# Enhancing EEG Signals Classification using LSTM-CNN Architecture

**SWALEH M. OMAR[1], MICHAEL KIMWELE[2], AKEEM OLOWOLAYEMO[3], AND DENNIS M. KABURU[4]**
1School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya (e-mail: omari.swaleh@students.jkuat.ac.ke).

2School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya (e-mail: mkimwele@jkuat.ac.ke ).

3Department of Computer Science, Kuliyyah of Information and Communication Technology, International Islamic University Malaysia (e-mail: akeem@iium.edu.my ).

4School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya (e-mail: dennis.kaburu@jkuat.ac.ke ).

Corresponding author: Swaleh M. Omar (e-mail: omari.swaleh@students.jkuat.ac.ke).
This project was partly funded by Africa-ai-Japan

## Abstract

Epilepsy is a condition that disrupts normal brain function and sometimes leads to seizures, unusual sensations, and temporary loss of awareness. Electroencephalograph (EEG) records are commonly used for diagnosing epilepsy, but traditional analysis is subjective and prone to misclassification. Previous studies applied Deep Learning (DL) techniques to improve EEG classification, but their performance has been limited due to dynamic and non-stationary nature of EEG structure. In this paper, we propose a multi-channel EEG classification model called LConvNet, which combines Convolutional Neural Networks (CNN) for spatial feature extraction and Long Short-Term Memory (LSTM) for capturing temporal dependencies. The model is trained using open source secondary EEG data from Temple University Hospital (TUH) to distinguish between epileptic and healthy EEG signals. Our model achieved an impressive accuracy of 97%, surpassing existing EEG classification models used in similar tasks such as EEGNet, DeepConvNet and ShallowConvNet that had 86%, 96% and 78% respectively. Furthermore, our model demonstrated impressive performance in terms of trainability, scalability and parameter efficiency during additional evaluations.

Keywords: EEG, LSTM-CNN, Spatial Features, Temporal Dependencies

## I INTRODUCTION

Epilepsy is a temporary disturbance in the normal functioning of the brain that often results in seizures or unusual sensation and occasionally, loss of awareness. The condition is diagnosed using an Electroencephalography (EEG) test, which involves placing metal electrodes on or inside the skull to detect electrical pulses representing brain activities. EEG test records brain's rhythm as continuous electrical wave frequencies over time, measured in seconds (Hertz) vs amplitude measured in microvolts. The presence of distinctive rhythmic waves with specific morphologies in the EEG, known as epileptiform activity, may indicate epilepsy [1].

Traditional interpretation of EEG relies on Human Expert Judgement, which is prone to bias and can lead to misdiagnosis. About 30% of patients seen at specialized epilepsy clinics in America are misdiagnosed [2]. In addition, due to time-limited nature of EEG, subsequent tests cannot cancel the previous ones exacerbating the consequences of misdiagnosis. This has led to increased interest in epilepsy monitoring, detection and prediction using Deep Learning (DL) techniques.

DL Sequence models such as RNN's, LSTM, GRUs and Transformer Networks gained popularity for language modelling due to their ability to analyze time series data. These models represent each word in a sequence as token and performs mapping to a low level fixed-length vector space, learning context information and place similar tokens closer to one another. Representing Continuous time series data like EEG proves challenging due to ambiguity in determining tokens. Researchers have attempted to resolve this issue using segmented windows, scalograms sequence, spectrograms and motifs in time series mining to learn relevant temporal features for EEG representation, but still, this fails when interesting pattern lies hidden within common patterns in the sequence [3]-[5]. Moreover, Sequence models lacks the capability for good generalization for spatial information inherit within EEG data.

Originally, Convolution Neural Network (CNN) was designed for computer vision due to its ability to learn spatial information, however, recent research have enable to apply CNN for classification of various data formats. In EEG signal processing, CNN have been used to learn spatial features that are characteristics of different EEG signals in classification tasks.

This study suggests a method to enhance the classification of EEG signals by combining an RNN-based sequence modelling technique such as LSTM with a CNN model. The aim is to enhance the classification performance of EEG tasks while taking advantage of LSTM's capacity to identify temporal dynamics and CNN's capability to extract spatial patterns.

### A. CONTRIBUTION OF THE STUDY

This study makes contribution to the fields of Brain Computer Interactions (BCI) and Signals processing by proposing LConvNet Model, a novel approach that combines the strengths of recurrent neural network (RNN) based sequence modeling techniques, such as Long Short-Term Memory (LSTM), and convolution operation. By leveraging LSTM's ability to capture temporal dynamics and Convolutional Neural Network (CNN) capability to detect spatial patterns. The proposed method proved successful in enhancing EEG classification by outperforming existing models developed for similar EEG classification task.

## II RELATED WORK

### A. EEG BRAIN RHYTHMS AND EPILEPTIFORM

An epileptic seizure, is a temporary occurrence of abnormal excessive brain activity, causing an imbalance of excitatory and inhibitory forces [6]. It can either be focal or generalized, with

symptoms such as loss of consciousness, uncontrolled movement and unusual sensations. Focal seizures are caused by abnormal activities in a specific part of the brain and is further classified as simple or complex partial seizure, whereas, Generalized seizures affect the entire brain and can be categorized into Tonic, Atonic, Clonic, Myoclonic and Tonic-Clonic (Convulsive) seizures [7],[8].

The primary test for detecting epilepsy is the Electroencephalography (EEG). EEG records electrical activities in the brain using metal electrode discs and presents the data as signals [9]. In a normal brain, EEG readings show four different patterns; Alpha waves (8 to 12 Hz) are seen in inattentive brain, drowsiness; Beta waves(15 to 60 Hz) depicts an alert brain; Delta waves (1 to 5 Hz) are common in childhood during waking up and in adults mostly in sleep. Theta waves (4-8 Hz) are witnessed in children below the age of five years [8]. Several EEG morphologies may suggest non-epileptic brain rhythms which may suggest abnormal cerebral functions without specific etiology, yet others are noise commonly known as artifacts.

Epileptiform activities are brain rhythm abnormalities that could be associated with seizures but usually require clinical correlation [10]. Misdiagnosis of epilepsy can have severe consequences since EEG test is time bound, subsequent test cannot rule out previous one.

### B. SIGNALS PREPROCESSING AND REPRESENTATION

Analyzing bio signals such as EEG is complex due to its nonstationary and nondeterministic nature [11], [12]. To improve the performance of DL models in subsequent stages of data analysis, thorough signal preprocessing is crucial. Various preprocessing techniques have been proposed including the use of; Hybrid feature extraction algorithms [13], Independent Component Analysis [14], Fast Fourier Transform [15], Discrete Wavelet Transform [16], Eigenvector [17] and Autoregressive Techniques [18] depending on subsequent analysis task and expected outcome.

Enhancement of approaches in DL have seen several studies applying different models in studying EEG data for purposes of understanding brain temporal and state dynamics [19]-[21]. DL algorithms are incapable of high-level language processing, hence, rely on numerical operations for purposes of representation. In Language modelling, tokenization techniques such as word embedding improved generalization of Sequence to Sequence models for Natural Language Processing applications [22]. Representation for non-obvious sequence data still presents challenges due to ambiguity in tokenization [23]. Attempts to resolve this include converting raw signals to meaning representation that can be used for downstream tasks [24], [25]. CNN have emerged as credible competitors in one dimensional tasks such as audio, text and time series analysis [17]. Some CNN-based models have proved successful in deriving good feature representation based on distinct characteristics of input signals in Brain-Computer Interface challenges [26], [27].

### C. CLASSIFICATION OF EEG SIGNALS

Earlier techniques used for EEG signal classification relied on raw time-domain EEG data manipulation with traditional statistical methods such as mean, standard deviation,

probability and normal distribution [28], [29]. Frequency Domain techniques such as power spectral density and spectral entropy improved classification tasks by leveraging on spatially mapping EEG features from different frequency bands [30], [31]. Frequency Domain Features yields richer representation of EEG signals by extracting more relevant latent features from its spatial space, however fails in representation of temporal information. With the advancement of Deep Learning (DL), various architectures such as CNN and RNN have been proposed for EEG classification. These have demonstrated high accuracy in classifying EEG signals for various applications [32].

For further improvement, some studies proposed hybrid approaches that combine multiple techniques and algorithms such as combination of time-domain and frequency-domain features, hybrid and even ensemble algorithms to achieve better performance [33]. However, despite many examples of impressive progress, there's still much room for improvement and research in this field to develop more accurate and efficient EEG classification methods.

Shallow Convolution Networks (ShallowConvNets) are a type of CNN architecture that performs image classification tasks using fewer layers and small filters making them computational efficient and deployable on mobile and embedded devices [34]. ShallowConvNets have also been applied to other applications including EEG decoding and classification tasks [35], [36]. Deep Convolution Neural Networks (DeepConvNets) architecture, on the other hand, have been highly successful in computer vision tasks, including image classification, object detection and segmentation. Notable DeepConvNets include Unet [37] and Inceptionet [38]. Unlike ShallowConvNets which has smaller number of layers, DeepConvNets are characterized by a large number of convolution layers which enable them to learn hierarchical representation of the input data with much finer detail.

Majority of CNN architectures have typical structure consisting of inputs, kernels, pooling layers, and output (feature maps) for analyzing image data. In the case of time series signals such as EEG, CNN performance is limited due to:

i. Specific characteristics of EEG data: EEG signals are time series whereas images are spatial in structure, rendering it difficult for extraction of useful features.
ii. High noise to signal ratio: EEG data are often contaminated with various types of noise and artifacts making it difficult for CNN to distinguish.
iii. Temporal Dynamics: EEG signals are dynamic in nature and the patterns of neural activity can change rapidly over time which make it difficult for static nature of CNN operations to capture temporal dynamics of EEG signals

To mitigate against the following limitations, a number of researchers resolved in application of CNN to EEG data by spatial mapping using spectrograms or images. However, such representations discards temporal information which is useful for classification purposes. Recent publications in Brain Computer Interface (BCI) field demonstrated that different types of convolutions operations can enable extraction of various important EEG

features. EEGNet [39] investigated the roles of Separable Convolution network for spatial mapping of EEG features using several BCI tasks and achieved impressive results.

Researchers such as [40], investigated the impact of design architectural choices and training strategies in shallow and deep convolution networks, along with the use of machine learning techniques like batch-normalization and exponential linear units, on the decoding accuracies of EEG signals. Their study demonstrated improved performance surpassing widely used spectral power modulation EEG algorithms.

In this paper we propose a multi-channel EEG classification model, called LConvNet that combines functionalities of CNN and LSTM algorithms for classification of EEG data. Whereas CNN learns spatial features; LSTM in conjunction with the time distributed flatten and dense layers, captures and processes long term dependencies from the temporal EEG data. Subsequently, global average pooling layer is applied to summarize the learned features from the input tensor into fixed length vector which are concatenated with outputs from CNN-LSTM for deeper spatial and temporal features' representation. We have demonstrated that, our hybrid model improve EEG classification by outperforming existing EEG classification models such as EEGNet [39], ShallowConvNet and DeepConvNet [40] in performance and efficiency. To the best of the author's knowledge, this is the first implementation of LSTM-CNN in binary classification of EEG data.

## III. MATERIALS AND TOOLS

### A. EXPERIMENTAL SETUP

Data access from TUH database was done through MobaXterm toolbox for remote computing. Data was saved in .EDF format. Visualization, preprocessing, feature extraction and classification was implemented through Python programming, MNE version 1.2.2 environment, with Tensorflow and Keras platforms. The machine components consisted of Paperspace Gradient IDE, NVIDIA A4000 Graphics, 45 GB storage space and 8CPU/16GPU RAM.

### B. DATASET AND PREPROCESSING

This study uses open source TUH EEG Epilepsy Corpus (TUEP) v2.0.0, all procedure conducted pertaining to the dataset is in accordance with the Declaration of Helsinki and conforms to the HIPAA Privacy rule [41]. EEG data was arranged in two classes of 49 epileptic and 49 non-epileptic sessions with varied durations, sampling frequencies and number of channels all saved in .EDF format.

During preprocessing, all EEG channels were first set to average reference to enable ease in approximation by obtaining the difference between the electric potential of signal in its location and the average of all channels using the following equation:

$$Y(t) = x_{i(t)} - mean\left(x_{1(t)}, x_{2(t)}, \dots, x_{n(t)}\right)$$

This was followed by resampling at the rate of 128Hz for the purpose of improving quality and reliability using:

$$y[i] = x[round(i * r)]$$

Where *x* is the original data, i is the index of the sample, *y* is the output resampled data, r is the ratio of new sampling rate to the original sampling rate, and *round()* is the nearest integer function.

We then apply band-pass filtering between 1 and 45Hz respectively for noise reduction and artifact correction using MNE's Finite Impulse Response [42] to selectively pass frequencies within a certain band , application of Independent Component Analysis (ICA) to achieve the same especially in removal of eye movement artifacts yielded no further improvement in the context of this study. Further, all data was cropped between 1 to 200 seconds for uniformity. None-EEG signals used for triggering seizure and measuring pulses such as Photic and Electrocardiogram (ECG) were removed. The data was then segmented into fixed equal duration windows of 2 seconds each referred to as epochs to improve interpretability of the data through detection of specific events or stimuli within the signals using:

$$y_{i[n]} = x[i * N + n]$$

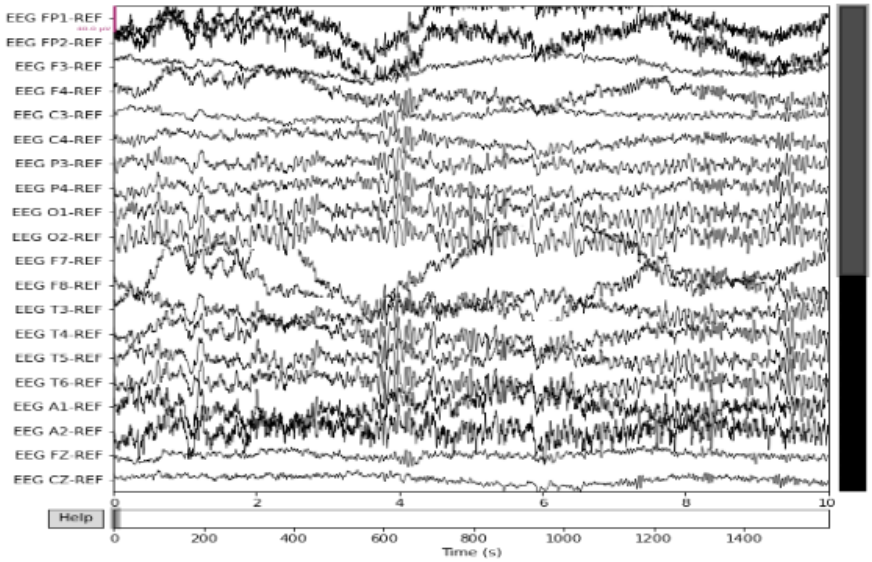
Where x represents continuous EEG data, *i* represents the index of the epoch or window, *N* represents the length or duration of each epoch or window and n represents the index of the sample within each epoch or window. Principal Component Analysis (PCA) was applied for dimensionality reduction at a standard 25 signals components using:

$$X = USV^T$$

Where X is the signal's matrix, U is matrix of principal components, S is the diagonal matrix and VT is the matrix of loadings.

Finally, the output from preprocessing stage is converted into a 3D array for subsequent analyzing steps. FIGURE 1 shows a sample of raw EEG data (A) and its output (B) after preprocessing operations from an epileptic class.

**(A)**




**(B)**

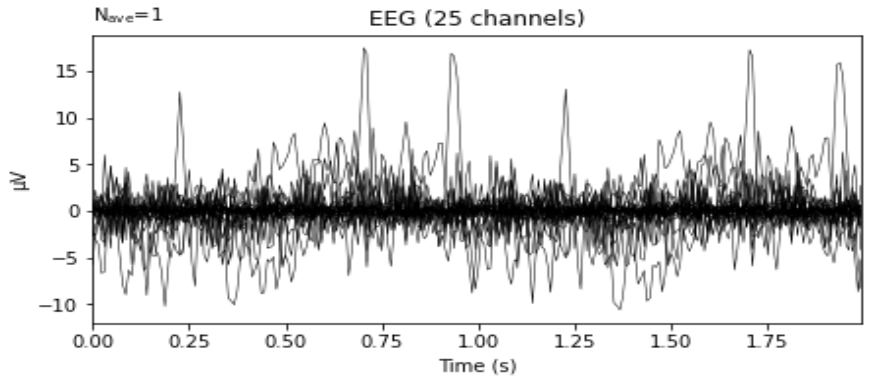



FIGURE 1 EEG Record. **(A)** Raw Data, **(B)** Preprocessed Data

### C. ARCHITECTURE OF LCONVNET MODEL

Our LConvNet model constitute CNN and LSTM components for extraction of spatial features and temporal dependencies respectively from preprocessed EEG data for the purpose of performing binary classification of epilepsy and healthy classes.

1) CNN Component

The CNN component takes EEG epochs converted to 3D input tensor X of shape (nsamples, ntimesteps, nchannels) where nsamples refers to number of EEG samples, ntimesteps refers to number of EEG time steps and nchannels is the number of EEG channels derived from previous preprocessing step where $X = \{x1, x2. . ., xn\}$. From this, we derive spatial rich feature maps using three convolution layers with incrementing filter sizes (16, 32, 64) and kernels (3x3, 5x5, 7x7) denoted using the following convolution equation, where $x$ is the input sample, $i$ is an instance of the samples, $w$ and $b$ are convolution weights and bias respectively, Relu activation function $f$ is used:

$$f\left(\sum_{\{i=1\}}^{\{N\}} * \ (w, x) + \ b\right)$$

We increment convolution filters by a factor of 2 as explained to allow capturing features from high temporal resolution inherited in sampling frequency of 128 used during preprocessing, this proved effective in past similar tasks [39], and in addition, this enables the model to capture patterns at different scales from the input data. Smaller filters such as 16, capture finer frequency details and edges, whereas larger filters such as 64 capture more global patterns and structures. The choice of the kernel sizes determine the receptive field, which is the area of the input that it considers when computing the output. The choice of kernel sizes is based on the tradeoff between capturing fine details and edges versus capturing larger patterns and structures of the input data. This, choice of filters and kernels is designed not only to capture data at different scales but, subsequently, allowing the model to learn representations that are robust to variation in the input. Relu activation function (f) is applied to convolution layers to reduce nonlinearity, dropout operation to reduce overfitting while same padding and max pooling to maintains spatial dimension.

The output from the CNN layer is then passed to a Time Distributed Flatten Layer (T) that flattens the output of the CNN component across time-steps to produce a 2D feature map then send the same to a Time Distributed Dense Layer that applies a dense neural layer with 64 units to each time step of the flattened feature map (M), hence

$$M = T(Y)$$

2) LSTM Component

Application of LSTM in this study is to enhance subsequent classification task by complementing CNN spatial features with EEG temporal dependencies. The input M derived from previous convolution and time distribution operations is split into 64 LSTM time steps *(t)*, we find 64 to produce best results. At each timestep t, the LSTM layer takes two inputs: the current input vector *x(t)* and the previous hidden state *a(t-1)*. Using these inputs, it computes three gates – the forget gate, the update gate and the candidate memory. The following equation depicts computation of unidirectional LSTM layer as is applicable to our model.

$$f_{r(t)} = \ \sigma\big(W_f[a(t-1), x(t)] + \ b_f\big)$$
$$u_{r(t)} = \ \sigma(W_u[a(t-1), x(t)] + \ b_u)$$
$$\tilde{c}(t) = \ tanh\big(W_{c[a(t-1),x(t)]} + \ b_c\big)$$
$$c(t) = \ f_{r(t)} \odot c(t-1) + \ u_{r(t)} \odot \tilde{c}(t)$$
$$o_{r(t)} = \ \sigma\big(W_{o[a(t-1),x(t)]} + \ b_o\big)$$
$$a(t) = \ o_{r(t)} \odot tanh\big(c(t)\big)$$

Where: $f_{r(t)}$ is the forget gate, $u_{r(t)}$ is the update gate, $\tilde{c}(t)$ is the candidate memory, $c(t)$ is the new memory, $o_{r(t)}$ is the output gate, $a(t)$ is the hidden state, $W_i$ and $b_i$ are corresponding weights and bias.

3) Concatenation

Subsequently, the input tensor X is passed to 1D Global Average Pooling Layer along the time axis to obtain an average of each feature across the time dimension using the following equation.

$$\mathrm{Pi} = \left(\frac{1}{T}\right) * \left(\sum_{\{t=1\}}^{\{T\}} \mathrm{X_{it}}\right)$$

The resulting vector $P_i$ represents the $i^{th}$ feature of the tensor P. T is the sequence length in the input tensor X. $X_{it}$ denotes the value of the $i^{th}$ feature at time step t in the input tensor X. P is of the same shape as the hidden state a(t) calculated previously by the LSTM operation. Besides producing informative information pertaining to input data, we discovered that, in addition, the global average pooling operation helped in inhibiting overfitting by reducing spatial dimensions of the input, hence introducing a form of regularization. The output tensor P is then concatenated with hidden state $a(t)$ to produce tensor C using equation: $C = (a(t) \& P)$

This step allows the spatial features captured by the CNN component to be combined with the temporal dependencies captured by the LSTM component.

4) Dense Neural Network

The resulting concatenated Tensor is then passed through a final dense neural network with a sigmoid function for binary classification. The dense layer's equation is:

$$Y = \sigma(w * C + b)$$

Where Y is a 2 dimension matrix, σ is a sigmoid activation function, w is a 3 dimension weight matrix of shape = (nlstmunits, nchannels, nclasses), C is a 2 dimension concatenation results of the CNN-LSTM components output and output from the global average pooling layer, while b is a 1 dimension vector of shape = (nclasses). Multiplication operation $(*)$ between w and C is achieved through broadcasting. FIGURE 2 shows the architecture of our LConvNet model as described in this section.

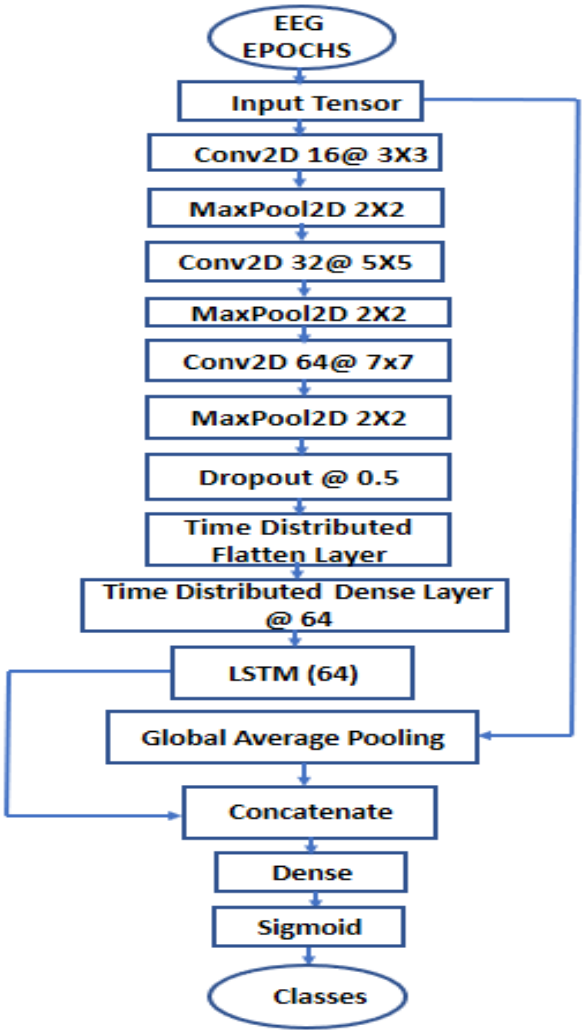



FIGURE 2. Architecture of LConvNet Model

The model is finally compiled using Adam optimizer of 0.0001 and binary cross entropy loss function as shown in the following equation.

$$-(y\ \log(p) + (1-y)\log(1-p))$$

Where y is the actual classification for an observation and p the predicted observation.

**D. MODEL TRAINING**

For the classification task, we employed a stratified approach to divide the data into training and validation sets with a 20% split, ensuring a balanced representation of both classes. The data was then normalized to have a mean of 0 and a standard deviation of 1. During training, a batch size of 128 and 200 iterations/epochs were chosen. These parameters were selected based on their ability to maximize efficiency, effectiveness, and prevent overfitting, resulting

in optimal performance. TABLE 1, shows the components of our LConvNet model after training.

**TABLE 1.** LConvNet Architecture Input and Output Components

| Layers | Input (nsamples ntimestep, n_channels) | Operations | Output Shape | No. of Parameters |
|---|---|---|---|---|
| | (1, 256, 25) | Reshape | (1, 25, 256, 1) | 0 |
| 1 | (1, 25, 256, 1) | Conv2D, Filter @ (3*3), Kernels @ 16, MaxPooling2D @ (2,2), Padding =same, Activation = Relu | (1, 12, 128, 16) | 160 |
| 2 | (1, 12, 128, 16) | Conv2D_1, Kernels @ 32, Filters @ 5*5, MaxPooling2D @ (2*2), Padding =same, Activation = Relu | (1, 6, 64, 32) | 12832 |
| 3 | (1, 6, 64, 32) | Conv2D_2, Kernels @ 64Filters (7*7),MaxPooling2D @ (2*2), Padding =same, Activation = relu Dropout (rate = 0.5) | (1, 3, 32, 68) | 106692 |
| 4 | (1, 3, 32, 68) | TimeDistributed(Flatten) | (1, 3, 2176) | 0 |
| 5 | (1, 3, 2176) | TimeDistributed(Dense)(32) | (1, 3, 32) | 69664 |
| 6 | (1, 3, 32) | LSTM(32) | (1,32) | 8320 |
| 7 | (1,32) | GlobalveragePooling1D | (1,25) | 0 |
| | | Concatenate | (1,57) | 0 |
| 8 | (1,57) | Kernel_Constraint=max_norm( 0.5) Dense (sigmoid) | (None, 1) | 58 |
| Total Parameters | | | | 197, 726 |

## E. EEG FEATURE EXTRACTION METHODS

This section discusses three existing models that proved successful in extraction and classification of EEG data from previous literatures. The three models are used for comparison and benchmarking while evaluating our LConvNet model.

### 1) Feature Extraction using EEGNet

The inputs shape for EEGNet [39] model is taken as (C, T) where C is the number of EEG channels and T refers to EEG time steps. The architecture of the EEGNet consists of two main parts: EEGNet module and classification layer. EEGNet module is composed of three layers:

i. Depthwise Convolution Layer with filter size (C, 1). The Kernel Length (K) is half the EEG Sampling Rate. This layer learns features that are specific to electrode channels, hence, this layer is capable of extraction of latent temporal features of the EEG even and up to low frequencies.

ii. Pointwise Convolution Layer with filter size (1, 1) that combines the output channels from previous layer. This reduces the dimensionality and allows for fewer training parameters making the model more efficient.
iii. A Pooling layer that performs temporal aggregation across time dimension of the input data. This further reduces dimensionality while allowing extraction of relevant features.

For the purpose of our study, we substituted C with the number of preprocessed channels which equals to 25, T equals to 256 and K is half the sampling frequency we use which is 64. We substitute softmax with sigmoid for binary classification.

2) Feature Extraction Using ShallowConvNet and DeepConvNet

The ShallowConvNet proposed by [40] consists of a single convolutional layer, followed by average pooling layer and a fully connected layer. The convolution layer has 40 filters with 25 time points, which allows the network to capture both spatial and temporal information from the EEG signals. Relu activation function is applied to prevent instability caused by vanishing gradient. After the convolution layer, the output is fed into a global average pooling layer, which averages the activation values across all of the temporal and spatial dimensions. This helps reducing the dimensionality of the data hence prevent overfitting. The output of the global average pooling layer is then passed to a fully connected layer with units which produces a probability distribution over the classes.

The exact DeepConvNet architecture as proposed by [40] varies depending on specific EEG decoding but a common configuration consists of 4 to 5 convolution layers with increasing filter sizes, followed by batch normalization and dropout layers to prevent overfitting. The number of filters in each convolution layer increased from 25 to 100, allowing. DeepConvNet captures increasingly complex features. The filter sizes in the first two convolutional layers are set to 1 time points, while those of subsequent layers increased to capture large-scale features. The output is then passed to a global average pooling and fully connected layers for classification.

## IV. RESULTS

### A. PERFORMANCE EVALUATION

To evaluate performance, we first tested models' accuracies, precision, recall, f1 score and Macro-average F1, with a threshold of 0.7 on validation dataset using the following equations to derive subsequent analysis.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$MF1 = \sum_{\{i=1\}}^{\{C\}} \mathrm{F1}_{\mathrm{i}}C$$

Where TP is True Positives, FP is False Positives, FN is False Negatives, TN is True Negatives, C is number of classes(2). TABLE 2 shows summary of the findings.

**TABLE 2.** Performance Metric Values

| | | LConvNet | ShallowConvNet | EEGNet | DeepConvNet |
|---|---|---|---|---|---|
| Accuracy | | 0.97 | 0.78 | 0.86 | 0.96 |
| Class-wise Precision | 0 | 0.96 | 0.69 | 0.78 | 0.93 |
| | 1 | 0.99 | 1.00 | 0.99 | 0.99 |
| Macro-Avg Precision | | 0.97 | 0.85 | 0.88 | 0.96 |
| Class-wise Recall | 0 | 0.99 | 1.00 | 0.79 | 0.99 |
| | 1 | 0.95 | 0.56 | 0.98 | 0.92 |
| Macro-Avg Recall | | 0.97 | 0.78 | 0.86 | 0.96 |
| Class-wise F1-Score | 0 | 0.98 | 0.82 | 0.88 | 0.96 |
| | 1 | 0.97 | 0.72 | 0.84 | 0.96 |
| Macro-Avg Recall | | 0.97 | 0.77 | 0.85 | 0.96 |

Accuracy measures the overall performance of the model in predicting both positive (1) and negative (0) classes. The higher the accuracy, the better the model's performance. From the given metrics, we can see that our model LConvNet and DeepConvNet have the highest accuracies of 97% and 96%, respectively. EEGNet has an accuracy of 86% followed by the ShallowConvNet at 78% as depicted in TABLE 2.

Precision measures the proportion of true positives among all predicted instances. A higher precision means the model is good at identifying the positive classes. From the given metrics, we can see that LConvnet has the highest class-wise and macro-average precision for identification of both positive and negative classes of the validation dataset followed by DeepConvNet. ShallowConvnet and EEGNet has class-wise precisions and the smallest macro-average precisions. Recall measures the proportion of true positives among all actual positives instances. The highest micro-average recall and proportional class-wise recalls are obtained by LConvNet and DeepConvNet.

F1 score is the harmonic mean of precision and recall and a measure of the models' overall performance. From the results shown in TABLE 2, the LConvNet has the macro-average F1 Scores, followed by DeepConvNet, EEGNet and ShallowConvNet.

The four models' confusion matrices in FIGURE 3 shows the proportion of True Positives (second row, second column), True Negatives (first row, first column), False Positives (first row, second column) and False Negatives (second row, first column) for LConvNet **(A),** DeepConvNet **(B),** EEGNet **(C)** and ShallowConvNet **(D).**

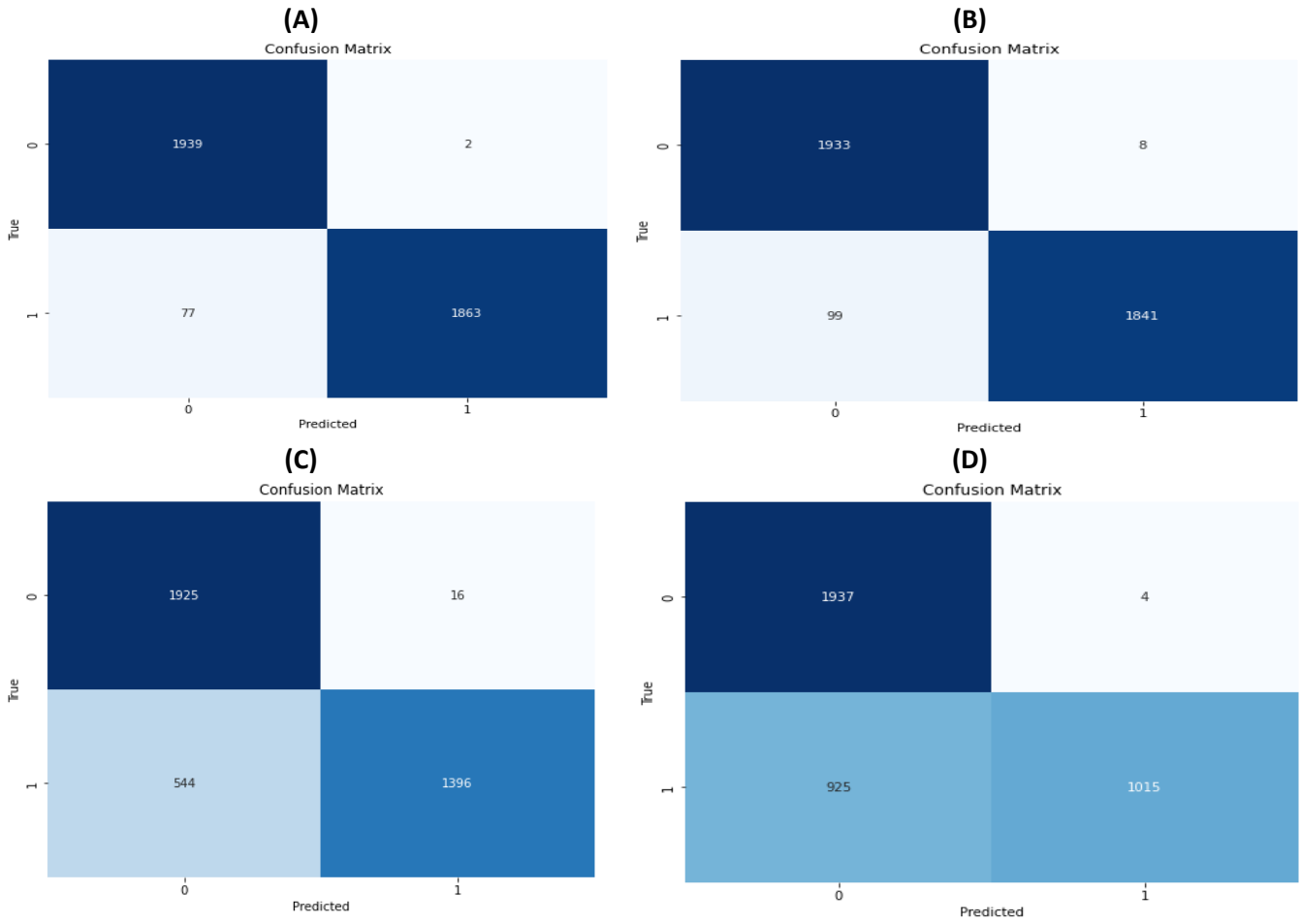



FIGURE 3 Confusion Matrices. **(A)** LConvNet, **(B)** DeepConvNet, **(C)** EEGNet and **(D)** ShallowConvNet

The Area Under a Curve (AUC) shows the overall performance of the model in distinguishing between positive and negative instances. AUC curve in FIGURE 4 shows comparison of the four models' performance (models distinguished by graph color as shown in legend). The higher the AUC value, indicates that the model has good discrimination ability and can separate the positive and negative samples well.

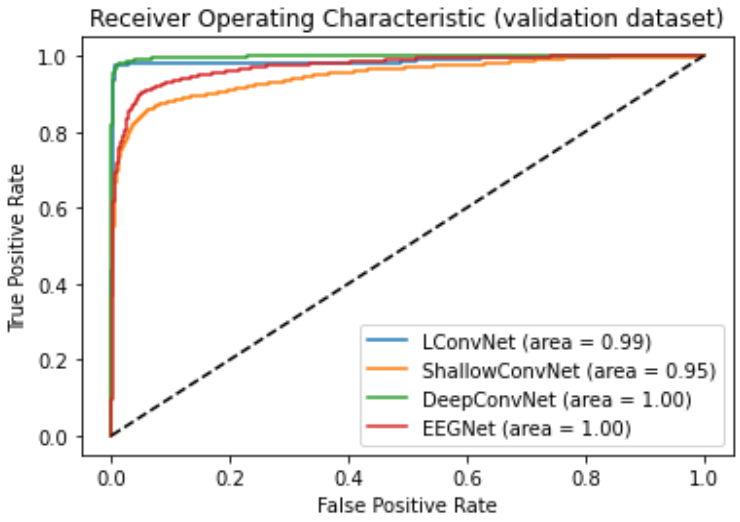



FIGURE 4 AUC for the four models for validation dataset

We further assess the level of agreement between the predicted and actual labels in each of the four models using Cohen's Kappa (CK) and Mathew Correlation Coefficient (MCC) using the following equations.

$$CK = \frac{P_o + P_e}{1 - P_e}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Cohen's Kappa measures the agreement between predicted and actual labels ($P_o$) , adjusted for chance ($P_e$), and ranges from -1(complete disagreement) and 1 (perfect agreement).

LConvNet, with a CK value of 0.9639 indicates a very high level of agreement between the predicted and true labels, with a small margin of error. The MCC value of 0.9642 confirms this high level of performance, indicating a very strong correlation between the predicted and true labels as shown in TABLE 3. DeepConvNet model indicates a relatively high level of agreement with CK of 0.9520, the MCC value of 0.9523 confirms this level of performance. EEGNet and ShallowConvNet indicate good and moderate level of performance respectively when distinguishing between the predicted and true labels with good values of 0.7995 and 0.7304 confirmed with MCC of 0.8058 and 0.7304 respectively.

**TABLE 3.** Model's agreement to actual and predicted labels

| | LConvNet | ShallowConvNet | EEGNet | DeepConvNet |
|---|---|---|---|---|
| Cohen Kappa | 0.9639 | 0.7304 | 0.7995 | 0.9520 |
| MCC | 0.9642 | 0.7527 | 0.8058 | 0.9523 |

In addition, we compare the models' robustness by assessing how well the models handle adversarial input samples. In this assessment, the adversarial samples are copies of the original input data manipulated by including perturbation generated using the Fast Gradient Sign Method (FGSM) attack with a perturbation epsilon ($\epsilon$) of 0.01, for the purpose of causing intentional misclassification to the model. The equation for generating the adversarial validation data is as follows:

$$y = x + \epsilon * \text{sign}\left(\nabla_x J\left(\Theta, x, y\right)\right)$$

Where *y* is the adversarial input, *x* is the input data, y is the ground truth label of the input data, $\epsilon$ denotes the degree for perturbation, a value of 0.01 is small enough to generate unnoticeable perturbation using human eye, which is desirable, *sign()* function ensures that the perturbation is aligned with the direction that maximizes the loss (i.e. the direction that makes the neural network most likely to make a mistake),$\nabla$ is the gradient of the loss function, $\Theta$ is our model, and J is the loss function.

**TABLE 4.** Robustness Evaluation

| | LConvNet | ShallowConvNet | EEGNet | DeepConvNet |
|---|---|---|---|---|
| Original data Accuracy | 0.96 | 0.48 | 0.98 | 0.90 |
| Adversarial d a t a Accuracy | 0.53 | 0.25 | 0.48 | 0.51 |
| Accuracy Difference | 0.43 | 0.23 | 0.49 | 0.39 |
| | | | | $\epsilon$ = 0.01 |

The models are evaluated based on their accuracy in correctly classifying the validation data. From TABLE 4, we can see that all four models have relatively higher accuracies on original validation dataset compared to adversarial validation dataset. This means that when the input data is not perturbed, the models can classify it with higher accuracy. However, on perturbed input data, the accuracy drops significantly, indicating lack of robustness against adversarial attacks. The degree of drops varies between models, with LConvNet and ShallowConvNet showing the least but significant drop in accuracy, while EEGNet and DeepConvNet performing relatively worst.

In summary all four models performed well on the classification task, but LConvNet and DeepConvNet stand out with very high levels of performance, while EEGNet and ShallowConvNet have well to moderate levels of performance.

**B. TRAINABILITY EVALUATION**

When assessing trainability of the models, we calculated the bias and variance by computing the mean predictions from actual values, and computing mean of the squared differences between the predictions and the mean predictions of the validation datasets respectively.

TABLE 5 shows LConvNet having the largest number of trainable parameters (197,726) among the four models. It has relatively low bias of 0.00130 which suggests that it is able to

fit the training data well, however, it has a relatively high variance (0.1812) which may indicate that it is overfitting the training data and may not generalize well to new data. ShallowConvNet has a relatively smaller number of trainable parameters (41,841) to LConvNet and a higher bias of 0.0777 which suggest it slightly underfitting the training data. However it has a relatively low variance of 0.0999, which suggest it may be more robust to generalize new data. EEGNet model has he smallest number of trainable parameters (753) among the models listed. It has relatively low bias (0.01741), which suggest that it is able to fit the training data well. However has a higher variance of 0.1619, which may indicate it is prone to overfitting. DeepConvNet has a larger number of trainable parameters (150,551) and a relatively low bias (0.0148), which suggests that it is able to fit the training data well, however, a higher variance of 0.20587 may indicate it is prone to overfitting.

**TABLE 5.** Trainability Evaluation

| | LConvNet | ShallowConvNet | EEGNet | DeepConvNet |
|---|---|---|---|---|
| Trainable Parameters | 197, 726 | 41,841 | 753 | 150,551 |
| Bias | 0.00130 | 0.0777 | 0.01741 | 0.01487 |
| Variance | 0.18124 | 0.0999 | 0.1619 | 0.20587 |

When evaluating the performance and generalization of the models, we gain a local perspective by considering variance and bias. Difference in outcomes can be attributed to factors like the number of trainable parameters, model architecture and complexity. To better understand the behavior of the models during training, we analyze Learning Curves (LC), which plot accuracy against training iterations (epochs) as shown in Figure 5. By examining the models' LC behaviors from FIGURE 5, we can observe that the training procedures for LConvNet (A) and EEGNet (C) models exhibit more stability and consistency in learning from the data compared to DeepConvNet and ShallowConvNet. This implies that LConvNet and EEGNet are likely to have better accuracy and generalization than DeepConvNet and ShallowConvNet.

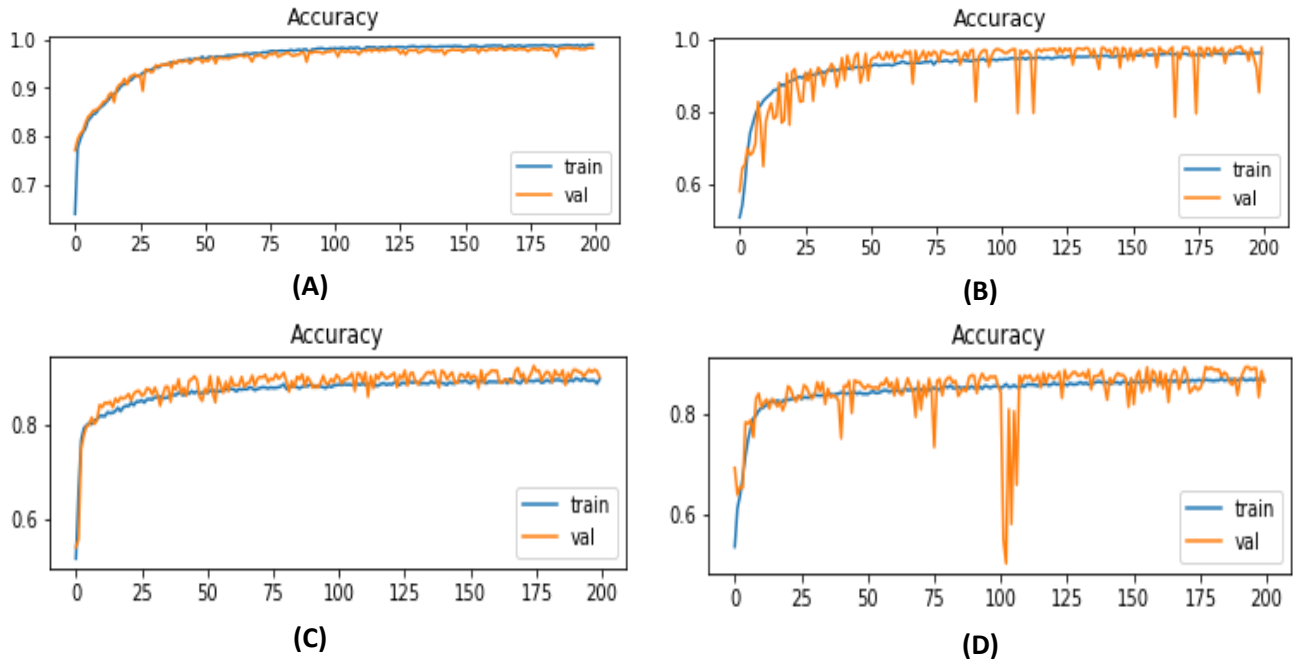



FIGURE 5 : Learning Curves LConvNet **(A),** DeepConvNet **(B)** , EEGNet **(C)** , ShallowConvNet **(D)**

**C. MODEL VISUALIZATION**

To access interpretability of the models, we performed feature visualization to understand the relevant features used for classification within layers of the models. All models except LConvNet displays highly diffuse and distinct features with high activation within layers, whereas LConvNet CNN layers (without LSTM component) displays more localized features with weak activation as shown in FIGURE 6. The colors in Figure 6 represents magnitude of activation, where, darker colors represent lower activation and lighter color represents higher activation as shown by the legend.

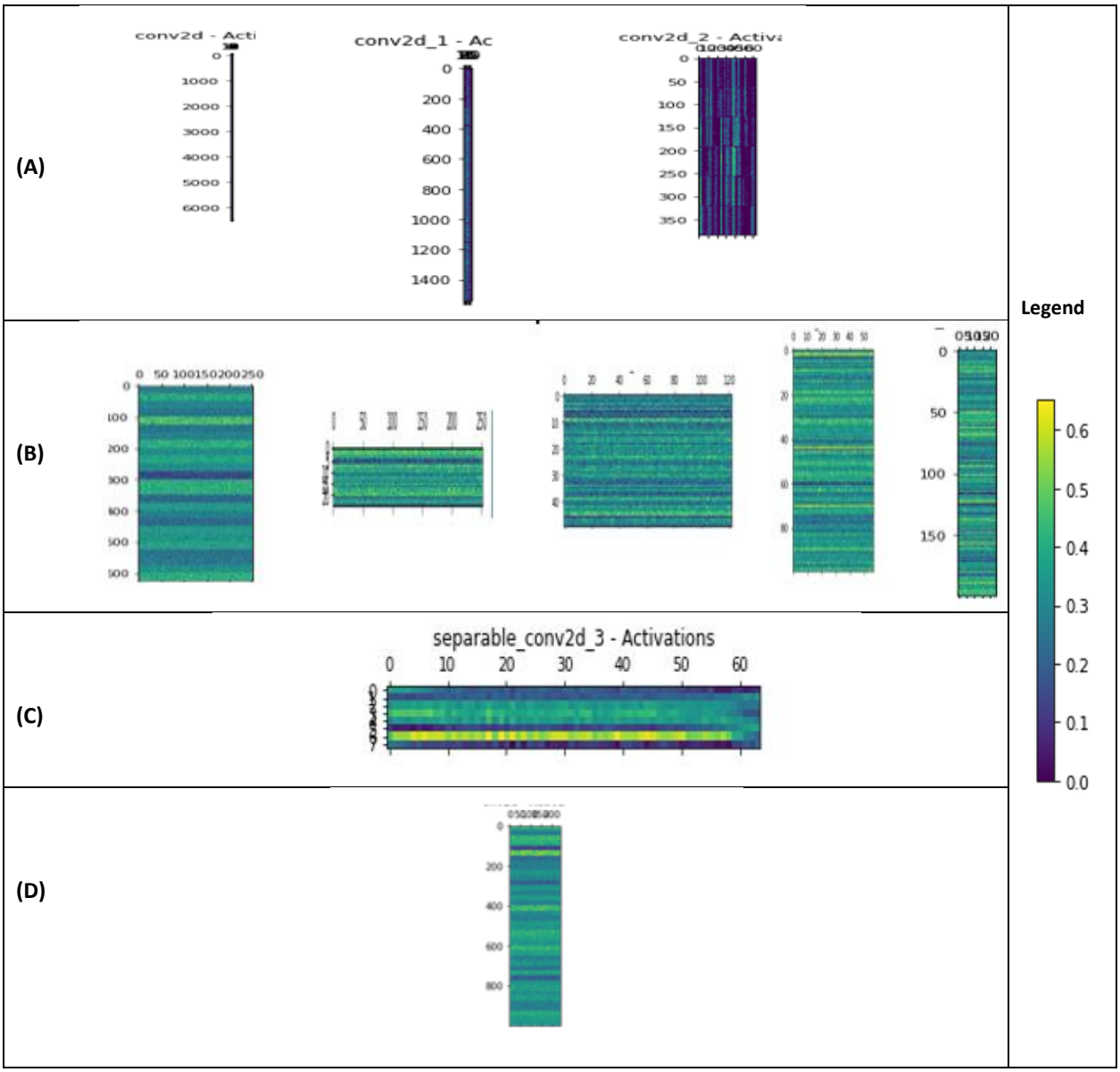



FIGURE 6 Layer Activation Heat Maps. LConvNet **(A),** DeepConvNet **(B),** EEGNet **(C),** ShallowConvNet **(D)**

## D. SCALABILITY AND EFFICIENCY METRICS

1) Training and Inference Times

To evaluate scalability of the models, we start by measuring models' efficiencies using the training and inference time. Training time refers to the amount of time taken to train machine learning model this includes learning the data and updating its parameters to minimize the loss function, inference time, on the other hand, refers to the amount of time it takes for the trained model to make predictions on new, unseen data.

TABLE 6. Scalability Evaluation

| | LConvNet | ShallowConvNet | EEGNet | DeepConvNet |
|---|---|---|---|---|
| Trainable Parameters | 197,726 | 41,841 | 753 | 150,551 |
| Training Time | 343.87 | 370.15 | 334.21 | 323.71 |
| Inference Time | 0.49 | 0.75 | 0.75 | 0.86 |

TABLE 6 shows LConvNet as having the highest number of trainable parameters among the four models. However, it has the fastest training time per parameter with a cumulative total of 343.87 seconds and the fastest inference time of 0.4924 seconds per parameter. This suggest that LConvNet may be highly scalable as it can train quickly on large datasets and make predictions in real time. With only 41,841 trainable parameters, ShallowConvNet has fewer parameters than LConvNet, however, it has a relatively slow training time of 370.15 seconds and relatively slow inference time of 0.7472 per parameter. This may suggest that ShallowConvNet may not be as scalable as LConvNet as it may take longer to train and make predictions on larger datasets. EEGNet has fewer numbers of trainable parameters at 753, however relatively slowest training time of 334.215 seconds and relatively slow inference time of 0.7493 seconds, hence may not be as scalable as LConvNet and may take longer to train and make predictions on larger datasets compared to LConvNet. With 150,551 trainable parameters, DeepConvNet has a moderate number of parameters compared to LConvNet and ShallowConvNet. DeepConvNet has a relatively fast training time of 323.71 seconds, but a slower inference time of 0.86 seconds compared to the rest. This suggest that DeepConvNet may be moderately scalable, as it can train relatively quickly on large datasets, but may take longer to make predictions in real-time applications compared to the rest.

In summary, LConvNet appear to be the most scalable of the four models, with the fastest training and inference times despite having the largest number of trainable parameters. ShallowConvNet and EEGNet have relatively fewer numbers of trainable parameters, but slower training and inference times compared LConvNet which may limit their scalability. DeepConvNet has a moderate number of trainable parameters and relatively fast training time, but its slower inference time may limit its scalability compared to LConvNet.

### 2) Parameter Efficiency

In this study, we assess Parameter Efficiency (PE) by measuring the average number of parameters processed by the models in a single training batch calculated using the following equation:

$$PE = \frac{N}{E * B}$$

For each input, where PE refers to parameter efficiency value, E is the number of training epochs and B is the training batch size. A higher efficiency value indicates that a model can handle a greater number of trainable units in a complex model in a single batch, which is desirable.

**TABLE 7.** Parameter Efficiency Evaluation

| | LConvNet | ShallowConvNet | EEGNet | DeepConvNet |
|---|---|---|---|---|
| Trainable Parameters | 197,726 | 41,841 | 753 | 150,551 |
| PE Value | 7.7236 | 1.6375 | 0.0294 | 5.8880 |
| Epoch=200, Batch Size = 128 | | | | |

TABLE 7 shows, LConvNet and DeepConvNet as the most parameter-efficient models among the four, as they can give better performance with a moderate number of trainable parameters. ShallowConvNet is also relatively efficient, while EEGNet requires a large number of epochs/or batch size to achieve same level of performance as the other three despite having very few trainable parameters.

## E. ANALYSIS OF LCONVNET TEMPORAL DEPENDENCIES

The preprocessing steps applied to raw data prior to classification, includes various activities that may have impact on the extraction of temporal dependencies by our model. In this section we highlight some possible temporal dependencies extracted by our LConvNet model.

### 1) Frequency Specific Temporal Patterns

The band pass filtering of raw EEG data between 1 to 45 Hz followed by subsequent segmentation of EEG data to a fixed duration epochs of 2 seconds with an overlap of 1 second, allows the model to capture relevant frequency specific temporal patterns.

We convert the preprocessed EEG data into frequency domains using bands ranging between 0.5-4.5 Hz (Delta), 4.5- 8.5Hz (theta), 11.5-15.5Hz (sigma), 15.5-30Hz (beta), 30-45Hz (gamma) using welch method to derive power spectral density (PSD) spectrograms [43]. Analysis of EEG data in these specific frequency waves is important since each frequency band is associated with different cognitive and neural processes in the brain as detailed in related literature section. We then deploy t-SNE (t-Distributed Stochastic Neighbor Embedding) with output space of 2 and perplexity of 5 to each of the spectrograms to enable visualization and gaining insight on our model's ability to distinguish between output classes. We fitted the results of each frequency band spectrogram to LConvNet model before application of t-SNE.

FIGURE 7 show that our LConvNet model distinguished clear separation between epileptic and non-epileptic classes with distinct properties given a specific timeframe epochs (2 seconds) with 1 seconds overlap. Gamma, Beta and Delta produced better classification results.

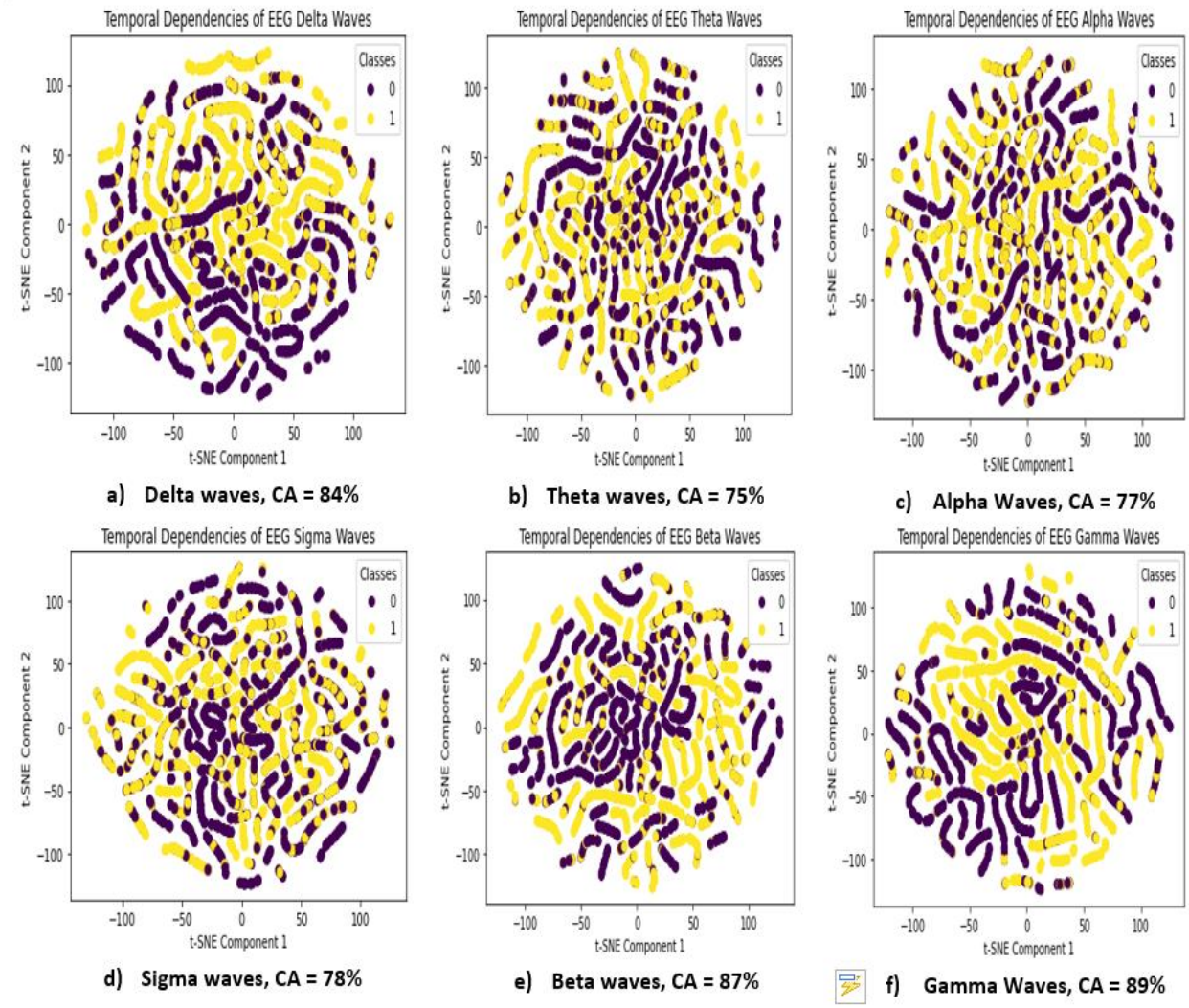



FIGURE 7 t-SNE Visualization of Frequency Specific Temporal Patterns. Delta Waves (a), Theta Waves (b), Alpha Waves (c), Sigma Waves (d), Beta waves (e), Gamma Waves (f). Yellow = epileptic class, Purple = Non-epileptic, CA=Classification Accuracy

2) Temporal Dynamics of EEG Activities

We initially sample EEG data at a rate of 128Hz, this means that for each second of EEG data, there are 128 data points, with sampling resolution t = 1 (sec)/128 (~ 8ms) that determines the level of temporal details to be captured within EEG signals. We assess our model using resolution of ~4ms and ~16ms. FIGURE 8, highlights the analysis of temporal dynamics of EEG data, a medium-high temporal resolutions of 8ms allows the model to capture fast brain processes that occur in milliseconds timescale, such as evoked potentials or transient oscillations, which are important in understanding cognitive processes and neurological disorders.

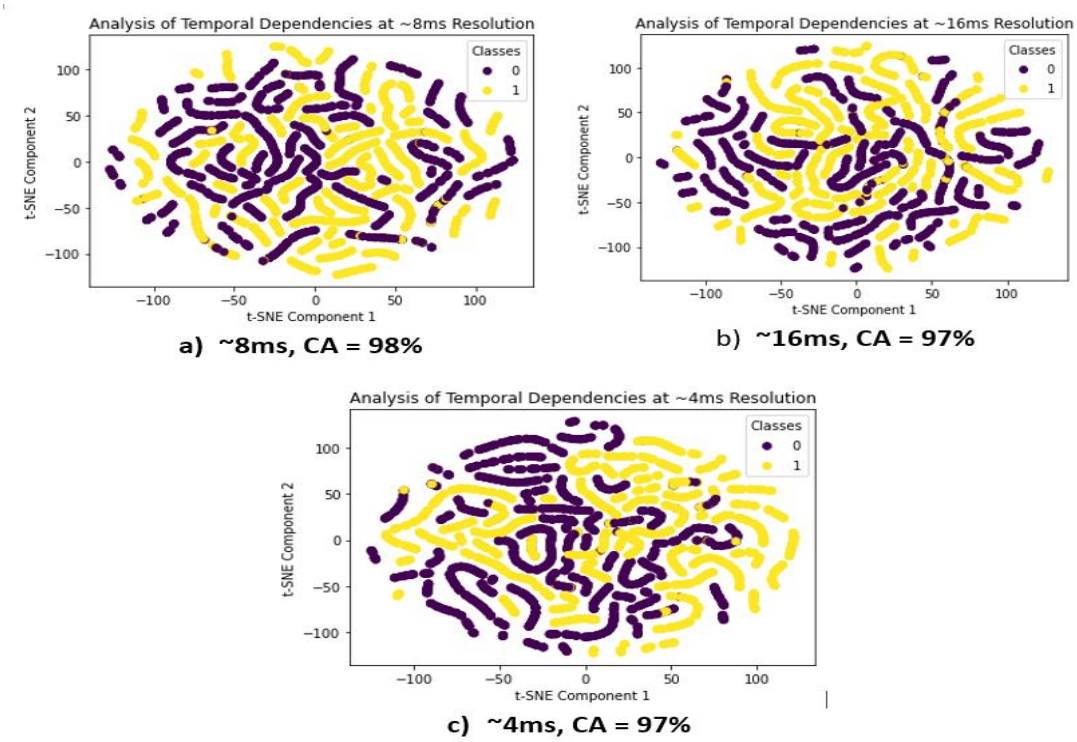



FIGURE 8 t-SNE Visualization of Temporal Dependencies at Varied Resolutions. (a) ~ 8ms, (b) ~ 16 ms, (c) ~ 4ms

**F. TEMPORAL CORRELATIONS BETWEEN EEG CHANNELS**

The EEG dimensionality reduction is preprocessed using PCA, which subsequently captures correlations between EEG channels such as brain regions and/or functional connectivity network. PCA operation standardized the EEG channels to 25 prior to segmenting data into epochs. The sample preprocessed EEG data is a 3D array of shape (nepochs, nchannels, ntimepoints) where nepochs is the number of epochs, nchannels the number of EEG channels and ntimepoints is the number of time steps.

$$r = \frac{\sum(x_i - x^-)(y_i - y^-)}{\sqrt{\sum(x_1 - x^-)^2 \sum(y^1 - y^-)^2}}$$

We assess the temporal correlations between EEG channels by calculating the Pearson Correlation Coefficient (r) between electrodes of EEG channels in an epoch as shown in the equation above where $x_i$ and $y_i$ are the individual observations values of the EEG channel's electrode, $x^-$ and $y^-$ are the means of EEG temporal dynamics for the respective EEG channels' electrodes. The results are depicted in FIGURE 9.

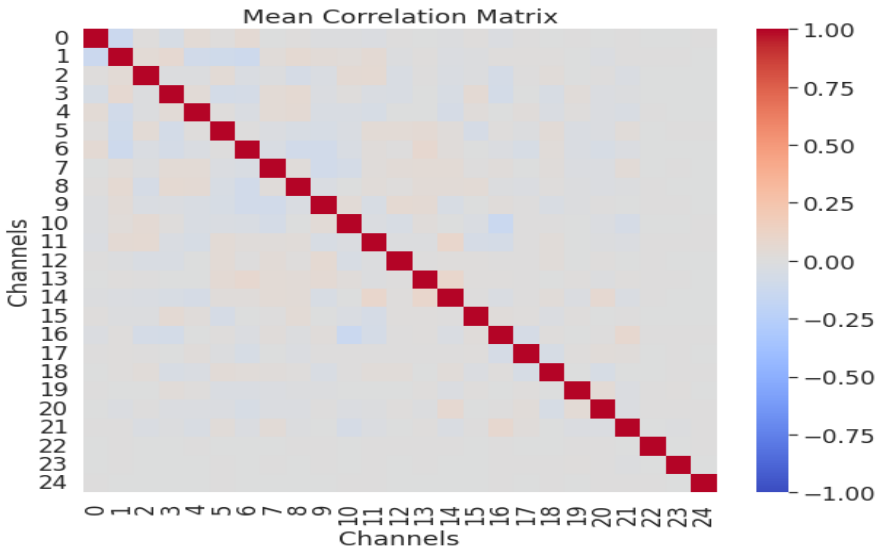



**FIGURE 9** Correlation Matrix of EEG Channels between EEG epochs

The Mean Correlation Matrix heatmaps shown in FIGURE 9, shows much stronger positive correlation between electrodes from similar channels of a sample preprocessed EEG data at a particular time point.

## V. DISCUSSIONS

### A. COMPARISON OF THE MODELS' BASED ON PERFORMANCE METRICS

In this study, we proposed LConvNet, a shallow LSTMCNN model that can generalize well in Time Domain EEG classification tasks. We first compare the model's performance against existing EEG classification models, highlighting strength and weaknesses using metrics such as accuracy, precision, recall, F1-Score, Mathew's Correlation Coefficient and Cohen's Kappa. We note that LConvnet has best overall performance followed by DeepConvNet, EEGNet and ShallowConvNet. Similarly, based on AUC shown in FIGURE 4, LConvNet has the best discriminating ability among the four models, which is consistent with its high accuracy and good balance between precision and recall. In summary, the AUC values support the performance metrics described earlier.

Subsequently, the model robustness was assessed using adversarial data, of concern, the accuracy of all models dropped significantly. This indicate that models are not robust to handle such attacks. This is critical, since adversarial attacks are a growing concern in sensitive domains like healthcare where the consequences of misclassification can be severe. It is therefore, essential to evaluate such models performance before deployment.

Another concern is that the level of perturbation used in this research study is relatively small and unnoticeable to the human eye, this raise possible concerns for malice and intentional sophisticated manipulation for purpose of misclassification. Researcher and

practitioners should focus on developing robust models to ensure their suitability for deployment in sensitive domain.

### B. TRAINABILITY OF THE MODELS

Overall, the findings suggest that each of the models has its own strength and weaknesses in terms of ability to fit the training data and generalize to new data. The LConvNet and DeepConvNet appear to fit the training data well, but may be prone to overfitting due to higher variances. The ShallowConvNet on the other hand, may be slightly underfitting the training data, but may be more robust to generalization. The EEGNet model appears to fit the training data well, but also prone to overfitting.

The comparison of the learning curves for the different models provides additional insights into their behavior during training. The more stable and consistent learning curves of LConvNet and EEGNet suggest that they may be more reliable models for predicting new data compared to the more erratic learning curves of DeepConvNet and ShallowConvNet. However, it is important to note that results may be influenced by the specific complexity of the models and further analysis and evaluation would be necessary to draw more definitive conclusions.

### C. SCALABILITY AND PARAMETER EFFICIENCY OF THE MODELS

The models' scalability is evaluated based on their training and inference times, which are important metrics in determining model's ability to handle larger datasets and making predictions in real-time applications. Derived results revealed that LConvNet has the highest number of trainable parameters, but with the fastest training and inference time per parameter. This indicates highest scalability among the four models.

Furthermore, we assess the models efficiency through the number parameters that can be processed by the models in a single training cycle. The results indicate that LConvNet and DeepConvNet are the most parameter efficient, meaning that they can handle more single units in a single training batch. These findings have practical implications. In particular, the results indicate that a moderate number of trainable parameters may be preferable to achieve performance with reasonable batch size and number of training epochs. Subsequently, the results suggest that even models with fewer trainable parameters may still require a large number of epochs or batch size to achieve optimal performance, highlighting the importance of careful model selection and hyperparameter tuning.

### D. LCONVNET TEMPORAL DEPENDENCIES

Derived results demonstrate that LConvNet is capable of extracting frequency-specific temporal patterns critical for classification tasks. This suggest that frequency domain analysis is a powerful technique for identifying features relevant for classification. Subsequently, it highlights the importance of preprocessing steps like band pass filtering and EEG segmentation in capturing temporal dependencies in EEG.

Moreover, the findings suggest LConvNet is effective in analyzing EEG data in high temporal resolutions and the preprocessing used is capable of capturing temporal correlations between EEG channels. The use of PCA to reduce dimensionality further enables our LConvNet model to capture more complex correlations between channels' electrodes.

Overall, the findings demonstrate the ability of LConvNet model to capture temporal dependencies in EEG data, which is critical for accurate classification of EEG signals in clinical settings. The findings may have implications for development of more advanced EEG signal processing techniques that can be used to diagnose and treat neurological disorders.

## AUTHOR CONTRIBUTIONS

**Swaleh Maulid:** Conceptualization (Equal); Data Curation (Equal); Formal Analysis (Equal); Investigation (Equal), Methodology (Equal); Writing-Original Draft (Equal); Writing-Review & Editing (Equal). **Michael Kimwele:** Conceptualization (Equal), Methodology (Equal); Supervision (Equal); Writing –review & editing (Supporting). **Akeem Olowolayemo:** Conceptualization (Equal); Methodology (Equal); Supervision (Equal); Writing-review & editing (Equal). **Dennis Kaburu:** Conceptualization (Equal); Methodology (Equal), Supervision (Equal), Writing-review & editing (Equal)

## CONFLICT OF INTEREST STATEMENT

Authors have no conflict of interest relevant to this article.

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Temple University Hospital repository at https://isip.piconepress.com/projects/tuh_eeg/html/downloads.shtml . The codes used during this study will be shared on request

## ORCID

*Swaleh Maulid* https://orcid.org/0009-0004-7452-332X

## REFERENCES

[1] S. D. Patel, A. Seifi, and H. V. Patel, "Electroencephalography (EEG) - StatPearls," StatPearls Publishing, March 8, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK470319/

[2] SR Benbadis and PW Kaplan. The dangers of over-reading an EEG. Journal of Clinical Neurophysiology, 36(4):249, 2019.

[3] Xiaoyi Li, Xiaowei Jia, Guangxu Xun, and Aidong Zhang. Improving EEG feature learning via synchronized facial video. In 2015 IEEE International Conference on Big Data (Big Data), pages 843–848. IEEE, 2015.

[4] Sahar Torkamani and Volker Lohweg. Survey on time series motif discovery. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 7(2):e1199, 2017.

[5] Guangxu Xun, Xiaowei Jia, and Aidong Zhang. Context-learning based electroencephalogram analysis for epileptic seizure detection. In2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 325–330. IEEE, 2015.

[6] Yan-Yun Sun, Wen-Jin Chen, Ze-Ping Huang, Gang Yang, Ming-Lei Wu, De-En Xu, Wu-Lin Yang, Yong-Chun Luo, Zhi-Cheng Xiao, Ru-Xiang Xu, et al. Trim32 deficiency impairs the generation of pyramidal neurons in developing cerebral cortex. Cells, 11(3):449, 2022.

[7] N Venkata Saichand et al. Epileptic seizure detection using novel multilayer lstm discriminant network and dynamic mode koopman decomposition. Biomedical Signal Processing and Control, 68:102723, 2021.

[8] William O Tatum IV. Handbook of EEG interpretation. Springer Publishing Company, 2021.

[9] Jiang Wu, Tengfei Zhou, and Taiyong Li. Detecting epileptic seizures in EEG signals with complementary ensemble empirical mode decomposition and extreme gradient boosting. Entropy, 22(2):140, 2020.

[10] Qiong Li, Jianbo Gao, Ziwen Zhang, Qi Huang, Yuan Wu, and Bo Xu. Distinguishing epileptiform discharges from normal electroencephalograms using adaptive fractal and network analysis: A clinical perspective. Frontiers in physiology, 11:828, 2020.

[11] Abeer Alzubaidi, Jonathan Tepper, and Ahmad Lotfi. A novel deep mining model for effective knowledge discovery from omics data. Artificial Intelligence in Medicine, 104:101821, 2020.

[12] Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis: a systematic review. Journal of neural engineering, 16(5):051001, 2019.

[13] Khakon Das, Debashis Daschakladar, Partha Pratim Roy, Atri Chatterjee, and Shankar Prasad Saha. Epileptic seizure prediction by the detection of seizure waveform from the pre-ictal phase of EEG signal. Biomedical Signal Processing and Control, 57:101720, 2020.

[14] Shrey Agarwal and Muhammad Zubair. Classification of alcoholic and non-alcoholic EEG signals based on sliding-ssa and independent component analysis. IEEE Sensors Journal, 21(23):26198–26206, 2021.

[15] Turker Tuncer, Sengul Dogan, Fatih Ertam, and Abdulhamit Subasi. A dynamic center and multi threshold point based stable feature extraction network for driver fatigue detection utilizing EEG signals. Cognitive neurodynamics, 15:223–237, 2021.

[16] Cyrille Feudjio, Victoire Djimna Noyum, Younous Perieukeu Mofendjou, Ernest Fokoue, et al. A novel use of discrete wavelet transform features in the prediction of epileptic seizures from EEG data. arXiv preprint arXiv:2102.01647, 2021.

[17] Xiaoqi Xu, Nicolas Drougard, and Raphaelle N Roy. Dimensionality reduction via the laplace-beltrami operator: Application to EEG-based BCI. In 2021 10th International IEEE/EMBS Conference on Neural Engineering (NER), pages 347–350. IEEE, 2021.

[18] Farah Nassif and Soosan Beheshti. Automatic order selection in autoregressive modeling with application in EEG sleep-stage classification. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5135–5139. IEEE, 2021.

[19] Lung-Chang Lin, Chen-Sen Ouyang, Rong-Ching Wu, Rei-Cheng Yang, and Ching-Tai Chiang. Alternative diagnosis of epilepsy in children without epileptiform discharges using deep convolutional neural networks. International journal of neural systems, 30(05):1850060, 2020.

[20] Zhiping Wang, Junying Na, and Baoyou Zheng. An improved KNN classifier for epilepsy diagnosis. IEEE Access, 8:100022–100030, 2020.

[21] Wei Qu, Zhiyong Wang, Hong Hong, Zheru Chi, David Dagan Feng, Ron Grunstein, and Christopher Gordon. A residual based attention model for EEG based sleep staging. IEEE journal of biomedical and health informatics, 24(10):2833–2843, 2020.

[22] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543, 2014.

[23] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in neural information processing systems, 33:12449–12460, 2020.

[24] Ye Yuan, Guangxu Xun, Qiuling Suo, Kebin Jia, and Aidong Zhang. Wave2vec: Deep representation learning for clinical temporal data. Neurocomputing, 324:31–42, 2019.

[25] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS), Vancouver, Canada, Dec. 2020, pp. 12449-12460.

[26] Xin Deng, Boxian Zhang, Nian Yu, Ke Liu, and Kaiwei Sun. Advanced tsgl-eegnet for motor imagery EEG-based brain-computer interfaces. IEEE Access, 9:25118–25130, 2021.

[27] Wenkai Huang, Yihao Xue, Lingkai Hu, and Hantang Liuli. S-eegnet: Electroencephalogram signal classification based on a separable convolution neural network with bilinear interpolation. IEEE Access, 8:131636–131646, 2020.

[28] I. Stancin, M. Cifrek, and A. Jovic, "A review of EEG signal features and their application in driver drowsiness detection systems," in Proceedings of the IEEE Sensors, vol. 21, no. 11, pp. 3786, 2021.

[29] F. Movahedi, J. L. Coyle, and E. Sejdić, "Deep belief networks for electroencephalography: A review of recent contributions and future outlooks," in IEEE Journal of Biomedical and Health Informatics, vol. 22, no. 3, pp. 642-652, May 2018.

[30] M. J. Hasan, D. Shon, K. Im, H. Choi, D. Yoo, and J. Kim, "Sleep state classification using power spectral density and residual neural network with multichannel EEG signals," in IEEE Applied Sciences, vol. 10, no. 21, pp. 7639, Nov. 2020.

[31] A. Ameera, A. Saidatul, and Z. Ibrahim, "Analysis of EEG spectrum bands using power spectral density for pleasure and displeasure state," in IOP Conference Series: Materials Science and Engineering, vol. 557, no. 1, IOP Publishing, 2019.

[32] S. Acharya, T. S. Bhat, and S. B. Pani. Deep learning based classification of EEG signal using wavelet transformation. In Advances in Intelligent Systems and Computing, volume 1029, pages 231–240. Springer, 2020.

[33] Bufang Yang, Xilin Zhu, Yitian Liu, and Hongxing Liu. A single-channel EEG based automatic sleep stage classification method leveraging deep one-dimensional convolutional neural network and hidden markov model. Biomedical Signal Processing and Control, 68:102581, 2021.

[34] Fangyuan Lei, Xun Liu, Qingyun Dai, and Bingo Wing-Kuen Ling. Shallow convolutional neural network for image classification. SN AppliedSciences, 2:1–8, 2020.

[35] Sung-Jin Kim, Dae-Hyeok Lee, and Seong-Whan Lee. Rethinking CNN architecture for enhancing decoding performance of motor imagery-based EEG signals. IEEE Access, 10:96984–96996, 2022.

[36] Mouad Riyad, Mohammed Khalil, and Abdellah Adib. Mi-eegnet: A novel convolutional neural network for motor imagery classification. Journal of Neuroscience Methods, 353:109037, 2021.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pages 234–241. Springer, 2015.

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818–2826, 2016.

[39] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. Journal of neural engineering, 15(5):056013, 2018.

[40] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for EEG decoding and visualization. Human brain mapping, 38(11):5391–5420, 2017.

[41] Iyad Obeid and Joseph Picone. The temple university hospital EEG data corpus. Frontiers in neuroscience, 10:196, 2016.

[42] William O Tatum IV. Handbook of EEG interpretation. Springer Publishing Company, 2021.

[43] Donald B. Percival and Andrew T.Walden. Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques. Cambridge University Press, 1993