Received: 19 June 2023 | Revised: 16 November 2023 | Accepted: 21 November 2023

DOI: 10.1002/eng2.12827

RESEARCH ARTICLE

Engineering Reports Open Access WILEY

Enhancing EEG signals classification using LSTM-CNN architecture

Swaleh M. Omar[1,2] | **Michael Kimwele**[2] | **Akeem Olowolayemo**[3] | **Dennis M. Kaburu**[2]

[1]Engineering and ICT Research Center, Kenya Industrial Research and Development Institute, Nairobi, Kenya

[2]School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya

[3]Department of Computer Science, Faculty of Information and Communication Technology, International Islamic University Malaysia, Selangor, Malaysia

Correspondence

Swaleh M. Omar, School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya.

Email: swaleh.mit@gmail.com

Funding information

Japan International Cooperation Agency; Jomo Kenyatta University of Agriculture and Technology; Pan African University

Abstract

Epilepsy is a disorder that interferes with regular brain activity and can occasionally cause seizures, odd sensations, and momentary unconsciousness. Epilepsy is frequently diagnosed using electroencephalograph (EEG) records, although conventional analysis is subjective and prone to error. The dynamic and non-stationary nature of EEG structure restricted the performance of Deep Learning (DL) approaches used in earlier work to improve EEG classification. Our multi-channel EEG classification model, dubbed LConvNet in this paper, combines Convolutional Neural Networks (CNN) for extracting spatial features and Long Short-Term Memory (LSTM) for identifying temporal dependencies. To discriminate between epileptic and healthy EEG signals, the model is trained using open-source secondary EEG data from Temple University Hospital (TUH). Our model outperformed other EEG classification models employed in comparable tasks, such as EEGNet, DeepConvNet, and ShallowConvNet, which had accuracy rates of 86%, 96%, and 78%, respectively. Our model attained an amazing accuracy rate of 97%. During additional testing, our model also displayed excellent performance in trainability, scalability, and parameter efficiency.

KEYWORDS

EEG, LSTM-CNN, spatial features, temporal dependencies

1 | INTRODUCTION

Epilepsy is a momentary disruption in the brain's regular function, frequently causing convulsions or other strange sensations, and on rare occasions, it can cause unconsciousness.

An electroencephalography (EEG) test, which includes placing metal electrodes on or within the skull to detect electrical pulses corresponding to brain functions, is used to diagnose the disorder. In contrast to amplitude, which is measured in microvolts, an EEG test records the brain's rhythm as continuous electrical wave frequencies across time, measured in seconds (Hertz). Epileptiform activity, or the appearance of distinct rhythmic waves with particular morphologies, is a sign that someone may have epilepsy.[1]

Traditional EEG interpretation relies on expert human judgment, which is subject to bias and can result in incorrect diagnosis. In the United States of America, misdiagnosis occurs in about 30% of patients who visit specialized epilepsy clinics.[2] Additionally, because EEG examinations are time-limited, the results of future tests cannot be reversed, which worsens the effects of misdiagnosis. This has spurred interest in applying Deep Learning (DL) approaches for epilepsy monitoring, detection, and prediction.

DL Sequence models such as RNNs, LSTM, GRU, and Transformer Networks gained popularity for language modeling due to their ability to analyze time series data. These models translate each word in a sequence to a token, learn context information, and perform mapping to a low-level fixed-length vector space. Because it might be confusing to decide which tokens to use, representing continuous time series data like EEG can be difficult. Researchers have tried to solve this problem by learning relevant temporal features for EEG representation using segmented windows, scalograms sequence, spectrograms, and motifs in time series mining, but this still fails when interesting pattern lies hidden within common patterns in the sequence.[3–5] Moreover, Sequence models lack the capability for good generalization for spatial information inherited within EEG data.

Due to its capacity to learn spatial information, convolution neural network (CNN) was initially developed for computer vision; however, more recent research has made it possible to use CNN to classify several data types. CNN has been applied to EEG signal processing to learn spatial features, which are properties of various EEG signals in classification tasks.

This paper provides a methodology to improve EEG classification by combining a CNN model with an RNN-based sequence modeling technique like LSTM. With the help of CNN's ability to extract spatial patterns and LSTM's ability to recognize temporal dynamics, it is intended to improve the classification performance of EEG tasks.

While the suggested method outperforms current DL techniques in comparable classification tasks, it is crucial to recognize the accomplishments of particular methods in Complex Science, that have demonstrated outstanding results in epileptiform discharge analysis and classification, frequently achieving nearly perfect accuracies by emphasizing interpretability over black-box methodologies.[6–8]

1.1 | Contribution of the study

This study makes a contribution to the fields of Brain-Computer Interactions (BCI) and Signal Processing by proposing the 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 tasks.

2 | RELATED WORK

2.1 | 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.[9] 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 are further classified as simple or complex partial seizures, whereas, Generalized seizures affect the entire brain and can be categorized into Tonic, Atonic, Clonic, Myoclonic, and Tonic–Clonic (Convulsive) seizures.[10,11]

The primary test for detecting epilepsy is Electroencephalography (EEG). EEG records electrical activities in the brain using metal electrode discs and presents the data as signals.[12] In a normal brain, EEG readings show four different patterns; Alpha waves (8–12 Hz) are seen in an inattentive brain, drowsiness; Beta waves (12–35 Hz) depict an alert brain; Delta waves (1–5 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 5 years.[11] Several EEG morphologies may suggest non-epileptic brain rhythms which may suggest abnormal cerebral functions without specific etiology, yet others are noises commonly known as artifacts.

Epileptiform activities are brain rhythm abnormalities that could be associated with seizures but usually require clinical correlation.[6] Misdiagnosis of epilepsy can have severe consequences since the EEG test is time-bound, subsequent tests cannot reverse the previous one.

2.2 | Signals preprocessing and representation

Due to their nonstationary and nondeterministic nature, biosignals like the EEG require complicated analysis.[13,14] Thorough signal preprocessing is essential to enhance the performance of DL models during training. A number of preprocessing methods have been proposed, depending on the task and desired outcome. These methods include the use of hybrid feature extraction algorithms,[15] independent component analysis,[16] Fast Fourier Transform,[17] discrete wavelet transform,[18] eigenvector,[19] and autoregressive methods.[20]

Studies utilizing various models to evaluate EEG data have been conducted as a result of advancements in DL techniques in order to understand better the temporal and state dynamics of the brain.[21–23] DL algorithms must rely on numerical operations to represent data since they are unable to process high-level language. The generality of Sequence-to-Sequence models for use in Natural Language Processing was boosted by tokenization techniques like word embedding.[24]

Due to tokenization's ambiguity, representing EEG signals is difficult.[25] This issue has been addressed by transforming raw signals into meaningful representations that can be applied to activities farther down the line.[26,27] CNN has established itself as a respectable rival[19] in one-dimensional tasks like audio, text, and time series analysis. Certain CNN-based models have shown promise in extracting useful feature representation based on distinguishing features of input signals in Brain-Computer Interface (BCI) challenges.[28,29]

2.3 | 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.[30,31] Frequency Domain techniques such as power spectral density and spectral entropy improved classification tasks by leveraging spatially mapping EEG features from different frequency bands.[32,33] Frequency Domain Features yield a richer representation of EEG signals by extracting more relevant latent features from its spatial space, however, fail in the 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.[34]

For further improvement, some studies proposed hybrid approaches that combine multiple techniques and algorithms such as a combination of time-domain and frequency-domain features, hybrid, and even ensemble algorithms to achieve better performance.[35] However, despite many examples of impressive progress, there is still much room for improvement and research in this field to develop more accurate and efficient EEG classification methods.

Shallow Convolution Networks (ShallowConvNets), a subtype of CNN architecture, are deployable on mobile and embedded devices and are computationally efficient since they execute image classification tasks with fewer layers and minimal filters.[36] Other applications involving EEG decoding and classification tasks have also been used with ShallowConvNets.[37,38] On the other hand, Deep Convolution Neural Networks (DeepConvNets) architecture has achieved outstanding results in computer vision applications, such as image classification, object recognition, and segmentation. UNet[39] and InceptionNet[40] are two notables DeepConvNets. Unlike ShallowConvNets, DeepConvNets are characterized by a large number of convolution layers which enable them to learn a hierarchical representation of the input data with much finer detail.

The majority of CNN architectures for evaluating image data share a typical structure of inputs, kernels, pooling layers, and output (feature maps). CNN's performance is constrained when dealing with time series signals, such as EEG, due to:

i. Particularities of EEG data: Images are spatially structured but EEG signals are time series, making it challenging to extract relevant features.
ii. High noise-to-signal ratio: EEG data are frequently tainted with different kinds of physiological and environmental noise, and artifacts, making it challenging for CNN to discern between different patterns.
iii. Temporal Dynamics of EEG Records: Since neural activities change quickly over time, convolution techniques have a difficult time capturing the temporal dynamics of EEG signals.

While applying CNN to EEG data, researchers have discovered techniques to mitigate the limitations noted, such as preprocessing EEG data to pixel images or spectrograms for convolution. Such representations, however, omit temporal information that is crucial for classification. Recent studies in the field of Brain-Computer Interface (BCI) have shown that various convolutional techniques can make it possible to extract a variety of significant EEG features. Excellent findings were found when EEGNet[41] evaluated the roles of separable convolution networks for the spatial mapping of EEG properties. Additionally, others looked into the effects of design architectural decisions, training strategies for shallow and deep convolution networks, and the application of machine learning techniques like batch-normalization and exponential linear units on the decoding accuracy of EEG signals.[42] They found that their study outperformed commonly used spectral power modulation EEG techniques.

In this study, we propose the LConvNet multi-channel EEG classification model, which combines CNN and LSTM features for the classification of EEG data. While CNN learns spatial properties, LSTM acquires and analyzes long-term dependencies from the temporal EEG data in conjunction with the Time-Distributed dense layers. After that, a global average pooling layer is used to condense the learned features from the input tensor into a fixed-length vector, which is then concatenated to the CNN-LSTM outputs for a more in-depth representation of the spatial and temporal data. We have shown that our hybrid model outperforms other EEG classification models, including EEGNet,[41] ShallowConvNet, and DeepConvNet,[42] in terms of performance, scalability, and efficiency. To the best of the authors' knowledge, this is the first implementation of LSTM-CNN in the binary classification of multichannel EEG data.

3 | MATERIALS AND TOOLS

3.1 | Experimental setup

Data access from the TUH database were done through the MobaXterm toolbox for remote computing. Data were saved in EDF format. Visualization, preprocessing, feature extraction, and classification were 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.

3.2 | Dataset and preprocessing

This research utilizes the TUEP (TUH EEG Epilepsy Corpus) v2.0.0, an open-source dataset. All procedures related to this dataset adhere to principles outlined in the Declaration of Helsinki and are compliant with HIPAA Privacy rule.[43] The EEG dataset is categorized into two groups: Epilepsy and Non-Epilepsy, based on criteria including clinical history, medication usage, and features associated with epilepsy.[43] The dataset comprises records with differing durations in seconds, various sampling frequencies, and a variable number of channels, with a minimum of 25 channels, all stored in EDF format. For the purpose of this study, the EEG data was organized into two classes, consisting of 49 sessions with epilepsy and 49 sessions without epilepsy (considered healthy), each exhibiting diverse durations, sampling frequencies, and channel configurations.

During preprocessing, all EEG channels were set to average reference (Y) to enable ease in approximation by obtaining the difference between the electric potential of input signal x in its location and the average of all channels ($mean$) as represented by Equation (1).

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

This was followed by resampling at the rate of 128 Hz for the purpose of improving quality and reliability using Equation (2), where x is the original data, i is the index of the sample, y is the output resampled data, r is the ratio of the new sampling rate to the original sampling rate, and the *round function* approximates to the nearest integer.

$$y[i] = x[round(i * r)] \tag{2}$$

Then, using MNE's Finite Impulse Response[11] to selectively pass frequencies within a certain band, we apply band-pass filtering for noise reduction and artifact correction. The application of Independent Component Analysis (ICA) to achieve the same, particularly in the removal of eye movement artifacts, yielded no further improvement in the context of this study. Additionally, for homogeneity, all data were chopped between 0 and 200 s.

Non-EEG signals like photic and electrocardiogram (ECG), which are utilized to provoke seizures and measure pulses, were eliminated. In order to improve the data's readability by identifying certain events or stimuli within the signals.

Data were then segmented into fixed equal duration epochs of 2 s each with an overlap of 1 s to allow a smoother transition between epochs. This process is represented by Equation (3); 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. This resulted in 200 epochs for each EEG subject.

$$y_{i[n]} = x[i * (N - overlap) + n] \tag{3}$$

Further, Principal Component Analysis (PCA) was applied for dimensionality reduction at a standard 25 signal components. The PCA operation is expressed in Equation 4; where X is the input signal's matrix, U is the matrix of principal components, S is the diagonal matrix containing squared singular values representing the variance explained by each principal component and V^T is the matrix of loadings.

$$X = USV^T \tag{4}$$

The EEG data have variable numbers of input channels, with a minimum of 25 channels, as was described in the prior section. We decided to keep 25 PCA components in order to accurately capture important data points. The conclusion is supported by the fact that when compared to different numbers of components, a Cumulative Explained Variance Ratio (CEVR) exceeding 90% is attained by summing the squared singular values and dividing by the total sum of squared singular values using 25 PCA components as illustrated in Figure 1. This high CEVR shows that these components successfully retain the majority of the important information found in the original data. Additionally, the choice of components helped avoid overfitting in the later stages of model training.

The preprocessing steps resulted in the creation of 200 epochs for each EEG subject, with each epoch lasting approximately 2 s. This means there were a total of 9800 epochs for each of the epileptic and healthy classes. Subsequently, these epochs were converted into 3D arrays for further analysis. Figure 2 shows a sample of unprocessed EEG data (labeled as A) from the epileptic class, along with its output (labeled as B) after undergoing the preprocessing steps.

3.3 | Architecture of the LConvNet model

Our LConvNet model constitutes 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.

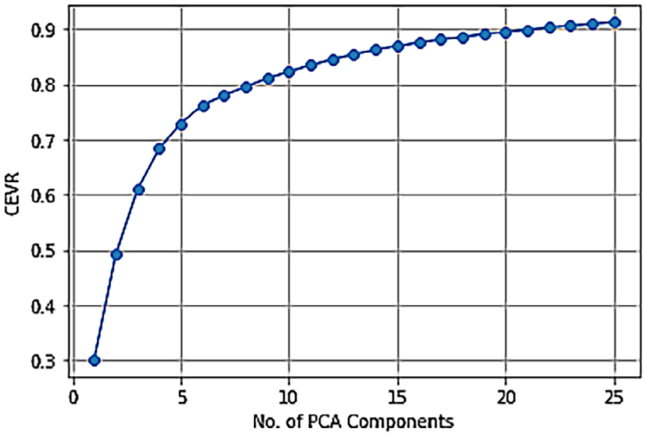


FIGURE 1 PCA explained variance (PCA EV) for EEG data: This figure illustrates the principal component analysis (PCA) explained variance (PCA EV) for EEG (Electroencephalogram) data. The chart showcases the contribution of each principal component to the overall variance in the EEG dataset. PCA is a statistical technique commonly used for dimensionality reduction, and the Explained Variance provides insights into how much information is retained by each principal component.

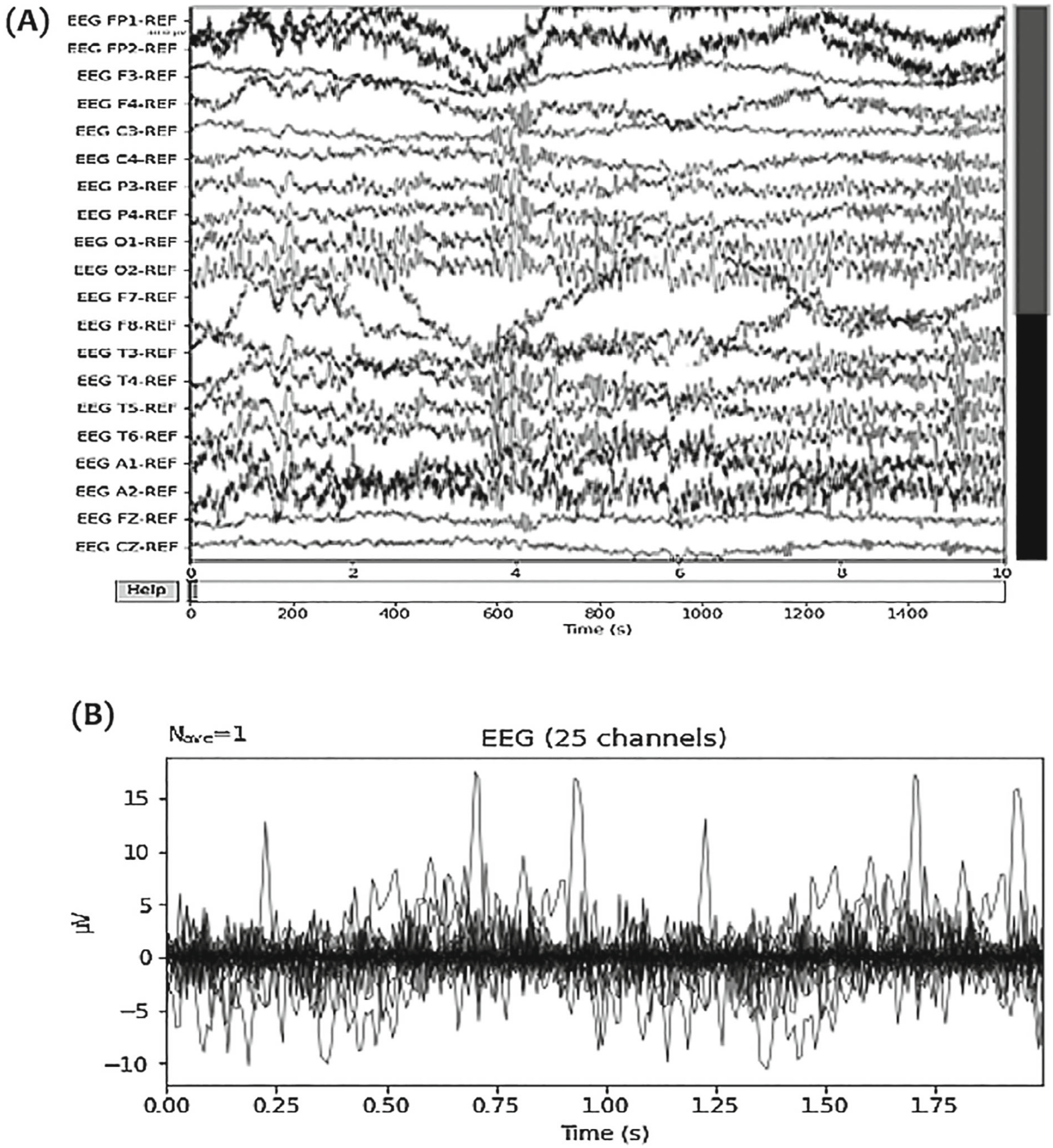


FIGURE 2 EEG record. (A) Raw data, (B) Preprocessed data: This figure shows a sample of unprocessed EEG data (labeled as A) from the epileptic class, along with its output (labeled as B) after undergoing the preprocessing steps.

3.3.1 | CNN component

The CNN component takes EEG epochs converted to 3D input tensor X of shape (ntimesteps, nchannels, nsamples) where ntimesteps refers to the number of EEG time-steps, nchannels is the number of EEG channels and nsamples refers to the number of EEG samples derived from the previous preprocessing steps, where $X = \{x1, x2 \ldots xn\}$. We derive spatial rich feature maps using three convolution layers with incrementing filter sizes (16, 32, 64) and kernels (3×3, 5×5, 7×7).

We increment convolution filters to allow capturing features from high temporal resolution inherited in varied sampling frequencies as applied during preprocessing, this proved effective in similar tasks,[41] in addition, this enables the model to capture input patterns at different scales. Smaller filters such as 16, capture finer frequency details and edges, whereas larger filters such as 64 capture more global patterns and structures. Kernels in CNN determine the receptive field. Hence, the combination of filters and kernels used in this study's model is designed not only to capture data at different scales but, subsequently, to allow the model to learn representations that are robust to variation in the input. The CNN component operation used in this study is described in Figure 3.

The CNN operation is expressed in Equation (5) where x is the input matrix derived from previous preprocessing steps; $M = \{m_1, m_2, m_3 \ldots m_n\}$ is the output feature map; f is a ReLU non-linearity activation function; Asterix $(*)$denotes convolution operation between w and x, this operation generates convolved output (feature maps) that accentuates certain pattern in the data aiding in the extraction of meaningful features; w and b are learned vectors representing convolution kernels (weights) and bias respectively, adjusted during backpropagation; N is the total number of feature maps; $\sum_{i=1}^{N}$ represents the summation of all N individual convolved feature maps. Dropout operation is applied to reduce overfitting while max pooling to maintain the spatial dimension.

$$M = f\left(\sum_{i=1}^{N} * (w, x) + b\right) \tag{5}$$

By running the output feature Map M through a Time Distributed Flatten Layer, it is transformed into a 2D format that spans many time-steps. Additionally, the flattened M vector is subjected to a Time Distributed Dense Layer (TDDL) with 64 units using Equation (6), where Y_t represents output at time step t of the TDDL, $f$is the activation function, w is the TDDL's weight matrix, $(*)$ is the multiplication operation, x_t is the input at time-step t and b_i is the bias vector.

$$Y_t = f(w * x_t + b) \tag{6}$$

TDDL allows individual processing of sequential data at each time-step using shared weights and biases, effectively capturing temporal dependencies.

3.3.2 | LSTM component

The application of LSTM in this study is to enhance subsequent classification tasks by complementing CNN spatial features with EEG temporal dependencies. The input Y derived from previous convolution and time distribution operations is split into 64 LSTM time steps denoted by t, we find 64 to produce the best results. At each timestep t, the LSTM layer takes two inputs: the current input vector $x(t)$and the previously hidden state $\alpha(t-1)$. Using these inputs, it computes three gates – the forget gate, the update gate, and the candidate memory. Equation (7) depicts the computation of the unidirectional LSTM layer as is applicable to our model.

$$f_{r(t)} = \sigma\left(W_f[\alpha(t-1), x(t)] + b_f\right)$$

$$u_{r(t)} = \sigma(W_u[\alpha(t-1), x(t)] + b_u)$$

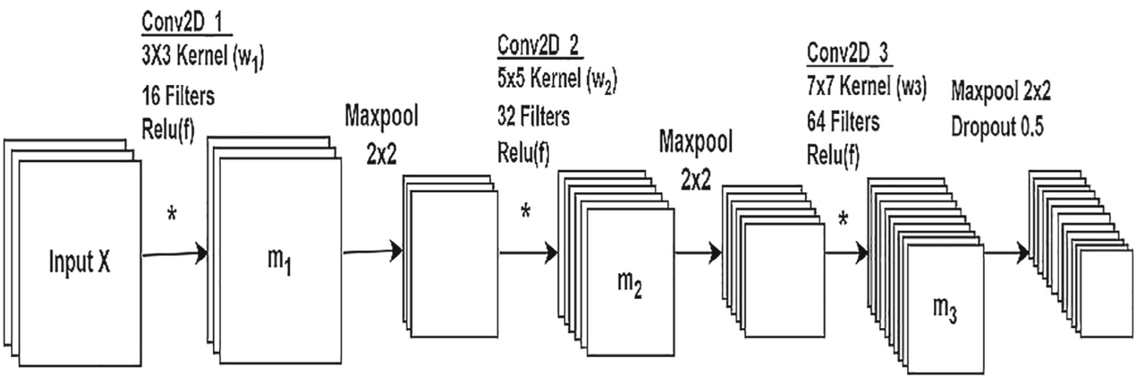


FIGURE 3 CNN component operations: This figure shows the CNN component operation as used in this study.

$$\tilde{c}(t) = tanh\left(W_{c[\alpha(t-1),x(t)]} + b_c\right)$$

$$c(t) = f_{r(t)} \odot c(t-1) + u_{r(t)} \odot \tilde{c}(t)$$

$$o_{r(t)} = \sigma\left(W_{o[\alpha(t-1),x(t)]} + b_o\right)$$

$$\alpha(t) = o_{r(t)} \odot tanh(c(t)) \tag{7}$$

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, $\alpha(t)$ is the hidden state, W_{i}, and b_{i} are corresponding weight matrix and bias vector, respectively.

3.3.3 | Concatenation

Subsequently, the input tensor X from preprocessing operations is passed to 1D Global Average Pooling Layer along the time axis to obtain an average of each feature across the time dimension as expressed in Equation (8).

$$P_i = \left(\frac{1}{T}\right) * \left(\sum_{t=1}^{T} \mathrm{X}_{\mathrm{it}}\right) \tag{8}$$

The resulting vector P_i represents the ith feature of the tensor P. T is the sequence length in the input tensor X. X_{it} denotes the value of the ith feature at time step t in the input tensor X. P is of the same shape as the hidden state $\alpha(t)$ 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 $\alpha(t)$ from the LSTM to produce tensor C as expressed in Equation (9).

$$C = (\alpha(t)\&P) \tag{9}$$

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

3.3.4 | Dense neural network

The resulting concatenated Tensor is passed through a final dense neural network with a sigmoid function for binary classification. The dense layer operation is expressed in Equation (10)

$$Y = \sigma(w * C + b) \tag{10}$$

where Y is a 2-dimension output matrix representing binary classes (healthy, epileptic), σ is a sigmoid activation function, w is a 3-dimension weight matrix of shape = (nlstmunits, nchannels, nclasses), C is a 2-dimension concatenation result 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 4 shows the architecture of our LConvNet model as described in this section.

The model is compiled using Adam optimizer with a learning rate $r = 1 \times 10^{-4}$, this value was found to give a good balance between training speed and convergence stability after experimentation and hyperparameter tuning. We further deploy binary cross entropy loss function ℓ as shown in Equation (11). Where y is the actual classification for an observation and p the predicted observation.

$$\ell = -(y\ \log(p) + (1-y)\log(1-p)) \tag{11}$$

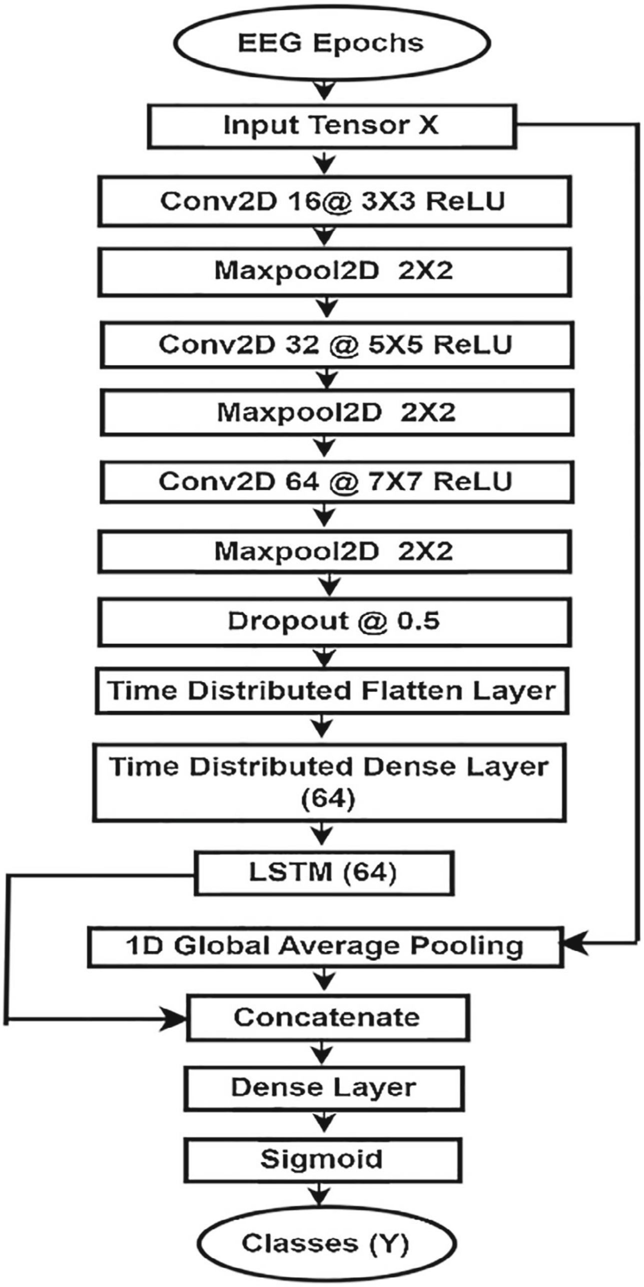


FIGURE 4 The architecture of LConvNet Model: Shows the architecture of our LConvNet model as described in the section.

3.4 | Model training

We used a stratified technique to split the data into training and validation sets with a 20% split for the classification job, guaranteeing a balanced representation of both classes. Next, the data were standardized mean and standard deviation set to 0 and 1 respectively for consistency. Batch sizes of 128 and 200 iterations/epochs were selected for training. These variables achieve the best performance by maximizing effectiveness, efficiency, and avoiding overfitting. The elements of our LConvNet model after training are displayed in Table 1.

3.5 | EEG feature extraction methods

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

TABLE 1 LConvNet architecture input and output components.

Layers	Input (nsamples ntimesteps, 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)	12,832
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)	106,692
4	(1, 3, 32, 68)	TimeDistributed (Flatten)	(1, 3, 2176)	0
5	(1, 3, 2176)	TimeDistributed (Dense)(32)	(1, 3, 32)	69,664
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)	(None, 1)	58
		Dense (sigmoid)		
Total parameters				197, 726

3.5.1 | Feature extraction using EEGNet

The inputs shape for EEGNet[41] 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: the EEGNet module and the classification layer. The 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 the 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 the time dimension of the input data. This further reduces dimensionality while allowing the extraction of relevant features.

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

3.5.2 | Feature extraction using ShallowConvNet and DeepConvNet

The ShallowConvNet proposed by Reference [42] consists of a single convolutional layer, followed by an 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 the 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 to reduce the dimensionality of the data hence preventing 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 Reference [42] 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.

4 | RESULTS

4.1 | Performance evaluation

To evaluate performance of the models, we first tested models' accuracies, precision, recall, and F1 score with a threshold of 0.7 using Equations (12)–(15).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{12}$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{13}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

where TP is true positives, FP is false positives, FN is false negatives, TN is true negatives, and *C* is the number of classes (2). Table 2 shows the summary of the findings.

Accuracy gauges how well a model performs overall when making predictions. The performance of the model improves with accuracy. Our models LConvNet and DeepConvNet have the highest accuracy rates, with 97% and 96%, respectively, according to the provided metrics, whereas, EEGNet has an accuracy of 86%, and ShallowConvNet, 78%. When comparing all expected occurrences, precision is defined as the percentage of true positives. The model is better at detecting the positive classes when precision is increased. As can be seen in Table 2 that LConvNet is the model with the highest precision, followed by DeepConvnet. The least precise networks are EEGNet and ShallowConvnet. Recall counts the number of actual positive instances that are actually true positives. Likewise, LConvNet and DeepConvNet exhibit the highest average recall. The F1 score, which is a gauge of the models' overall effectiveness, is the harmonic mean of precision and recall. According to Table 2 results, LConvNet, DeepConvNet, EEGNet, and ShallowConvNet have the greatest F1 Scores.

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) is shown in the confusion matrices for the four models in Figure 5 where LConvNet performs better than other models.

The total effectiveness of the model in differentiating between positive and negative cases is displayed by the Area Under a Curve (AUC). The AUC curve in Figure 6 compares the performance of the four models (models are identified by graph color as shown in legend). The better the AUC value, the better the model's ability to discriminate between positive and negative data is.

TABLE 2 Performance metric values.

	LConvNet	ShallowConvNet	EEGNet	DeepConvNet
Accuracy	0.97	0.78	0.86	0.96
Precision	0.97	0.85	0.88	0.96
Recall	0.97	0.78	0.86	0.96
F1-Score	0.97	0.77	0.85	0.96

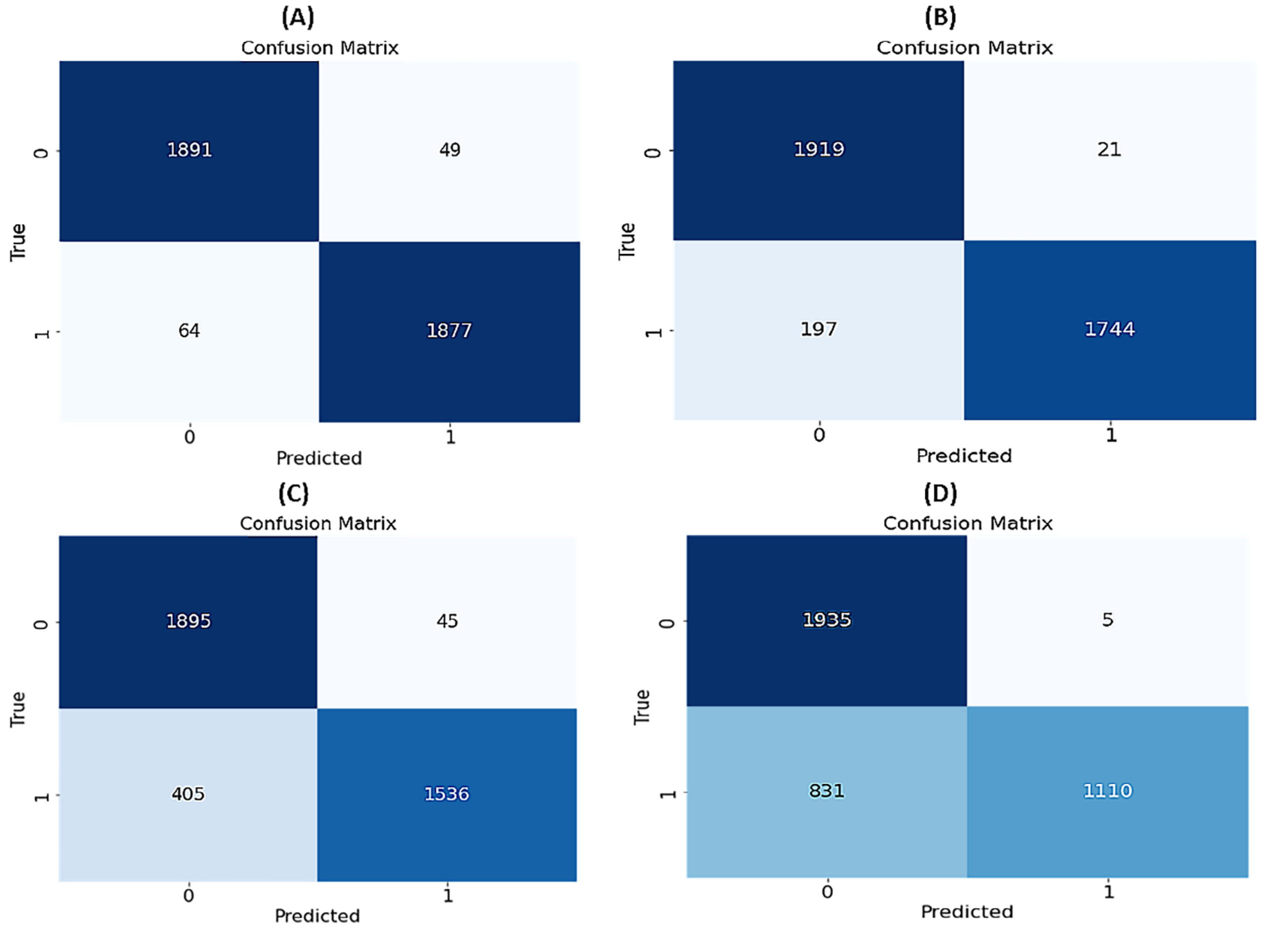


FIGURE 5 Confusion Matrices. (A) LConvNet, (B) DeepConvNet, (C) EEGNet and (D) ShallowConvNet: Confusion Matrices shows the proportion of True Positives True Negatives, False Positives, and False Negatives (second row, first column) for LConvNet (A), DeepConvNet (B), EEGNet (C), and ShallowConvNet (D).

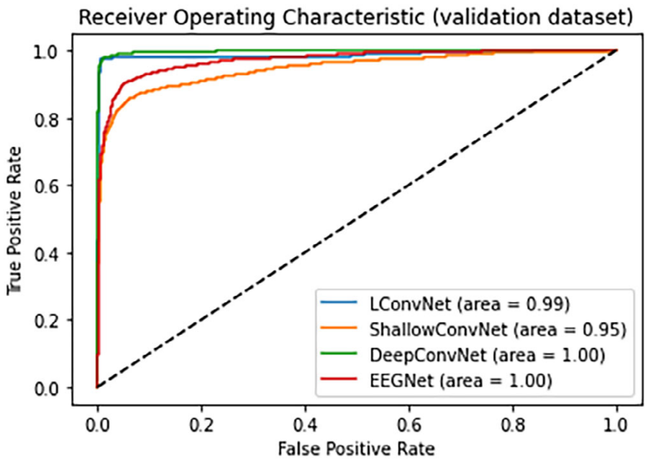


FIGURE 6 AUC for the four models for the validation dataset: The total effectiveness of the model in differentiating between positive and negative cases is displayed by the Area Under a Curve (AUC).

We then use Cohen's Kappa (CK) and Mathew Correlation Coefficient (MCC) as expressed in Equations (16) and (17) respectively to evaluate the degree of agreement between the predicted and actual labels in each of the four models.

$$\text{CK} = \frac{P_\text{o} + P_\text{e}}{1 - P_\text{e}} \tag{16}$$

$$\text{MCC} = \frac{\text{TP} * \text{TN} - \text{FP} * \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{17}$$

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 levels 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.

We further assess how well the models can tolerate hostile input samples in order to compare robustness. The adversarial samples used in this evaluation are duplicates of the original input data that have been altered to include perturbation created using the Fast Gradient Sign Method (FGSM) attack depicted in Equation (18). This was done with the intention of purposely misclassifying the model.

$$Y = x + \epsilon * \text{sign}(\bigtriangledown x\text{J}(\Theta, x, y)) \tag{18}$$

In this context, Y is the adversarial output, x is the input data, y is the ground truth label of the input data, ϵ is the magnitude for perturbation applied to the input data x, we selected a small perturbation value, specifically $\epsilon = 0.01$ to ensure any perturbations generated are imperceptible to the human eye which is desirable for deception, *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), $\bigtriangledown$ is the gradient of the loss function, Θ is our model, and J is the loss function.

The models are assessed according to how well they classified the validation data with accuracy. According to Table 4, all four models perform relatively better on the original data than they do on the adversarial data. This indicates that the models can classify the input data more accurately when it has not been altered. However, accuracy suffers dramatically when input data is disrupted, showing that the models are not strong enough to withstand hostile attacks. The degree of accuracy declines differs between models, with EEGNet and DeepConvNet performing considerably worse while LConvNet and ShallowConvNet show the least but still noticeable accuracy decline.

In conclusion, 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.

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

TABLE 4 Robustness evaluation.

	LConvNet	ShallowConvNet	EEGNet	DeepConvNet
Original data accuracy	0.96	0.48	0.98	0.90
Adversarial data accuracy	0.53	0.25	0.48	0.51
Accuracy difference	0.43	0.23	0.49	0.39

4.2 | Trainability evaluation

By computing the mean predictions from actual values and, accordingly, the mean of the squared differences between the predictions and the mean predictions of the validation datasets, we were able to determine the bias and variance when evaluating the models' trainability.

LConvNet has the most trainable parameters (197,726) of the four models, according to Table 5. Although it has a reasonably high variance (0.1812) and a relatively low bias (0.00130), suggesting that it is able to fit the training data well, however, it may overfit when introduced to new data. ShallowConvNet has a bias of 0.0777 and fewer trainable parameters (41,841) than LConvNet, which indicate that it slightly underfits the training set of data. The fact that it has a low variance of 0.0999, however, suggests that it might be more reliable to generalize new data. Of the models listed, the EEGNet model has the fewest trainable parameters (753). Its low bias (0.01741) indicates that it can adequately fit the training set of data. 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.

We obtain a local perspective when assessing the models' performance and generalizability by taking variance and bias into account. Variations in results can be related to elements like complexity, model architecture, and the number of trainable parameters. We examine learning curves (LC), which plot accuracy versus training iterations (epochs), to better understand how the models behave during learning process, as shown in Figure 7. The plots show, DeepConvNet and ShallowConvNet to have greater instability in determining accuracies as they update validation parameters during learning compare to LConvNet and EEGNet.

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

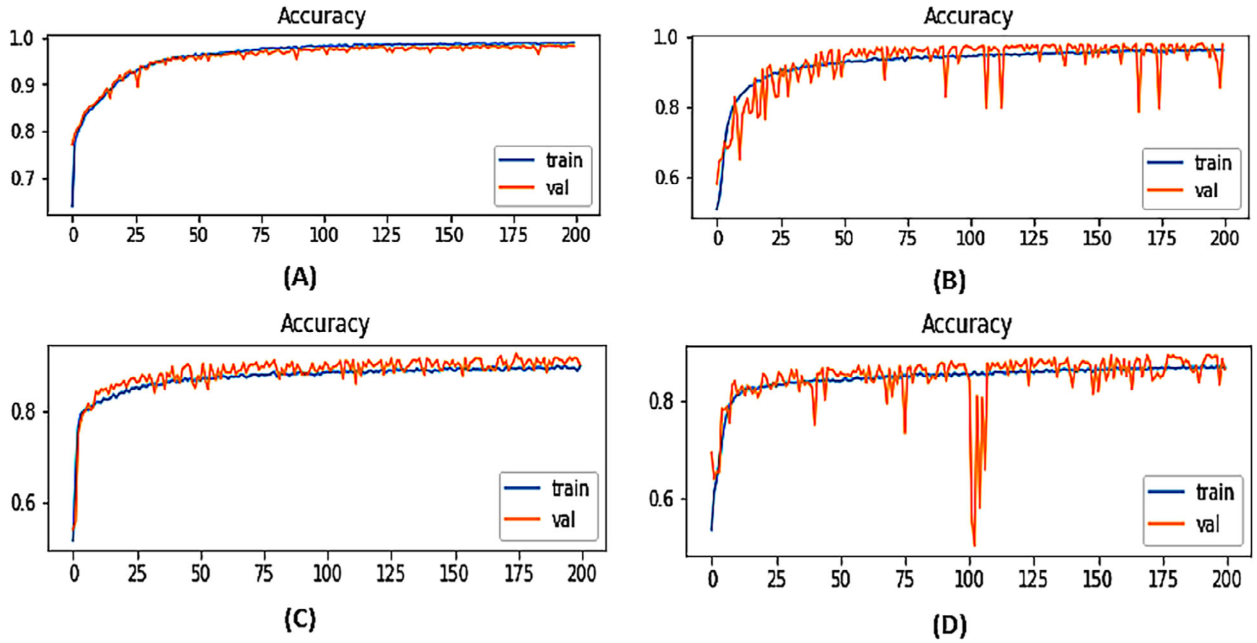


FIGURE 7 Accuracy LCs. LConvNet (A), DeepConvNet (B), EEGNet (C), ShallowConvNet (D): This figure presents the accuracy learning curves for the different models. These curves are crucial for assessing the models' convergence and performance trends, offering insights into how well each architecture learns and improves accuracy over the course of training.

Similarly, the Loss LCs shown in Figure 8, depict how well the four models learn and iteratively improve as they each update parameters, they confirm the results given by Accuracy LCs in Figure 7. Hence, the training processes for the LConvNet (A) and EEGNet (C) models demonstrate greater stability and consistency in learning from the data as compared to DeepConvNet and ShallowConvNet.

4.3 | Model visualization

We employed feature visualization to comprehend the pertinent characteristics used for classification within the layers of the models in order to access the interpretability of the models. As demonstrated in Figure 9, all models except LConvNet show extremely distinct, diffuse features with high layer activation, whereas LConvNet CNN layers (without the LSTM component) show more localized features with low-layer activation. The colors of the heatmaps in Figure 9 show the degree of activation; darker colors indicate lower activation and lighter colors indicate more activation.

4.4 | Scalability and efficiency metrics

4.4.1 | Training and inference times

By calculating the efficiency of the models using the training and inference time, we can assess the scalability of the models. Training time refers to the amount of time spent learning the data and updating the model's 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 fresh, unstudied data.

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 s and the fastest inference time of 0.49 s per parameter. This suggests that LConvNet may be highly scalable as it can train quickly on large datasets and make

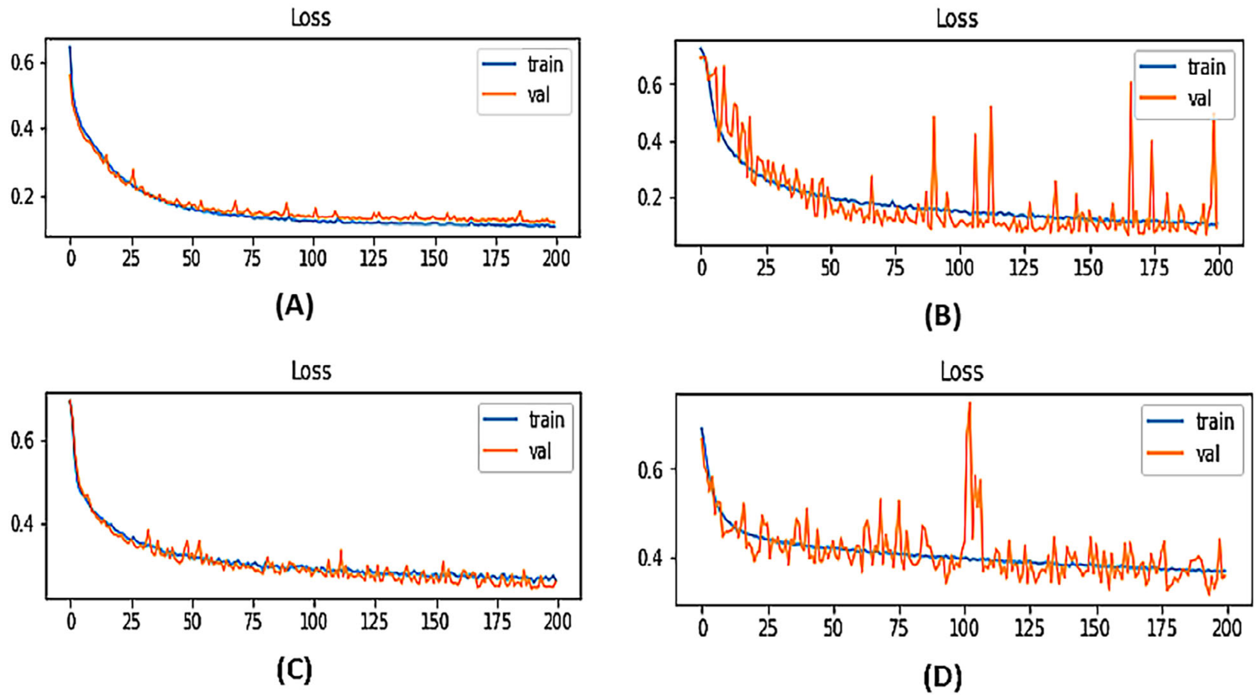


FIGURE 8 Loss LCs. LConvNet (A), DeepConvNet (B), EEGNet (C), ShallowConvNet (D): Loss learning curves depict how the loss function, which measures the difference between predicted and actual values, changes over training epochs. Monitoring these curves is essential for assessing the convergence and optimization of neural network models. Figure 8 provides insights into how each architecture minimizes its loss during the training process, aiding in the evaluation and comparison of their training dynamics.

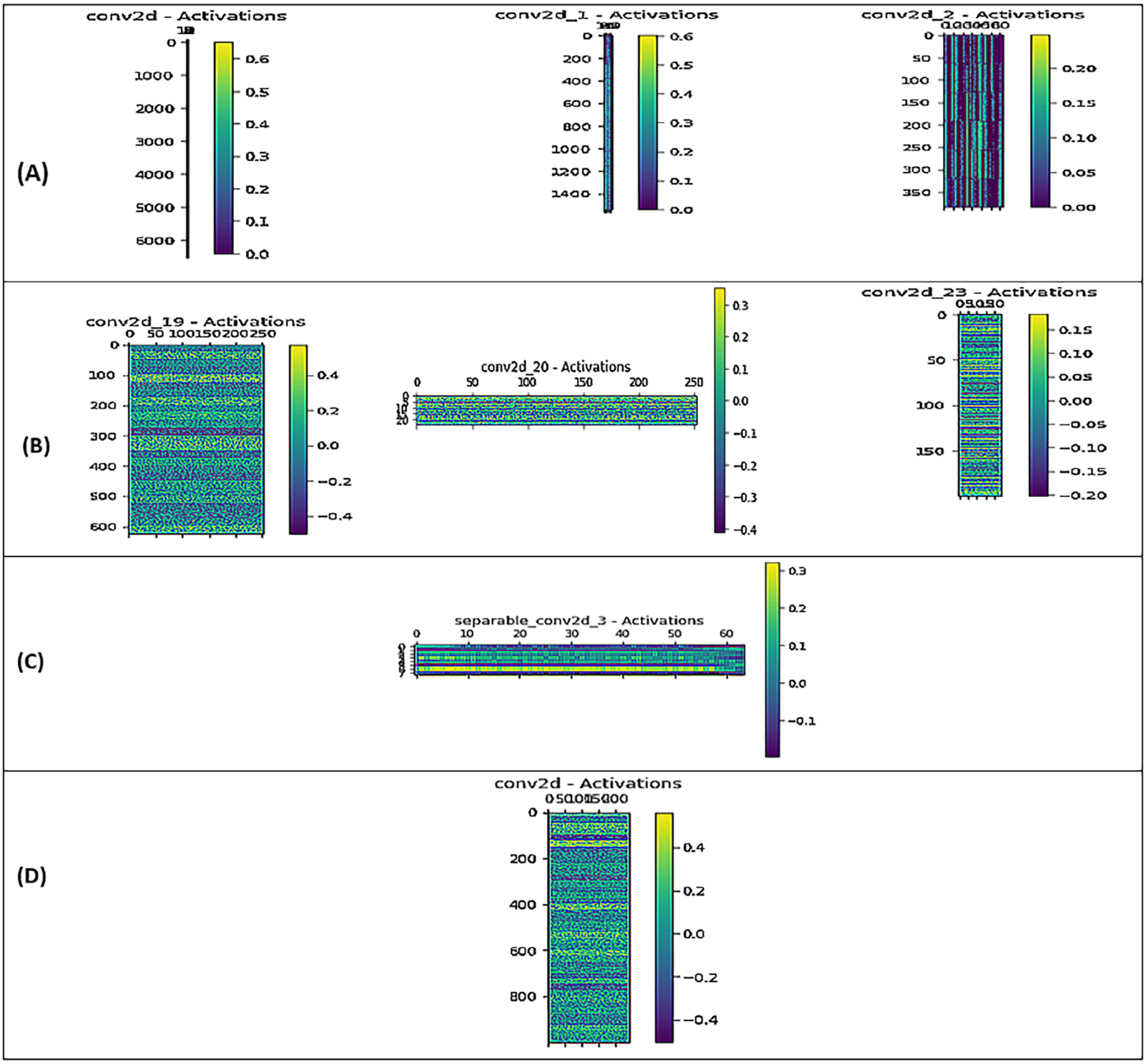


FIGURE 9 Layer activation heat maps. LConvNet (A), DeepConvNet (B), EEGNet (C), ShallowConvNet (D): This figure showcases layer activation heat maps for various neural network architectures. Layer activation heat maps visually represent the activation levels across different layers of each neural network during the processing of input 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

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 s and a relatively slow inference time of 0.75 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, a relatively slowest training time of 334.21 s and a relatively slow inference time of 0.75 s, 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 s, but a slower inference time of 0.86 s compared to the rest. This suggests 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 appears 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.

4.4.2 | 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 Equation (19).

$$\mathrm{PE} = \frac{N}{E * B} \tag{19}$$

For each input, where PE refers to the parameter efficiency value, N is the number of parameters, 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 single batch, which is desirable.

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 the same level of performance as the other three despite having very few trainable parameters.

4.5 | Analysis of LConvNet temporal dependencies

Before classification, preprocessing was conducted on the raw data and involved a number of processes that could affect how our model extracts temporal connections. This section highlights a few potential temporal relationships that our LConvNet model discovered.

4.5.1 | Frequency-specific temporal patterns

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

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

Note: Epoch = 200; Batch size = 128.

We convert the preprocessed EEG data into frequency domains using bands ranging between 0.5 and 4.5 Hz (Delta), 4.5 and 8.5 Hz (theta), 11.5 and 15.5 Hz (sigma), 15.5 and 30 Hz (beta), 30 and 45 Hz (gamma) using welch method to derive power spectral density (PSD) spectrograms.[44] Analysis of EEG data in these particular frequency bands is crucial since, as explained in the related literature section, each frequency band corresponds to a different set of cognitive and neurological processes in the brain. Then, in order to enable visualization and get insight into our model's propensity to distinguish between different output classes, we transformed findings of corresponding frequency bands to spectrograms and then applied t-Distributed Stochastic Neighbor Embedding (t-SNE) with output space of 2 and perplexity of 5 to each of the spectrograms and fit to our model.

Figure 10 shows that our LConvNet model distinguished clear separation between epileptic and healthy classes with distinct properties given a specific timeframe epoch (2 s) with a 1-s overlap. Gamma, Beta, and Delta produced better classification results.

4.5.2 | Temporal dynamics of EEG activities

During preprocessing, we initially sample EEG data at a rate of 128 Hz, which implies that for every second of EEG data, there are 128 data points, hence, the level of temporal information to be caught in EEG signals is determined by $t = 1$

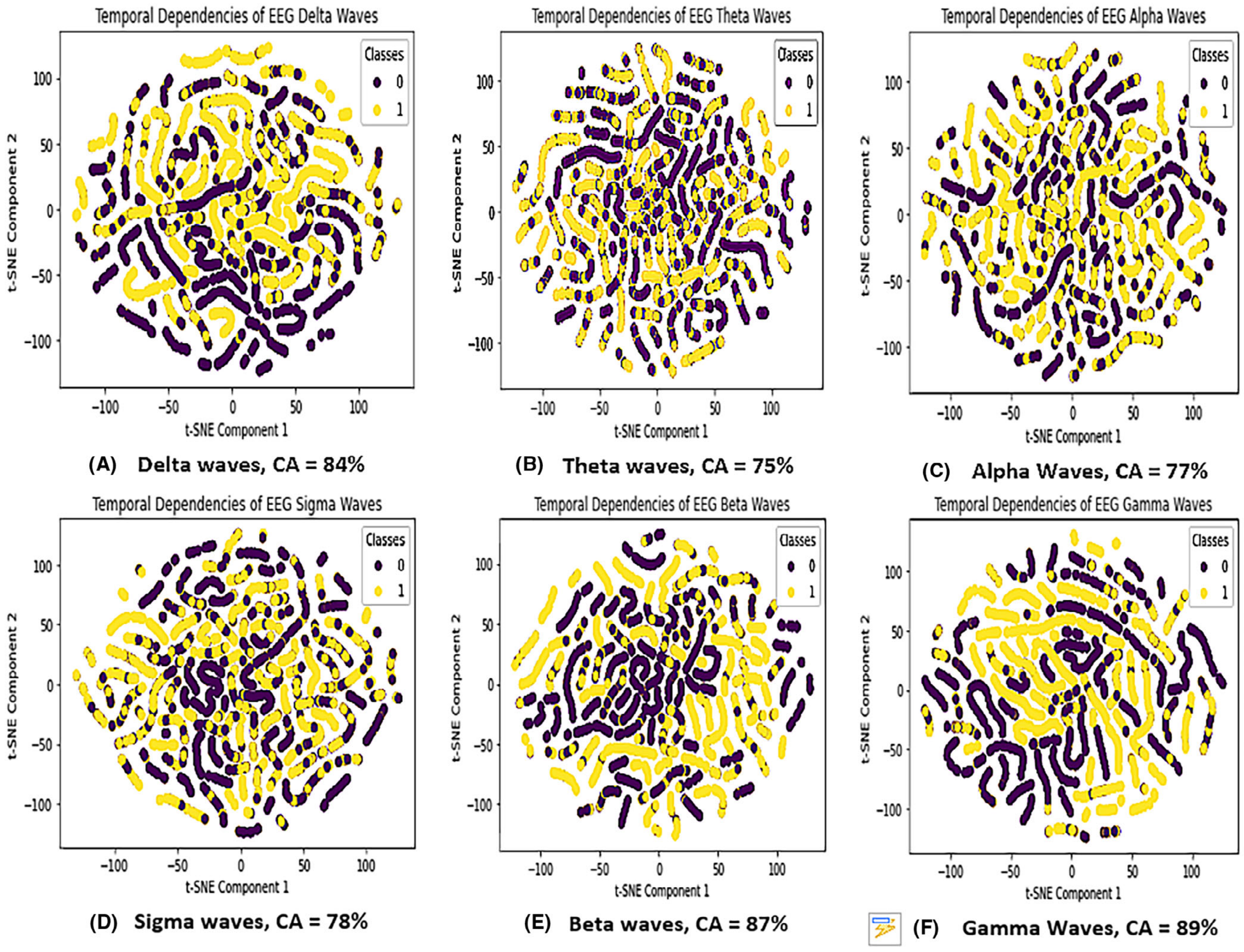


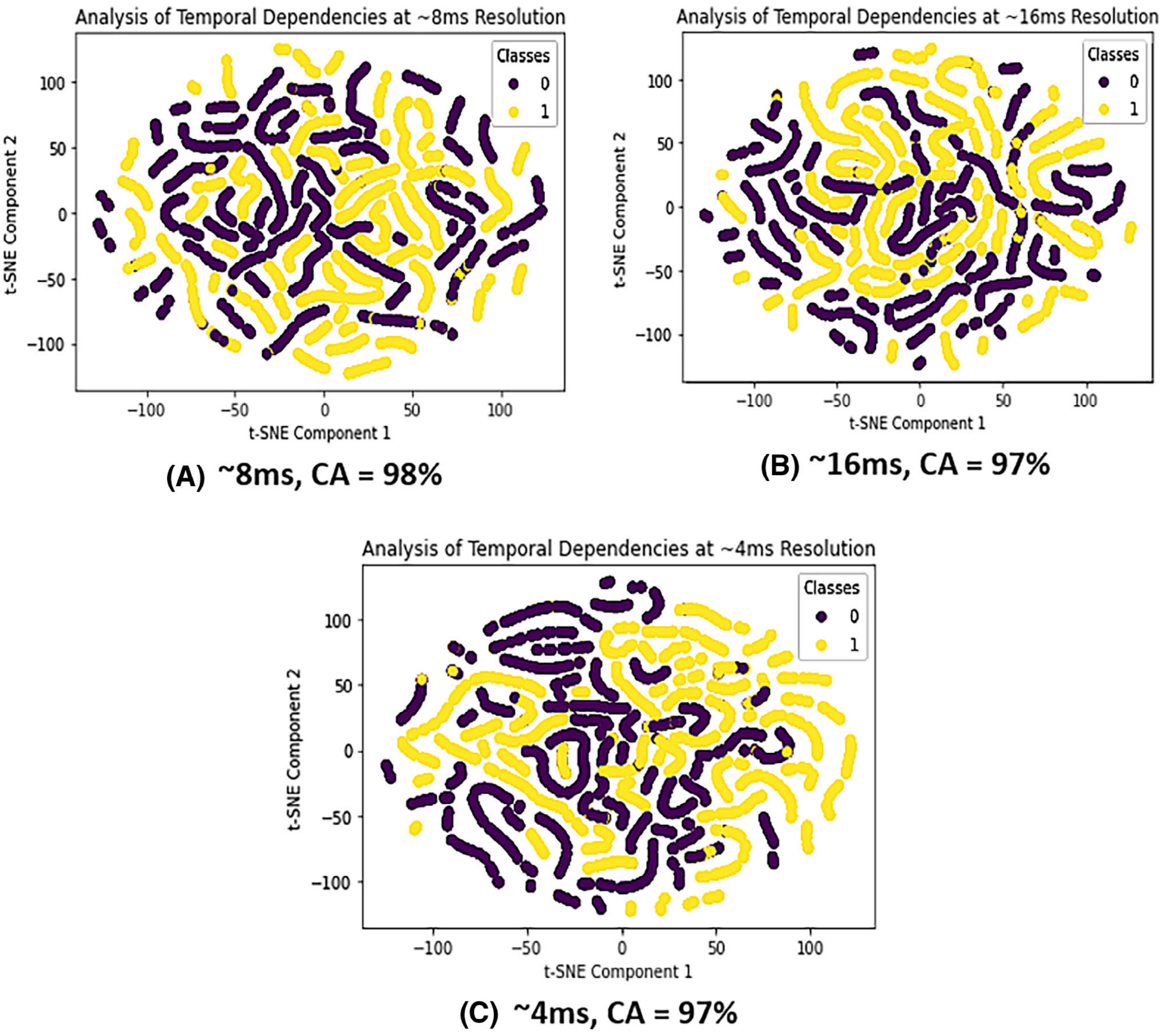
FIGURE 10 t-SNE visualization of frequency specific temporal patterns. Delta waves (A), theta waves (B), alpha waves (C), sigma waves (D), beta waves (E), gamma waves (F). Yellow = epileptic class, Purple = Non-epileptic, CA = Classification Accuracy: This figure displays frequency-specific temporal patterns of the four models.

(s)/128 (~8 ms). We evaluate our model using resolutions of ~4 and ~16 milliseconds (ms). A medium-high temporal resolution of ~8 ms allows the model to capture quick brain processes that occur in millisecond timescales, such as evoked potentials or transient oscillations, which are important in understanding cognitive processes and neurological disorders. Figure 11 highlights the analysis of temporal dynamics of EEG data.

4.6 | Temporal correlations between EEG channels

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

$$r = \frac{\sum (x_i - x^{-})\,(y_i - y^{-})}{\sqrt{\left(\sum (x_i - x^{-})^2 \sum (y_i - y^{-})^2\right)}} \tag{20}$$



(A) ~8ms, CA = 98%

(B) ~16ms, CA = 97%

(C) ~4ms, CA = 97%

FIGURE 11 t-SNE visualization of temporal dependencies at varied resolutions. (A) ~8 ms, (B) ~16 ms, (C) ~4 ms: This figure displays highlights from the analysis of temporal dynamics of EEG data.

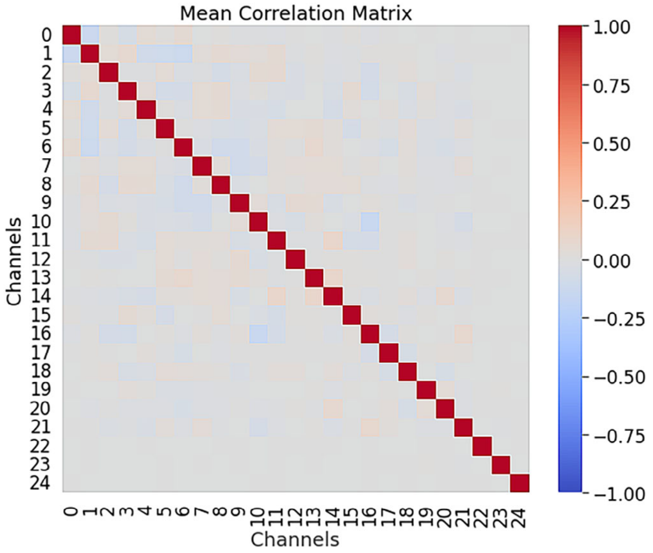


FIGURE 12 Correlation matrix of EEG channels between EEG epochs: This figure displays heatmaps as a result of assessing the temporal correlations between EEG channels by calculating the Pearson Correlation Coefficient (r) between electrodes of EEG channels in an epoch.

We assess the temporal correlations between EEG channels by calculating the Pearson Correlation Coefficient (r) between electrodes of EEG channels in an epoch as expressed in Equation (20) 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 12.

The Mean Correlation Matrix heatmap shown in Figure 12 shows a much stronger positive correlation between electrodes from similar channels of a sample preprocessed EEG data at a particular time point.

5 | DISCUSSION

5.1 | 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 strengths and weaknesses using metrics such as accuracy, precision, recall, F1-Score, Mathew's Correlation Coefficient, and Cohen's Kappa. We note that LConvNet has the best overall performance followed by DeepConvNet, EEGNet, and ShallowConvNet. Similarly, based on the AUC curve shown in Figure 6, all models are seen to possess good discriminating abilities, which is consistent with good accuracy and a good balance between precision and recall.

Subsequently, the models' robustness was assessed using adversarial data, of concern, the accuracy of all models dropped significantly. This indicates that the models are not robust to handle malicious 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 a model's 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 raises possible concerns for malice and intentional sophisticated manipulation for the purpose of misclassification. Researchers and practitioners should focus on developing robust models to ensure their suitability for deployment in sensitive domains.

5.2 | Trainability of the models

Overall, the findings suggest that each of the models has its own strength and weaknesses in terms of the 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 is 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.

5.3 | 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 the 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 the highest scalability among the four models.

Furthermore, we assess the models' efficiency through the number of parameters that can be processed by the models in a single training cycle. The results indicate that LConvNet and DeepConvNet are the most parameters 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 a 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.

5.4 | LConvNet temporal dependencies

Derived results demonstrate that LConvNet is capable of extracting frequency-specific temporal patterns critical for classification tasks. This suggests that frequency domain analysis is a powerful technique for identifying features relevant for classification. Subsequently, it highlights the importance of preprocessing steps like bandpass 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 an accurate classification of EEG signals in clinical settings. The findings may have implications for the development of more advanced EEG signal processing techniques that can be used to diagnose and treat neurological disorders.

6 | CONCLUSION

This study adeptly confronts the challenges associated with binary classification of EEG bio-signals, particularly the subjectivity and error-proneness inherent in conventional analysis methods due to the dynamic and non-stationary nature of EEG data. The innovative LConvNet model proposed in this research strategically harnesses the capabilities of Convolutional Neural Networks (CNN) for spatial feature extraction and Long Short-Term Memory (LSTM) networks for capturing temporal dependencies. Through the Utilization of open-source EEG data from Temple University Hospital, the model achieved remarkable success, outperforming existing EEG classification models such as EEGNet, DeepConvNet, and ShallowConvNet. Additionally, the model exhibited exceptional performance in terms of trainability, scalability, and parameter efficiency during further testing.

AUTHOR CONTRIBUTIONS

Swaleh M. Omari: Conceptualization (equal); data curation (equal); formal analysis (equal); investigation (equal); methodology (equal); writing – original draft (equal); writing – review and editing (equal). **Michael Kimwele:** Conceptualization (equal); methodology (equal); supervision (equal); writing – review and editing (equal). **Akeem Olowolayemo:** Conceptualization (equal); methodology (equal); supervision (equal); writing – review and editing (equal). **Dennis Kaburu:** Conceptualization (equal); methodology (equal); supervision (equal); writing – review and editing (equal).

CONFLICT OF INTEREST STATEMENT

The authors have no conflict of interest relevant to this article.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/eng2.12827.

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 M. Omar https://orcid.org/0009-0004-7452-332X

REFERENCES

1. Patel SD, Seifi A, Patel HV. *Electroencephalography (EEG)—StatPearls*. StatPearls Publishing; 2022. https://www.ncbi.nlm.nih.gov/books/NBK470319/
2. Benbadis SR, Kaplan PW. The dangers of over-reading an EEG. *J Clin Neurophysiol*. 2019;36(4):249.
3. Li X, Jia X, Xun G, Zhang A. Improving EEG feature learning via synchronized facial video. Paper presented at: 2015 IEEE International Conference on Big Data (Big Data), pp. 843–848. IEEE, 2015.
4. Torkamani S, Lohweg V. Survey on time series motif discovery. *Wiley Interdiscip Rev: Data Mining Knowl Discov*. 2017;7(2):e1199.
5. Xun G, Jia X, Zhang A. Context-learning based electroencephalogram analysis for epileptic seizure detection. Paper presented at: 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 325–330. IEEE, 2015.
6. Li Q, Gao J, Zhang Z, Huang Q, Wu Y, Xu B. Distinguishing epileptiform discharges from normal electroencephalograms using adaptive fractal and network analysis: a clinical perspective. *Front Physiol*. 2020;11:828. doi:10.3389/fphys.2020.00828/full
7. Li Q, Gao J, Huang Q, Wu Y, Xu B. Distinguishing epileptiform discharges from normal electroencephalograms using scale-dependent Lyapunov exponent. *Front Bioeng Biotechnol*. 2020;8:1006. doi:10.3389/fbioe.2020.571064/full
8. Stiehl A, Flammer M, Anselstetter F, et al. Topological analysis of low-dimensional phase space trajectories of high-dimensional EEG signals for classification of Interictal Epileptiform discharges. Paper presented at: 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), pp. 1–5, 2023. https://ieeexplore.ieee.org/document/XXXXXXX
9. Sun Y-Y, Chen W-J, Huang Z-P, et al. Trim32 deficiency impairs the generation of pyramidal neurons in developing cerebral cortex. *Cell*. 2022;11(3):449.
10. Saichand NV. Epileptic seizure detection using novel multilayer lstm discriminant network and dynamic mode koopman decomposition. *Biomed Signal Process Control*. 2021;68:102723.
11. Tatum WO IV. *Handbook of EEG Interpretation*. Springer Publishing Company; 2021.
12. Jiang W, Zhou T, Li T. Detecting epileptic seizures in EEG signals with complementary ensemble empirical mode decomposition and extreme gradient boosting. *Entropy*. 2020;22(2):140.
13. Alzubaidi A, Tepper J, Lotfi A. A novel deep mining model for effective knowledge discovery from omics data. *Artif Intell Med*. 2020;104:101821.
14. Roy Y, Banville H, Albuquerque I, Gramfort A, Falk TH, Faubert J. Deep learning-based electroencephalography analysis: a systematic review. *J Neural Eng*. 2019;16(5):51001.
15. Das K, Daschakladar D, Roy PP, Chatterjee A, Saha SP. Epileptic seizure prediction by the detection of seizure waveform from the pre-ictal phase of EEG signal. *Biomed Signal Process Control*. 2020;57:101720.
16. Agarwal S, Zubair M. Classification of alcoholic and non-alcoholic EEG signals based on sliding-ssa and independent component analysis. *IEEE Sens J*. 2021;21(23):26198-26206.
17. Tuncer T, Dogan S, Ertam F, Subasi A. A dynamic center and multi threshold point based stable feature extraction network for driver fatigue detection utilizing EEG signals. *Cognit Neurodyn*. 2021;15:223-237.
18. Feudjio C, Noyum VD, Mofendjou YP, 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.

19. Xiaoqi X, Drougard N, Roy RN. Dimensionality reduction via the Laplace-Beltrami operator: application to EEG-based BCI. Paper presented at: 2021 10th International IEEE/EMBS Conference on Neural Engineering (NER), pp. 347–350. IEEE, 2021.
20. Nassif F, Beheshti S. Automatic order selection in autoregressive modeling with application in EEG sleep-stage classification. Paper presented at: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5135–5139. IEEE, 2021.
21. Lin L-C, Ouyang C-S, Rong-Ching W, Yang R-C, Chiang C-T. Alternative diagnosis of epilepsy in children without epileptiform discharges using deep convolutional neural networks. *Int J Neural Syst*. 2020;30(5):1850060.
22. Wang Z, Na J, Zheng B. An improved KNN classifier for epilepsy diagnosis. *IEEE Access*. 2020;8:100022-100030.
23. Wei Q, Wang Z, Hong H, et al. A residual based attention model for EEG based sleep staging. *IEEE J Biomed Health Inform*. 2020;24(10):2833-2843.
24. Pennington J, Socher R, Manning CD. Glove: global vectors for word representation. Paper presented at: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, 2014.
25. Baevski A, Zhou Y, Mohamed A, Auli M. wav2vec 2.0: a framework for self-supervised learning of speech representations. *Adv Neural Inf Process Syst*. 2020;33:12449-12460.
26. Yuan Y, Xun G, Suo Q, Jia K, Zhang A. Wave2vec: deep representation learning for clinical temporal data. *Neurocomputing*. 2019;324:31-42.
27. Baevski A, Zhou Y, Mohamed A, Auli M. wav2vec 2.0: a framework for self-supervised learning of speech representations. Paper presented at: Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS), Vancouver, Canada, Dec. 2020, pp. 12449–12460.
28. Deng X, Zhang B, Nian Y, Liu K, Sun K. Advanced tsgl-eegnet for motor imagery EEG-based brain-computer interfaces. *IEEE Access*. 2021;9:25118-25130.
29. Huang W, Xue Y, Lingkai H, Liuli H. S-eegnet: electroencephalogram signal classification based on a separable convolution neural network with bilinear interpolation. *IEEE Access*. 2020;8:131636-131646.
30. Stancin I, Cifrek M, Jovic A. A review of EEG signal features and their application in driver drowsiness detection systems. *Proc IEEE Sens*. 2021;21(11):3786.
31. Movahedi F, Coyle JL, Sejdić E. Deep belief networks for electroencephalography: a review of recent contributions and future outlooks. *IEEE J Biomed Health Inform*. 2018;22(3):642-652.
32. Hasan MJ, Shon D, Im K, Choi H, Yoo D, Kim J. Sleep state classification using power spectral density and residual neural network with multichannel EEG signals. *IEEE Appl Sci*. 2020;10(21):7639.
33. Ameera A, Saidatul A, Ibrahim Z. 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, p. 12030.
34. Acharya S, Bhat TS, Pani SB. Deep learning-based classification of EEG signal using wavelet transformation. *Adv Intell Syst Comput*. 2020;1029:231-240.
35. Yang B, Zhu X, Liu Y, Liu H. A single-channel EEG based automatic sleep stage classification method leveraging deep one-dimensional convolutional neural network and hidden markov model. *Biomed Signal Process Control*. 2021;68:102581.
36. Lei F, Liu X, Dai Q, Ling BW-K. Shallow convolutional neural network for image classification. *SN Appl Sci*. 2020;2:1-8.
37. Kim S-J, Lee D-H, Lee S-W. Rethinking CNN architecture for enhancing decoding performance of motor imagery-based EEG signals. *IEEE Access*. 2022;10:96984-96996.
38. Riyad M, Khalil M, Adib A. Mi-eegnet: a novel convolutional neural network for motor imagery classification. *J Neurosci Methods*. 2021;353:109037.
39. Ronneberger O, Fischer P, Brox T. 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, pp. 234–241. Springer, 2015.
40. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. *Proc IEEE Conf Comput Vis Pattern Recognit*. 2016;2818-2826.
41. Lawhern VJ, Solon AJ, Waytowich NR, Gordon SM, Hung CP, Lance BJ. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *J Neural Eng*. 2018;15(5):056013.
42. Schirrmeister RT, Springenberg JT, Fiederer LDJ, et al. Deep learning with convolutional neural networks for EEG decoding and visualization. *Hum Brain Mapp*. 2017;38(11):5391-5420.
43. Obeid I, Picone J. The temple university hospital EEG data corpus. *Front Neurosci*. 2016;10:196.
44. Percival DB, Walden AT. *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge University Press; 1993.

How to cite this article: Omar SM, Kimwele M, Olowolayemo A, Kaburu DM. Enhancing EEG signals classification using LSTM-CNN architecture. *Engineering Reports*. 2023;e12827. doi: 10.1002/eng2.12827