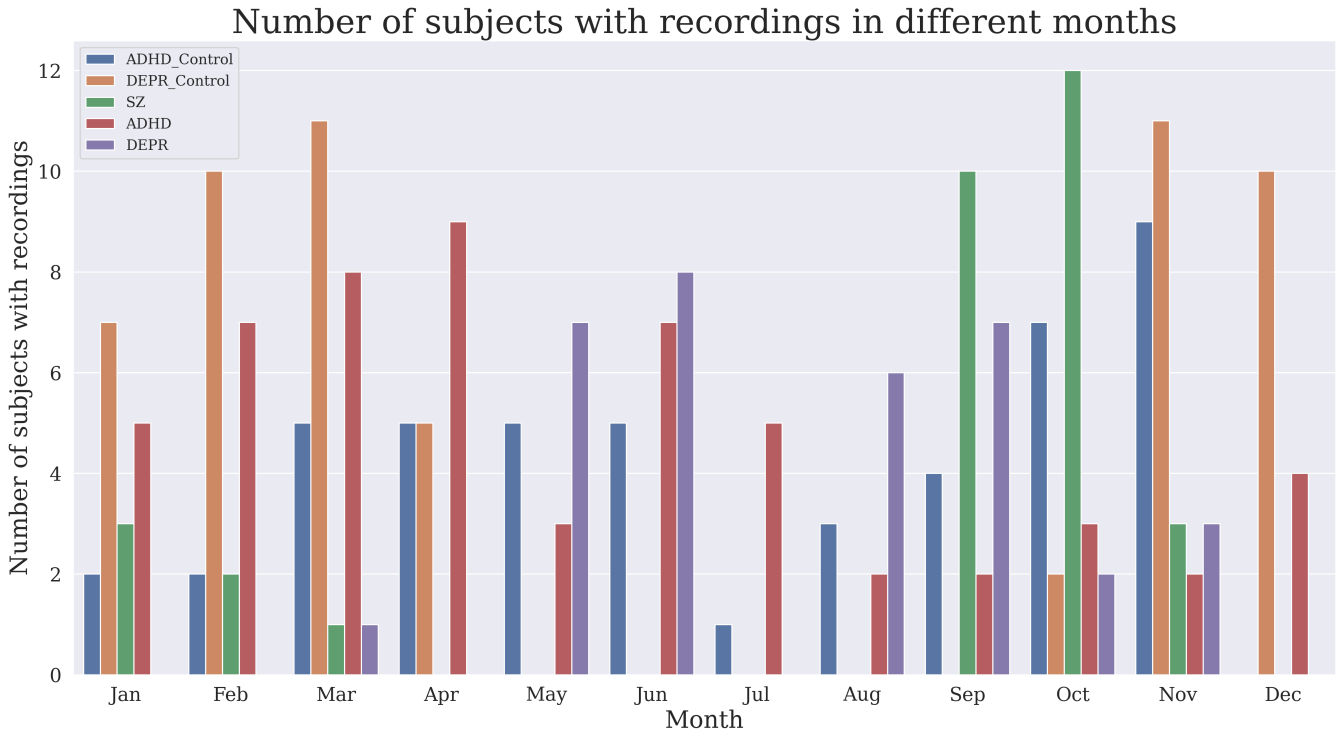


Fig. 18: Plot showing in which months data was recorded from subjects.