IdMind – EEG biometrics

A flexible multi-model system for biometric authentication using electroencephalography.

Michele Romani
EEMCS, Interaction Technology
University of Twente
Enschede, Overijssel, Netherlands
m.romani@student.utwente.nl

Ricky Walsh
EEMCS, Data Science
University of Twente
Enschede, Overijssel, Netherlands
r.walsh@student.utwente.nl

ABSTRACT

The inevitable entry of Brain-Computer Interfaces into the market has opened the way for many new opportunities and challenges to be explored. One such opportunity is the use of electroencephalography (EEG) devices for biometric authentication. This study focused on solving the scalability problem of an enterprise authentication system by developing a flexible multi-model system that could handle the addition of extra participants without retraining the entire system. The solution combines the use of a convolutional autoencoder to generalise features from the EEG signal of 20 participants and then feed the encoded features to a one-vs-all SVM classifier for identification.

In the first part, the problem and the context are introduced, followed by the description of the dataset used. Then follows the description of the methods applied: pre-processing of the data, the architecture of the autoencoder and the SVM classifier, and then the optimisation and evaluation of the models. In the final part the results are discussed and compared with related works, and finally conclusions are drawn around possible alternatives to carry on this study in the future.

KEYWORDS

eeg, biometric, autoencoder, svm, identification, cae, convolution

1 INTRODUCTION

From the first computer mouse to voice commands and fingerprint authentication, the ease with which we communicate with technology has gradually increased. Recent developments have made technologies like EEG and fNIRS very affordable and EEG has become a popular choice in the study field of Brain-Computer Interfaces thanks to its non-invasive setup and its temporal resolution in the millisecond range. The idea of this interaction between a human and a computer at the speed of thought and using only one's mind has inspired a myriad of new research topics for the application of BCI devices outside of the medical field. One of these topics is the identification of users based on the patterns of their brain activity.

Recent studies suggest unique brain wave patterns in everyone, therefore like other biometric traits they could be used as a mean of authentication [1]. There are many issues outstanding with current biometric identifiers such as fingerprint, iris, or face recognition. In particular, spoofing, where an attacker could, for example, present a fake fingerprint or face mask in order to fool the system has received much interest [2]–[4]. The benefit of the use of EEG signals here is that they are private so cannot be captured at a distance [5]. There is also some evidence that the use of EEG may prevent an attacker from forcing a user to authenticate, since the stress signals detected in the EEG could result in a denial of access [6].

In most studies in this area thus far, the focus has been on discriminating between a defined set of participants. The system is trained on 20 participants, for example, who are represented by 20 classes in the classifier. The system is then tested on separate data from those same 20 participants. It is not difficult to envision a situation in a real-world application where a new person must be added to the identification system, for example if a new employee of a company needs to be added to their authentication system. In this case, re-training the whole system would be affected by a critical scalability problem as the number of employees increases.

This problem is posed in [7] as a challenge which remains open in EEG-based biometric systems and is more relevant for the identification task (given an EEG signal, identify the person) rather than the verification task (given an EEG signal, is this person X?). In summary, the goal was to build an identification model trained on a set of participants in the attempt to capture general information from the EEG signal which makes each person broadly unique, rather than discriminating just from the other participants. The model was then tested by adding an unseen participant as an extra class and assessing the classification performance.

2 RELATED WORK

The idea of using EEG signals as a biometric identifier is not a new idea, and in fact was proposed as early as 1980 [8]. A seminal work was published in 1999 making progress in investigating the feasibility of such an approach [1]. In this study, auto-regressive (AR) parameters were estimated from the EEG signals and used in a Learning Vector Quantiser to identify individuals out of a set of four participants. The accuracies of 72-84% supported the idea that EEG signals contain genetic information [1], [5]. This lay the groundwork for further research which has accelerated over the last

decade thanks to improved signal processing & classification techniques and advancing technology for capturing EEG signals [9].

Traditionally this task was split into two distinct stages: feature extraction and classification. Examples of feature extraction techniques employed in similar studies include AR modelling [1], [5], [10], [11], Power Spectral Density (PSD) [11]–[13], and Wavelet Transform (WT) [14]. The extracted features are then used to train a classifier. Again, many different models have been used for the classification task [7], including Linear Discriminant Analysis [10], [12], Artificial Neural Networks (ANN) [15], [16] or simply using distance functions (Euclidean or Manhattan, for example) [11], [12]. More recently however, there has been a wider study of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) which have the potential to manage both the feature extraction & classification tasks. [17], [18] have applied CNNs for the task, while [19], [20] combined CNNs with Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) RNNs. The application of these deep learning techniques has demonstrated the possibility of remarkable results, in many cases exceeding 99% classification accuracy [7].

Recently, criticism has emerged which points out flaws in the methodologies of many of the studies discussed above. Because of the difficulties in running experiments and gathering data, the data used in most of the studies discussed above contains only one recording session per participant. However, this means that the reported accuracies do not reflect real-world conditions, since as Campisi & Rocca point out, the EEG signals may "depend on the time of the day as well as on the time of the year they are acquired, thus reflecting both circadian and seasonal influences respectively."[5, p. 789] Furthermore, EEG signals can be affected by both physiological artifacts and exogenous noise caused by the hardware which "may be significantly different between different acquisition sessions"[11, p. 165]. This calls into question the reliability of the high accuracy estimates given in most related studies, and in fact [21] found a drop in accuracy from 98-99% for predictions using data from the same session to 66-72% when predicting across multiple sessions for the ten participants.

In all the aforementioned studies, the focus has been on discriminating between a defined set of participants. Data from all participants are used to both train *and* test the identification system. Moreover, the number of participants involved in these studies is often quite small due to the cost and effort involved to gather data. In fact, among the many studies reviewed by [7], 81% used data from less than 50 participants. This leads to questions over the true uniqueness of the EEG signals, since the deep learning methods have the potential to learn characteristics that discriminate between the limited set of participants but are not unique more generally. In particular, this means that each time a new participant needs to be added to the identification system, the whole system needs to be re-trained, which is impractical.

3 DATA

The dataset selected for the study follows specific criteria that were carefully chosen to support the validity of the model. EEG signal comes in the form of time-series and the experimental conditions in which they are recorded as well as the number of participants and sessions recorded are relevant for the study purpose. In general, the greater the number of participants, the better the model will generalise. However, due to the technical difficulties, EEG experiments usually involve fewer participants compared to other studies. In addition, as highlighted by [5] and [21], to ensure the uniqueness and the utility of EEG as a biometric signature, the results should be consistent over different sessions and over time.
The dataset used for this study is called EEG Dataset of 7-day motor imagery BCI [22], collected for a study on motor imagery classification and it contains data from 20 healthy subjects (11 males, mean age 23) over a time span of two weeks. The participants were asked to attend 7 sessions over the course of the experiment, each session lasting around 40 minutes and organized in 6 runs with short breaks in between. During each run, participants were instructed to perform 4 different motor imagery tasks, performed 10 times and in random order, for a total of 40 trials and each trial lasting 9 seconds. instructions were given in a symbol format using a cross to mark the beginning of the trial and then 4 arrows, respectively corresponding to a motor imagery task as shown in figure 1.

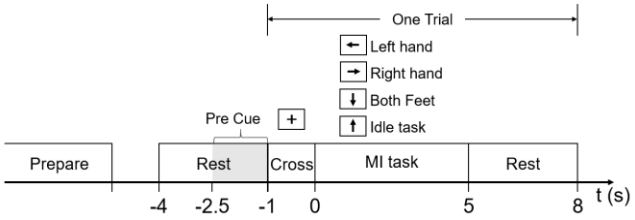

Figure 1 – Scheme of the experimental protocol.

The EEG recordings were performed using a Synamps2[1] system with 64 channels and a sampling frequency of 500Hz and electrodes were selected in two different layouts according to the 10-20 system, including either 26 or 41 channels placed above the motor cortex, represented in Figure 2.

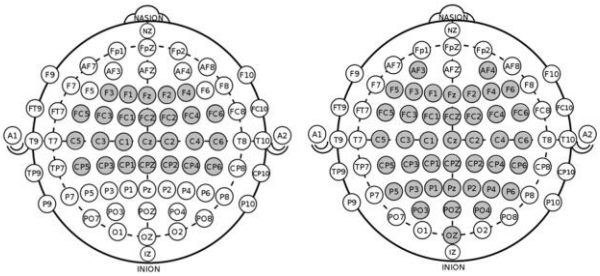

Figure 2 - The two possible layouts for the experiment.

[1] Neuroscan, Inc.

For our study, the raw dataset of 50GB was too large to be easily managed, so to simplify both the handling and later the model training we extracted a subset with the following method: Right Hand task was arbitrarily chosen, then for each trial only the 5 seconds span (see Figure 1) of the task was considered. This resulted in a dataset that was ~1/4 the size of the original dataset. The dataset included all the 26 channels in common between the two possible layouts, to standardise the study between all participants.

A further split for training purposes divided the dataset in the subsequent parts: a training dataset comprising 79.2% of the data; a validation dataset comprising 15.8% of the data and explicitly chosen to be a separate run within each session for each participant rather than a random selection of samples, to minimise the possibility of temporal correlations and ensure generalisability across all sessions and participants; lastly a test dataset comprising the remaining 5% of the data and corresponding to the data of a single participant that has been deliberately excluded from the training to avoid possible biases in the final classification.

The initial goal was to perform the study on the greatest possible number of channels to take advantage of any possible discrimination derived from the data between the users. However, the high dimensionality of the data increased the difficulty of developing a suitable architecture for the autoencoder as well as causing interminable training times. A final downsizing in the dataset was performed with a 9-channel selection shown in Figure 3, using as a main criterion the symmetry of the position on the two brain hemispheres.

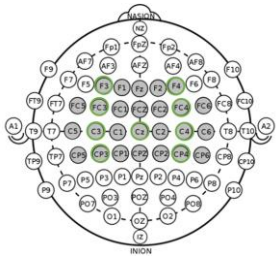

Figure 3 - Channels selected for the study.

The final size of the training data is 6,665 samples, for the validation data is 1,300 samples and for the test data is 419 samples. Each sample is composed of 2,500 timesteps (5 seconds of recording with sampling frequency of 500Hz) and 9 channels.

4 METHODS

4.1 Pre-processing

Upon early exploration of the data, outliers were noticed in several channels. In particular, the 1st/99th percentiles for most channels lay between ±100, but the maximum and minimum values reached 10,121 and –5,971, respectively, as can be seen in Table 1 below. When using a MinMax scaler to transform the data to [0,1] prior to training the autoencoder, this meant that for some channels most signals were compressed into a small range around 0.5. They thus received lower priority by the autoencoder since predicting a straight line would still result in a low loss relative to the other channels.

Table 1 – Summary Statistics for Raw Data in 4 Channels

Statistic	F3	F4	FC3	FC4
min	-5,661	-5,971	-483	-409
max	1,458	10,121	468	521
mean	0.88	1.96	0.44	0.72
variance	1,217	16,969	389	458
1st percentile	-78	-120	-54	-59
99th percentile	85	138	59	64

Initially, a method to deal with the outliers was implemented by replacing any value that fell below the 1st percentile with the 1st percentile value, and similarly for any point exceeding the 99th percentile. This eliminated outliers and led to a better reconstruction by the autoencoder. However, this introduced 'flat' periods into the signals which violated the proper waveform of the EEG. To avoid this, alternative methods of dealing with these outliers were investigated.

4.1.1 Band-pass Filtering

It was observed that many of the extreme values in the signals resulted from either 1) slow trends, or 2) very quick spikes/oscillations. Thus, a band-pass filter seemed a good approach to deal with these, since removing low frequencies would fix slow trends and removing high frequencies might remove quick, extreme spikes. Since brain waves mostly lie between the range (0.5Hz, 40Hz) [7], then it is possible that frequencies outside this range are artefacts left by the recording hardware or physiological processes [23]. As such, a Butterworth filter was applied, which is the most common band-pass filter applied to EEG according to [7]. The signals were filtered between the range (0.1Hz, 50Hz) to capture the range of EEG frequencies mentioned above, and also using suggestions from [23] and [24]. The summary statistics of the resulting data can be seen in Table 2 below for four out of the nine channels.

Table 2 – Summary Statistics after BP Filter

Statistic	F3	F4	FC3	FC4
min	-2,485	-8,705	-276	-266
max	2,649	5,271	185	194
mean	0.01	0.08	-0.001	0.01
variance	277	4,789	56	61
1st percentile	-34	-49	-20	-21
99th percentile	34	48	20	21

4.1.2 Quantile Transformer

While the band-pass filter reduced the variance of all channels and smoothed out the most extreme values, the table above shows that outliers persisted which would affect any scaling done to the data. Over 20% of the data samples had at least one value outside the $1^{st}/99^{th}$ percentile, so samples with extreme values could not just be removed from the dataset. Similarly, the extreme values could not be 'cut' out of each sample since this would distort the waveform and frequency information.

Thus, the quantile transformer method of scaling was used, which can reduce the impact of outliers and spreads out the scale of the most frequent values [25]. The scikit-learn implementation of this technique was used, and the output distribution was set to be gaussian because the raw data was approximately gaussian (within the $1^{st}/99^{th}$ percentiles).

4.2 Convolutional Autoencoder

An autoencoder was used to conduct feature extraction on the raw signals. In the original proposal for the study the idea was to use a LSTM autoencoder because of their ability to model time series data well. For example, LSTM was used by [26] for motor imagery classification using EEG signals, a similar task to the one presented in this study. Unfortunately, initial attempts with the LSTM autoencoder performed poorly, failing to learn useful information about the signal (see Appendix A.3) while requiring a huge number of trainable parameters. A simple fully connected autoencoder was then implemented and showed a noticeable improvement over the results of the LSTM; a CNN approach inspired by [27] was then used to test the feasibility of using convolutions on our dataset to reduce it to a low-dimensional representation.

The final decision was to use a convolutional autoencoder (CAE) to take advantage of its ability to learn the invariant features of a signal combined with the lower number of trainable parameters that would reduce training time, with the only downside of it being very application specific [27]. In our case the main purpose of the CAE is to compress the high-dimensional data into a feature vector to be fed to the classifier for identification. The CAE thus takes as input the data in the form of a 3-dimensional array (*n_samples, n_timesteps, n_channels)* and feeds it into 3 convolutional layers, each one followed by a single max pooling layer to reduce the dimensionality over the time-axis by taking the max value of each pool. The hyperparameters & layer dimensionality are discussed in section 5.2.

The convolutional layers use *ReLU* as activation function to rectify any negative value to zero and guarantee the correct behaviour of the network. Relu was selected since it is commonly used in CNNs according to [28]. *Mean Absolute Error* is used as the loss function to prevent eventual outliers, mostly caused by noise, from unduly influencing the reconstruction. The decoder is symmetrical to the encoder, with the max pooling layers substituted with up-sampling layers. Typically, a *sigmoid* or *ReLU* activation function would be applied to the output layer, but in this case the quantile transformer scaled the data in the range (-5, 5), therefore a *linear* activation function was chosen instead.

The architecture of the CAE is represented in Figure 4.

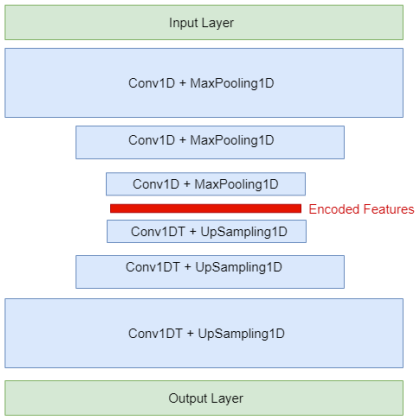

Figure 4 - Architecture of the autoencoder

4.3 SVM Classification

The primary objectives for the choice of classifier were 1) a flexible method which allows a new participant to be easily added to the identification system without retraining, and 2) to minimise the potential for overfitting, given the training set contains less than 7,000 samples. A support vector machine (SVM) seemed a good candidate for these criteria.

4.3.1 Pairwise SVM

Initially, the plan was to take a pairwise approach whereby pairs of samples would be compared and a binary classifier trained to determine whether they came from the same class or not, similar to the approach in [29]. A voting system across all pairwise comparisons could then be used to predict the class of a test sample. However, this approach proved infeasible due to excessive training times and poor classification performance. Training on only 10% of the samples, and 20% of the possible pairwise combinations between these samples, took more than one hour to train and yielded a classification accuracy on the validation set of between 10-15%. This is not much higher than the expected accuracy of 5.3% when guessing randomly across the 19 classes.

4.3.2 SVM One-vs-all

The method implemented as an alternative was a more straightforward version of the SVM classifier. One binary classifier was trained per class in a one-vs-all approach. To predict the class of a sample, the probabilities from each classifier were calculated and the class with the maximal probability was chosen for that sample. This is essentially the Bayesian approach, but taking the prior probabilities for each class to be equal, i.e. given a sample $\boldsymbol{x}$, predict class $\boldsymbol{C_k}$ where $\boldsymbol{k = \underset{i}{\operatorname{argmax}}\, P(x|C_i)}$.

The scikit-learn SVC implementation [30] was used to train each binary classifier and calculate the predicted probabilities. A radial basis function (rbf) kernel was used as linear decision boundaries were unlikely to be sufficient to separate 19 classes in the latent

space in a one-vs-all approach. Increasing the degree of the rbf kernel from 2 to 3 yielded no improvement on accuracy for the validation set, so the degree was set to 2, and a grid search was conducted across values of C and gamma.

While each binary classifier was trained on an imbalanced dataset with only 5% approx. of positive examples, this did not influence the final class output since probabilities were used rather than class predictions from the individual binary classifiers. Indeed, while undersampling the negative class for each binary classifier resulted in quicker training times, the final classification accuracy on the validation set deteriorated by up to 5 percentage points.

Taking the one-vs-all approach described above means that when a new participant is added to the system, then just one binary classifier needs to be trained for the new class and added to the existing classifiers rather than re-training the whole system. In this way, it met the objectives set based on the motivating problem of adding new participants.

5 OPTIMISATION & EVALUATION

5.1 Assessing Candidate Architectures

The exploratory study of autoencoder variations led to the parallel development of two CAE candidate architectures, both able to reconstruct the oscillatory pattern of the signals, but with different layers. The first architecture had 8 layers before the encoding, 4 of which performed a one-dimensional convolution, and the size of the encoded layer was 250 values; the second architecture had 7 layers before the encoding, 3 of which performing a one-dimensional convolution, and the size of the encoding was also 250 values. The assessment was conducted with an empirical method by running an experiment in parallel on both architectures for 20 and then 50 epochs, and finally collecting the following metrics: the total reconstruction loss, the loss distribution on the whole dataset, the loss distribution for each channel and the ability to reconstruct a single sample. While the two models performed similarly, the second achieved a lower loss in many of the samples as noted by the loss distribution in Figures 5 and 6.

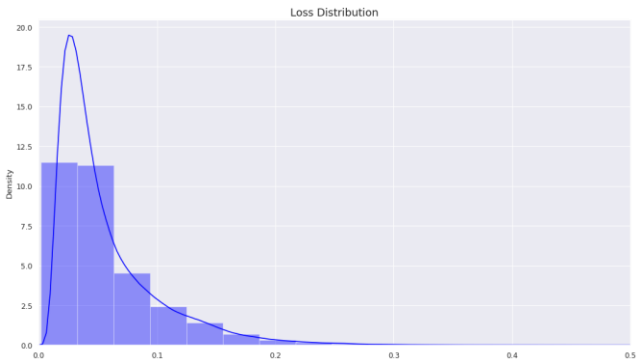

Figure 5 - Loss distribution (across samples) of the first architecture

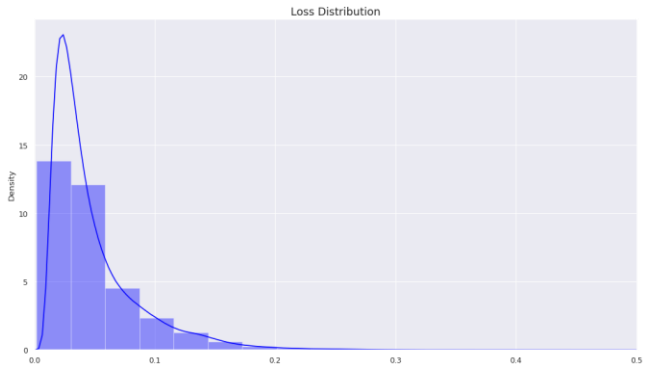

Figure 6 - Loss distribution (across samples) of the 2nd architecture

It was interesting that the second architecture was able to learn better despite having a simpler architecture. The second architecture, with fewer layers, was thus chosen for further experiments and hyperparameter tuning.

5.2 Hyperparameter Tuning

A grid search was chosen as the preferred method for the optimization of the hyperparameters of the convolutional autoencoder for its ability to quickly generate multiple variants of the model. The parameters included in the optimization process were the *kernel size* and the *number of filters* for the convolutional layers and the *pool size* for the max pooling layers. For each parameter 3 possible combinations of triplets, to be plugged in the respective layers, were provided for a total of 3^3=27 possible variants and 3 different sizes for the *encoding*: 1250, 500 and 250 parameters.

Table 4 – Possible values of hyperparameters

Kernel Sizes	Filters	Pool Sizes
11, 7, 5	27, 18, 2	2, 2, 1
9, 5, 3	18, 6, 2	2, 5, 1
5, 5, 5	27, 9, 2	2, 5, 2

A larger *kernel size* means bigger steps over the data, resulting in a "less rigorous reading of the data, but may result in a more generalized snapshot of the input"[31]. *Filters* and *pool sizes* instead were chosen accordingly with the dimensionality of the data.

The experiment was run over all the models sequentially for 25 epochs and a batch size of 20, and an experiment tracker was automatically generated to keep track of each model, its parameters, the size of encoding and the loss on the training and the validation datasets.

5.3 CAE Model Selection

While the experiment tracker highlighted, as expected, a lower loss in almost all models with an encoding size of 1,250 values, it was interesting to observe that the validation loss increased only by 0.02

for an encoding size of 500 values and by 0.06 for an encoding of 250. The final decision was to keep the best performing model for each of the encoding sizes, as shown in the following table.

Table 3 – Best model for each encoding size
(note when considering loss that data scale is [-5,5])

Model	Kernel Sizes	Filters	Pool Sizes	Enc. Size	Train Loss	Val Loss
19	5, 5, 5	27,18,2	2,2,1	1250	0.42	0.41
27	11,7, 5	27,18,2	2,5,1	500	0.44	0.43
17	5,5,5	27,18,2	2,5,2	250	0.48	0.47

5.4 Classification Layer

The autoencoders with the lowest loss for each of the three encoding sizes were evaluated in the classification layer, to compare the effect of different factors of compression. Three models were trained on each set of encodings, with the C parameter of the SVM varying across {0.1, 1, 10}, and using the default value for gamma[2]. The results across each set of encodings were similar. The less compressed data with 1,250 features achieved a slightly higher accuracy on the training set but a lower accuracy on the validation set compared to the other encoding sizes. It seems the greater number of features caused the model to overfit. Since the results were broadly similar, the smallest encoding size of 250 values was chosen in line with Occam's Razor.

A grid search was conducted using the chosen encoded data, fitting SVM models with varying values of C and gamma. Upon final selection and training of the best performing SVM model, the unseen participant was added. This meant that some of the test data had to be used to train the final binary classifier for this new class, so to obtain robust results a cross-validation approach was taken in fitting & testing in this phase. Each of the six 'runs' (see section 3 for data description), were left out of training and then used for testing. The mean of the six testing accuracies obtained is reported as the test accuracy.

6 RESULTS

6.1 Training & Validation

Table 4 contains a subset of results from the hyperparameter tuning of the SVM. The worst performance across all attempts was found with a third degree rbf kernel with C=1 and Gamma=10^{-5}. This model performed poorly on both the training set and validation set, indicating that the model was too simple and the value of C and/or gamma should be increased. Indeed, increasing the value for either parameter resulted in higher accuracies, but gamma had by far the most impact. Increasing the gamma parameter to a more suitable value in the order of 10^{-3} enabled an accuracy on the training set above 90%.

A large gap exists between the training accuracy and validation accuracy indicating that the model is learning a good deal of information in training which is not generalisable. Often, this gap can be reduced by applying regularisation to the model. However, regularisation here in the form of a lower value for C or gamma decreased accuracies for both the training *and* validation sets. Thus, the best performing model achieved an accuracy on the validation set of 49.9%, but yielded a training accuracy of 99.95%.

Table 4 – SVM Hyperparameter Tuning (subset of results)

Model	Kernel	Degree	C	Gamma	Train Acc	Validation Acc
1	rbf	3	1	0.00001	36.13%	30.62%
2	rbf	2	0.1	scale	92.59%	46.77%
3	rbf	2	1	scale	92.99%	46.08%
4	rbf	2	10	scale	99.74%	48.54%
5	rbf	2	10	0.006	99.97%	49.00%
6	rbf	2	50	0.006	99.99%	48.23%
7	rbf	2	100	0.006	99.95%	49.85%

Since the individual class likelihoods were used in the one-vs-all approach, a threshold could be set on the probabilities on whether to accept a class prediction or not. E.g. given a sample x, predict class C_k [where $k = \underset{i}{\operatorname{argmax}} P(x|C_i)$], if and only if $P(x|C_k) > threshold$. Otherwise, return no class prediction. A higher threshold thus results in a higher mean precision across classes but a lower mean recall. This trade-off can be seen in Fig 7 below.

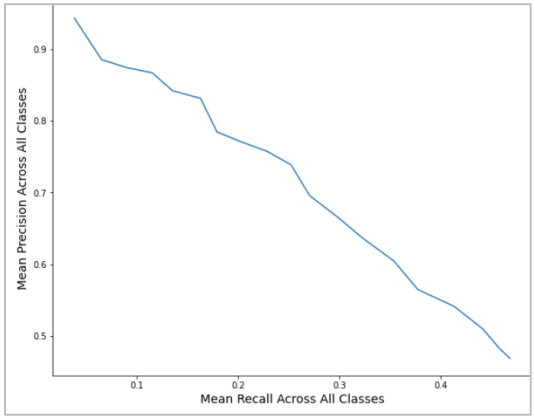

Figure 7 - Mean Precision-Recall Curve

6.2 Results on Test Set

After training the model successively on each set of five 'runs' for the test participant and testing the classification accuracy on the left out run, the mean accuracy was 78%. Since there was only one class in this test set, the accuracy here is equivalent to the recall for this class, and precision is not relevant.

[2] Gamma is set to 'scale' in sklearn by default. This is calculated as 1/(n_features * variance of the data). This equated to roughly 0.003 for the data here.

7 DISCUSSION

Clearly, an accuracy of 49.9% on the validation set indicates serious shortcomings for this approach. Further analysis is required to determine how much of this low accuracy is due to the autoencoder + SVM methodology, and how much is inherent in the data itself. Time constraints meant that neither a baseline traditional approach nor a full CNN classifier could be implemented and tested. Of course, with more classes (participants) for the model to discriminate between, we expect a lower accuracy and indeed small experiments on subsets of the data showed potential validation accuracies of 80-90% using only two-three participants or 65-75% accuracies using five-eight participants. However, this performance is still not sufficient for a real-world system and it is unrealistic for such a system to be limited to a low number of participants.

By varying the probability threshold, it was possible to tweak the precision-recall trade-off. Some identifications systems like this may require a higher security level at the expense of user experience [32], meaning a higher precision is preferred. Thus, the ability of the model to achieve a mean precision across classes of 80-90% is notable. However, the mean recall at which it achieves this is much too low at 10-20%. This would make for an acceptable user experience since, on average, it would work for the user only once out of five attempts. Adjustments to the system could improve this aspect, however, such as multiple identification attempts over the course of several seconds without notifying the user of failures.

Perhaps the most surprising result of the study was the accuracy on the test set, given that there is a much lower accuracy on the validation set. A possible explanation for this borrows ideas from the way autoencoders are used to detect anomalies [33], which relies on the autoencoder performing poorly when it sees a previously unseen pattern. For this study, the autoencoder may be overfitting on the patterns of the 19 participants in the training set, thus when the samples from the unseen participant are encoded, the encodings are quite different from the training data. This allows the classifier to separate them more easily and to identify the test participant with 78% accuracy.

As mentioned in Section 2, many studies report classification accuracies close to 99%. However, as has been pointed out, these accuracies are achieved using only one recording session per participant, so do not provide a fair comparison. Perhaps, the best study with which we can compare results is [21] which achieved accuracies between 66-72% when predicting across different sessions using a pool of ten participants. The accuracy achieved in the current study is much lower, but this is partly due to the higher number of participants. Unfortunately, this is the first time this dataset has been used for this particular task, so the predictive potential of the dataset is unclear.

Further improvements could be made to the current study by including more channels in the dataset, using more powerful techniques such as a Multi-layer Perceptron in the classification layer, or adding LSTM layers to the autoencoder in a similar way to [17], [18]. Of course, replacing the autoencoder with a CNN for feature extraction & classification would be expected to yield higher classification accuracies but as discussed previously, would not be flexible to the addition of new users. Nonetheless, ideally a state-of-the-art classification technique would be applied to the data to determine the predictive potential in the data, and to compare to the autoencoder-SVM approach.

8 CONCLUSION

The process followed in this study dealt explicitly with several of the existing issues in classifying EEG data. Despite the results being below the initial expectations and worse than some of the comparable studies, the methodology applied demonstrated the potential of this innovative approach to EEG classification. Starting with the selection of the dataset, relevant criteria such as the number of participants, the length of the sessions and their distribution over a span of two weeks already diverges from many related works. The use of CNN/RNN and their variants have great potential, but the accuracies reported in similar studies are unrealistic given the biases involved in the data used. Even when these biases are avoided, the problem of scalability remains, and this can be tackled with a more general multi-model solution such as the one presented here.

In conclusion, this study explores an alternative autoencoder-SVM approach which encourages the investigation of multi-model systems for biometric authentication using EEG signals. The proposed improvements for the methods presented invites others to experiment with alternative techniques, to improve the feature extraction and classification layers, and eventually to collect EEG data in a dedicated experiment with the specific purpose of performing biometric authentication using EEG signals.

REFERENCES

[1] M. Poulos, M. Rangoussi, V. Chrissikopoulos, and A. Evangelou, 'Person identification based on parametric processing of the EEG', in *ICECS'99. Proceedings of ICECS '99. 6th IEEE International Conference on Electronics, Circuits and Systems (Cat. No.99EX357)*, Sep. 1999, vol. 1, pp. 283–286 vol.1, doi: 10.1109/ICECS.1999.812278.

[2] J. Galbally, S. Marcel, and J. Fierrez, 'Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition', *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 710–724, Feb. 2014, doi: 10.1109/TIP.2013.2292332.

[3] W. Yang, S. Wang, J. Hu, G. Zheng, and C. Valli, 'Security and Accuracy of Fingerprint-Based Biometrics: A Review', *Symmetry*, vol. 11, no. 2, Art. no. 2, Feb. 2019, doi: 10.3390/sym11020141.

[4] Z. Akhtar and A. Rattani, 'A Face in any Form: New Challenges and Opportunities for Face Recognition Technology', *Computer*, vol. 50, no. 4, pp. 80–90, Apr. 2017, doi: 10.1109/MC.2017.119.

[5] P. Campisi and D. L. Rocca, 'Brain waves for automatic biometric-based user recognition', *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 5, pp. 782–800, May 2014, doi: 10.1109/TIFS.2014.2308640.

[6] J. Klonovs, C. K. Petersen, H. Olesen, and A. Hammershoj, 'ID Proof on the Go: Development of a Mobile EEG-Based Biometric Authentication System', *IEEE Veh. Technol. Mag.*, vol. 8, no. 1, pp. 81–89, Mar. 2013, doi: 10.1109/MVT.2012.2234056.

[7] A. Jalaly Bidgoly, H. Jalaly Bidgoly, and Z. Arezoumand, ‘A survey on methods and challenges in EEG based authentication’, *Comput. Secur.*, vol. 93, p. 101788, Jun. 2020, doi: 10.1016/j.cose.2020.101788.

[8] H. H. Stassen, ‘Computerized recognition of persons by EEG spectral patterns’, *Electroencephalogr. Clin. Neurophysiol.*, vol. 49, no. 1, pp. 190–194, Jul. 1980, doi: 10.1016/0013-4694(80)90368-5.

[9] I. Jayarathne, M. Cohen, and S. Amarakeerthi, ‘Survey of EEG-based biometric authentication’, in *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*, Nov. 2017, pp. 324–329, doi: 10.1109/ICAwST.2017.8256471.

[10] M. Abo-Zahhad, S. M. Ahmed, and S. N. Abbas, ‘A New EEG Acquisition Protocol for Biometric Identification Using Eye Blinking Signals’, *Int. J. Intell. Syst. Appl.*, vol. 7, no. 6, pp. 48–54, May 2015, doi: 10.5815/ijisa.2015.06.05.

[11] E. Maiorana, D. L. Rocca, and P. Campisi, ‘On the Permanence of EEG Signals for Biometric Recognition’, *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 1, pp. 163–175, Jan. 2016, doi: 10.1109/TIFS.2015.2481870.

[12] Y. Di, X. An, F. He, S. Liu, Y. Ke, and D. Ming, ‘Robustness Analysis of Identification Using Resting-State EEG Signals’, *IEEE Access*, vol. 7, pp. 42113–42122, 2019, doi: 10.1109/ACCESS.2019.2907644.

[13] T. Pham, W. Ma, D. Tran, P. Nguyen, and D. Phung, ‘Multi-factor EEG-based user authentication’, in *2014 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2014, pp. 4029–4034, doi: 10.1109/IJCNN.2014.6889569.

[14] Z. A. A. Alyasseri, A. T. Khader, M. A. Al-Betar, J. P. Papa, O. A. Alomari, and S. N. Makhadme, ‘An Efficient Optimization Technique of EEG Decomposition for User Authentication System’, in *2018 2nd International Conference on BioSignal Analysis, Processing and Systems (ICBAPS)*, Jul. 2018, pp. 1–6, doi: 10.1109/ICBAPS.2018.8527404.

[15] Q. Gui, Z. Jin, W. Xu, M. V. Ruiz-Blondet, and S. Laszlo, ‘Multichannel EEG-based biometric using improved RBF neural networks’, in *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, Dec. 2015, pp. 1–6, doi: 10.1109/SPMB.2015.7405418.

[16] T. Waili, G. Johar, K. Sidek, N. Nor, H. Yaacob, and M. Othman, *EEG Based Biometric Identification Using Correlation and MLPNN Models*. International Association of Online Engineering, 2019, pp. 77–90.

[17] T. Schons, G. J. P. Moreira, P. H. L. Silva, V. N. Coelho, and E. J. S. Luz, ‘Convolutional Network for EEG-Based Biometric’, in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Cham, 2018, pp. 601–608, doi: 10.1007/978-3-319-75193-1_72.

[18] X. Zhang, L. Yao, S. S. Kanhere, Y. Liu, T. Gu, and K. Chen, ‘MindID: Person Identification from Brain Waves through Attention-based Recurrent Neural Network’, *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 3, p. 149:1-149:23, Sep. 2018, doi: 10.1145/3264959.

[19] T. Wilaiprasitporn, A. Ditthapron, K. Matchaparn, T. Tongbuasirilai, N. Banluesombatkul, and E. Chuangsuwanich, ‘Affective EEG-Based Person Identification Using the Deep Learning Approach’, *IEEE Trans. Cogn. Dev. Syst.*, vol. 12, no. 3, pp. 486–496, Sep. 2020, doi: 10.1109/TCDS.2019.2924648.

[20] Y. Sun, F. P.-W. Lo, and B. Lo, ‘EEG-based user identification system using 1D-convolutional long short-term memory neural networks’, *Expert Syst. Appl.*, vol. 125, pp. 259–267, Jul. 2019, doi: 10.1016/j.eswa.2019.01.080.

[21] O. Özdenizci, Y. Wang, T. Koike-Akino, and D. Erdoğmuş, ‘Adversarial Deep Learning in EEG Biometrics’, *IEEE Signal Process. Lett.*, vol. 26, no. 5, pp. 710–714, May 2019, doi: 10.1109/LSP.2019.2906826.

[22] Zhou, Qing, ‘EEG dataset of 7-day Motor Imagery BCI’. IEEE DataPort, Nov. 29, 2020, doi: 10.21227/F1C7-7X89.

[23] M. Teplan, ‘Fundamentals of EEG measurement’, *Meas. Sci. Rev.*, vol. 2, no. 2, pp. 1–11, 2002.

[24] N. P. Subramaniyam, ‘Pitfalls of Filtering the EEG Signal’, *Sapien Labs | Neuroscience | Human Brain Diversity Project*, Nov. 12, 2018. https://sapienlabs.org/pitfalls-of-filtering-the-eeg-signal/ (accessed Jan. 23, 2021).

[25] ‘sklearn.preprocessing.QuantileTransformer — scikit-learn 0.24.1 documentation’. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html#sklearn.preprocessing.QuantileTransformer (accessed Jan. 19, 2021).

[26] R. H. Elessawy, S. Eldawlatly, and H. M. Abbas, ‘A Long Short-Term Memory Autoencoder Approach for EEG Motor Imagery Classification’, in *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, Jan. 2020, pp. 79–84, doi: 10.1109/ICCAKM46823.2020.9051489.

[27] X. Lun, Z. Yu, T. Chen, F. Wang, and Y. Hou, ‘A Simplified CNN Classification Method for MI-EEG via the Electrode Pairs Signals’, *Front. Hum. Neurosci.*, vol. 14, 2020, doi: 10.3389/fnhum.2020.00338.

[28] S. SHARMA, ‘Activation Functions in Neural Networks’, *Medium*, Feb. 14, 2019. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 (accessed Jan. 24, 2021).

[29] J. R. Bock and D. A. Gough, ‘Predicting protein–protein interactions from primary structure’, *Bioinformatics*, vol. 17, no. 5, pp. 455–460, May 2001, doi: 10.1093/bioinformatics/17.5.455.

[30] ‘sklearn.svm.SVC — scikit-learn 0.24.1 documentation’. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (accessed Jan. 23, 2021).

[31] J. Brownlee, ‘1D Convolutional Neural Network Models for Human Activity Recognition’, *Machine Learning Mastery*, Sep. 20, 2018. https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/ (accessed Jan. 24, 2021).

[32] ‘FAR and FRR: security level versus user convenience’. https://www.recogtech.com/en/knowledge-base/security-level-versus-user-convenience (accessed Jan. 24, 2021).

[33] B. Larzalere, ‘LSTM Autoencoder for Anomaly Detection’, *GitHub*. https://github.com/BLarzalere/LSTM-Autoencoder-for-Anomaly-Detection (accessed Jan. 24, 2021).

[34] S. Koelstra *et al.*, ‘DEAP: A Database for Emotion Analysis ;Using Physiological Signals’, *IEEE Trans. Affect. Comput.*, vol. 3, no. 1, pp. 18–31, Jan. 2012, doi: 10.1109/T-AFFC.2011.15.

[35] M. Hunter *et al.*, ‘The Australian EEG database’, *Clin. EEG Neurosci.*, vol. 36, no. 2, pp. 76–81, Apr. 2005, doi: 10.1177/155005940503600206.

[36] G. B. Moody, R. G. Mark, and A. L. Goldberger, ‘PhysioNet: a Web-based resource for the study of physiologic signals’, *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 70–75, May 2001, doi: 10.1109/51.932728.

APPENDIX

A.1 – Candidate datasets

These suitable datasets were initially identifier according to the minimum requirements for our study:

- The DEAP dataset [34], containing emotion analysis of 32 participants with ~40 minutes trials.
- The Australia EEG Database [35] recorded over a span of 11 years with ~20 minutes trials and 40 participants.
- The Physionet Motor Movement/Imagery dataset [36] with 109 participants performing 14 various tasks.

These datasets are publicly available for use and are known in the research community, so they were good candidates also for future comparisons with other studies. However, we finally opted for a more compromise option, that would still satisfy our requirements but with a more manageable structure and size given our time and computational constraints.

A.2 – Effects of Band-pass Filter

While the distribution of the data was approximately Gaussian between the 1st and 99th percentiles, the application of the band-pass filter served to 1) reduce the total variance of the data by reducing the effect of outliers, and 2) provide a wider spread of data between the 1st and 99th percentiles, as can be seen from the differences between Fig A.1 and Fig A.2.

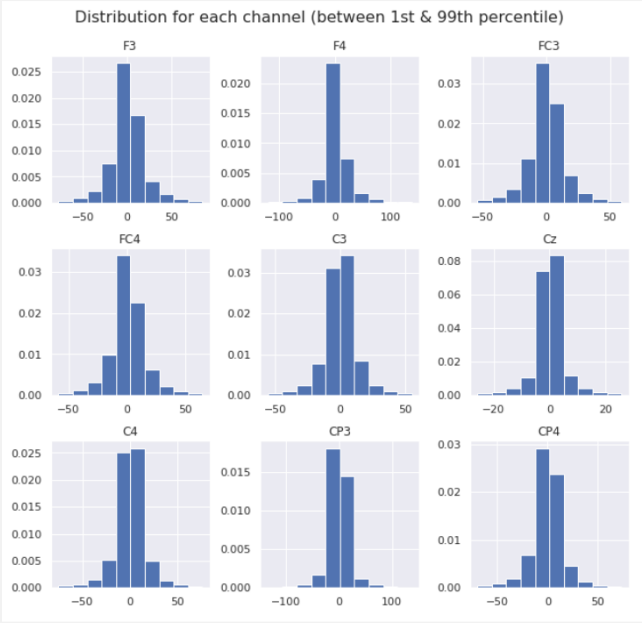

Fig A.1 – Distribution of Raw Data by Channel

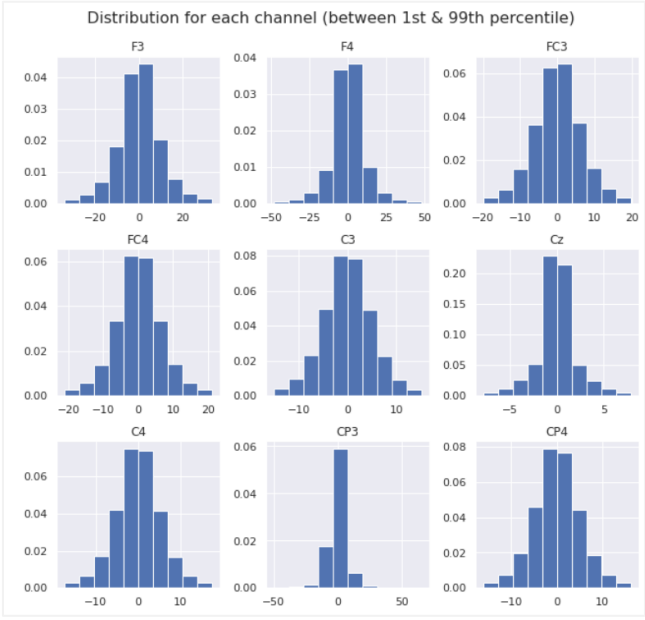

Fig A.2 – Distribution of Data by Channel after Band-pass Filter

We can the effect of the band-pass filter on a typical example of a signal sample for one channel in Fig A.3. The blue line is the original signal and the orange is the reconstructed signal. Removing the low frequencies serves to remove the slow drifts, like in the latter quarter of the original signal. Removing frequencies above 50Hz serves to remove any very high frequency patterns, but has only a marginal impact on this particular sample.

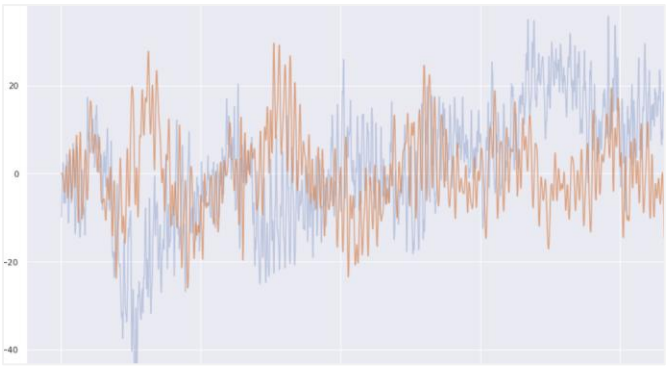

Fig A.3 – Sample signal before (blue-grey) and after (orange) band-pass filter

A.3 – Autoencoder Learning

The first attempts with a LSTM layers indicated the model was far too simple, as it was learning to fit and predict the mean of each signal sample but no other useful information, as can be seen from Fig A.4. (Note in most of the below plots, the data scale is [0,1] as these were created using the MinMax scaler before switching to Quantile Transformer. The reconstructions were similar, however).

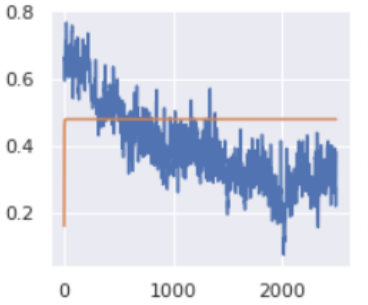

Fig A.4 – LSTM Prediction Example. Original signal (blue) vs. the predicted reconstruction (orange).

Building simple fully connected autoencoders with one or two layers in the encoder allowed the model to learn a simple moving average of the signal. Similar results were achieved with convolutional autoencoders with one or two convolutional and max pooling layers in the encoder. Fig A.5 gives an example of this reconstruction.

Fig A.5 – Simple Convolutional Autoencoder Prediction Example. Original signal (blue) vs. the predicted reconstruction (orange).

Adding an extra convolutional layer & max pooling layer to the encoder finally allowed it to learn the oscillatory pattern of the signals, as can be seen in Fig A.6.

Fig A.6 – Convolutional Autoencoder Prediction (3 conv layers) Example. Original signal (blue) vs. the predicted reconstruction (orange).

A.4 – Assessing CAE architectures

During the assessment of the candidate architectures two parallel experiments were run to quantify the goodness of fit of the models.

For each candidate architecture we plot how a single sample is reconstructed channel-wise. An interesting discovery is that both architectures failed to reconstruct the channel Cz. Central channels are positioned over the corpus callosum and do not represent either hemisphere adequately, therefore will not necessarily reflect or amplify lateral hemispheric cortical activity, making them suitable candidates as 'grounds' or 'references'. Replacing Cz with a pair of lateral channels could possibly improve the overall prediction.

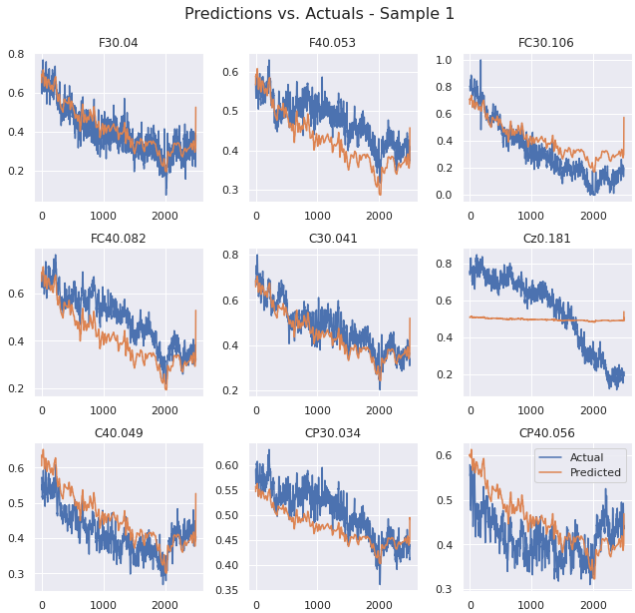

Fig A.6 – Plot of the prediction (in orange) over each channel of the first architecture

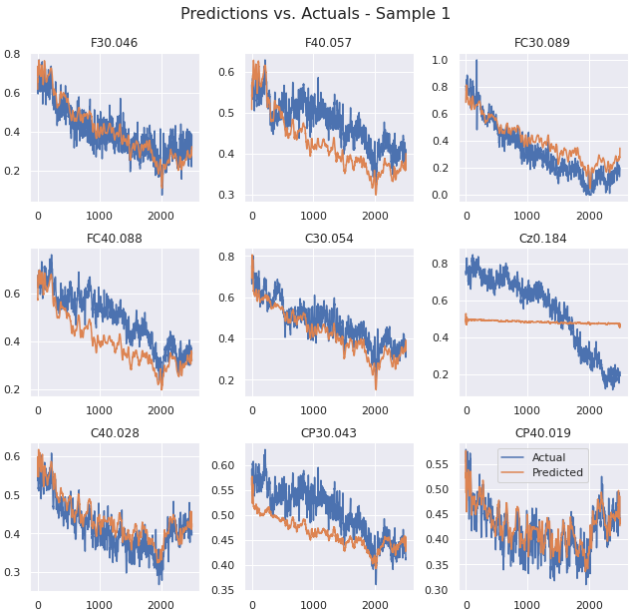

Fig A.7 – Plot of the prediction (in orange) over each channel of the second architecture