A Comparative Study of Time-Series and Image-Based Models for EEG Binary Classification

Anonymous

Australian National University

Abstract. Electroencephalography (EEG) signals represent time series data, providing an opportunity to develop classification models using Deep Neural Network (DNN) architectures. This report presents four deep learning-based models based on the nature of the dataset. For a time-series dataset, Long Short-term Memory (LSTM) and a Deep Neural Network (DNN) are implemented, and for an image dataset, a Convolutional Neural Network (CNN) and Residual Neural Network ResNet-18 are employed. All these architectures are trained and tested on the UCI EEG dataset and also compared with the results of a simple DNN network architecture with and without Principal Component Analysis (PCA) and with a reference implementation. Results show that the four network architectures achieve a good performance on the within-subject split and cross-subject split accuracy and F1 score.

Keywords: EEG Signals · Time series data · UCI EEG Dataset · DNN · LSTM · CNN· ResNet-18· Binary Classification · Machine Learning Algorithms · PCA · Deep Learning.

1 Introduction

EEG signals are the estimates of human brain electrical activity. They are measured by mounting EEG sensors on the human brain scalp. They have a major application in the field of health science. It is used in a variety of applications, especially in monitoring the alertness and mental engagement level, exploration of chronic conditions, and serving as signals for biofeedback or assistive devices [16]. It is challenging to analyze them due to the complexity of brain function, inherent signal variability between individuals, sensitivity to noise, and artifacts. On the Other hand, for a computer EEG signals are considered to be time series signals. Graphically, they have voltage levels on the y-axis and time on the x-axis as in 2. Time series representation of EEG signals is very crucial in the context of medical and neurological research, as well as applications in fields like brain-computer interfaces (BCIs) and cognitive neuroscience. It is also important due to the following factors:-

Temporal Nature of EEG Signals: - EEG records electrical activity in the brain over time, capturing neural activity in a highly time-dependent manner. This temporal aspect is vital for understanding brain functioning, cognitive processes, and neurological disorders.

Real-Time Monitoring: - In clinical and research settings, real-time monitoring of EEG data is important. Time series data are necessary for continuous monitoring and making immediate assessments, especially in scenarios like epilepsy monitoring units.

Pattern Recognition: - Time series analysis is crucial for recognizing specific patterns or states, such as identifying the onset of seizures, sleep stages, or cognitive states from EEG data. Machine learning and pattern recognition techniques often rely on time series data. With the advancement in deep learning, DNN architectures are very capable of analyzing the intricate temporal dependencies within the data.

Neural Networks are a subset of Machine learning that consists of node layers. The input layer is where the data is fed into the network followed by a sequence of hidden layers with hidden neurons and an output layer where the labels are predicted. Recurrent Neural Networks (RNNs) are stacks of multi-layer neural networks that are used in predicting the output of sequential data. It is used for speech recognition, time series data prediction, etc. Long Short Term Memory networks on the other hand are RNNs with memory units and feedback loops from output to input layers. These are crucial in predicting time-series data prediction as it has information about the previous data. Considering the time-series nature of EEG signals, the application of time series-based networks will make the task of predicting whether a subject is alcoholic or a control subject possible.

Convolution is a mathematical operation performed on two signals where one of the signals is the input and the other one is the kernel or filter to get a new representation. In the context of deep learning, CNN (Convolutional

Neural Networks) is used for feature extraction. For instance, it is used to get important features like the eyes of a face, edges, and textures of an image, etc. ResNet-18 is a convolutional neural network architecture widely used for image classification tasks. It is part of the ResNet (Residual Network) family, known for addressing the vanishing gradient problem. ResNet-18 consists of 18 layers 1, including convolutional layers, batch normalization, and residual blocks. The key innovation is residual connections that skip over one or more layers, allowing for the training of very deep networks. These shortcuts facilitate the flow of gradients during training, enabling the efficient training of deep neural networks. ResNet-18 has been successful in various computer vision tasks and is known for its superior performance and ease of training.

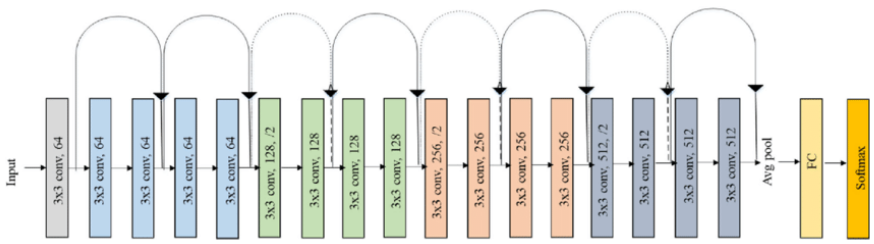

Fig. 1: ResNet-18 Architecture

Therefore if we can transform the time series EEG signal to an image, we can perform feature extraction using CNN and use traditional neural networks for binary classification.

Problem Definition: - The problem definition is formulated in two ways.

- Given a time series-based EEG signal and binary labels, how accurate can a time series-based neural network be used to for binary classification?
- Given an image and binary labels, how accurate a CNN network is used for feature extraction and label prediction?

2 Related Work

The field of EEG Signal classification has involved a lot of researchers' contributions and methodologies. Farsi et al.[4] employed two algorithms for EEG Classification. Algorithm 1 is about using PCA (Principal Component Analysis) for dimensionality reduction and a Deep Neural Network and Algorithm 2 is about implementing a two-layered Long short-term memory network (LSTM). P. Nagabushanam et al.[11] also developed a two-layered LSTM and four-layered Deep Neural Network on the EEG time series signals. Convolutional Neural Networks are considered as the feature extractors of a dataset, particularly images. LeCun et al.[7] used CNNs as feature extractors for images and time-series data. Other CNN network approaches include Deep4Net [15] and SyncNet [8]

3 EEG Dataset

The dataset I worked on for the research purpose is the alcoholism EEG database. It is from the UCI Machine Learning repository found in [3]. I worked on the Large Dataset format which has information about 127 subjects out of which 77 are diagnosed with alcoholism are the remaining ones are control subjects. Each subject has undergone 120 trials to check whether the subject is alcoholic or not. The signals are recorded by placing 64 electrodes on the subject's scalp and recorded for 1 second. Figure 2 shows the EEG signals for alcoholic and control subjects where the first 10 channels were considered. As we can see It is challenging to distinguish between alcoholic and non-alcoholic subjects due to the noisy nature of the data, and upon inspection, the signal representations appear quite similar. The dataset consists of two different representations.

EEG as Time-series Data: - Time series signals representation in machine learning refers to the way data is structured and organized for the analysis and modeling of temporal data, where the order of data points matters. Time series data typically consists of observations collected at successive time intervals. EEG data is represented in a time series format by recording electrical signals from the scalp electrodes over time. Each electrode measures the voltage fluctuations in the electrical activity of the brain, and these measurements are sampled at specific time intervals which is 256Hz in this case.

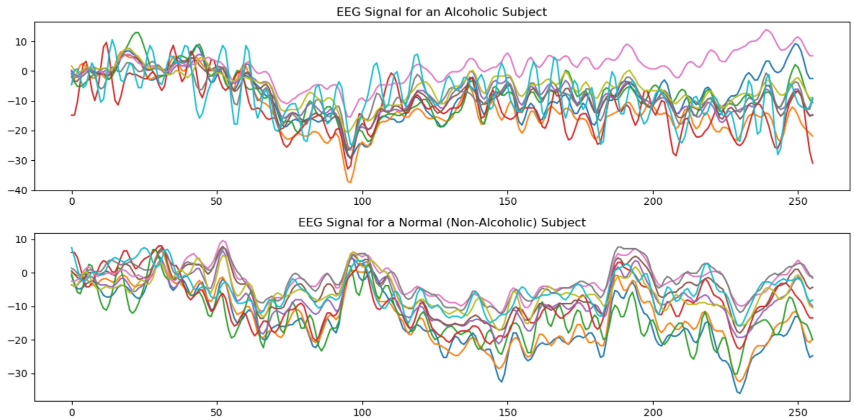

Fig. 2: Representation of EEG signals. The first one is the EEG signal for the alcoholic subject and the latter one is for the control subject

EEG time series data to Image: - This approach, based on Bashivan's research [2], involves integrating both temporal and spatial information from EEG signals recorded across the scalp during a single trial. It begins by performing a Fast Fourier Transform (FFT) on the time series data to compute the power spectrum for each trial, resulting in a 64 x 256 matrix. Subsequently, three distinct frequency bands, namely theta (4–7 Hz), alpha (8–13 Hz), and beta (13–30 Hz), are isolated, and their respective squared absolute values are summed to create a 64 x 3 map.

To construct an RGB EEG image, the theta frequencies are assigned to the red channel, alpha frequencies to the green channel, and beta frequencies to the blue channel. Additionally, the Azimuthal Equidistant Projection (AEP), also known as Polar Projection, is employed to project the original three-dimensional positions of the 64 EEG channels onto a two-dimensional flat surface. This projection process ensures that all EEG electrode positions are uniformly represented in a consistent 2-D space, despite their initial three-dimensional distribution across the scalp. Consequently, each 64 x 1 frequency band is then mapped onto a 32 x 32 grid, generating a 32 x 32 x 3 dataset. The Clough-Tocher interpolation scheme is used to estimate values between the electrodes across the 32 x 32 grid.

In summary, this methodology transforms a set of EEG signals with dimensions 64 x 256 into a series of 32 x 32 x 3 color images, effectively encapsulating both spectral and spatial information in a visual format.

3.1 Description of the Data set: -

data or X: - This tensor stores information related to EEG signals. In the context of time-series data, it follows the format N x T x C, where N corresponds to the total number of trials (11057), T indicates the time-series frequency (256Hz), and C represents the number of channels (64). However, when transitioning from time-series data to images, the variable "data" or "X" is reshaped to the dimensions N x 32 x 32 x 3, with three channels representing alpha, beta, and gamma frequency bands.

y_alcoholic: - This is a binary label with values 0 or 1. A value of 1 signifies an alcoholic subject, while 0 indicates a control subject. From the histogram representation shown in 3, it becomes evident that subjects labeled as "1" (indicating alcoholism) make up 70% of the dataset, highlighting the dataset imbalance.

y_stimulus: - This is an N-vector categorical variable that specifies which stimulus was presented to the subject during the trial. In this study, five different stimuli were used, resulting in labels ranging from 1 to 5. These labels correspond to the following stimuli:

- 1: S2nomatch
- 2: S1obj
- 3: S2match
- 4: S2matcherr
- 5: S2nomatcherr

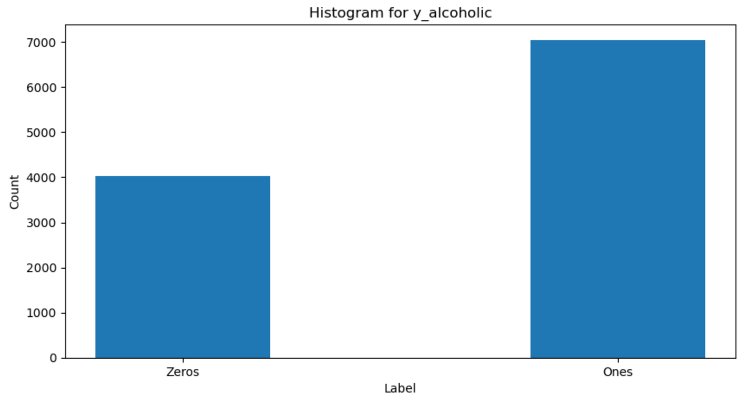

Fig. 3: Histogram distribution of Binary label y_alcoholic

***subjectid:* -** To maintain subject-specific distinctions, a unique identifier, is assigned to each of the 122 subjects participating in the study, with sequential numbering from $1 \ldots 122$.

***trialnum:* -** It is the trial identifier. It is used in tracking individual trials, with each subject typically experiencing approximately 120 trials. The numbering ranges from 0 to approximately 119 for each subject.

3.2 Data Preprocessing

The dataset used here is already a pre-processed version given by the Research School of Computer Science, ANU. However, the dataset has undergone normalizing to convert each sample in the range of [0-1] to prevent overfitting. In the context of time series datasets, Principal Component Analysis (PCA) is implemented to effectively reduce the dimensionality of channels and streamline the model. This approach simplifies the data while maintaining its fundamental characteristics. Specifically, 75% of the data's variance is captured by the first 30 principal components. This dimensionality reduction thus retains its essential characteristics while reducing complexity. It is worth noting that for image datasets, preprocessing steps like noise removal do not significantly impact the model's accuracy. However, the utilization of raw images in neural networks yielded promising results. The dataset undergoes within-subject and cross-subject splitting as in [6]. Within-subject classification involves training a model using a subset of the data from a particular subject, tailoring the model to that specific individual. In contrast, cross-subject classification entails using data from different subjects to train a model that is not specific to any one subject.

4 Methodology

This paper provides the following deep learning models tailored to the specific characteristics of our data. For time series data, we utilized Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM) networks. On the other hand, the time series data were converted to images and Convolutional Neural Networks (CNN), ResNet, and Autoencoders were applied to analyze image data.

4.1 Time series Architectures: -

***Deep Neural Networks**:* - DNNs are one of the sub-branches of Machine Learning used Binary Classification tasks [10]. It is created by stacking more than one hidden layer between the input and output layers. In the context of a two-layered DNN, we can now explore the mathematical principles behind forward and backward propagation.

***Forward Propagation:* -** If we assume W1, b1 are weights and biases of the first layer and W2, b2 are the weights and biases for the second layer and Input as the input matrix, then the Output of the model during the forward pass is calculated from the equation 1. Later, the loss function which is the absolute difference between the outputs and the labels is calculated.

$$\begin{aligned} \text{Output} &= ((\text{Input} \cdot W_1^T) + b_1)W_2^T + b_2 \\ \text{Output} &= \text{Activation}(z) \end{aligned} \tag{1}$$

Reverse Propagation: - The purpose of this method is to minimize the error between the predicted and the true labels. In this step, all the weights and biases from the output layer to the first hidden layer of the network are updated. The changes are performed by computing the gradients of the loss function. The above process is performed for a certain number of iterations called epochs for model convergence. For time series-based data, I first used a four-layered binary classifier with 128, 64, 32, and 16 hidden neurons respectively.

LSTM: - In addition to DNN, another crucial component of our time series architecture implementation, is the utilization of Long Short-Term Memory (LSTM) networks. LSTM uses Recurrent Neural networks with memory blocks in each hidden layer to avoid the problem raised due to vanishing gradients. In this paper, the raw time series signals are fed to the LSTM layer. The structure of the LSTM network is shown below 4. As we can see from the figure the raw time series signals are applied to the Dense layer with 64 hidden neurons. Later it is applied to the LSTM network [12] with 64 input neurons representing the number of channels or electrodes for EEG signals.

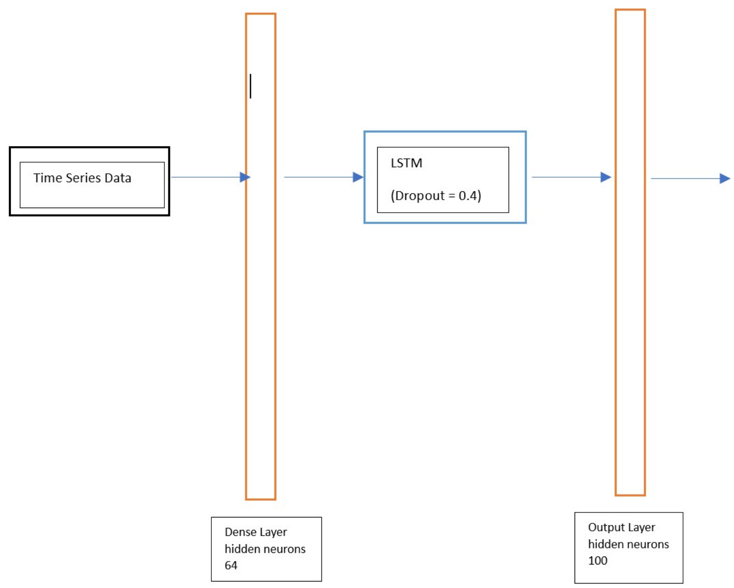

Fig. 4: LSTM structure with 64 hidden neurons in layer 1 and 100 hidden neurons in output layer

For the time series architectures, The number of neurons in the input layer is the Time series frequency (256Hz) X The number of channels (64) which are 16,384 neurons. For PCA, since the number of channels is reduced from 64 to 30, the total number of input neurons is 7680.

4.2 Image based architectures: -

In this method, the time-series EEG signals are converted to 32 x 32 x 3 images using [12] and the following architectures are implemented.

Convolution Neural Network: - In this the EEG images of size 32 x 32 x 3 are sent to the CNN which has 3 input channels representing the alpha, beta, and theta frequency bands. From the figure5, a convolution operation is performed using a 3 x 3 kernel with a stride of 1 in the convolution layer [12]. Later the output of the convolution layer is sent to a 2D max pooling layer with a 2 x 2 kernel and a stride of 2. Finally, it is passed to a fully connected layer with 64 hidden neurons and the output layer for performing binary classification.

ResNet 18 Architecture: - In this approach, a deeper neural network, ResNet-18 [14], is used for binary classification without pre-training. The ResNet-18 architecture from the figure 6 takes a 32 x 32 x 3 image as input. After passing through ResNet-18, the resulting vector has a dimension of 1000 x 1. This vector is then forwarded through three fully connected layers with 256, 128, and 64 hidden neurons.

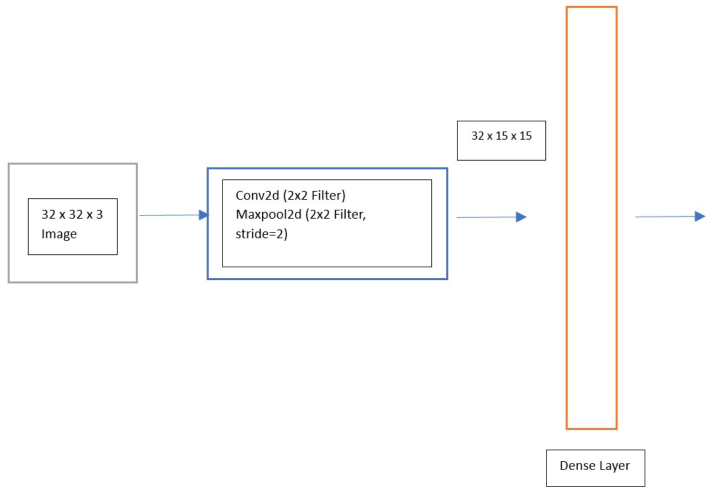

Fig. 5: CNN Architecture with input image 32x 32 x 3.

For the time series and image datasets, the number of output neurons of the output layer is always 1 representing the binary variable 1 for alcoholic and 0 for the control subject.

4.3 Network Training and Testing: -

The dataset has undergone an 8:2 split into training and testing samples using within-subject and cross-subject splitting techniques as mentioned in [6]. The dataset is loaded using the CoustomDatset and Data Loader classes from the pytorch framework with a batch size of 64 [12]. All the network architectures are trained for 200 epochs with a learning rate of 1e-4 1. The models are trained on a Windows-10 compatible PC with 16GB of RAM and RTX-3060 GPU. In addition to that, a learning rate scheduler is used to adjust the learning rate as the number of epochs increases. It is used to make the learning rate adaptive for gradient descent and improve the performance of classification. For all the networks, The Adam optimizer [5] is used for applying gradient descent. For binary classification tasks, the Binary Cross Entropy with Logit Loss [9] (BCEWithLogitsLoss()) function is used to compute the loss, between the predicted and true labels. After every subsequent layer, the ReLU activation function [1] is used for non-linearity.

Table 1: Hyper-Parameters for the Network

Split Size,	8:2
Batch Size	64
No. of Epochs	200
Learning Rate	1e-4
Lr Scheduler	step_size = 25, gamma = 0.1
Optimizer	Adam
Loss	BCEWithLogitsLoss()
No. of principal components	30

Metrics: - The accuracy and F1 score are used a performance metrics for the model. For every batch, the accuracy 2 is calculated by summing the number of occurrences of the predicted label and true label being equal and dividing by the total number of samples in the batch. Similarly, F1 score [13] for each batch is calculated using the formula 3

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples in the Batch}} \tag{2}$$

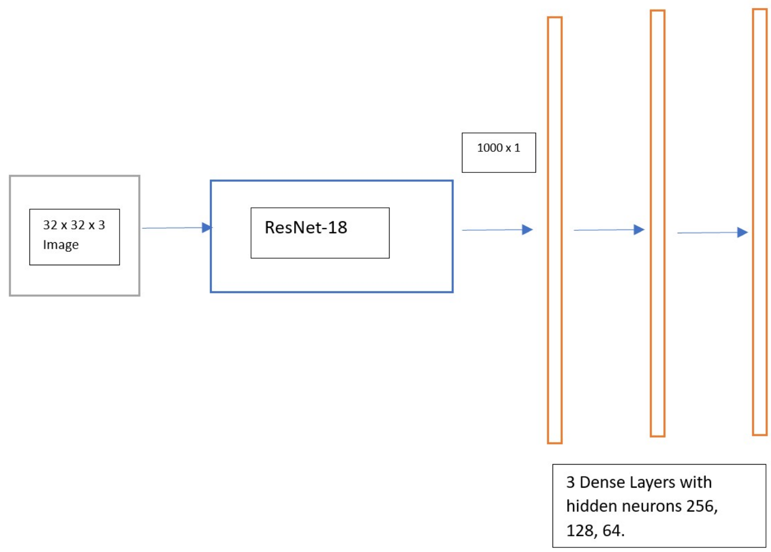

Fig. 6: ResNet-18 Architecture.

$$F1\_\text{score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

The results for every epoch during the training of the model are recorded. To select the best model for evaluating the test accuracy, I relied on the highest F1 training score, as it serves as a robust indicator of whether the labels are separated.

5 Results and Discussion

In this stage a comparative analysis of the within-subject and cross-subject test accuracy, F1 score is performed by applying DNN and LSTM on time series data and CNN, ResNet on image data, and their results are compared with the reference implementation [17] on the binary label y_alcoholic. Also, the accuracy metrics are compared with and without the application of PCA dimensionality reduction only on the time-series data.

Table 3 represents the evaluation of within-split test accuracy and F1 scores of time-series and EEG image-based models, whereas Table 3 is centered on results related to cross-subject splits.

Table 2: Within-subject split results

Model	Test Accuracy	Train Accuracy	Best test F1 score	Final Training loss
DNN	70.01%	73.5%	0.878	0.49
LSTM Classifier	**72.81**%	75.90%	**0.923**	0.49
CNN Classifier	77.22%	79.15%	0.902	0.42
ResNet18	**87.71**%	99.96%	**0.952**	0.0010

Table 3: Cross-subject split results

Model	Test Accuracy	Train Accuracy	Best test F1 score	Final Training loss
DNN	58.38%	65.8%	0.769	0.48
LSTM Classifier	**70.91**%	76.95%	**0.876**	0.48
CNN Classifier	61.70%	83.17%	0.81	0.36
ResNet18	**68.96**%	99.96%	**0.819**	0.0002

In the context of within-split accuracy of time series data, the LSTM classifier demonstrates superior performance when compared to the DNN, achieving a within-split accuracy of 75.90% and an F1 score of 0.923. This can be due to the LSTM's memory units, which effectively address the issue of vanishing gradients, ultimately leading to its impressive accuracy. However, it's worth noting that the DNN, despite its relatively simpler four-layered structure, also delivers considerable results. The within-split shared weight accuracy achieved by the reference implementation in [17] is 85.8%. It's important to mention that this reference architecture involves the construction of channel-wise autoencoders. Consequently, both my LSTM and DNN architectures yield substantial results when compared to the reference implementation. Also, the accuracy metrics for the binary label with PCA is 69.32%. Additionally, I have also performed PCA to reduce the dimensionality by taking the top 30 principal components of the data. The test accuracy generated by PCA is 69.32%. We can see that the accuracy scores of models utilizing reduced feature sets do not closely align with those using all available features. However, the computation time is extremely fast as the number of channels dropped significantly from 64 to 30. On the other hand, for EEG image data, ResNet18 has given the best test accuracy and F1 score of 87.71% and 0.952 respectively. This is due to its deeper architecture. Notably, the training loss has undergone a substantial reduction, reaching a minimal value of 0.0010, and it has achieved the highest training accuracy among all the models. Furthermore, the CNN classifier has demonstrated a decent performance, yielding an F1 score that closely aligns with that of ResNet18. This states that the labels have undergone a good boundary separation. The reference within-split accuracy of Normal Image wise encoders [17] is 0.917 which closely aligns with the within-split accuracy of my models.

In the context of cross-subject split accuracy on time series data, LSTM performs better than the DNN architecture with test accuracy being 76.95% and F1 score of 0.876. However, DNN's result is very close to that of LSTM. Similarly, ResNet-18 performs better than the CNN network on the EEG image data with an accuracy of 68.96% and 0.819 F1 score. The reference cross-subject accuracies for time-series and image datasets are 73.1% and 75.6% respectively.

Overall, LSTM and ResNet-18 architectures are performing well on time-series and EEG image datasets. However, simple neural network architectures like DNN and CNN are also giving a considerable performance on the test set.

6 Conclusion and Future Work

To conclude this report talks about LSTM and Deep Neural Networks for a time-series dataset, CNN, and ResNet for EEG Images. It also gave a comparative analysis of within-subject and cross-subject binary label accuracy and F1 score metrics on the dataset and with the reference implementation. It was observed that the LSTM and ResNet structures outperform the traditional neural network for time series and the CNN-based classifier for the EEG images dataset. Additionally, the introduction of Principal Component Analysis (PCA)[13] as a preprocessing step on the time-series data, has substantially reduced the model's complexity.

As a next step in the future, we can perform Multi-class classification on the label y_stimulus using the above DNN, LSTM, CNN, and ResNet architectures. As the number of trials is comparatively low, primary research should take place in data augmentation. However, the proper consent of the subject should be taken as a reflection of Ethics, and data anonymity should be maintained.

References

1. Agarap, A.F.: Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375 (2018)
2. Bashivan, P., Rish, I., Yeasin, M., Codella, N.: Learning representations from eeg with deep recurrent-convolutional neural networks. arXiv preprint arXiv:1511.06448 (2015)
3. Begleiter, H.: EEG Database. UCI Machine Learning Repository (1999), DOI: https://doi.org/10.24432/C5TS3D
4. Farsi, L., Siuly, S., Kabir, E., Wang, H.: Classification of alcoholic eeg signals using a deep learning method. IEEE Sensors Journal **21**(3), 3552–3560 (2020)
5. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. Lawhern, V.J., Solon, A.J., Waytowich, N.R., Gordon, S.M., Hung, C.P., Lance, B.J.: Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. Journal of neural engineering **15**(5), 056013 (2018)
7. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks **3361**(10), 1995 (1995)
8. Li, Y., Dzirasa, K., Carin, L., Carlson, D.E., et al.: Targeting eeg/lfp synchrony with neural nets. Advances in Neural Information Processing Systems **30** (2017)
9. Lisa, M., Bot, H.: My awesome research software. Journal Title **1**(1), 1–10 (9 2021). https://doi.org/10.0000/00000

10. Mahmood, A., Bennamoun, M., An, S., Sohel, F., Boussaid, F., Hovey, R., Kendrick, G., Fisher, R.B.: Deep learning for coral classification. In: Handbook of neural computation, pp. 383–401. Elsevier (2017)
11. Nagabushanam, P., Thomas George, S., Radha, S.: Eeg signal classification using lstm and improved neural network algorithms. Soft Computing **24**, 9981–10003 (2020)
12. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32. pp. 8024–8035. Curran Associates, Inc. (2019), bluehttp://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
14. Ramzan, F., Khan, M.U.G., Rehmat, A., Iqbal, S., Saba, T., Rehman, A., Mehmood, Z.: A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using resting-state fmri and residual neural networks. Journal of medical systems **44**, 1–16 (2020)
15. Schirrmeister, R.T., Springenberg, J.T., Fiederer, L.D.J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., Ball, T.: Deep learning with convolutional neural networks for brain mapping and decoding of movement-related information from the human EEG. CoRR **abs/1703.05051** (2017), bluehttp://arxiv.org/abs/1703.05051
16. Tran, Y.: Eeg signal processing for biomedical applications (2022)
17. Yao, Y., Plested, J., Gedeon, T.: Deep feature learning and visualization for eeg recording using autoencoders. In: International Conference on Neural Information Processing. pp. 554–566. Springer (2018)