



General and patient-specific seizure classification using deep neural networks

Yasmin M. Massoud¹ · Mennatallah Abdelzاهر⁵ · Levin Kuhlmann^{2,3} · Mohamed A. Abd El Ghany^{1,4}

Received: 6 April 2022 / Revised: 4 January 2023 / Accepted: 22 February 2023
© The Author(s) 2023

Abstract

Seizure prediction algorithms have been central in the field of data analysis for the improvement of epileptic patients' lives. The most recent advancements of which include the use of deep neural networks to present an optimized, accurate seizure prediction system. This work puts forth deep learning methods to automate the process of epileptic seizure detection with electroencephalogram (EEG) signals as input; both a patient-specific and general approach are followed. EEG signals are time structure series motivating the use of sequence algorithms such as temporal convolutional neural networks (TCNNs), and long short-term memory networks. We then compare this methodology to other prior pre-implemented structures, including our previous work for seizure prediction using machine learning approaches support vector machine and random under-sampling boost. Moreover, patient-specific and general seizure prediction approaches are used to evaluate the performance of the best algorithms. Area under curve (AUC) is used to select the best performing algorithm to account for the imbalanced dataset. The presented TCNN model showed the best patient-specific results than that of the general approach with, AUC of 0.73, while ML model had the best results for general classification with AUC of 0.75.

Keywords Electroencephalogram · Temporal convolutional network · Machine learning · Support vector machine · Random-under sampling boost · Area under curve · False positive rate

✉ Yasmin M. Massoud
yasmin.massoud@guc.edu.eg

✉ Mennatallah Abdelzاهر
menatalah.abdelzاهر@student.guc.edu.eg

✉ Levin Kuhlmann
levin.kuhlmann@monash.edu

✉ Mohamed A. Abd El Ghany
mohammed.abdel-ghany@guc.edu.eg

¹ Electronics Engineering Department, German Univeristy in Cairo, Cairo, Egypt

² Biomedical Engineering Department, Melbourne Univeristy, Parkville, Australia

³ Brain Dynamics Lab, Swinburne University, Melbourne, Australia

⁴ Integrated Electronics Systems Lab, Technische Universität Darmstadt, Darmstadt, Germany

⁵ Networks Engineering Department, German Univeristy in Cairo, Cairo, Egypt

1 Introduction

Neurological disorders are diseases targeting the brain, spine, and the nerves connecting them. These diseases occur in nearly 1 billion of the world's population, so offering adequate help is of the utmost importance to improve the quality of life of those affected. Epilepsy is a neurological disorder that impacts the human nervous system. Ranked as the fourth most common neurological disorder, epilepsy occurs in one of 26 people during their lifetime [1]. The World Health Organization states that around 50 million individuals around the world are impacted by epilepsy [2]. Of the world's population, 0.4–1% are diagnosed with active epilepsy. Some of its side effects include sudden uncontrollable seizures, lack of energy, and migraines. For example, if an epileptic seizure begins while an epileptic person is driving or swimming, they could be in fatal danger. The sudden onset of seizures seriously endangers the individual's life.

In recent years, there have been advancements in the field of seizure prediction, and that has allowed intervention during seizures, such as medications or invasive electrodes [3–5]. The drawbacks of such methods include

the harmful effects of epileptic drugs on the liver and kidneys [6]; as well as adverse effects on female reproductive organs [7], also using invasive electrodes causes irreversible damage to brain tissues.

To provide a solution for epilepsy two main steps have to be taken into consideration, the first being recording seizure signals to study their nature, then predicting these signals before they happen to allow safe intervention by the epilepsy patient's caretaker.

Electroencephalograms (EEGs) reflect the brain activity in milliseconds in temporal resolution, which allows accurate recording of seizure signals. Several parameters need to be considered upon the recording of seizure data such as the headset to use, the patient's history, and the quality, timing, and file formats of EEG signals. Recent progress in this area has yielded several databases that have established major advances in the field of seizure treatment, such as the Epilepsiae database [8], IEEG.org [9], the Physionet database [10], and the Epilepsy Ecosystem database of long continuous intracranial (iEEG) data [11].

Machine learning (ML) and deep learning (DL) algorithms have gained a reputation in the past few years for seizure classification [12–15]. One of the main challenges of machine learning (ML) and deep learning (DL) algorithms is the availability of long-term EEG data. Up until 2008, the longest EEG data recordings were about 2 weeks in duration [16], which doesn't aid in delivering a sufficient number of seizures for ML and DL applications. In this work, we use DL algorithms to develop an accurate seizure prediction system using the data from the [11] platform. This platform and data were chosen for two main reasons. Primarily, the data recorded on the ecosystem platform contains continuous iEEG recordings between 6 months and 3 years in length, thus overcoming the problem of the short-term data. Secondly, the platform is used to crowdsource current state-of-the-art algorithms, which ensures fair competition of the best performing algorithms on this data.

This paper aims to present an accurate seizure classification system. DL and ML models are implemented, and best-performing algorithms are compared for performance as general and patient-specific approaches. The patient-specific approach is heavily dependent on patient-specific biomarkers; consequently, the general approach struggles due to the lack of said specificity. That mentioned, most seizure prediction systems are carried out for each patient independently. In this work, both approaches are implemented and compared for the best-performing algorithms. We present a TCNN model for the task of seizure prediction, and it's compared to other competing DL algorithms in this field. We also compare the performance of the TCNN model to SVM and RBT model that we published in [17], which showed high classification accuracy on the ecosystem data.

The rest of this paper is structured as follows: Sect. 2 “Related Work” presents relevant work previously done about seizure prediction. In Sect. 3 “Proposed Method”, the input data and its required preprocessing are discussed alongside the different Neural Network architectures and finally the evaluating performance metrics. Section 4 “Results and Discussion” recounts the hyperparameters and the effects of their tuning, including data balancing methods. It shows the results of our proposed approaches. In Sect. 5 “Conclusion”, we summarize our methodology and results.

2 Related work

The last two decades have witnessed a surge of research work related to seizure prediction. The primary success of seizure prediction lies in differentiating between the interictal (seizures) and pre-ictal periods (just before the seizure) on time.

Most seizure prediction algorithms, presented in the past few years, relied on applying pre-processing techniques and extraction of hand-crafted EEG features [18]. Work in [19, 20] included the use of online recursive independent component analysis and enhanced automatic wavelet independent component analysis as a preprocessing step for feature extraction. Feature extraction techniques included genetic algorithms which were used to enhance a selection of the most effective features [21], extraction of Spatio-temporal features [22], and, spectral power [23]. Another group of features for the task of EEG classification are the wavelet and PCA components, which were evaluated in [24, 25], which are successful for extracting spikes in EEG signals and thus signal classification.

The use of machine learning algorithms has shown major success in EEG seizure classification. The success of ML methods depends on the selection of the most deterministic features to differentiate between seizure segments. Several ML algorithms have shown major success, such as support vector machines (SVM) [26–28], and random forests [29, 30]. The boosted trees ensemble is another type of ML classification algorithm, which has benefits such as being decisive in data where there is a class imbalance, like in seizure classification problems [17, 31]. The disadvantage of using ML algorithms for the task of seizure prediction, is that they heavily depend on the extracted features. Moreover, these features vary in prominence as per the dataset. That makes producing a generalized seizure prediction algorithm more precarious.

Medical practitioners can predict seizure by classifying different shapes and values of the seizure signal. The same approach motivated the use of deep learning algorithms to extract seizure features automatically in the same fashion a medical practitioner would. Several researchers have

experimented with different deep neural network architectures for the task of seizure prediction. The authors of [32, 33] used a spectrogram to extract seizure features and a Convolutional Neural Network (CNN) to learn the extracted features. Another approach that has recently been published, is the use of wavelet features in a deep recurrent CNN, such models are known for their success in classifying sequence data such as EEG signals [34]. The shortcomings of such a method, are the short memory and vanishing gradient problems [35–37]. Moreover, several types of recurrent neural networks have shown success with seizure prediction, such as bidirectional long-short term memory (Bi-LSTM), which solves the problem of short memory by storing sequencing of necessary data and throwing away unneeded data [38–40]. Additionally, raw EEG signals are converted into images and used in CNNs, which act as a classifier [41, 42], this method is closest to practice by medical practitioners where visual features of seizures could be extracted using various image classifiers such as ImageNet [43] and DenseNet [44]. An approach based on raw EEG signals would be more beneficial to the task of seizure prediction, as it's able to generalize to different datasets and patients.

Other techniques that could be used in seizure classification, can be inspired from other work using (DL) algorithms for hardware fault detection. These techniques include, using Fast Fourier Transform (FFT) as a pre-processing step and then further reducing noise in dataset using principal component analysis (PCA) and feeding feature vector into CNN model [45]. The work is further developed to implement a full hardware system using Altera FPGA. The suggested approach helps to provide high accuracy to a diagnostic fault in an acceptable amount of time. The work in [46], implements an abnormal heartbeat detector using DL, also a complementary metal-oxide-semiconductor (CMOS) design is presented to produce a wearable device. Raw ECG signals are fed into the CNN classifier and very large-scale integration (VLSI) chip is implemented accordingly.

In this paper, several deep learning models are tested on raw EEG data from the Kaggle 2016 contest [11]. Results are compared to an SVM and random under-sampling with boosted trees ensemble (RBT) ML model that reported high classification results on epilepsy-ecosystem data to check the DL models' performance.

3 Proposed method

As formerly stated, the purpose of this paper is to employ DL algorithms to construct a clinically reliable seizure prediction system. Within this section, DL algorithms used in this work will be introduced and discussed, alongside the metrics using which we gauged the quality of the computed results. In addition, the DL algorithms' performances' are compared to those of the Support Vector Machine and boosted trees ensemble (RBT). Figure 1 represents the methodology used in this work, starting with sampling data, followed by dividing data into different sets, and finally patient-specific and general training are carried out using different models to assess prediction.

3.1 Data used

In 2016 the Melbourne University seizure prediction competition [11] aimed to predict seizures utilizing solely long-term human intracranial EEG (iEEG) recordings. Recordings were subdivided into ten-minute iEEG clips sampled from 16 electrodes at 400 Hz. The EEG clips are either pre-ictal or inter-ictal data. Pre-ictal data is collected before the occurrence of a seizure or stroke, while inter-ictal data refers to that which is within the interval between seizures or convulsions, more specifically- non-seizure data. Figure 2 shows a 1-second sample of training data used from the 3 patients, for both pre-ictal and inter-ictal samples. The number of files used for training, testing, and the percentage of inter-ictal files in training data are shown in Table 1.

3.2 Data preprocessing

As a pre-processing step, the 10-minute files are sampled into window sizes of 75 s and 300 s. Different window sizes were used by competing algorithms on the kaggle data, ranging from 20 s to 600 s. The highest AUC scores were produced with windows sizes from 20 to 600 s, with 60 and 75 s windows sizes showing the most discriminant seizure features. It was prominent to try different window sizes, as no information was provided at what timescale predictive features would be present. Several values for window size were attempted to find the highest AUC, where window size of 75 s produced the best results.

Fig. 1 Proposed system architecture

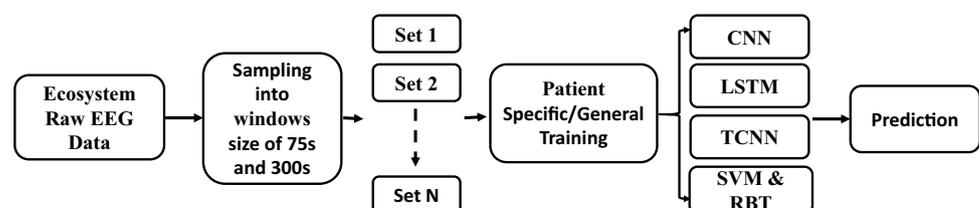


Fig. 2 Pre-ictal segment of training data, sampled at 400 HZ

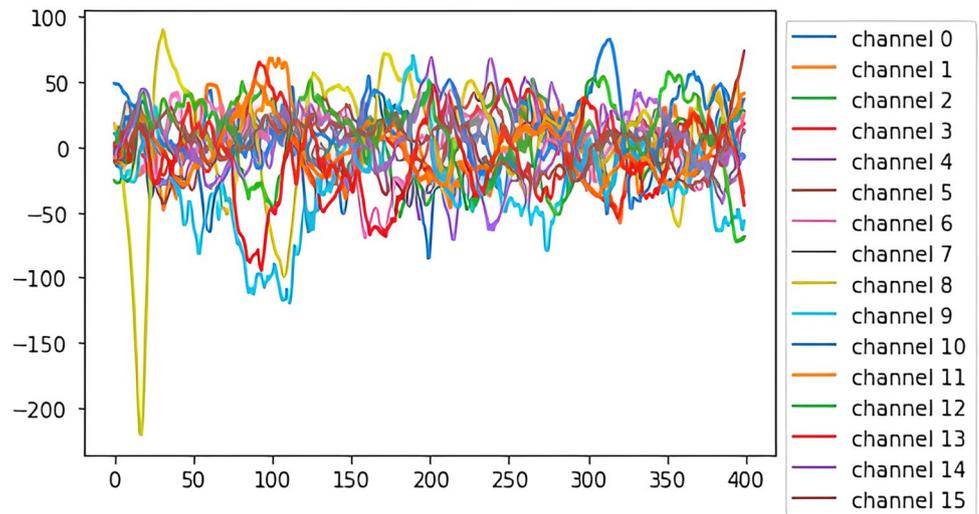


Table 1 Details of data from epilepsy-ecosystem

Patient	Train files	%Interictal	Test files
1	826	69.1	216
2	2058	89.3	1002
3	2963	88.2	690

Moreover, one of the complications of the DL network is training large amounts of data, like the one used in this work. To be able to carry out training for the 200 GB dataset, transfer learning is performed on divisions of the dataset. The 75 s window size (30,000) sampling rate ensured the result from sampling produces an integer number of files to be able to carry out training. Knowledge from training previous sets is transferred to the following sets. This method allowed the training of the data on an RTX 2060 GPU with 16 GB RAM. The use of transfer learning in this context allowed for the successful completion of training on the available hardware resources.

3.3 Deep neural networks

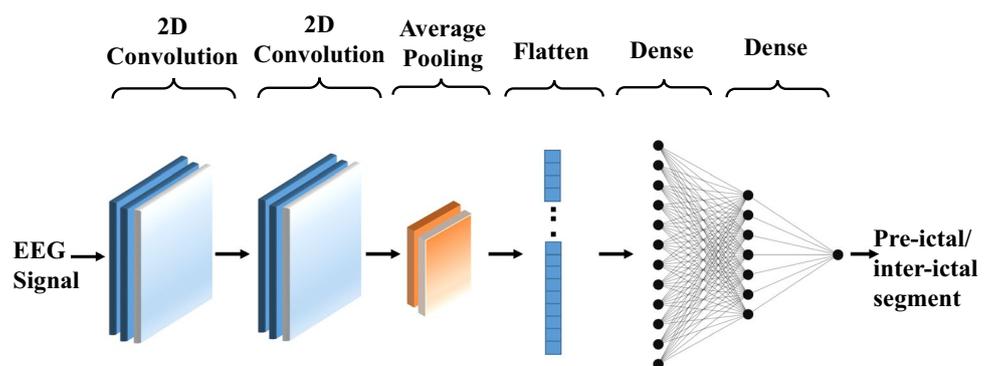
3.3.1 Convolutional neural networks

CNNs are among the most significant and powerful DL methods; they are a subset of multi-layer neural networks. By leveraging principles of linear algebra, they pass two-dimensional data to hierarchical layers for feature extraction. The convolution of two discrete signals x_n and w_n is defined in Eq. 1, this operation is easily broadened to be a multidimensional operation. The \otimes represents the convolution operation.

$$x_n \otimes w_n = \sum_{m=-\infty}^{\infty} x_m w_{n-m} \tag{1}$$

CNN's structure makes them ideal for EEG dual dimensionality in the form of time and channels. Figure 3 shows CNN architecture used in this work, which is similar to that of [47]. The input of size $240,000 \times 16$ of EEG data is convolved with a filter for feature extraction. The structure consists of two convolutional layers, with kernel size set to 40,

Fig. 3 CNN architecture for seizure classification



an average pooling layer that is used to reduce the feature maps' size. Finally, a fully connected layer and a sigmoid function is added to classify seizure and non-seizure signals.

The sigmoid activation function is used in the last layer. Moreover, the loss function used in this model is binary cross-entropy. The function is given by Eq. 1, where p is the target distribution and q is the observed distribution.

$$H(p, q) = - \sum_i p_i \log(q_i) \tag{2}$$

The training AUC of this model produced an AUC of 0.8. Model layers, output shape, and a number of parameters are shown in Table 2. The results of this model are discussed in Sect. 4.

3.3.2 Temporal convolutional neural networks

CNNs are a generic architectural concept that uses convolutions and dilations, making it most useful with sequential data with its temporality and large, flexible receptive fields. Its novelty comes in the form of two features; the first being that its architectural convolutions are causal, leading to zero information “leakage” from future to past; the second is its

ability to accept as input a sequence of any length and map it to an identically sized output sequence.

The structure of TCNN in this work is inspired by work in [48, 49]. Table 3 presents TCNN architecture details used for the seizure classification task. TCNNs consists of three main structures, causal convolutions, dilated convolution, and residual blocks as represented in Fig. 4. To achieve its causality, TCNNs use a one-dimensional fully-convolutional network (FCN) architecture. It contains hidden layers, where each has a size identical to that of the input layer; to ensure subsequent layers have the same length as those prior, zero-padding of length (kernel size-1) is added. This ensures that an element in the output sequence depends on only elements that came before in the input sequence. For seizure prediction, this satisfies the condition that predictions are made based on past seizure data only.

TCNNs also consist of dilated convolutions, which have the advantage of increasing the receptive field exponentially, while keeping the number of layers relatively small. This allows for full history coverage, where an input sequence of size input length affects a selected output entry. The following are mathematical equations depicting the required operations in the layers to construct the TCNN model. k stands for the Kernel size, C stands for the total number of channels and H and W are the height and width of the tensors, respectively.

$$Conv2D = k_1 \cdot k_2 \cdot C_{in} \cdot C_{out} \cdot H_{out} \cdot W_{out} \tag{3}$$

$$Conv1D = k \cdot C_{in} \cdot C_{out} \cdot W_{out} \tag{4}$$

$$SeparableConv2D = (k_1 \cdot k_2 + C_{out}) \cdot C_{in} \cdot H_{out} \cdot W_{out} \tag{5}$$

$$DepthWiseConv2D = k_1 \cdot k_2 \cdot C_{in} \cdot D \cdot H_{out} \cdot W_{out} \tag{6}$$

ReLU activations are used after the convolution layers, and regularization is deployed as a method to prevent overfitting

Table 2 CNN model summary

Layer type	Output shape	Param #
Conv2D	(None, 30,000, 16, 40)	1240
Conv2D	(None, 30,000, 1, 40)	25,640
AveragePooling2D	(None, 1999, 1, 40)	0
Flatten	(None, 79,960)	0
Dense	(None, 80)	6,396,880
Dense	(None, 1)	81
Total parameters		6,423,481

Fig. 4 TCNN architecture for seizure classification

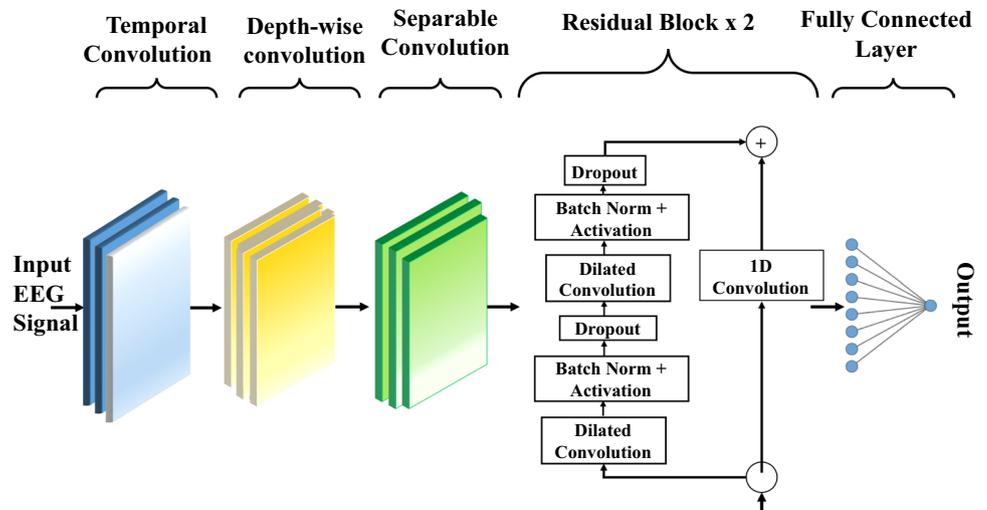


Table 3 TCNN model summary

Block	Layer type	Output shape	Param #
Temporal convolution	InputLayer	(None, 1, 16, 3000)	0
	Permute	(None, 30,000, 16, 1)	0
	Conv2D	(None, 30,000, 16, 8)	256
	BatchNormalization	(None, 30,000, 16, 8)	32
Depth-wise convolution	DepthwiseConv2D	(None, 30,000, 1, 16)	256
	BatchNormalization	(None, 30,000, 1, 16)	64
	Activation	(None, 30,000, 1, 16)	0
	AveragePooling2D	(None, 3750, 1, 16)	0
	Dropout	(None, 3750, 1, 16)	0
Separable convolution	SeparableConv2D	(None, 3750, 1, 16)	512
	BatchNormalization	(None, 3750, 1, 16)	64
	Activation	(None, 3750, 1, 16)	0
	AveragePooling2D	(None, 468, 1, 16)	0
	Dropout	(None, 468, 1, 16)	0
1st Residual block	Lambda	(None, 468, 16)	0
	Conv1D	(None, 468, 12)	780
	BatchNormalization	(None, 468, 12)	48
	Activation	(None, 468, 12)	0
	Dropout	(None, 468, 12)	0
	Conv1D	(None, 468, 12)	588
	BatchNormalization	(None, 468, 12)	48
	Activation	(None, 468, 12)	0
	Dropout	(None, 468, 12)	0
	Conv1D	(None, 468, 12)	204
2nd Residual block	Add	(None, 468, 12)	0
	Activation	(None, 468, 12)	0
	Conv1D	(None, 468, 12)	588
	BatchNormalization	(None, 468, 12)	48
	Activation	(None, 468, 12)	0
	Dropout	(None, 468, 12)	0
	Conv1D	(None, 468, 12)	588
	BatchNormalization	(None, 468, 12)	48
	Activation	(None, 468, 12)	0
	Dropout	(None, 468, 12)	0
Output	Add	(None, 468, 12)	0
	Activation	(None, 468, 12)	0
Output	Lambda	(None, 12)	0
	Dense	(None, 1)	13
Output	Softmax	(None, 1)	0
	Total parameters		4,137

after each residual block. Also used within this model is dropout after every convolutional layer, which is an effective technique commonly used to regularize neural networks by randomly removing a subset of hidden node values and setting them to zero.

Followed by depthwise and separable convolutions, residual blocks are used, which takes as an input the output of convolutional layers. To adjust the input and output channel width of residual blocks, a 1×1 convolution is used. The

minimum number of residual blocks required for full history coverage is given by Eq. 7, where, b is the dilation base and k is the kernel size, where in the architecture used in this work $n=2$.

$$n = \left\lceil \log_b \left(\frac{(l-1)(b-1)}{k-1} \right) + 1 \right\rceil \quad (7)$$

3.3.3 Long short term memory

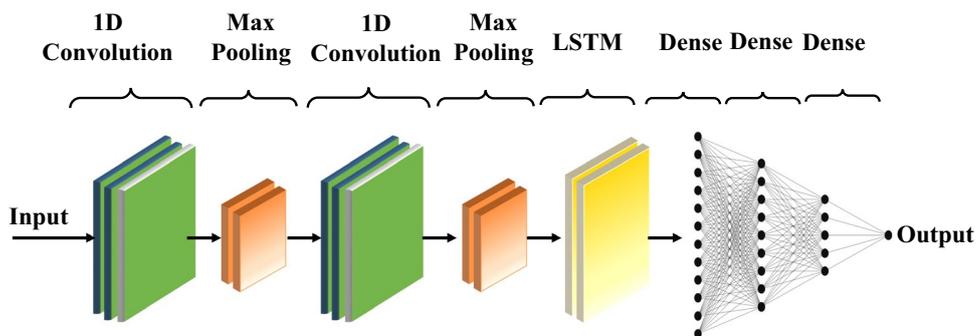
LSTM is a gated recurrent network architecture joint with a suitable gradient-based learning algorithm, created to avoid the long-term dependency problem and overcome the error back-flow problems and unstable gradient problems. Long-term dependencies are generally obtained from sequentially aligned input data, considering only forward dependencies. Unstable gradient problems come in the form of either the vanishing gradient problem or the exploding gradient problem. In a gradient-based training sense, the vanishing gradient problem denotes when the gradients of network weight are near zero. Thus impeding the reception of updates in the network weights by the gradients [50]. As for the exploding gradient issue, it is most aptly described as when the gradient back propagating through the network increases exponentially, layer to layer [51].

Figure 5 shows the architecture of the LSTM network, which is based on the CNN model described in Sect. 3.3.1. The first convolutional layer performs linear channel filtering through temporal convolution. The second layer deals with channels that share the same time instances. The kernel size is set to 16, to match the number of channels available, reducing channel dimension and producing a 1-D time series to be further reduced using an average pooling layer. Furthermore, a time distributed layer is used to produce an output for each time step, as the LSTM layer deals with time-series sequential data. Finally, a fully connected layer, together with a sigmoid function, is implemented to produce

Table 4 LSTM model structure

Layer type	Output shape	Param #
Conv2D	(None, 30,000, 16, 40)	1240
Conv2D	(None, 30,000, 1, 40)	25,640
AveragePooling2D	(None, 60,000, 40)	0
TimeDistributed	(None, 6000, 40)	0
LSTM	(None, 40)	12,960
Dense	(None, 1)	41
Total parameters		39,881

Fig. 5 LSTM architecture for seizure classification



seizure probabilities. Layer output and number of parameters are illustrated in Table 4.

3.4 Differences between DNN methods

The proposed models above are vastly different, albeit sharing a few commonalities. The LSTM operates on a largely different basis, it uses gates to decide whether the input is to be forwarded to the next layers, and which inputs that is. The gates are not present in any other model. In the topology, it is the only model that uses a Time Distributed layer, which is placed there to cater to the input data’s temporal nature.

The prime distinctions between TCNNs and CNNs are threefold: Causal Convolutions, Dilated Convolutions, and Residual Blocks; all three are present in TCNNs only. Causal Convolutions are in place to guarantee that the produced output at a point in time is dependent only on the input at that same point and those before it. Each hidden layer has a size identical to that of the input layer, and zero-padding of length (kernel size – 1), creating a 1D fully-convolutional network architecture to create this causal effect. Dilated Convolutions use a method of vastly increasing dilation factors, so the network’s receptive field can grow exponentially in size proportional to the network depth. Finally, the Residual Block is composed of a stack of 2 Dilated Convolution layers, a dropout layer in the middle, paired with batch-normalization. The below equation shows the receptive field size (RFS) of the TCN, with K_T as the kernel size and L as the number of residual blocks.

$$RFS = 1 + 2(K_T - 1)(2^L - 1) \tag{8}$$

The hyperparameters used are different as well, an example of that is the activation function used. The TCNN uses the exponential linear unit (ELU) due to it produces superior results, whereas the CNN uses the rectified linear unit (ReLU). Another major disparity is the number of trainable parameters, where the total parameters in the CNN vastly outnumber all other models, which is visualized in Fig. 6 (results and discussion Sect. 4.2).

3.5 Machine learning approach

To further better the performance of deep learning, algorithms extract seizure features automatically. ML algorithms like SVM and RBT classifiers are used to elevate performance. Data is first sampled into windows sizes of 80 s, 160 s, and 180 s respectively. Next, after dividing the data into the selected epochs of different window sizes, features are extracted from each of these groups independently. The extracted features are frequency power of bands, power of frequency, mean, root-mean-square, standard deviation, kurtosis, and correlation between all channel pairs in both time and frequency domain. Afterward, the extracted features from different epochs are joined together for classification. Following feature extraction, SVM and RBT are used for the classification of the dataset of extracted features into pre-ictal and inter-ictal segments.

3.5.1 Support vector machines(SVM)

Support Vector Machines (SVM) are a supervised machine learning technique extensively used for classification, regression, and feature reduction of labeled data. It puts its extensive mathematical foundation to use in the construction of a solution in terms of a subset of the training input, namely the *support vectors*. The *support vectors* are stored in memory during the training phase, after their selection post parameter identification; then moving forward they are employed for future prediction.

An advantage of SVMs is their ability to classify non-linearly separable classes. They do so using the *kernel technique*, which is a method of mapping data into novel hyper-dimensional spaces and then locating the ideal hyperplane in whichever dimension that may be. The ideal hyperplane is the border between classes that maximizes the margin (distance between classes-usually Euclidean distance). Albeit computationally expensive and complex, it yields exceptional results for small datasets. For SVM hyperparameters, a polynomial kernel that accounts for the non-linear nature of the EEG signals is used. A seed is used to generate a random number to scale the polynomial kernel.

3.5.2 Random under sampling (RBT)

The tree ensemble model consists of a set of classification and regression trees (CART); it combines the decisions from numerous ML models to achieve optimum results. The RBT model is developed by using random under-sampling and adaptive boosting (Adaboost). This algorithm is effective when working with imbalanced data, as it adjusts

the class distribution of the used dataset by removing samples from the majority class.

Often, a single tree is insufficient for decent results, which is why an ensemble approach is used where the sum of predictions of multiple trees together is the output. The boosting technique follows a *sequential order*; here, trees are fitted consecutively. If the classifier misclassifies an input, its weight will be incremented (over-weighting) and the next learner will classify with increased accuracy. It gives more weight to the model with the best performance, it reduces bias at the cost of overfitting-which can then be combatted by hyperparameter tuning.

Hyperparameter tuning was performed using *K-fold Cross-Validation*, with $k=6$. A higher k means that each model is trained on a larger training set and tested on a smaller test fold. Theoretically, this produces a reduced prediction error as the models view a larger amount of the available data. The learning rate was selected to be 0.1 for tree learners, as this increased accuracy during training.

3.6 Performance metrics and statistics

Evaluation metrics are used to measure the quality of the DL model quantitatively, which is vital to its performance and optimization. A wide variety of metrics are available, and often one could use a culmination of individual metrics to put a model to the test.

In this paper, four evaluation metrics were used: Area Under Curve (AUC), False Positive Rate (FPR), and sensitivity and accuracy.

Area Under Curve AUC helps the visualization of the model's performance, it is primarily utilized with binary classification problems, which makes it ideal for our model. AUC is considered the most important metric in this work, as the data is highly imbalanced. AUC is calculated by plotting the true positive rate against the false-positive rate. The higher AUC is close to 1 the better the model.

False Positive Rate The false-positive rate corresponds to the proportion of the cases which were predicted positive where they should've been correctly identified as negative concerning all the negative outputs. Its output is within the range $[0,1]$.

Sensitivity Sensitivity determines the model's ability to predict true positives of each available category (pre-ictal and inter-ictal segments).

Accuracy Accuracy is the ratio of correct to total predictions that the model has produced as output. However, sometimes it will be misleading, especially when dealing with imbalanced dataset. As high accuracy can be produced by the model being able to predict correctly majority class only.

4 Results and discussion

In this section, the results of deployed deep learning and machine learning algorithms are presented and discussed. The performance of CNN, TCNN, and LSTM architectures is evaluated on raw seizure data. We also compare the mentioned DL algorithms to our previous work published [17], which focused on using an RBT and SVM classifier that had a competing score on the epilepsy-ecosystem platform. The algorithm was based on the third-place ranked model in the Kaggle 2016 seizure competition contest, and so far its only algorithm able to have a general classification score higher than that of a patient-specific model.

Below are highlights of results to be summarized in this section, architecture descriptions are presented in Sect. 3.

- Evaluate TCNN model performance on raw seizure data.
- Hyperparameter Tuning for the TCNN model.
- Comparing performance to other DL models, namely: CNN, LSTM, ConvNet, EEGNet, and state-of-the-art models.
- Compare the performance of patient-specific format and general classification for TCNN and ML Models.

The superiority of such DL methods over ML algorithms and other DL algorithms is that in this work we train these algorithms on raw EEG data. This qualifies these algorithms to generalize on any seizure data, without the need to use specific pre-processing techniques. Finally, an evaluation of the best models is performed using a patient-specific and general classification approach. Training on patient-specific data has proven to be far superior to training on general data. This is due to patient-unique seizure biomarkers in the patient-specific iEEG files. A general model would be more beneficial to medical practitioners and is obtained by creating one model that classifies well for any patient.

For the results stated below, the AUC is split into 3 categories: overall AUC, public AUC, and private AUC. Public AUC is calculated by the individual(s) who have written the code as a metric to check the code's quality. The private AUC is calculated after the submission of the final algorithm by the epilepsy ecosystem platform; that is, to ensure that the algorithm will work on unseen data moving forward in future applications. Also, this ensures that the algorithm is competitive with state-of-the-artwork, as all algorithms are tested on the same dataset, using the same metrics. That being said, private AUC is the most important metric, as it is determined by the epilepsy-ecosystem platform on completely unseen data. This step qualifies the algorithm for clinical use on any seizure data.

4.1 TCNN algorithm

The time-series nature of EEG signals motivated the use of sequence-to-sequence models such as RNNs and LSTMs, the problem with such methods is vanishing gradients and the long training times. TCNN can be used for sequence modeling as its ability to predict future instances from the previous input sequence while keeping a structure of simple convolutional networks. Table 5 shows results for hyperparameter tuning of the TCNN architecture described in Sect. 3. The results are discussed below:

- **Epochs and batch size** Epoch value optimization was carried out to reach a value that would prevent both overfitting and underfitting. For this model, the number of epochs was initially set as 100, then upped to 1000. This led to a decrement in both overall AUCs, from 0.60 to 0.58, and public AUC, 0.64 to 0.53. In conclusion, this increase in epoch value is deemed ineffective. Oscillations in the number of epochs were performed yet ultimately the optimum value remained 100. For batch size, the starting point was 32 which was then upped to 64 and yielded better results. Higher values for batch size weren't successful during training due to the large data size.
- **Window size** The data was tested on a window size of 300 s, in the form of 120,000×16 channels, then on a window size of 75 s, in the form of 30,000×16 channels. The latter produced an increase in the majority of the AUC values, both patient-specific and general. The latter was retained for hyperparameter tuning moving forward. Details of choosing window size are given in Sect. 3.
- **Learning rate (LR)** determines the step size at each iteration while minimizing the loss function. It also decides the rate at which the model weights are changed during training. The learning rate was decremented to 0.0001 and all 3 general AUC values (overall, public and private) increased but all patient-specific AUC values were almost constant. The FPR decreased significantly, from 0.85 to 0.21 and overall accuracy increased to 53%. This value was chosen for LR.
- **Dropout** is one of the most used and efficient regularization techniques for neural networks, it randomly removes a section of hidden node values by setting them to 0. It can be applied to all types of layers except for the output layer. Each unit is kept/dropped with a fixed probability p independent of other units, where p can be chosen randomly or can simply be set at a definite value; 0.5, for example, seems to be close to optimal for a wide range of networks and tasks. Its main aim is to prevent overfitting and decrease noise while learning.

Initially setting dropout to 0.5, all patient-specific and general AUC values dropped, however, overall accuracy

Table 5 This table shows a comprehensive look at the various hyperparameters alongside their effects on the evaluation metrics

Model	Overall AUC	Private ⁰ /public ⁰	Private ¹ AUC	Public ¹ AUC	Private ² AUC	Public ² AUC	Private ³ AUC	Public ³ AUC	Overall FPR	Overall accuracy (%)	S ⁴ (%)
Initial	0.60	0.56/0.64	0.69	0.75	0.45	0.65	0.60	0.55	0.85	21	90
Epochs (1000)	0.58	0.60/0.53	0.37	0.29	0.61	0.46	0.46	0.52	0.48	46	63
BS (64)	0.61	0.58/0.65	0.45	0.44	0.44	0.60	0.46	0.51	0.22	74	37
WS (300 s)	0.57	0.57/0.58	0.35	0.32	0.46	0.51	0.43	0.51	0.41	59	60
LR(0.0001)	0.59	0.60/0.61	0.40	0.36	0.71	0.57	0.47	0.62	0.48	53	60
Dropout (0.5)	0.55	0.54/0.57	0.68	0.68	0.44	0.54	0.39	0.43	0.33	63	32
Dropout (0.2, 0.3)	0.62	0.46/0.44	0.59	0.43	0.50	0.50	0.51	0.46	0.22	74	83
Dropout (0.1, 0.2)	0.61	0.60/0.63	0.45	0.45	0.42	0.59	0.62	0.61	0.33	65	52
Cross Validation	0.60	0.60/0.60	0.44	0.38	0.36	0.50	0.54	0.38	0.41	59	57
compute weights	0.56	0.58/0.58	0.34	0.43	0.47	0.39	0.58	0.58	0.71	34	74
SMOTE	0.61	0.60/0.63	0.33	0.41	0.47	0.55	0.54	0.53	0.19	76	27
div2effect	0.61	0.59/0.65	0.46	0.44	0.44	0.60	0.46	0.53	0.22	74	37
SGD	0.60	0.56/0.64	0.70	0.76	0.35	0.51	0.58	0.47	0.0005	90	0.5
ADAM	0.68	0.67/0.69	0.75	0.66	0.72	0.66	0.70	0.71	0.42	60	70

Each row represents the hyperparameter value effect on 3 patients' public and private leaderboard AUC, FPR, and accuracy. The results show patient-specific performance

Overall AUC, accuracy, and FPR are calculated over the entire set of files independent of the patient

AUC= area under curve, BS=batch size, WS= window size, LR= learning rate, SMOTE= synthetic minority oversampling technique, SGD= stochastic gradient descent, ADAM= adaptive moment estimation

⁰ Private and Public AUC

¹ Patient 1

² Patient 2

³ Patient 3

⁴ Sensitivity

rose from 21% to 63%. Dropout was then set to 0.2 and 0.3 leading to an increase in overall AUC to 0.61, and private AUC to 0.6. FPR decreased as opposed to overall accuracy. The best value of AUC was found at dropout values 0.1 and 0.2, which had an overall AUC of 0.62.

- **Cross-validation:** The average training AUC for the TCNN algorithms was 0.98, as another attempt to make classification results similar to that high AUC of training, *k-fold cross-validation* was used with $k = 5$. The test AUC wasn't affected after cross-validation, though cross-validation takes 5X normal training time, and this is very computationally expensive. It was confirmed that the use of cross-validation for deep neural networks with huge data isn't liable.
- **Balancing methods** Data imbalance is a prominent issue facing classification tasks, where the classes are not equally represented. It is extremely common in binary classification problems. In the case of the ecosystem dataset, pre-ictal files (labeled 0) transcend the presence of inter-ictal files (labeled 1) in the dataset. Data balancing methods were employed to attempt to level the percentages. Three approaches are carried out to produce a balanced dataset and present the best solution:
 1. *compute_class_weight*: This method depends on a python class that estimates class weights for unbalanced datasets. It balances the weights of pre-ictal and inter-ictal segments by calculating the label count of each and setting weights of lesser labels at a higher value. Improvements were seen only in the private AUC, patient 2's private AUC, and patient 3's public AUC, while all other AUC values deteriorated. FPR and accuracy increased. To achieve better results, another method was implemented.
 2. *Synthetic Minority Oversampling Technique (SMOTE)* is a popular method for over-sampling. Over-sampling is when the minority class is over-sampled by generating "synthetic" examples rather than by over-sampling with replacement. By operating in "feature space" rather than "data space", new instances are created [52]. In applying SMOTE, both overall and private AUC increased, and so did patient 2's private AUC. Otherwise, the remaining AUC values decreased, some noticeably such as patient 1's private and public values: from 0.70 to 0.34 and from 0.75 to 0.41 respectively. Overall accuracy and sensitivity increased, with a sensitivity value of 74%, however, the AUC is the governing metric here to account for the data imbalance problem.
 3. *Div2effect* a novel technique called "Div2effect". It is described in the data pre-processing section but to reiterate: training and testing data were divided into sets of N number of files where each set was trained

than the trained model saved, the next training set was inputted into the trained, saved model and this process repeated until all the sets were all trained. The sets were chosen in such a way that each set has both preictal and interictal files. To guarantee that each training iteration will have both types of files to learn from and thus offer a balanced training. The last trained model is now trained on all sets. The method is implemented using the checkpoint class in Keras to save intermediate models. This approach led to an increase in all general AUC values, the overall AUC rising to 0.61. FPR decreased from 0.85 to 0.22 and overall accuracy rose to 74% from 21%.

Finally, the "Div2effect" approach was used alongside hyperparameter tuned parameters, as it showed the best results in comparison to the other two approaches.

- **Optimizer** Optimization is the technique used to speed up a model's training time, thus optimizing training. Two optimizers were examined to choose one with better performance. Stochastic gradient descent (SGD) and adaptive moment estimation (ADAM) optimizers were tested, and the ADAM optimizer showed a better performance. Overall AUC increased to 0.68 and the patient-specific AUC of each patient also increased, overall sensitivity increased to 70%.

4.2 Comparison to other DL algorithms

To further evaluate the performance of the enhanced TCNN method, its performance is compared to other DL algorithms that are well known within seizure classification boundaries. The neural networks' structure details are described in the previous Sect. 3. Moreover, we also compare the performance of these DL methods to a deep convolutional network structure that is known to generalize well on raw EEG signals [53] and EEGNet [54] by training those models on the available seizure data. In addition, state-of-the-art algorithms [55, 56] that were tested on the kaggle dataset are compared with the obtained results. The results of overall, public, and private AUC, as well as FPR, accuracy and sensitivity are summarized in Table 6.

The TCNN model was originally implemented to classify motor imagery data [49], alas in this work it has been generalized to classify EEG seizure signals. With only a linear growth in the number of parameters, in contrast to traditional CNNs' linear expansions; it is also able to exponentially enlarge its receptive field size. Another important advantage is its lack of unstable gradient issues- particularly while dealing with long input sequences like those in

Table 6 Comparison between our current model and other DL models

Model	Overall AUC	Public AUC	Private AUC	Overall FPR	Overall accuracy	Overall sensitivity
Proposed method	0.68	0.69	0.67	0.41	60%	70%
EEGNet [54]	0.60	0.60	0.58	0.02	80%	20%
DeepConvNet [53]	0.61	0.66	0.59	0.3	65%	55%
CNN [47]	0.59	0.63	0.60	0.76	28%	78%
LSTM [47]	0.54	0.54	0.52	0.08	85%	26%
Image to CNN [55]	–	0.84	–	–	–	–
Semi-dilated CNN [55]	–	0.88	–	–	–	–
Two-layer LSTM [56]	–	0.92	–	–	–	–

All models were of a patient-specific approach

The best result for each metric are given in bold

this scenario. These advantages make TCNNs superior to CNNs and LSTMs when dealing with time-series data such as EEG signals. The reason for these outstanding results are hyperparameter tuning and data balancing. The ideal hyperparameter values are as follows: epochs = 100, batch size = 64, learning rate = 0.0001, optimizer: ADAM, dropout = 0.1–0.2, and finally div2effect for data balancing.

Private AUC is the most liable metric to evaluate the performance of a seizure prediction system since it is calculated on unseen data, this supports the selected algorithm to be used in the clinical field. The results show that our TCNN model has the best overall AUC of 0.68. Compared with models that process raw input data such as EEGNet, ConvNet, CNN, and LSTM, the model shows the best AUC and follows LSTM in sensitivity (70%). The work [55, 56] uses scalogram, spectrogram, time, and frequency domain features to be fed into DL algorithms. The results show promising public AUC, with the highest being 0.92, when using time and frequency domain features with a two-layer LSTM network. However, a thorough comparison won't be applicable as overall, private AUC, FPR, and overall sensitivity weren't calculated in these papers. The advantage of using models that can predict seizure states on raw-EEG data without the need for any pre-processing technique lies in the ability to use these algorithms on any data and still obtain high classification results.

To compare the computational efficacy of the different deep learning models, the number of parameters of each model is calculated as shown in Fig. 6. The number of parameters of a DL model is weights that are learned during the training process and contribute to the models' classification accuracy.

A model which has the highest performance and least number of parameters would cause usage of fewer resources and accurate prediction would be the best performing model. The TCNN model used shows the least number of parameters, 4137. Following is the LSTM model with 39881 parameters,

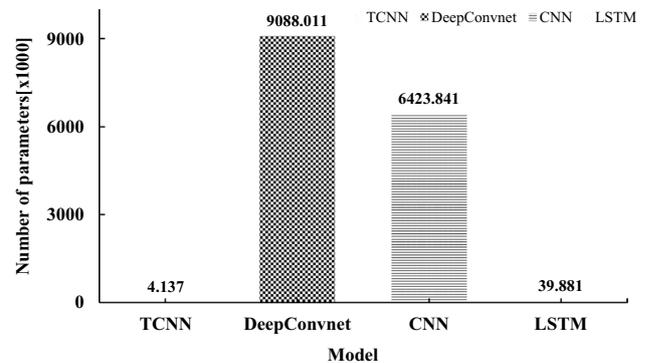


Fig. 6 Number of parameters for different deep learning models

the DeepConvNet and CNN models had the highest parameter values, although not contributing to better classification accuracy. The TCNN model shows the highest classification scores and lowest number of parameters used, along with other DL models used.

4.3 Patient-specific and general classification

The performance of the TCNN model proposed in this work is of a patient-specific approach, meaning training was run for each patient separately. This is the widely used approach in seizure prediction, as it yields better results. To address the challenge of general prediction, we retrain our model on all patients to achieve a general seizure classification TCNN model. Results are shown in Table 7, for the TCNN model the AUC was significantly dropped. Private AUC is used to select the best performing algorithm, as it's calculated using the epilepsy-ecosystem platform on completely held-out data, which makes the model applicable for medical use.

In this section, the TCNN model is compared with our previous attempts of applying ML classification on the epilepsy ecosystem platform. The work in [57] performed pre-processing by applying Butterworth filter and feature

Table 7 General classification comparison

Model	Total AUC	Private AUC	Public AUC	Sensitivity
Proposed Method	0.55	0.54	0.57	70%
SVM	0.83	0.75	0.73	–
RBT	0.75	0.61	0.62	–

AUC= area under curve; SVM= support vector machine; RBT = random boosted trees ensemble

The best result for each metric are given in bold

Table 8 Patient-specific approach comparison

Patient	Proposed method AUC	ML AUC
1	0.75	0.65
2	0.72	0.75
3	0.70	0.75
Average AUC	0.73	0.72

AUC= area under curve; ML= machine learning

The best result for each metric are given in bold

reduction using principal component analysis (PCA). Models used were SVM and ANN respectively. Although, the algorithm has high accuracy of 97% for SVM and 92% for ANN, it was implemented on public dataset only. The algorithm took more than 3 days to run feature extraction and thus deemed computationally inefficient, AUC results weren't presented to be able to compare results with this work. Nevertheless, AUC is more descriptive to ensure capability of successfully classifying preictal and interictal seizure segments. In our paper [17], the general approach method had a successful performance on AUC for both RBT and SVM classifiers, features extracted were frequency power of bands, mean, standard deviation, kurtosis and correlation between all channels in time and frequency domain. The algorithm scored a private AUC of 0.75, a total AUC of 0.82 using SVM. RBT model had total AUC of 0.75, private AUC of 0.61. The ML general model has a preceding position in private AUC using SVM model than any other approach. The model details are described in Sect. 3.

For the patient-specific approach, we also compare the TCNN algorithm performance with SVM and RBT classifiers. Results are presented in Table 8 showing the best algorithm for predicting patient 1 was the current TCNN approach, and for patient 2 and patient 3 SVM and TCNN have comparable AUC values. The mentioned results show that the enhanced TCNN model can compete with SVM and RBT classifiers, showing higher AUC for some patients and almost equal for other patients. The average AUC for the TCNN model is higher than that of the ML model, with a value of 0.73. This highlights the advantage

of using TCNNs on raw EEG data, as the method proposed in this work doesn't require hand-crafted features extraction. However, the SVM algorithm achieved the best AUC results for the general seizure classification.

5 Conclusion and future work

In this work, comprehensive studies were applied to sequential, continuous EEG data. This work aims to enhance the quality of life of epileptic individuals, who comprise a seventh of the world's population. The EEG data signals are pre-sampled to a window size of 75 s and various DL and ML algorithms are tested to develop the best classification algorithm.

The deep learning techniques applied to the data were Convolutional Neural Networks (CNNs), Temporal Convolutional Neural Networks (TCNNs), and Long Short Term Memory (LSTM). According to the results, and out of the aforementioned DL architectures our current method outperformed all others, with a total AUC value of 0.68, an FPR of 0.41, 70% sensitivity and overall accuracy of 60%.

Finally, patient-specific and general prediction approaches for seizure classification are compared. To qualify the total efficacy of these two approaches, we compare them to SVM and RBT classifiers that showed high accuracy in classifying seizures on this data. For the general classification approach, SVM still had the best performance with the highest AUC of 0.75. However, for the patient-specific approach, the TCNN model had the highest AUC of 0.75 for patient 1 and almost equal AUC for patients 2 and 3. The average AUC for the patient-specific approach was higher using the proposed TCNN model, reaching an AUC of 0.72, proving its competency in the field of seizure prediction. To conclude, the results showed TCNN's success in predicting EEG seizure data over other DL algorithms. Also, TCNN performs well as a patient-specific classifier, however SVM classifier has the best performance in the general classification.

Future work includes, hardware implementation of the seizure classification model on an Altera Field Programmable Gate Array (FPGA). Enhancement of the model in terms of computational efficiency and complexity will be required to facilitate the process of successful hardware design. Furthermore, to operate at reduced voltages when using FPGA, self-supervised learning (SSL) and Algorithm based self-tolerance (BFT) can be utilized.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Milligan, T. A. (2021). Epilepsy: A clinical overview. *The American Journal of Medicine*, 134(7), 840–847.
2. Kirch, W. (ed.) (2008). *World Health Organization World Health Organization (WHO)* (pp. 1477–1477). Springer, Dordrecht. https://doi.org/10.1007/978-1-4020-5614-7_3809.
3. Haddad, T., Ben-Hamida, N., Aouini, S. & Rezgui, J. (2020). Seizure prediction with a single i EEG electrode using non-linear techniques. In *2020 International symposium on networks, computers and communications (ISNCC)* (1–6). IEEE.
4. Stacey, W. C., & Litt, B. (2008). Technology insight: Neuroengineering and epilepsy-designing devices for seizure control. *Nature Clinical Practice Neurology*, 4(4), 190–201.
5. Vetkas, A., et al. (2022). Deep brain stimulation targets in epilepsy: Systematic review and meta-analysis of anterior and centromedian thalamic nuclei and hippocampus. *Epilepsia*.
6. Ismail, O. I., et al. (2022). The teratogenic effect of pregabalin on heart, liver and kidney in rats: A light microscopic, electron microscopic and immunohistochemical study. *BMC Pharmacology and Toxicology*, 23(1), 1–19.
7. Parekh, K., Kravets, H. D., & Spiegel, R. (2022). Special considerations in the management of women with epilepsy in reproductive years. *Journal of Personalized Medicine*, 12(1), 88.
8. Klatt, J., et al. (2012). The epilepsiae database: An extensive electroencephalography database of epilepsy patients. *Epilepsia*, 53(9), 1669–1676.
9. Kini, L. G., Davis, K. A., & Wagenaar, J. B. (2016). Data integration: Combined imaging and electrophysiology data in the cloud. *Neuroimage*, 124, 1175–1181.
10. Shoeb, A. H. (2009). *Application of machine learning to epileptic seizure onset detection and treatment*. Ph.D. thesis, Massachusetts Institute of Technology .
11. Kuhlmann, L., et al. (2018). Epilepsyecosystem.org: Crowdsourcing reproducible seizure prediction with long-term human intracranial EEG. *Brain*, 141(9), 2619–2630.
12. Anuragi, A., Sisodia, D. S., & Pachori, R. B. (2022). Epileptic-seizure classification using phase-space representation of fbse-ewt based eeg sub-band signals and ensemble learners. *Biomedical Signal Processing and Control*, 71, 103138.
13. Cleatus, T. S. & Thungamani, M. (2022). Epileptic seizure detection using spectral transformation and convolutional neural networks. *Journal of The Institution of Engineers (India): Series B*, 1–11.
14. Thakur, M., et al. (2022). Epileptic seizure detection using deep bidirectional long short-term memory network. *Sentimental Analysis and Deep Learning*, 893, 1.
15. Abdelhameed, A., & Bayoumi, M. (2021). A deep learning approach for automatic seizure detection in children with epilepsy. *Frontiers in Computational Neuroscience*, 15, 29.
16. Gadhouri, K., Lina, J.-M., Mormann, F., & Gotman, J. (2016). Seizure prediction for therapeutic devices: A review. *Journal of neuroscience methods*, 260, 270–282.
17. Massoud, Y. M., Kuhlmann, L. & Abd El Ghany, M. A. (2021). Comparison of patient specific and general classification of epileptic seizure prediction. In *International Conference on Microelectronics (ICM)* (pp. 116–119). IEEE.
18. Shoeibi, A., et al. (2021). Epileptic seizures detection using deep learning techniques: A review. *International Journal of Environmental Research and Public Health*, 18(11), 5780.
19. Bansal, D. & Sharma, R. (2017). EEG signal artifact removal using ORICA algorithm. *2017 International conference on trends in electronics and informatics (ICEI)* (pp. 758–761). IEEE.
20. Mammone, N. (2018). Preprocessing the EEG of Alzheimer's patients to automatically remove artifacts. *Multidisciplinary Approaches to Neural Computing*, 279, 1.
21. Harpale, V., & Bairagi, V. (2019). FPGA based architecture implementation for epileptic seizure detection using one way ANOVA and genetic algorithm. *Biomedical and Pharmacology Journal*, 12(3), 1543–1553.
22. Williamson, J. R., Bliss, D. W., Browne, D. W., & Narayanan, J. T. (2012). Seizure prediction using EEG spatiotemporal correlation structure. *Epilepsy & behavior*, 25(2), 230–238.
23. Park, Y., Luo, L., Parhi, K. K., & Netoff, T. (2011). Seizure prediction with spectral power of EEG using cost-sensitive support vector machines. *Epilepsia*, 52(10), 1761–1770.
24. Chaurasiya, R., Londhe, N., & Ghosh, S. (2015). Statistical wavelet features, PCA, and SVM based approach for EEG signals classification. *International Journal of Electrical, Computer, Electronics and Communication Engineering*, 9(2), 182–186.
25. Rahman, M. A., Hossain, M., Khanam, F., Alam, M. K., & Ahmad, M. (2019). Four-class motor imagery EEG signal classification using PCA, wavelet, and two-stage neural network. *International Journal of Advanced Computer Science and Applications*, 10(5), 1.
26. Zhang, J.-H., et al. (2018). Personalized prediction model for seizure-free epilepsy with levetiracetam therapy: A retrospective data analysis using support vector machine. *British Journal of Clinical Pharmacology*, 84(11), 2615–2624.
27. Gao, Y. et al. (2022). Abnormal degree centrality as a potential imaging biomarker for right temporal lobe epilepsy: A resting-state FMRI study and support vector machine analysis. *Neuroscience*.
28. Chisci, L., et al. (2010). Real-time epileptic seizure prediction using AR models and support vector machines. *IEEE Transactions on Biomedical Engineering*, 57(5), 1124–1132.
29. Basri, A., & Arif, M. (2021). Classification of seizure types using random forest classifier. *Advances in Science and Technology Research Journal*, 15(3), 167–178.
30. Messaoud, R. B. & Chavez, M. (2021). Random forest classifier for EEG-based seizure prediction. arXiv preprint [arXiv:2106.04510](https://arxiv.org/abs/2106.04510).
31. Tanveer, M. A. & Salman, A. (2019). Epileptic seizure classification using gradient tree boosting classifier.
32. Kumari, N., Anwar, S., & Bhattacharjee, V. (2022). A combi-national deep learning approach to visually evoked EEG-based image classification. *Fundamentals and Methods of Machine and Deep Learning: Algorithms, Tools and Applications*, 1, 259–276.
33. Jana, G. C., Sharma, R., & Agrawal, A. (2020). A 1d-CNN-spectrogram based approach for seizure detection from EEG signal. *Procedia Computer Science*, 167, 403–412.

34. Rout, S. K., Sahani, M., Dora, C., Biswal, P. K., & Biswal, B. (2022). An efficient epileptic seizure classification system using empirical wavelet transform and multi-fuse reduced deep convolutional neural network with digital implementation. *Biomedical Signal Processing and Control*, 72, 103281.
35. Kaur, K. & Kaur, P. (2022). Sabdm: A self-attention based bidirectional-rnn deep model for requirements classification. *Journal of Software: Evolution and Process* e2430.
36. Ma, H. *et al.* (2022). Knowledge distillation-based performance transferring for lstm-rnn model acceleration. *Signal, Image and Video Processing* 1–8.
37. Vennerød, C. B., Kjærran, A. & Bugge, E. S. (2021). Long short-term memory rnn. arXiv preprint [arXiv:2105.06756](https://arxiv.org/abs/2105.06756).
38. Nosouhian, S., Nosouhian, F. & Khoshouei, A. K. (2021). A review of recurrent neural network architecture for sequence learning: Comparison between LSTM and GRU.
39. Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, 31(7), 1235–1270.
40. Hu, X., *et al.* (2020). Scalp EEG classification using deep BI-LSTM network for seizure detection. *Computers in Biology and Medicine*, 124, 103919.
41. Ozcan, A. R., & Erturk, S. (2019). Seizure prediction in scalp EEG using 3d convolutional neural networks with an image-based approach. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(11), 2284–2293.
42. Hussein, R., Lee, S., Ward, R., & McKeown, M. J. (2021). Semi-dilated convolutional neural networks for epileptic seizure prediction. *Neural Networks*, 139, 212–222.
43. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1.
44. Zhang, K., Guo, Y., Wang, X., Yuan, J., & Ding, Q. (2019). Multiple feature reweight densenet for image classification. *IEEE Access*, 7, 9872–9880.
45. Khalil, K., Eldash, O., Kumar, A., & Bayoumi, M. (2020). Machine learning-based approach for hardware faults prediction. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(11), 3880–3892.
46. Chen, Y.-H., *et al.* (2022). A vlsi chip for the abnormal heart beat detection using convolutional neural network. *Sensors*, 22(3), 796.
47. Schirmer, R. T., *et al.* (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, 38(11), 5391–5420.
48. Ingolfsson, T. M. *et al.* (2020). Eeg-tenet: An accurate temporal convolutional network for embedded motor-imagery brain-machine interfaces. arXiv e-prints [arXiv–2006](https://arxiv.org/abs/2006).
49. Bai, S., Kolter, J. Z. & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint [arXiv:1803.01271](https://arxiv.org/abs/1803.01271).
50. Hu, Z., Zhang, J., & Ge, Y. (2021). Handling vanishing gradient problem using artificial derivative. *IEEE Access*, 9, 22371–22377.
51. Philipp, G., Song, D. & Carbonell, J. G. (2017). The exploding gradient problem demystified-definition, prevalence, impact, origin, tradeoffs, and solutions. arXiv preprint [arXiv:1712.05577](https://arxiv.org/abs/1712.05577).
52. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
53. Altan, G., Yayık, A., & Kutlu, Y. (2021). Deep learning with convnet predicts imagery tasks through EEG. *Neural Processing Letters*, 53(4), 2917–2932.
54. Lawhern, V. J., *et al.* (2018). Eegnet: A compact convolutional neural network for eeg-based brain-computer interfaces. *Journal of Neural Engineering*, 15(5), 056013.
55. Hussein, R. *et al.* (2019). Human intracranial eeg quantitative analysis and automatic feature learning for epileptic seizure prediction. arXiv preprint [arXiv:1904.03603](https://arxiv.org/abs/1904.03603).
56. Varnosfaderani, S. M. *et al.* (2021). A two-layer lstm deep learning model for epileptic seizure prediction. *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (pp. 1–4). IEEE.
57. Kamel, E. M., Massoud, Y. M., El Ghany, M. A. & Salem, M. A.-M. (2021). Eeg classification for seizure prediction using svm vs deep ann. *2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)* (pp. 389–395). IEEE.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Yasmin M. Massoud PhD, is a LecturerAssistant in Information Engineering and Technology, GermanUniversity in Cairo. Her research areas are biomedicalengineering, data science, machine learning, neuroscience, hardwareimplementation of biomedical systems.



Mennatallah Abdelzاهر is a bachelorstudent, in Networks Engineering, German University in Cairo. Herresearch interests are seizure prediction, biomedical engineeringand machine learning.



Dr. Levin Kuhlmann Ph.D., is a Senior Lecturerin Data Science and AI and Deputy Director of the Master of AIprogram at Monash University. His research areas include datascience, machine learning, signal processing, control theory andcomputational neuroscience applications to digital health, neuralengineering and neuroimaging.



Dr. Mohamed A. Abd El Ghany received the B.S. degree (Hons.) in electronics and communications engineering and the master's degree in electronics engineering from Cairo University, Cairo, Egypt, in 2000 and 2006, respectively, and the Ph.D. degree in high-performance VLSI/IC design from German University in Cairo (GUC), New Cairo, Egypt, in 2010. From 2003 to 2006, he was with the National Space Agency of Ukraine's EGYPT-SAT-1 Project. From 2008

to 2010, he was an International Scholar with the Electrical Engineering Department, Ohio State University, Columbus, OH, USA. From 2012

to 2014, he was awarded the Alexander von Humboldt Foundation Post-doctoral Fellowship at Technische Universität (TU) Darmstadt, Darmstadt, Germany. He is currently an Associate Professor with GUC. He is a Project Manager of four international projects between TU Darmstadt, Ruhr-University Bochum, and GUC. He is the author of about 90 articles, two book chapters, and two books in the fields of high throughput and low-power VLSI/IC design and NoC/SoC. His research interests include low-power design, embedded system design, network on chip design and related circuit level issues in high-performance VLSI circuits, clock distribution network design, digital ASIC design, and SoC/NoC design and verification. He is a reviewer and a program committee member of many IEEE international journals and conferences.