Effects of Varying Time Windows of EEG Recordings on Accuracy of ML Identification of Abnormal EEG Signals

Benjamin Kaplan and Olga Soldatenko

Introduction

Electroencephalograms (EEGs) are noninvasive measurements of electrical activity on the surface of the scalp which is representative of brain activity. EEGs have a variety of applications, including diagnosis of neural disorders. Abnormal EEG readings can be indicative of a variety of potential neurological disorders, but are most frequently used to diagnose patients suffering from epilepsy. [1] However, analysis of EEG readings can be difficult to perform. In addition to the resources required to set up and record the EEG reading, analysis is time-intensive and must be performed by trained physicians. [2] Machine learning (ML) is a promising avenue for reducing the workload on the limited number of experts qualified to perform these analyses and allowing them to better prioritize which readings to analyze.

Many ML algorithms have been applied to the problem of sorting normal EEG readings from abnormal EEG readings and include time-based, spectral, and feature-based analyses. [2] A popular database used for training these models is the Temple University Hospital (TUH) Abnormal EEG Corpus. Its popularity is due to its very large dataset, which includes over 1,000 hours of EEG recording from over 2,000 different patients, nearly half of whom were identified as having abnormal EEG readings. [3]

While many different approaches to sorting the data from the TUH Abnormal EEG corpus (TUAB) have been attempted, there appears to be no common consensus of how much time of each patient's recording should be used for analysis. Some studies used as few as one minute of each patient's data since trained clinicians require only a few minutes of the recording to make their diagnosis. [4] Roy, et al. note that when they used a small subset of the TUAB data and took different amounts of time from each recording for training, they found no drop in the validation set accuracy after the 11th minute (Figure 1). [4] They therefore conclude that 11 minutes is the optimal amount of recording time to be taken for analysis. However, it is unclear which algorithm they used for this experiment and they also do not show any gains in validation set accuracy by taking more time from each recording to explicitly justify their time of 11 minutes. Additionally, Gemein et al. use even longer recording times (up to 20 minutes) to train their ML models, [2] without any obvious loss in accuracy as observed by Roy et al.

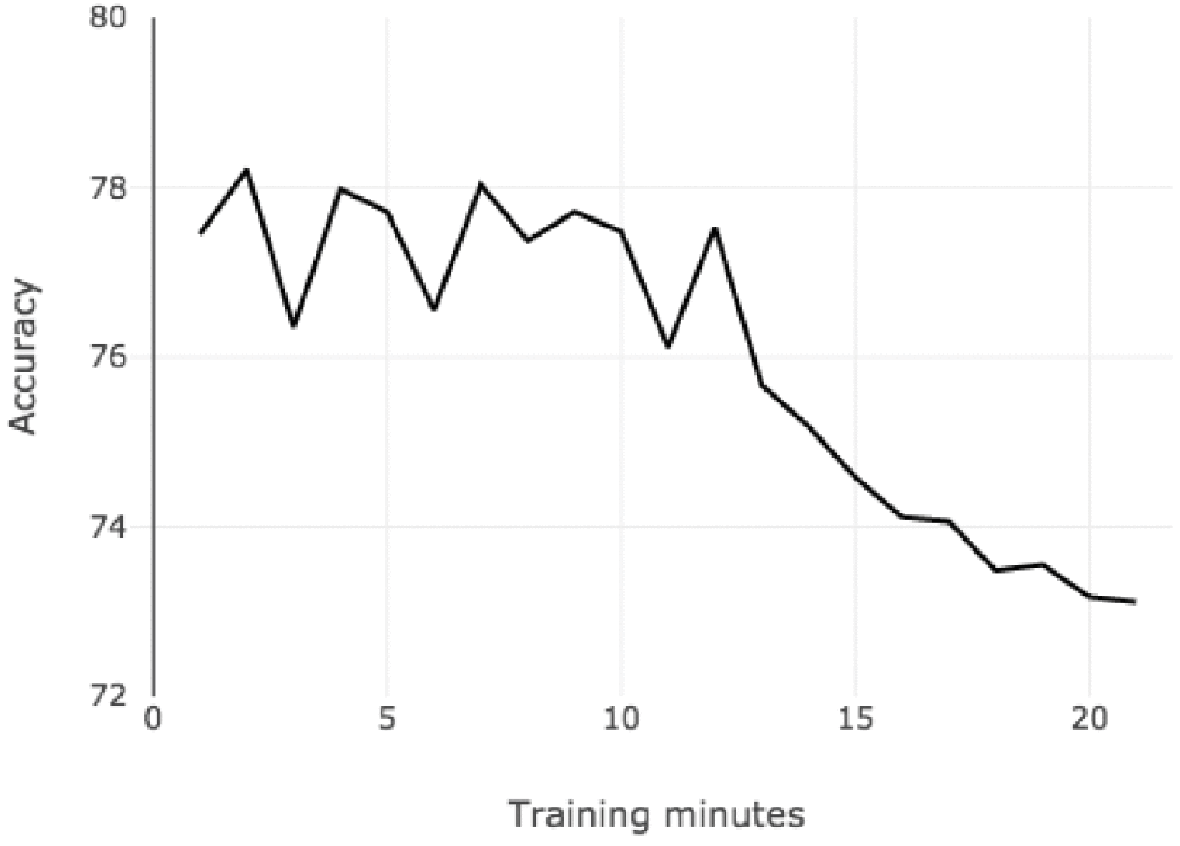


Figure 1. Graph of accuracy vs training minutes from Roy, et al.

Gemein et al. also note that they drop the first 60 seconds of each EEG recording since they saw many artifacts during that period. However, Roy, et al. make no mention of dropping the first minute of recording. Additionally, while artifacts can be seen during the first minute of many of the EEG recordings in TUAB (see Figure 2), Gemein, et al. do not provide any quantitative evidence that dropping the first minute improves the accuracy of the ML algorithms they use for sorting.

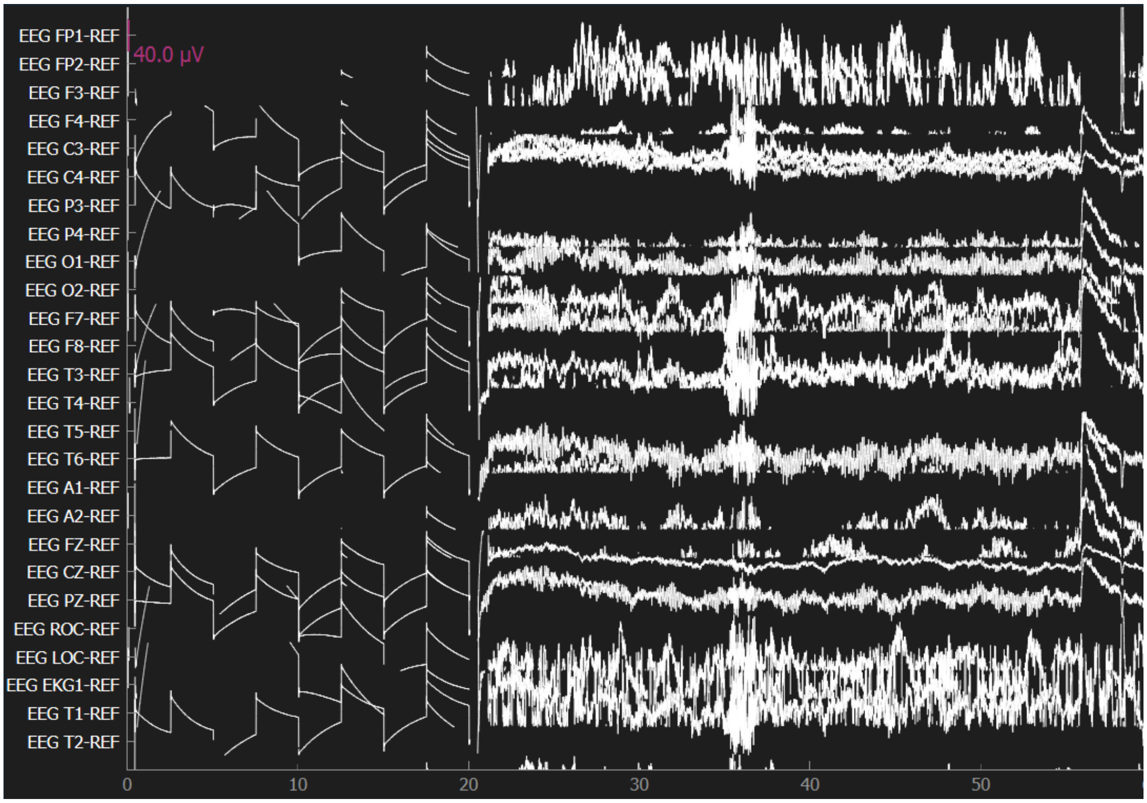


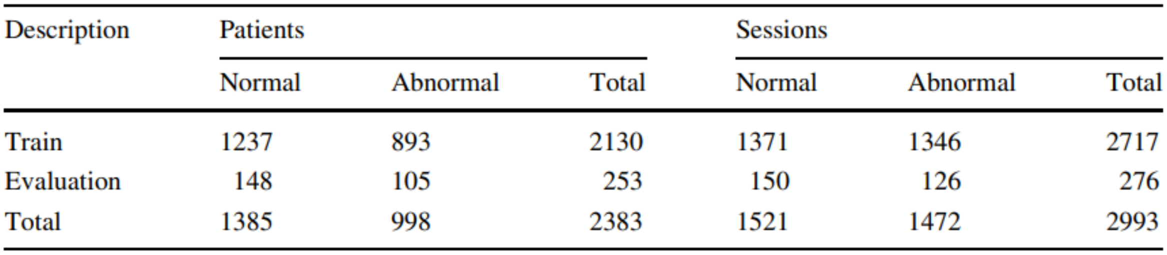
Figure 2: Example of artifacts seen in first minute of EEG readings in the TUAB database

The goal of this project is to measure the effect of the time window used for each EEG recording on the accuracy of various algorithms. This includes both the total length of the EEG recording used and whether or not the first minute of recording is dropped during preprocessing. Our approach improves on that of Roy et al.'s by using the entire TUAB dataset for this purpose rather than just a subset of undefined size and by testing the effect of different time windows on the accuracy of three distinct time-based ML models: the "shallow" and "deep" ConvNet architectures described by Schirrmeister et al. [5] and the EEGNet architecture described by Lawhern et al. [6] Our approach also attempts to test the assertion of Gemein, et al. that dropping the first minute of each recording improves sorting accuracy.

Methods

We have worked with the TUH Abnormal EEG Corpus [7] provided by Temple University. Due to the medical nature of the data, most of the sets of this type are kept private, this is the biggest publicly available dataset of EEG sessions, which is also constantly improved by the owners. The data holds one or more sessions of varied length for each patient, each record consists of EEG data and a report provided by the clinician that includes the patient's clinical history and summary of the medication, though we did not take this metadata into account in our analysis.

In table 1 we can see the data description. It's important to emphasize that there is no overlap between patients in the evaluation and training sets, the set is demographically balanced with respect to gender and age.



Description	Patients			Sessions		
	Normal	Abnormal	Total	Normal	Abnormal	Total
Train	1237	893	2130	1371	1346	2717
Evaluation	148	105	253	150	126	276
Total	1385	998	2383	1521	1472	2993

Table 1: Data description

The EEG readings in the TUAB database use the international 10–20 placement [8] of 21 electrodes as shown in fig 3.

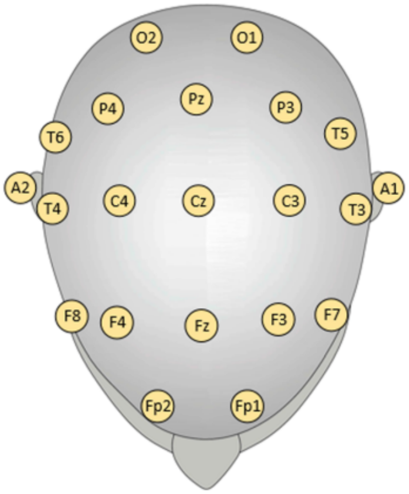


Figure 3: Electrodes' placement

We attempted as best as we could to keep our preprocessing steps in line with Gemein, et al., though they do not record their preprocessing steps in detail. They only mention downsampling all recordings to 100 Hz and clipping each recording to ±800 µV. We therefore also downsampled all recordings to 100Hz and clipped them at ±800 µV to avoid extreme values. The "A1-REF" and "A2-REF" electrodes, those placed on the patient's ears, were used as reference electrodes. No frequency-based filtering methods were applied to the data during preprocessing.

For all operations we have used the MNE and Braindecode Python libraries. MNE is a general purpose library for processing EEG data and Braindecode is an implementation of a number of popular ML architectures to be used for sorting EEG signals. It took a fair bit of trial and error to set up a Python environment in which Braindecode could successfully run and we have included the environment.yaml file used to set up the environment in Conda in the Github Repository. It should be noted that the Braindecode library divides up the training and testing datasets into compute windows. We used the recommended settings for the compute windows from the Braindecode developers and the inconsistent absolute quantities in the confusion matrices (Table S2 in the appendix) are a result of this functionality of the Braindecode library.

For our analysis we have used 2 different ConvNet models. The two architectures, called BD-Deep4 and BD-Shallow by Braindecode, are both introduced by Schirrmeister et al.[5]

These two models are characterized by a temporal-spatial convolution layer that handles the EEG input. The temporal layer performs a convolution over time, while the spatial one performs a spatial filtering with weights for all possible pairs of electrodes with filters of the preceding temporal convolution.

In the BD-Deep4 network the temporal-spatial layer is followed by another 3 blocks consisting of regular convolution and max-pooling, whereas the **BD-Shallow** network has only the temporal and spatial convolution layers. The architectures of these networks are shown in fig 4 and fig 5.

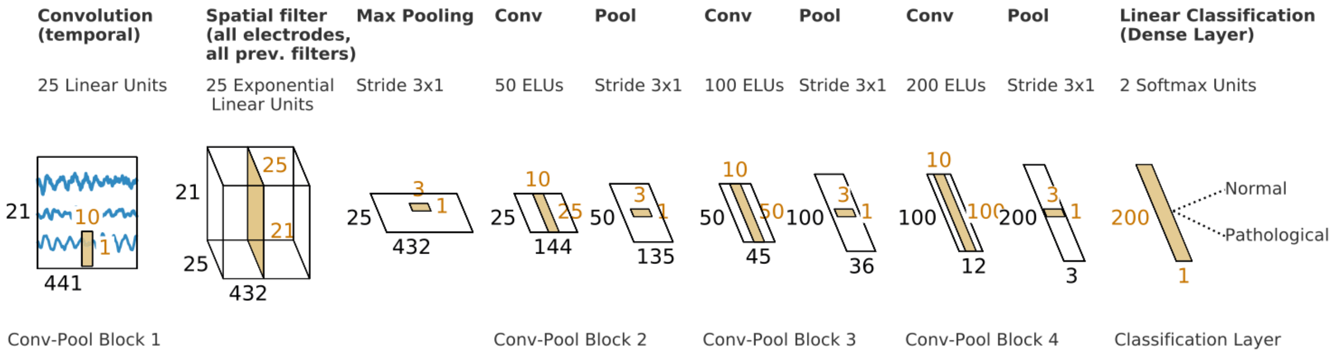


Figure 4: BD-Deep4 network architecture

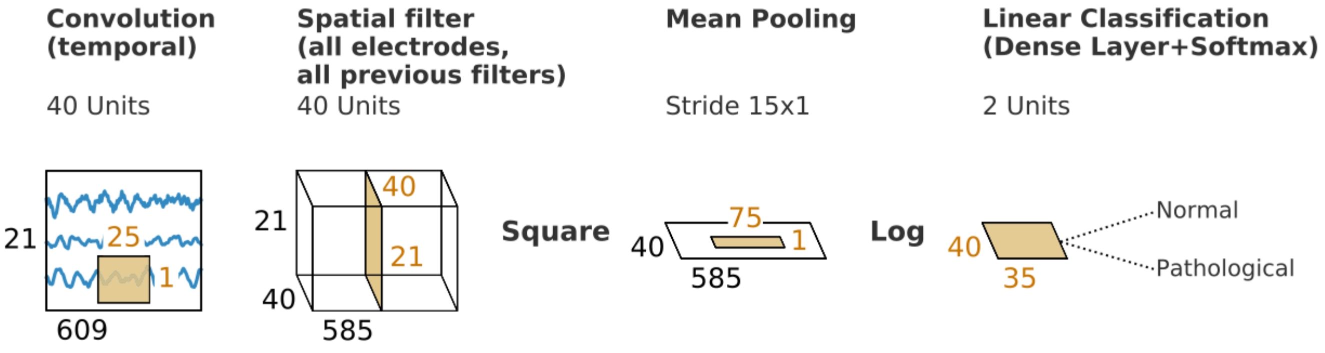


Figure 5: BD-Shallow network architecture

The third network that was used in the analysis, called **EEGNet,** was introduced by Lawhern et al [6]. This network is also convolution-based, but its architecture is different from the previous ones, see fig 6. It consists of a 2D convolutional layer half the sampling rate of the data(temporal layer), followed by a depthwise convolution layer and a separable convolution that is a combination of a depthwise convolution, which learns a temporal summary for each feature map individually, followed by a pointwise convolution, which learns how to optimally mix the feature maps together. The advantage of this architecture is that it uses a smaller number of parameters.

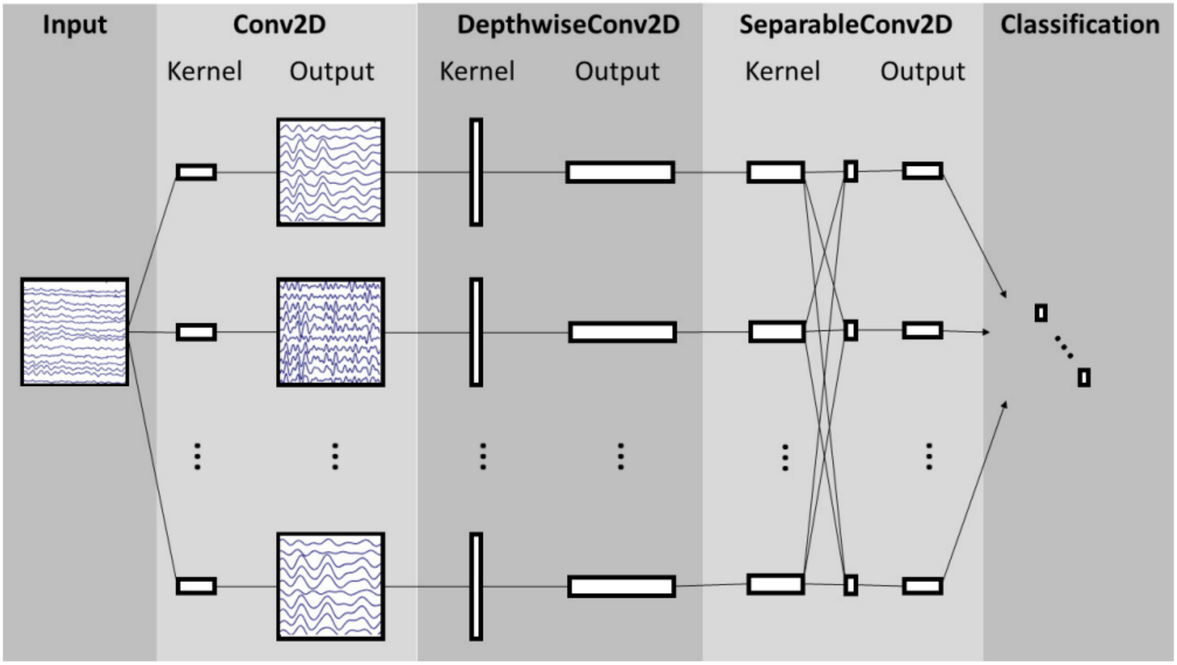


Figure 6: EEGnet network architecture

Our goal was to estimate accuracies of these models on different time windows of the EEG recordings. We have examined the windows of length: 1, 5, 11 and 20 minutes. We have also looked at both at windows that include the first minute of the recording and those that don't.

Due to limited calculation capacities we weren't able to examine the 20 minute window for all of the models so we have trained only the BD-Deep4 network in order to analyze the trend.

For the simplicity of notation we call each model by the combination of its name, whether the first minute was dropped and the length of the time window, for example deep 1+5 is a BD-Deep4 network trained on 5 minutes of data without the first minute - minutes [1,6].

The hyperparameters for all the models were chosen in agreement with those presented in Gemein et al.(2020)[3], see in Table 2

Hyperparameter	BD-Deep4	BD-Shallow	EEGNet
Batch size	16	16	16
Max epochs	35	35	35
Number of channels	21	21	21
Learning rate	1 * 0.01	0.06 * 0.01	0.11 * 0.01
Weight decay	0.5 * 0.001	0	5.8 e-07
Dropout	0.5	0.5	0.25

Table 2: Training hyperparameters

We used Adam optimizer and trained the networks for 35 epochs, as we have seen that further training doesn't improve the results, see the first lines of Appendix table S1, deep 0+1.

All the code relevant for the project is saved at the Github Repository.

Results

Time windows in min	Shallow	Deep	EEGNet
0+1: [0, 1]	78.5	79.8	76.3
1+1: [1, 2]	82.8	82.5	80.4
0+5: [0, 5]	81.5	82.3	80.8
1+5: [1, 6]	82.2	83.1	80.8
0+11: [0, 11]	82.2	82.8	80.2
1+11: [1, 12]	82.8	83.3	81.5
1+20: [1, 21]	x	82.4	x

Table 3: Models' accuracies

Detailed confusion matrices can be found in the Appendix, Table S2.

Discussion

Our data appears to be the first which directly compares the effect of using different time windows on the accuracy of ML for sorting the TUAB database. The most notable takeaways from our results are the quantitative effect of dropping the first minute of each EEG recording for a variety of different ML architectures and total time windows as well as the effect of those total time windows on the accuracy of multiple different architectures.

The best overall accuracies we observed for all three ML architectures are comparable with the highest accuracies for the test sets as reported by Roy et al. (achieved using their "1D-CNN-RNN" architecture) and only slightly below the accuracies for these same architectures as reported by Gemein, et al. The discrepancies between our results and Gemein, et al.'s results could be explained by the seed chosen, though the consistently lower accuracies just below the standard deviations for their results indicate there was likely a difference in either our preprocessing methods or hyperparameters, though we attempted to keep them as consistent as possible.

Our results validate Gemein, et al.'s assertion that the noise in the first minute of the EEG readings causes lower overall sorting accuracies across all three architectures. While this drop in accuracy is particularly notable when using a time window of one minute, dropping the first 60 seconds of each recording yields higher accuracies on average for larger time windows as well.

Enlarging the overall time window effects only a minor accuracy increase for the Shallow and Deep architectures, though the EEGNet seems to respond better to larger time windows. Due to restraints in training the ML models with larger time windows (training a single model with 20-minute time windows took multiple days) we only trained the model with the best results with the shorter time windows, the BD-Deep model with the first minute dropped. While we did observe a slight drop in accuracy as compared to the 11-minute window, it was not nearly as dramatic as what Roy, et al. observed. This is likely due to the more advanced ML models being better able to handle the larger quantities of data. While it does seem that 11 minutes still remains a more ideal time window than 20 minutes, researchers using BD-Deep and other advanced ML models should be unconcerned about significant losses in accuracy resulting from using larger time windows.

Based on our results, a time window of one minute should be enough time to obtain accuracies comparable to larger time windows. While increasing the amount of data used for training is generally desirable, our results indicate that the gains it provides are minor. This is particularly relevant to those with limited computational resources since comparable performance can be obtained by processing a far smaller amount of data.

Additionally, given the sharp rise in accuracy resulting from dropping the first minute of each reading, especially at lower total time windows, the practice of dropping the first minute should be recommended as standard for anyone using the TUAB database. This is especially true when choosing to use smaller overall time windows. While Gemein, et al. did this based on their own qualitative observations of the data, other papers using the database have not mentioned doing so and it is unclear whether this property of the TUAB data is known. Our quantitative demonstration of the advantage of this practice should cement it as standard.

References

[1] J. W. C. Medithe and U. R. Nelakuditi, “Study of normal and abnormal EEG,” Jan. 2016. Accessed: Sep. 21, 2022. [Online]. Available: http://dx.doi.org/10.1109/icaccs.2016.7586341

[2] S. Lopez, G. Suarez, D. Jungreis, I. Obeid, and J. Picone, “Automated identification of abnormal adult EEGs,” Neural Engineering Consortium, Temple University.

[3] L. A. W. Gemein *et al.*, “Machine-learning-based diagnostics of EEG pathology,” *NeuroImage*, vol. 220, p. 117021, Oct. 2020, doi: 10.1016/j.neuroimage.2020.117021.

[4] S. Roy, I. Kiral-Kornek, and S. Harrer, “Deep Learning Enabled Automatic Abnormal EEG Identification,” Jul. 2018. Accessed: Sep. 21, 2022. [Online]. Available: http://dx.doi.org/10.1109/embc.2018.8512756

[5] R. T. Schirrmeister *et al.*, “Deep learning with convolutional neural networks for EEG decoding and visualization,” *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, Aug. 2017, doi: 10.1002/hbm.23730.

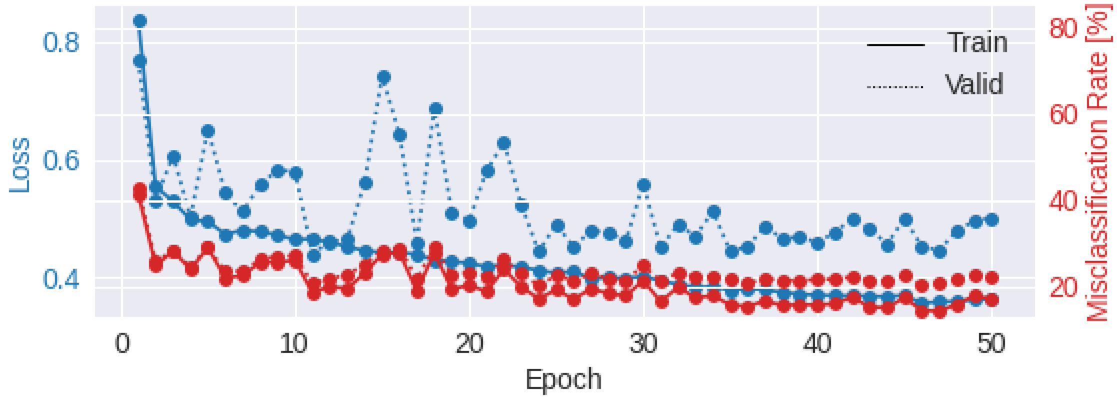
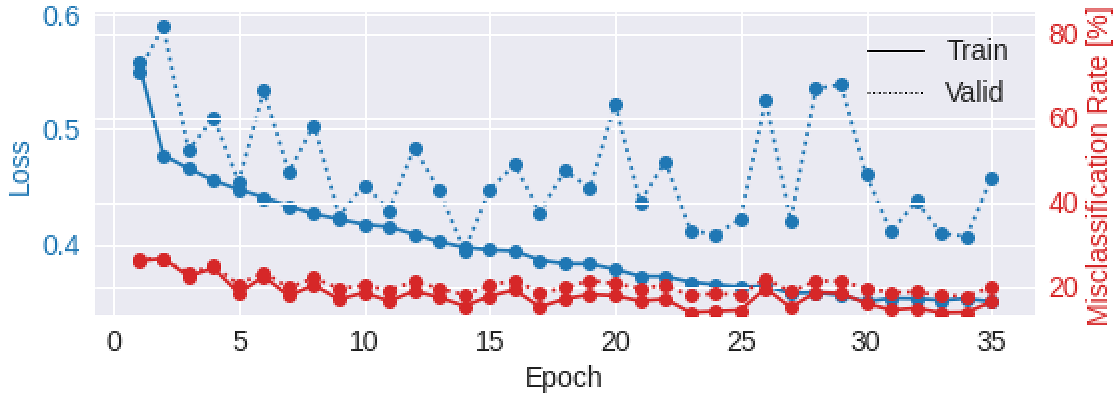
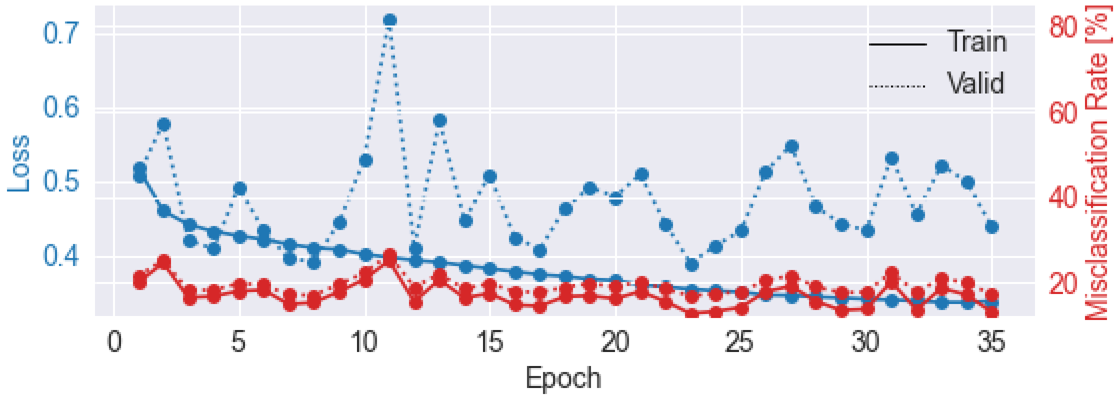
[6] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, “EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces,” *Journal of Neural Engineering*, vol. 15, no. 5, p. 056013, Jul. 2018, doi: 10.1088/1741-2552/aace8c.

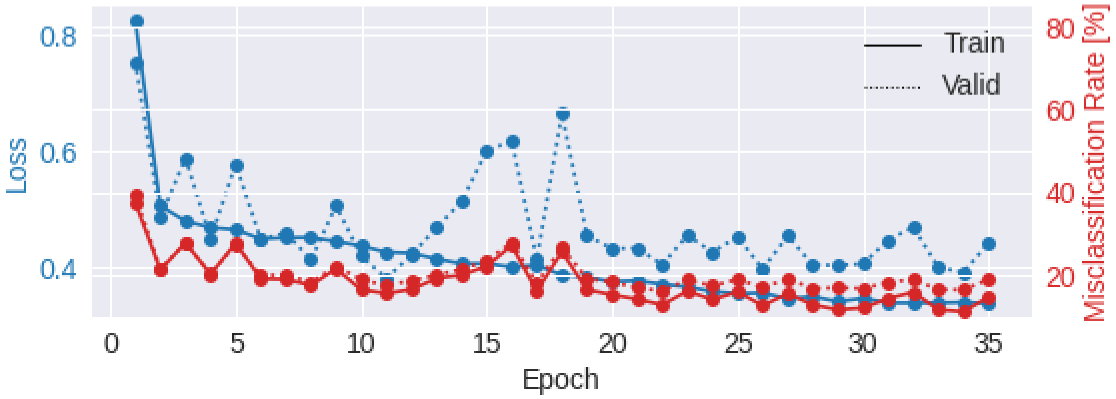
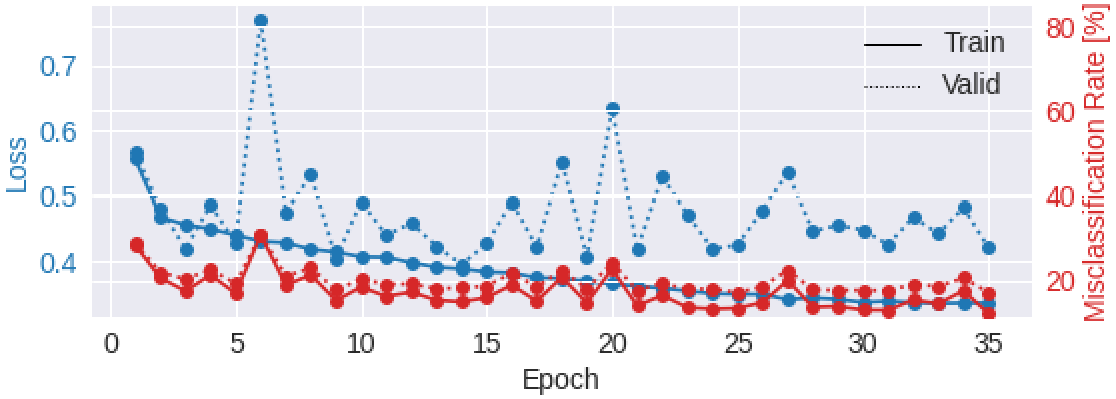
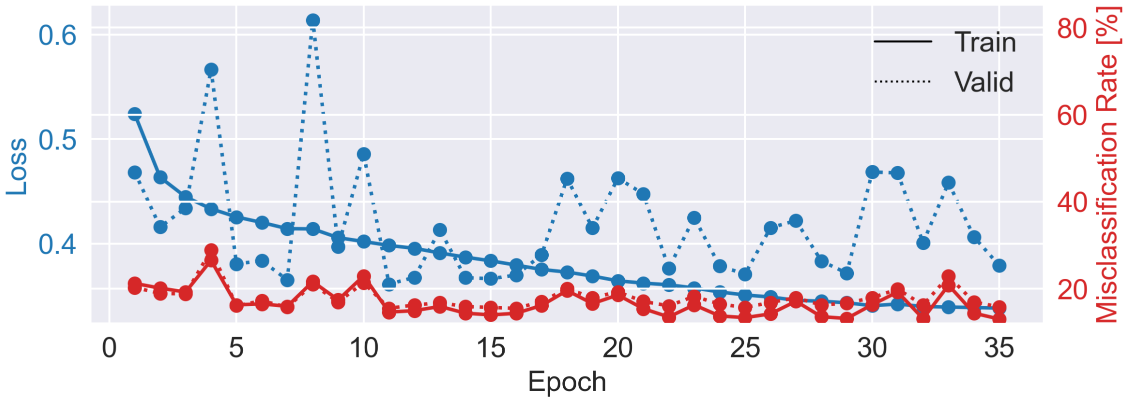
[7] J. Picone, “Temple University EEG Corpus,” *Downloads*. https://isip.piconepress.com/projects/tuh_eeg/html/downloads.shtml (accessed Sep. 21, 2022).

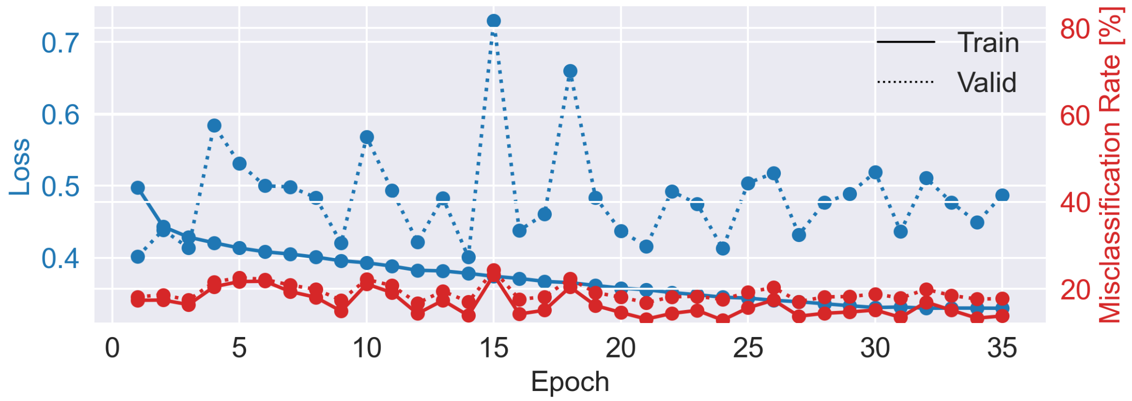
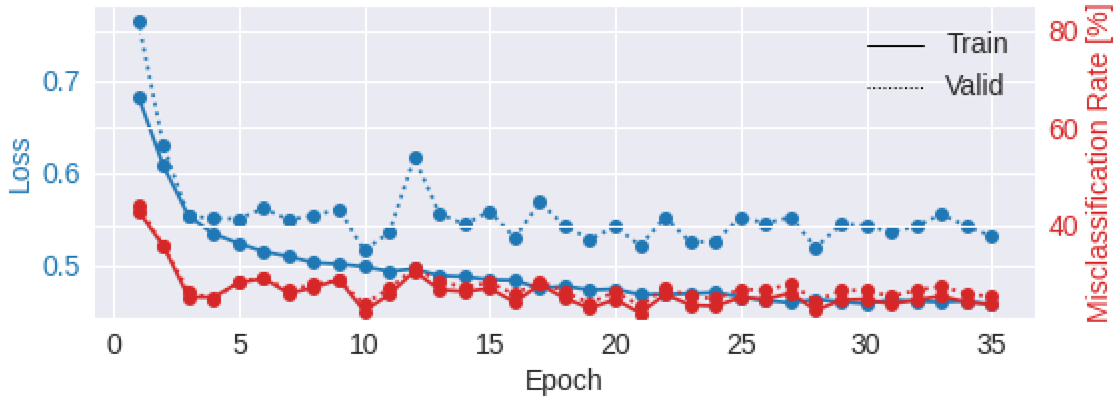
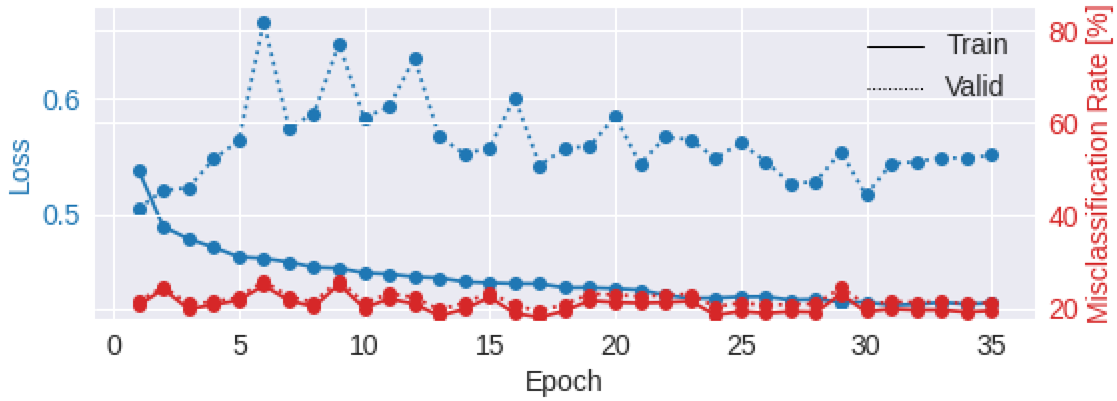
[8] “Report of the committee on methods of clinical examination in electroencephalography: 1957,” *Electroencephalography and Clinical Neurophysiology*, vol. 10, no. 2, pp. 370–375, 1958, doi: https://doi.org/10.1016/0013-4694(58)90053-1.

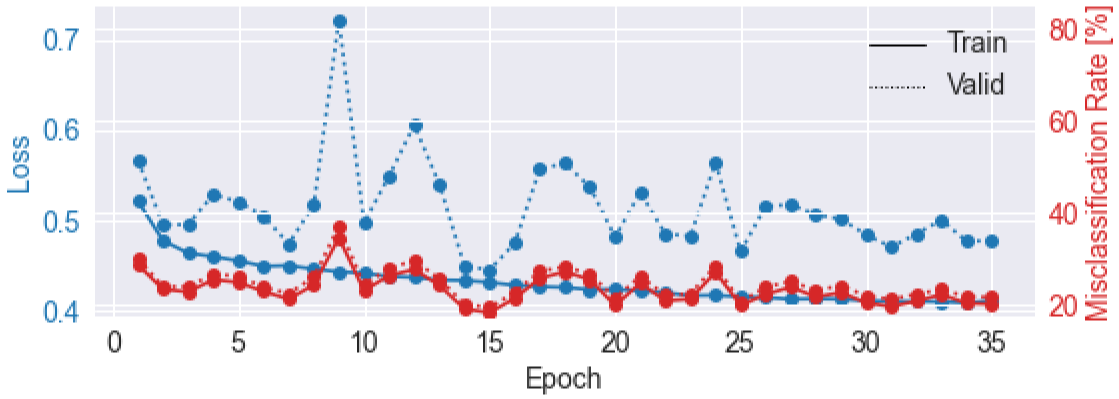
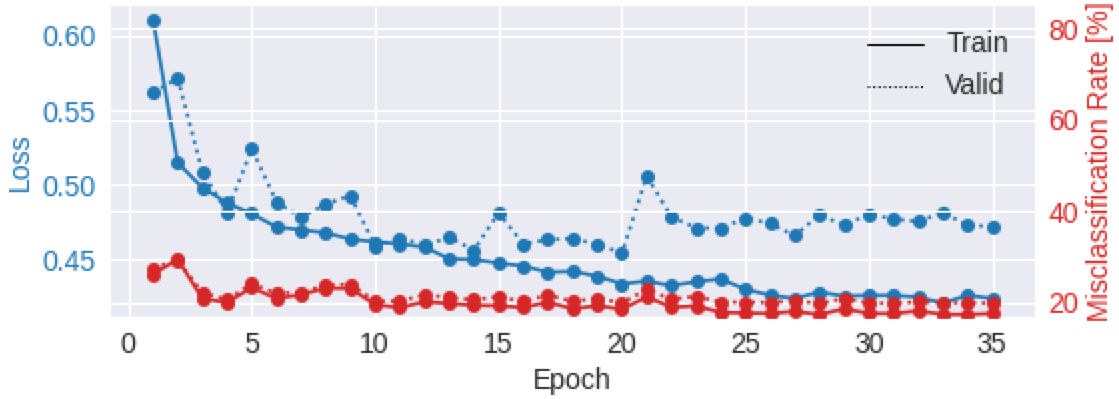
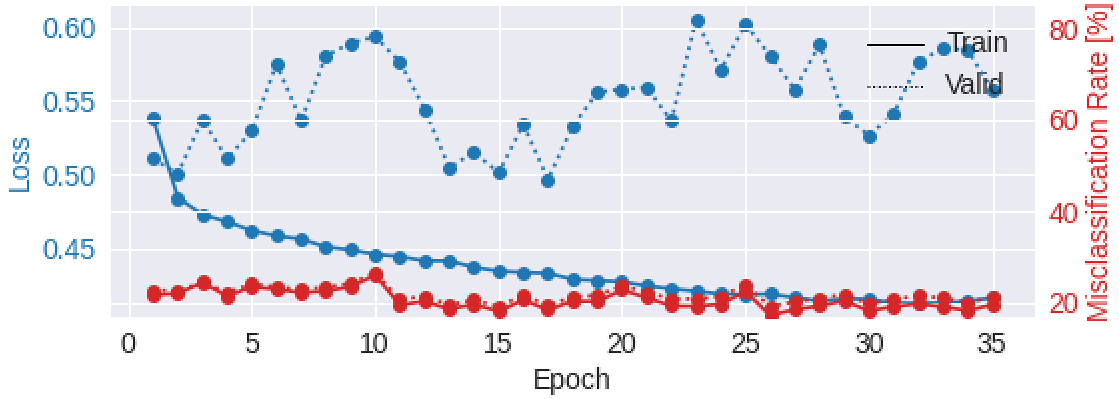
Appendix

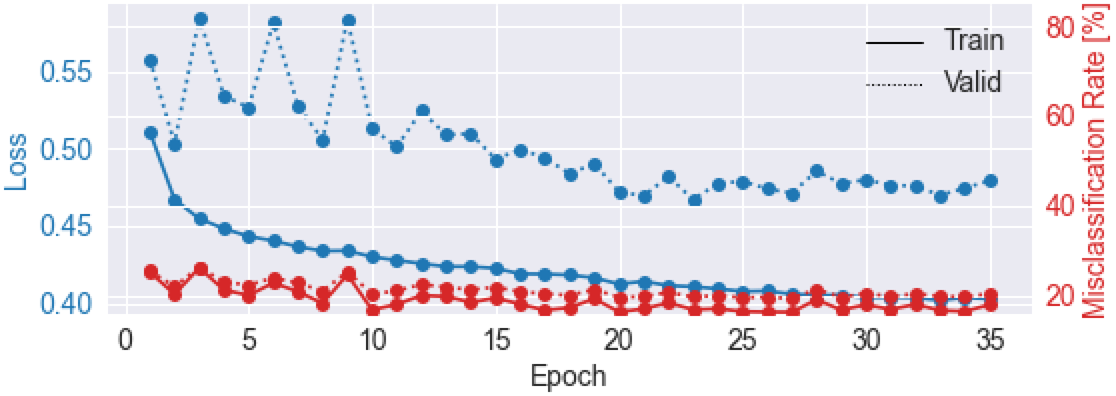
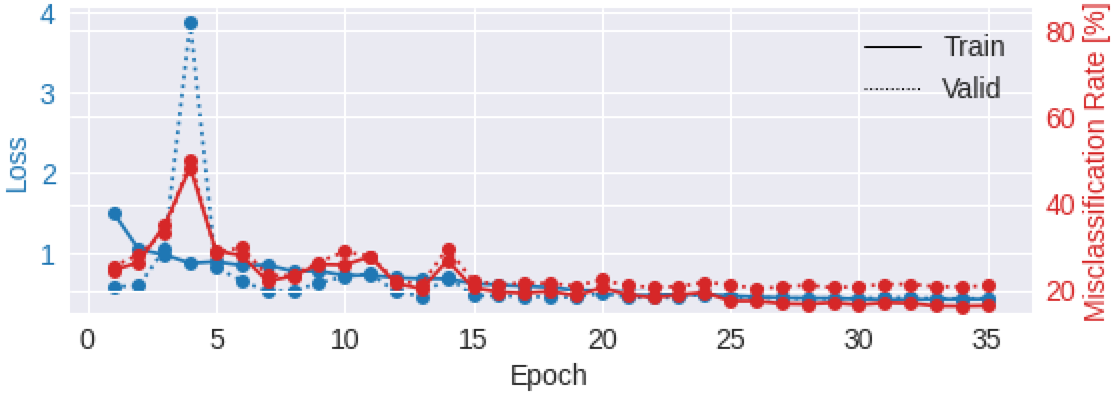
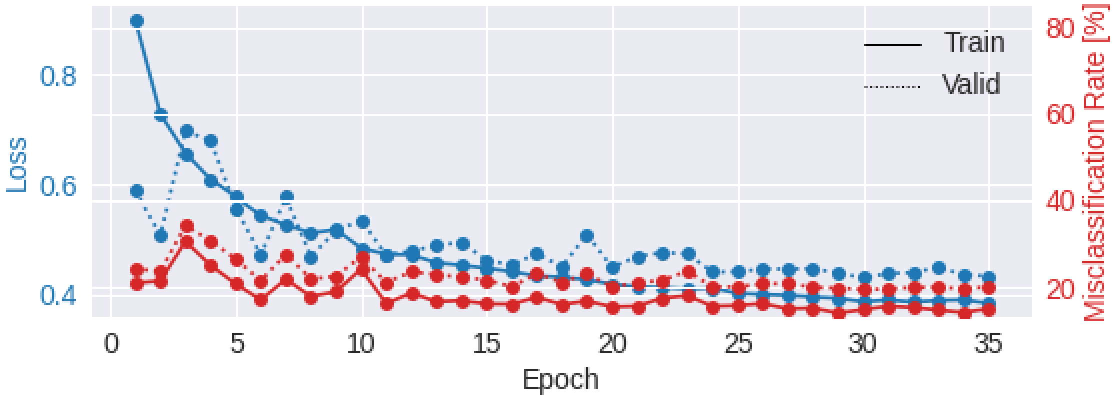
Training history

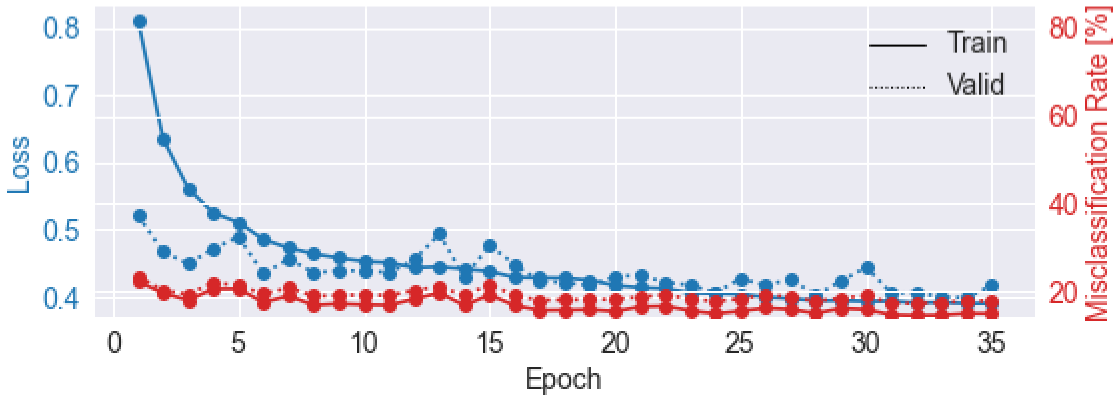
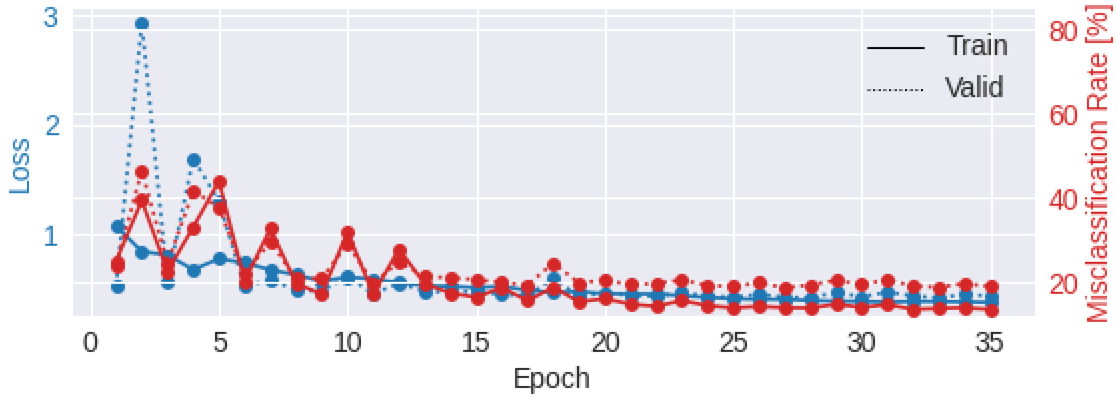
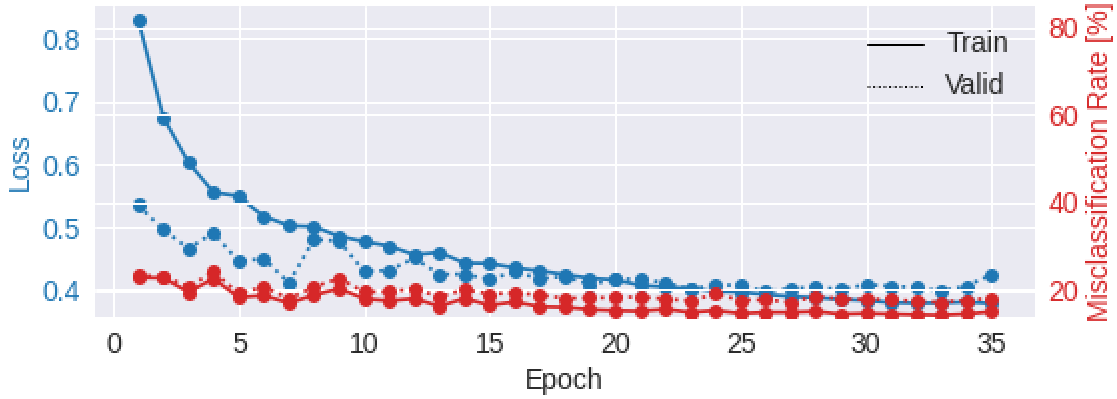
Deep 0+1	
Deep 0+5	
Deep 0+11	

Deep 1+1	
Deep 1+5	
Deep 1+11	

Deep 1+20	
EEGnet 0+1	
EEGnet 0+5	

EEGnet 0+11	
EEGnet 1+1	
EEGnet 1+5	

EEGnet 1+11	
Shallow 0+1	
Shallow 0+5	

Shallow 0+11	
Shallow 1+1	
Shallow 1+5	

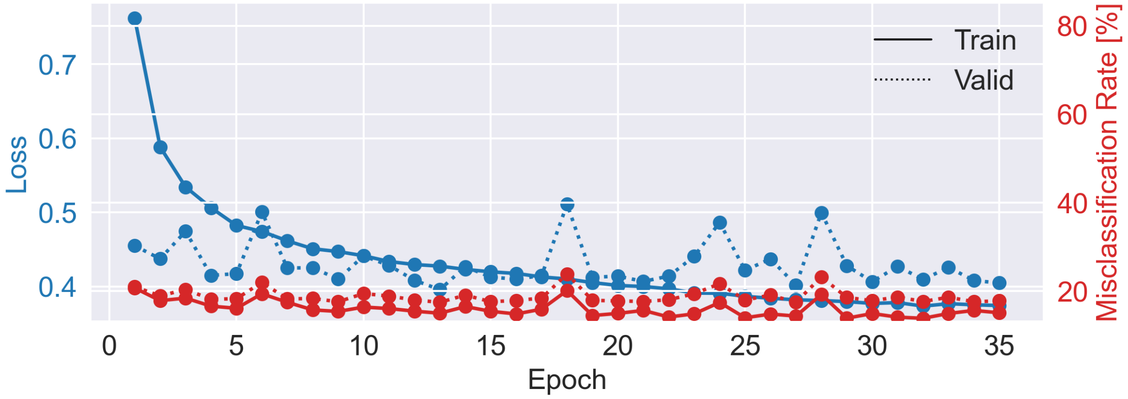
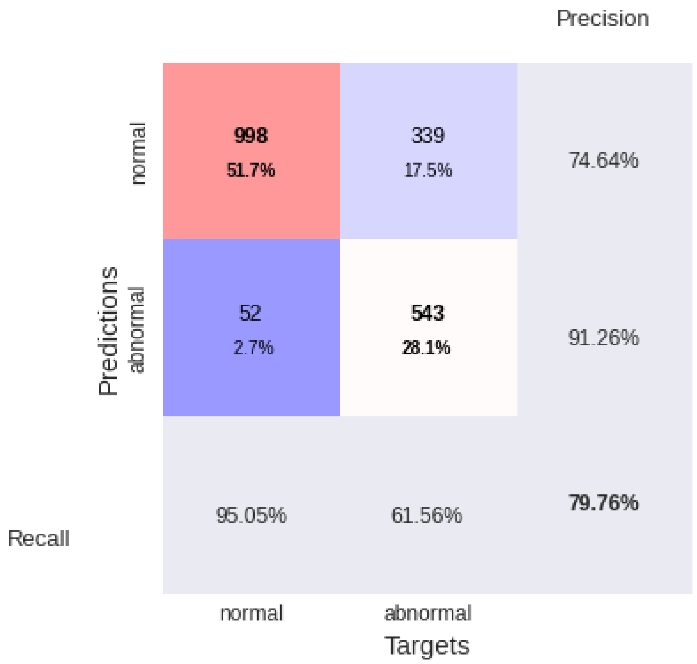
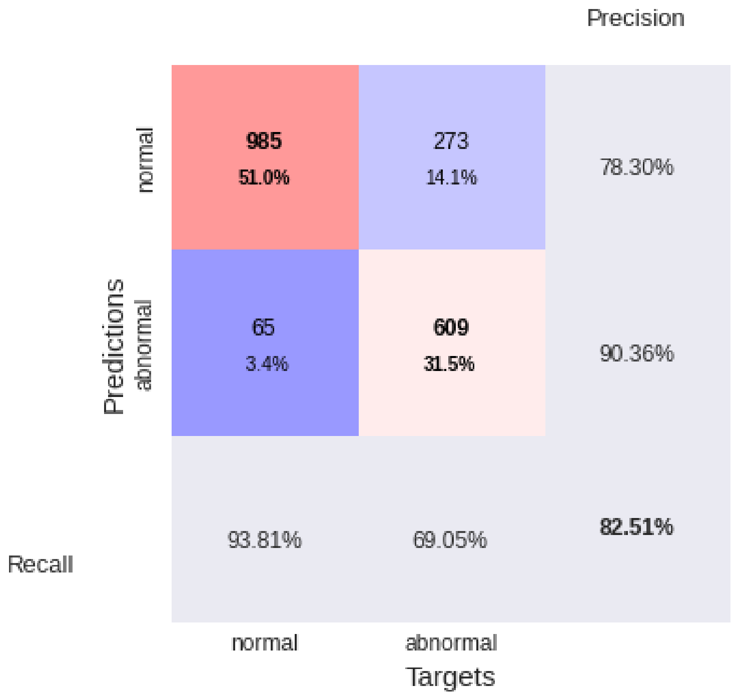
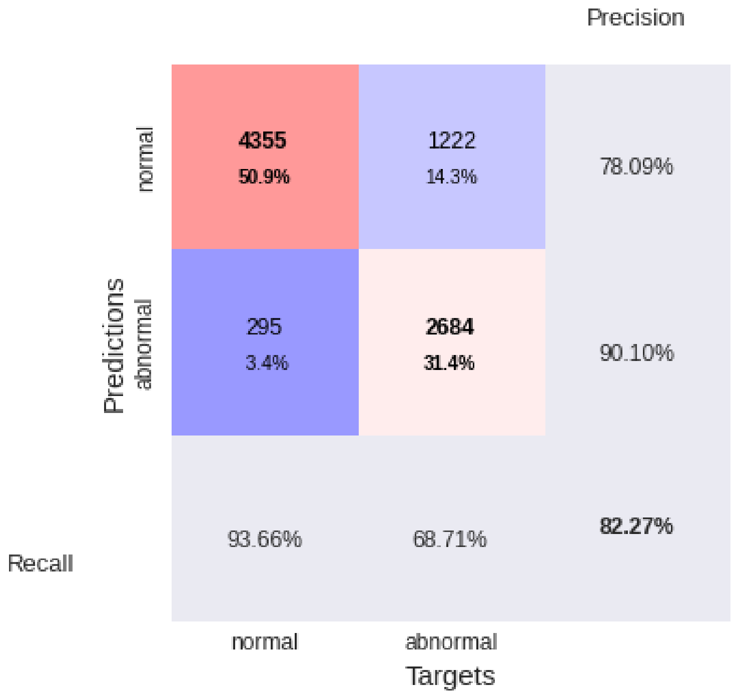
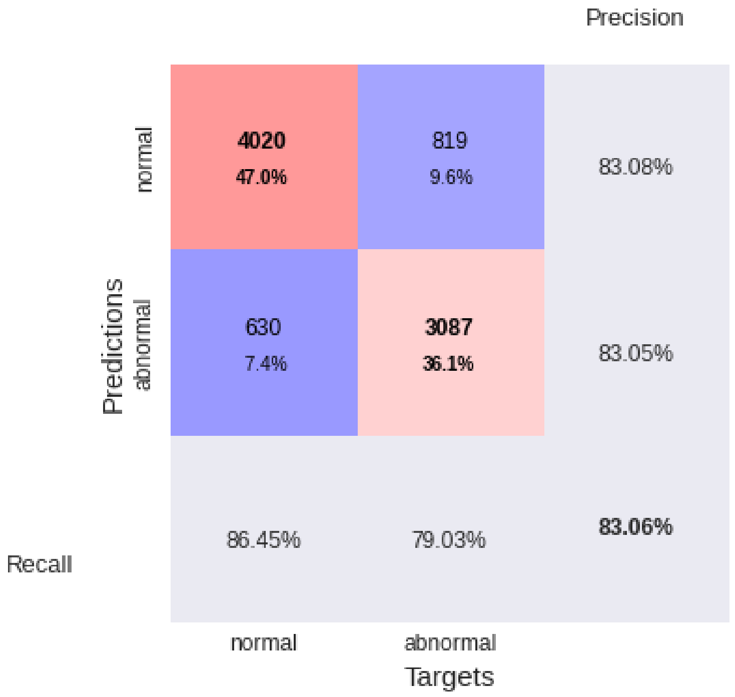
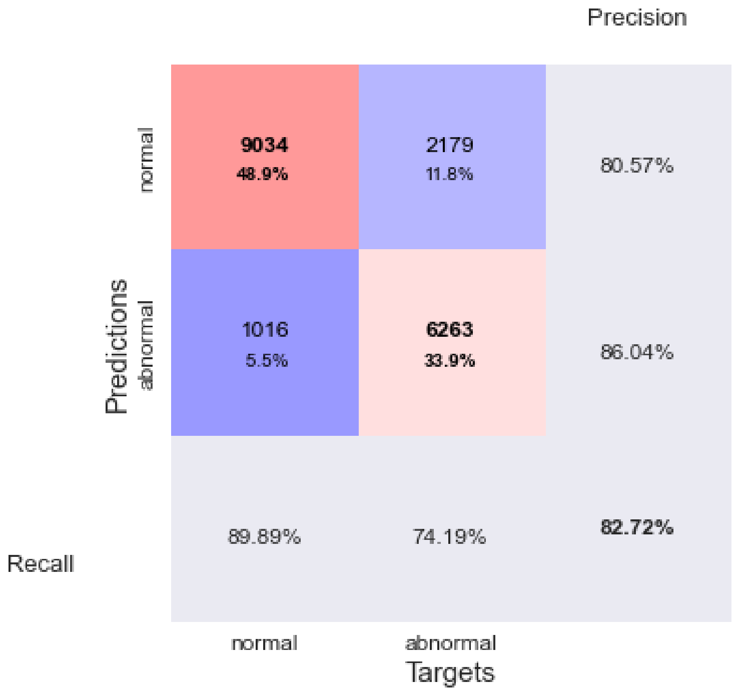
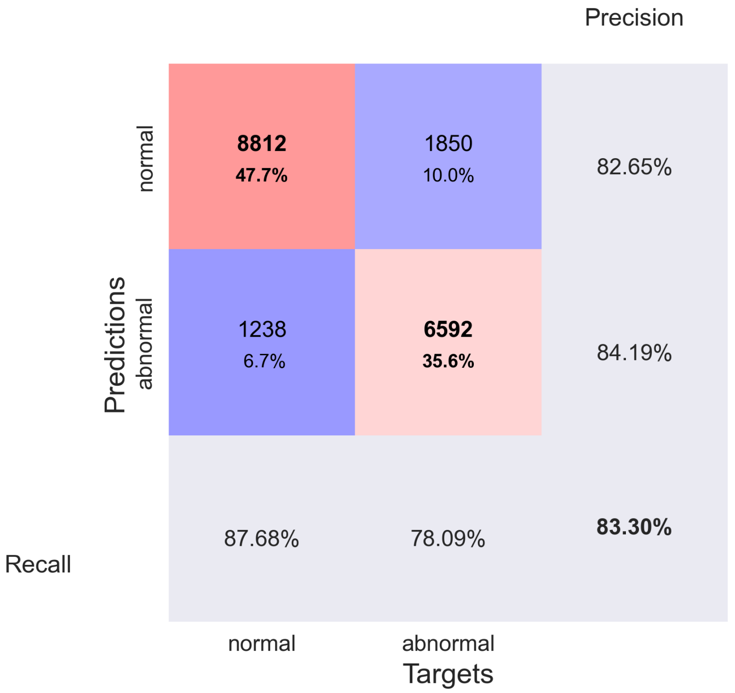
Shallow 1+11	

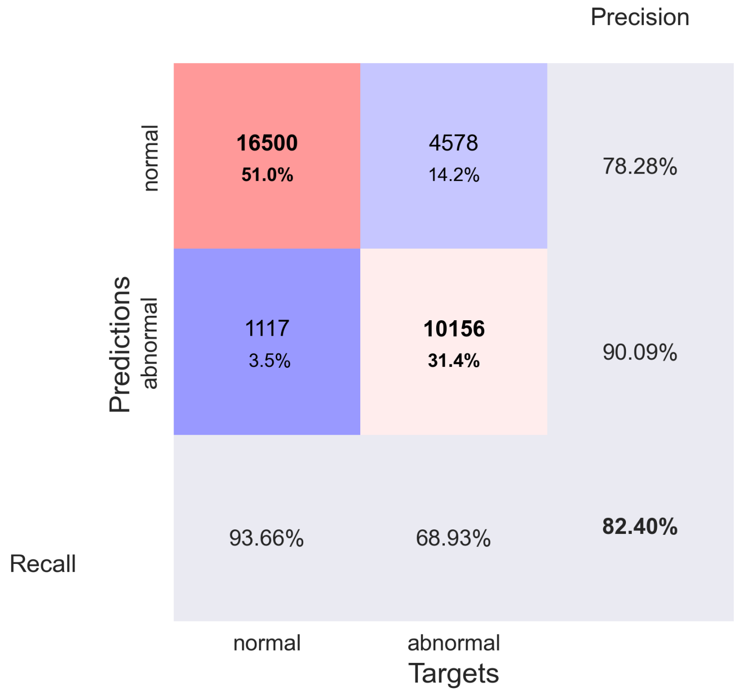
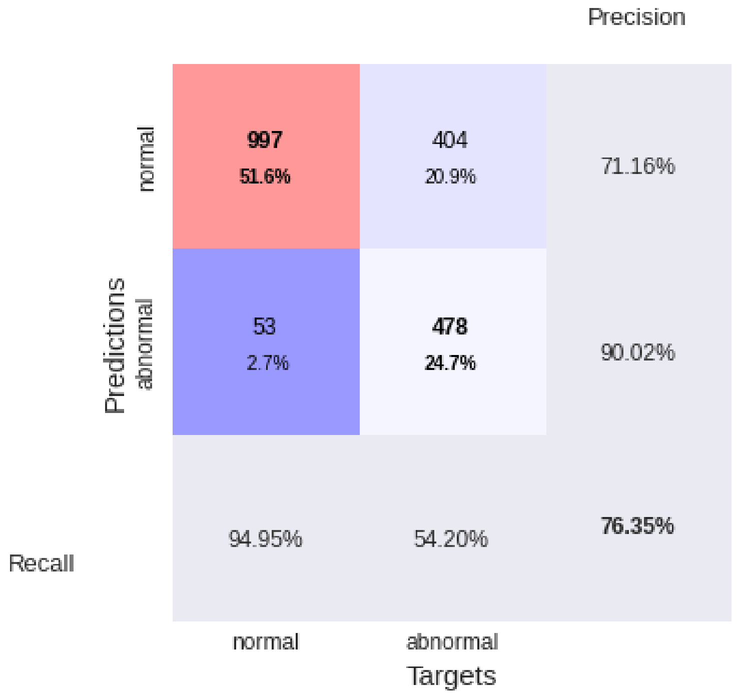
Table S1: Training history of all the models

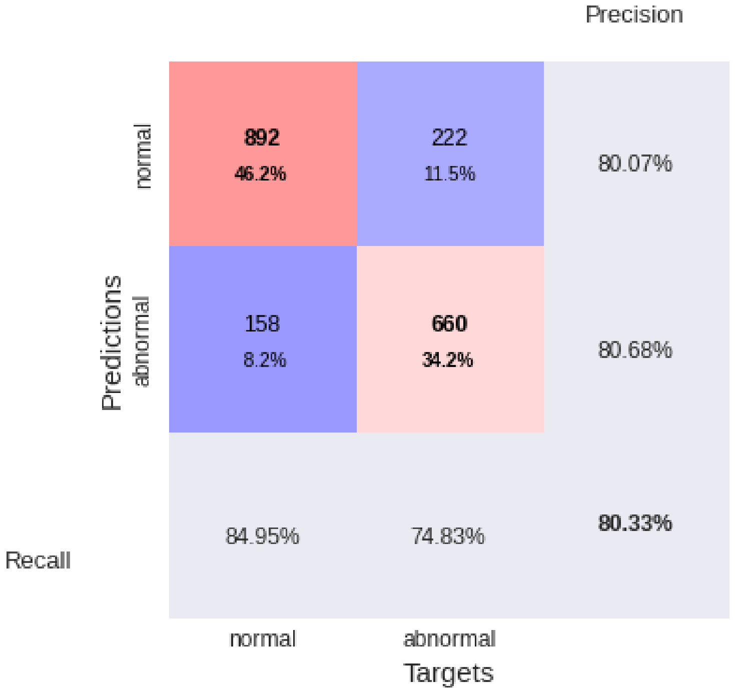
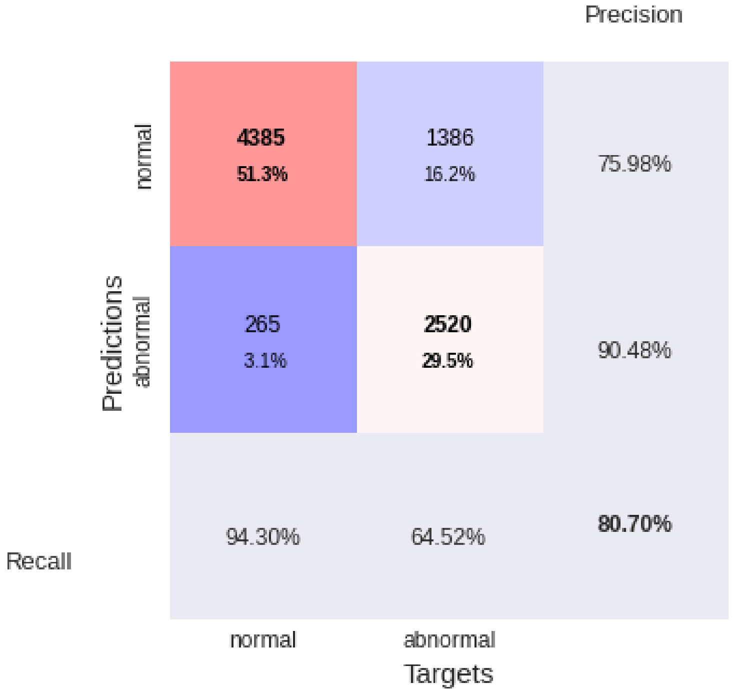
Confusion matrices

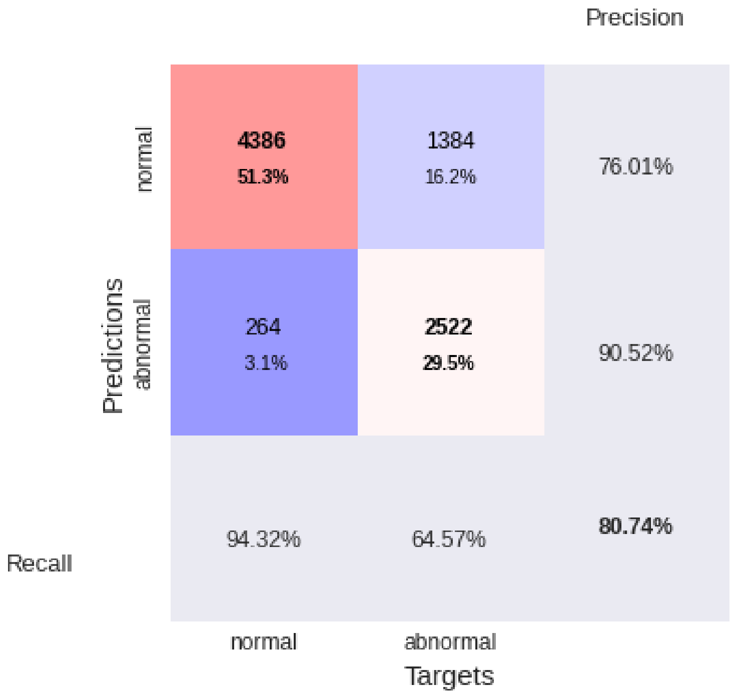
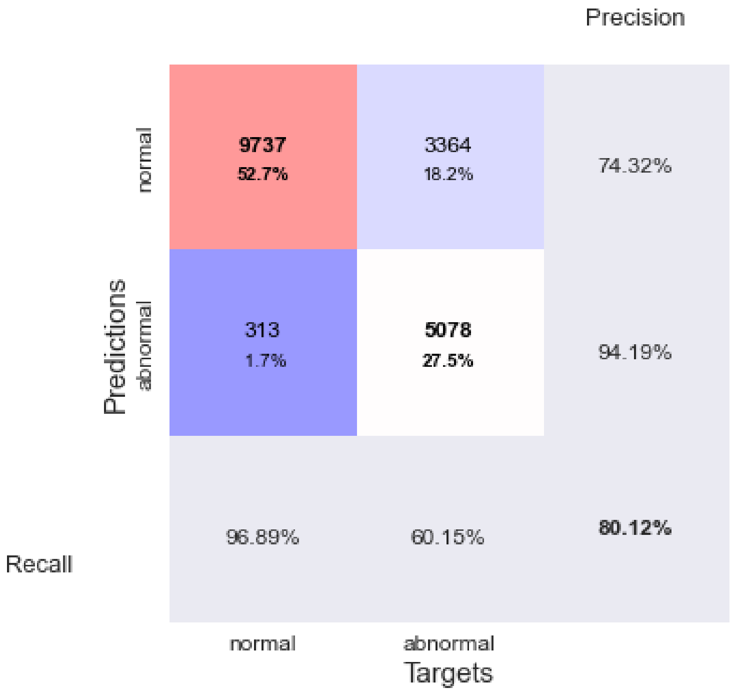
<table>
<tr>
<td>Deep 0+1</td>
<td>
</td>
</tr>
<tr>
<td>Deep 1+1</td>
<td>
</td>
</tr>
</table>

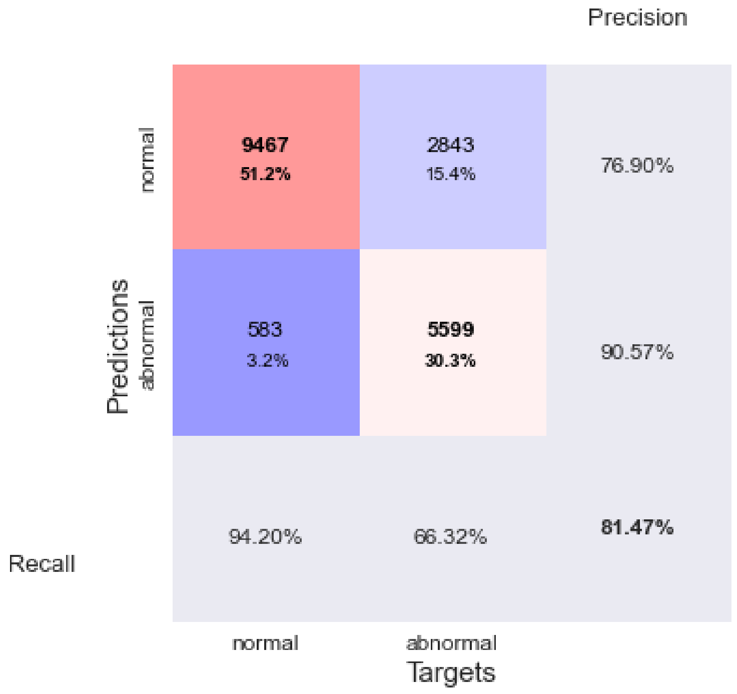
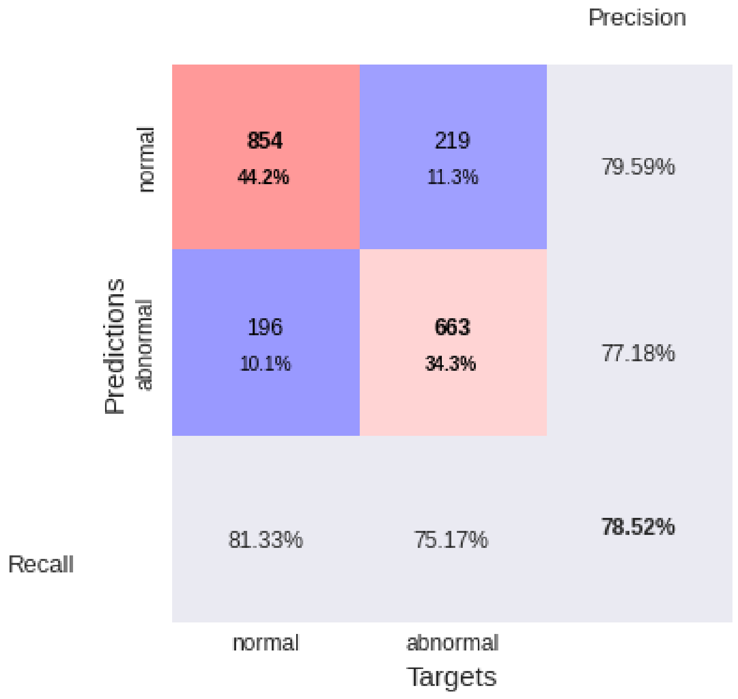
Deep 0+5	
Deep 1+5	

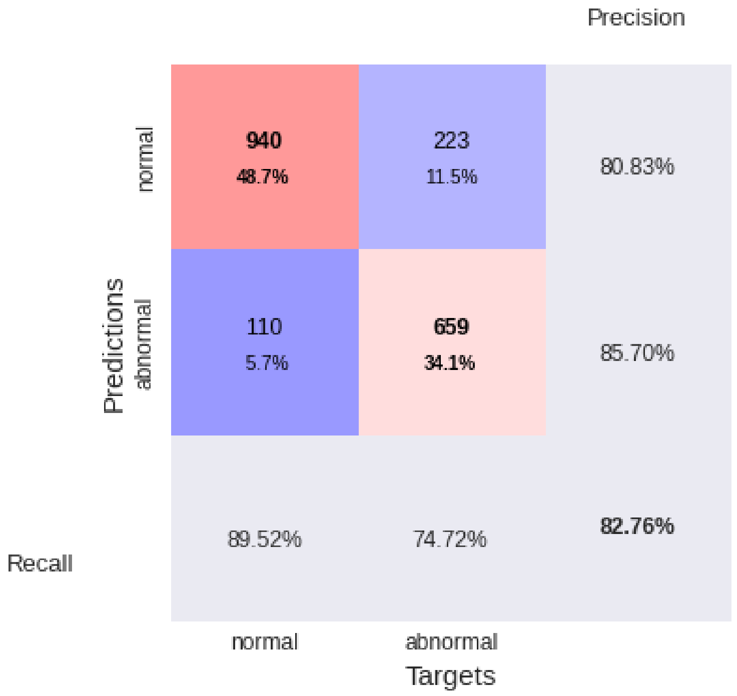
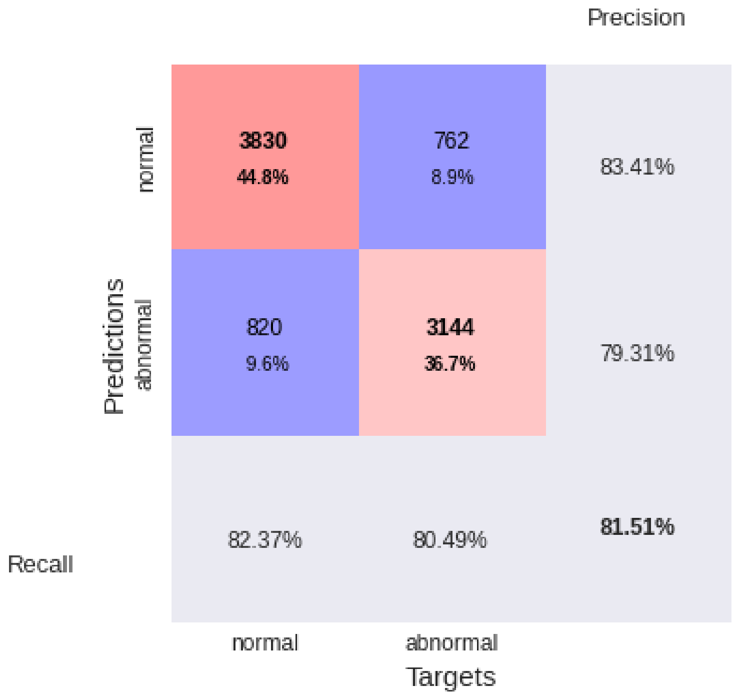
Deep 0+11	
Deep 1+11	

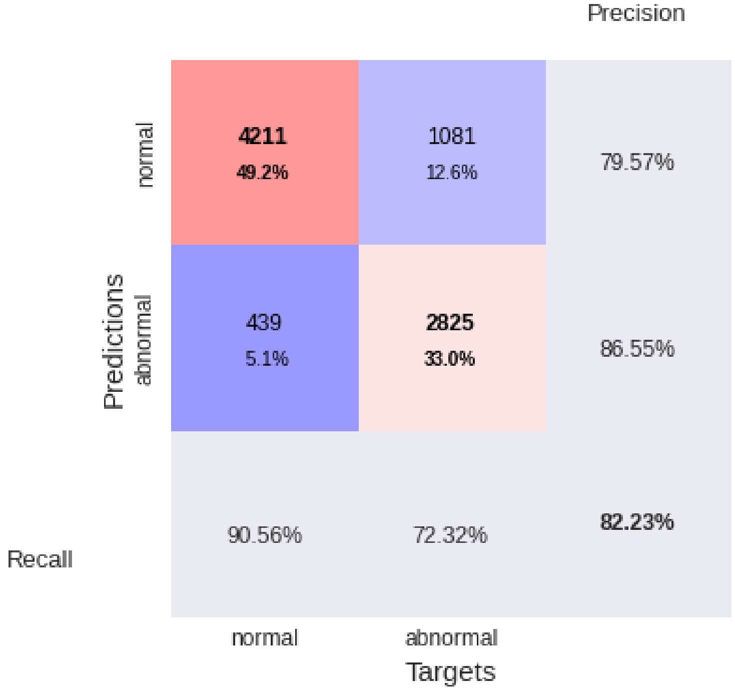
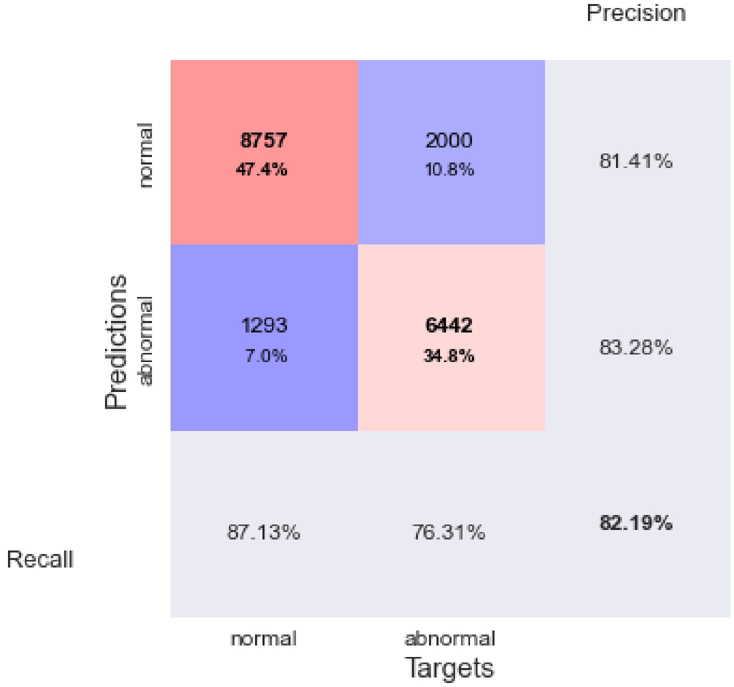
Deep 1+20	
EEGNet 0+1	

EEGNet 1+1	
EEGNet 0+5	

EEGNet 1+5	
EEGNet 0+11	

EEGNet 1+11	
Shallow 0+1	

Shallow 1+1	
Shallow 0+5	

<table>
<tr>
<td>Shallow 1+5</td>
<td>

<table>
<tr><th></th><th></th><th>normal</th><th>abnormal</th><th>Precision</th></tr>
<tr><td rowspan="2">Predictions</td><td>normal</td><td><b>4211</b>
<b>49.2%</b></td><td>1081
12.6%</td><td>79.57%</td></tr>
<tr><td>abnormal</td><td>439
5.1%</td><td><b>2825</b>
<b>33.0%</b></td><td>86.55%</td></tr>
<tr><td>Recall</td><td></td><td>90.56%</td><td>72.32%</td><td><b>82.23%</b></td></tr>
<tr><td></td><td></td><td colspan="2">Targets</td><td></td></tr>
</table>
</td>
</tr>
<tr>
<td>Shallow 0+11</td>
<td>

<table>
<tr><th></th><th></th><th>normal</th><th>abnormal</th><th>Precision</th></tr>
<tr><td rowspan="2">Predictions</td><td>normal</td><td><b>8757</b>
<b>47.4%</b></td><td>2000
10.8%</td><td>81.41%</td></tr>
<tr><td>abnormal</td><td>1293
7.0%</td><td><b>6442</b>
<b>34.8%</b></td><td>83.28%</td></tr>
<tr><td>Recall</td><td></td><td>87.13%</td><td>76.31%</td><td><b>82.19%</b></td></tr>
<tr><td></td><td></td><td colspan="2">Targets</td><td></td></tr>
</table>
</td>
</tr>
</table>

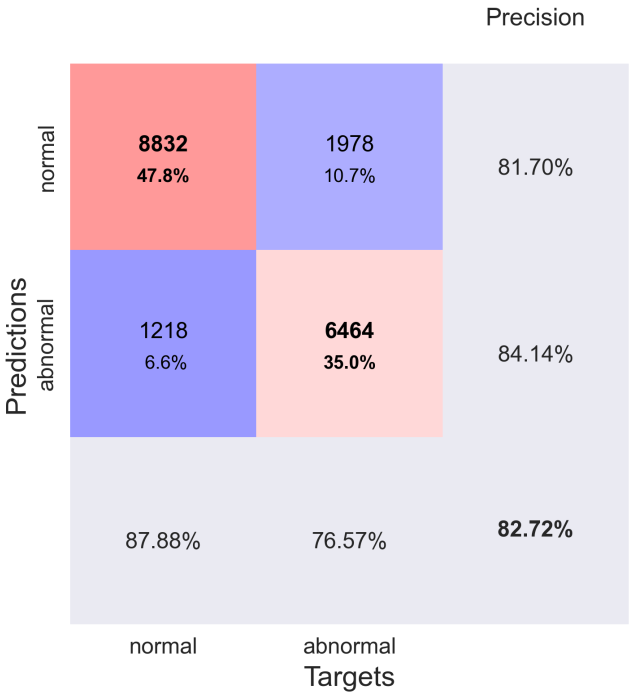
Shallow 1+11	

Table S2 : Confusion matrices for all the models