MAULANA AZAD
NATIONAL INSTITUTE OF TECHNOLOGY
BHOPAL INDIA, 462003

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Epilepsy Seizure Classification Through EEG Signals Using 1D CNN

Minor Project Report

Semester VI

Submitted by:

Satyam Singh	191112444
Mohammad Areeb	191112478
Faisal Jamil	191112476
Rishabh Pandey	191112461

Under the Guidance of

Dr. Mitul Kumar Ahirwal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Session: 2021-2022

MAULANA AZAD
NATIONAL INSTITUTE OF TECHNOLOGY
BHOPAL INDIA, 462003

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project report carried out on "**Epilepsy Seizure Classification Through EEG Signals Using 1D CNN**" by the 3rd year students:

Satyam Singh	191112444
Mohammad Areeb	191112478
Faisal Jamil	191112476
Rishabh Pandey	191112461

Have successfully completed their project in partial fulfillment of their Degree in Bachelor of Technology in Computer Science and Engineering.

Dr. Mitul Kumar Ahirwal
(Minor Project Mentor)

DECLARATION

We, hereby declare that the following report which is being presented in the Minor Project Documentation Entitled "**Implementation of One Dimensional Convolutional Neural Networks (CNN) for EEG Classification to detect Epilepsy Seizure**" is authentic documentation of our own original work and to the best of our knowledge. The following project and its report, in part or whole, has not been presented or submitted by us for any purpose in any other institute or organization. Any contribution made to the research by others, with whom we have worked at Maulana Azad National Institute of Technology, Bhopal or elsewhere, is explicitly acknowledged in the report.

Satyam Singh	191112444
Mohammad Areeb	191112478
Faisal Jamil	191112476
Rishabh Pandey	191112461

ACKNOWLEDGEMENT

With due respect, we express our deep sense of gratitude to our respected guide and coordinator Dr. Mitul Kumar Ahirwal, for his valuable help and guidance. We are thankful for the encouragement that he has given us in completing this project successfully.

It is imperative for us to mention the fact that the report of a minor project could not have been accomplished without the periodic suggestions and advice of our project guide Dr. Mitul Kumar Ahirwal and project coordinators Dr. Dhirendra Pratap Singh and Dr. Jay Trilok Choudhary.

We are also grateful to our respected director Dr. N. S. Raghuwanshi for permitting us to utilize all the necessary facilities of the college.

We are also thankful to all the other faculty, staff members and laboratory attendants of our department for their kind cooperation and help. Last but certainly not the least; we would like to express our deep appreciation towards our family members and batch mates for providing the much needed support and encouragement.

ABSTRACT

Our brain cells communicate through electrical impulses but at certain times temporary change in the electrical functioning of the brain may lead to sudden alteration of behavior leading to a seizure. EEG signals can be used to detect this unusual activity. Electroencephalogram (EEG) is widely used to monitor electrical brain activity. In EEG, electrodes are attached to the scalp and are used to record the electrical activity of the brain. It is very time consuming and laborious to examine and inspect these signals manually. Thus, machine learning algorithms can be used to improve the accuracy of detection. Nowadays, deep learning techniques have been used in the medical field for accurate diagnosis of these unusual brain activities. In this study, we have proposed a one dimensional convolutional neural network for detection of epileptic seizure based on EEG classification. In this study, we have used the EEG signals from a single channel obtained from UCI Machine Learning Repository EEG dataset to develop the 1D CNN model. The implemented network is able to classify between seizures (Abnormal) and no seizure (Normal) condition with about 98% accuracy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

Epilepsy is a rare chronic neurological disorder affecting millions of people in the world. Over 50 million people have had an epileptic attack, according to the World Health Organization (WHO). [1]. It is life-threatening for humans as it can cause spontaneous seizures, these attacks disrupting the part or whole body is caused by unusual electrical activity of the brain. Epileptic patients may lose consciousness, become confused, or experience sudden convulsive movements that can even cause very serious physical injury. Moreover, it leads to emotional distress and low self esteem for epileptic patients [1-2]. That's why it is of utmost crucial to diagnose this crucial health problem. Thus early detection of the occurrences of epileptic seizures can improve the quality of life of epileptic patients.

Electroencephalography (EEG) is a method to measure voltage changes between electrodes on a subject's scalp produced by ionic changes in the brain. It provides temporal and spatial information about the subject's brain. EEG signals are commonly used for the identification of epilepsy [1-2]. Commonly, identification of EEG signals is done manually by experts and doctors. This traditional approach, on the other hand, takes a long time and frequently results in mistakes. Fig-1 shows the comparison between normal and abnormal EEG signals [3].

In this study we have tried to classify Epileptic Seizure by training the dataset on two different architectures - CNN and CNN + LSTM. We have done both binary class classification as well as multi class classification and observed the accuracy of the models in each case.

Fig-1: Comparison of Normal (A) and Abnormal (B) EEG signal [3]

2. Literature review and survey

Epilepsy is a rare neurological disorder that is estimated to affect 10% of the global population. Most people who are suffering from epilepsy are unaware of the prevalence of epilepsy as a serious medical condition [1-2]. Unlike many other diseases or disorders, it is usually difficult to detect the possibility of having an epileptic seizure. As a result, the control and management of epilepsy is limited. Many scientific methods have been implemented to accurately detect epileptic seizures from EEG signals including machine learning algorithms and their results have been good so far [3]. Prior to the ascent of deep learning, feature extraction was performed by conventional machine learning algorithms. This restricted their performance to the capacity of those handcrafting the features. But with the development of deep learning, the extraction of features and classifications are completely automated [4]. Research shows how deep learning algorithms have significantly increased the accuracy of detection of epilepsy seizures over the time.

3. Gaps Identified

The accuracy of 1D CNN for EEG classification has always been the main concern and many models have been employed with different methods to increase the accuracy of the model, but there had been no significant improvement in accuracy of models, there could be many reasons behind this - one reason could be the imbalance of the data which generally results in the reduction of accuracy. In our study we have implemented several sampling techniques like SMOTE (Synthetic Minority Oversampling Technique) to balance the number of samples which enhanced our accuracy.

4. Proposed Work and Methodology

4.1 Dataset

The original dataset from the reference consists of 5 different folders, each with 100 files, with each file representing a single subject/person [5]. We have recorded the brain activity for every 23.5 seconds. We have sampled the corresponding time series into 4097 data points. The value of the EEG recording for each data point is taken at different points in time. So we have a total of 500 individuals with each having 4097 data points for 23.5 seconds and that corresponds to a sampling frequency of 173.60 Hz. 4097 data points were divided and shuffled into 23 chunks, with each chunk containing 178 data points for 1 second, and each data point is the value of the EEG recording at a different point in time. After this we have 11500 pieces of information(row) (23 x 500) , each row contains 178 data points recorded for 1 second(column), the last column represents the label y {1,2,3,4,5}[5].

In the dataset the Explanatory variables are X1, X2, …, X178 and the response variable is y in the column 179. y contains the category of the 178-dimensional input vector. Specifically y in {1, 2, 3, 4, 5}:

Class 1: People having Epileptic seizures.

Class 2: Recording the EEG from the area where the tumor was located.

Class 3: Region of tumor identified but EEG taken from the healthy brain area.

Class 4: People are healthy but reading of EEG taken from an eye close state.

Class 5: People are healthy but reading of EEG taken from an eye open state.

The comparison of EEG signals for all five classes is shown in Fig-2.

- Sampling Frequency**:** 173.60Hz
- Number of Classes: 5 [1: Abnormal, 2,3,4,5: Normal]
- Number of Samples in each class: 2300
- Total Number of Samples: 11500

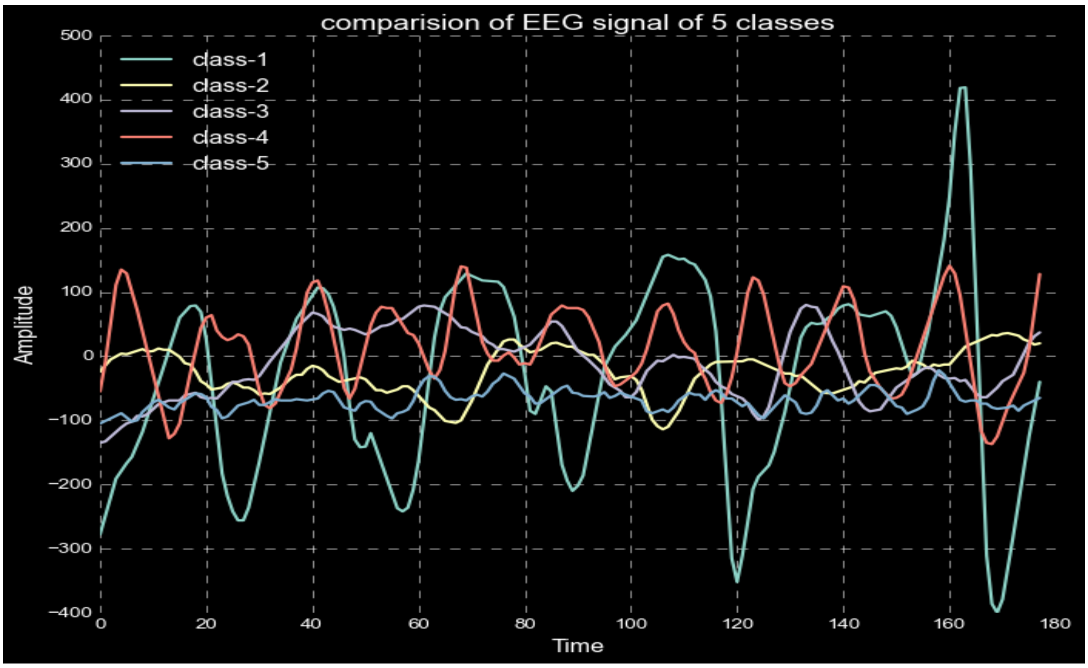

Fig-2: Comparison of EEG Signal of all 5 Classes

4.2 Data modification

We have assigned label 1 for class 1 and label 0 for the rest of the classes(2,3,4,5) as those subjects did not have epileptic seizures. After labeling, we have 9200 samples of label 0 and 2300 samples of label 1.

Number of samples:

- Normal 2300
- Abnormal 9200

Since the data is imbalanced, we balanced the data of both the labels using different sampling techniques. We used the following sampling methods for data balancing.

4.2.1 Undersampling

In this technique we balance the dataset by reducing the number of samples in the majority class.

Number of samples after balancing data:

- Normal 2300
- Abnormal 2300

4.2.2 Oversampling

In this sampling method we balance the dataset by increasing the number of samples in the minority class by just creating copies of already existing samples.

Number of samples after balancing data -

- Normal 9200
- Abnormal 9200

4.2.3 SMOTE

It’s basically an oversampling method which uses k-nearest neighbors for generating new samples in order to balance the data.

Number of samples after balancing data:

- Normal 9200
- Abnormal 9200

4.2.4 ADASYN

ADASYN or Adaptive Synthetic Sampling Approach is a sampling technique which is a generalization of SMOTE.It adjusts the decision constraints adaptively depending on difficult samples.

Number of samples after balancing data -

- Normal 9200
- Abnormal 9200

4.2.5 Random Oversampler

It creates duplicates by randomly selecting samples from minority class for data balancing.

Number of samples after balancing data -

- Normal 9200
- Abnormal 9200

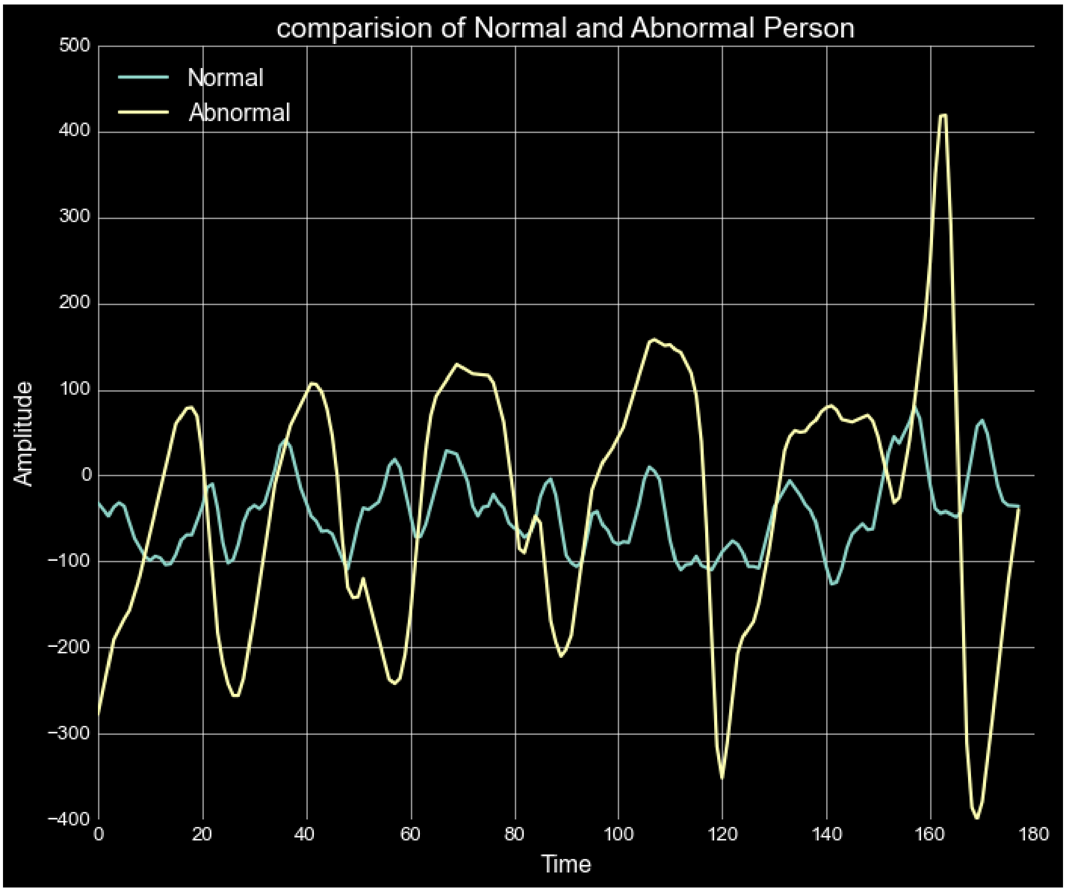

Fig-3: comparison of normal and abnormal EEG signal

4.3 Overview of 1-D CNN

A CNN works to identify patterns within your data which will then be used to form more complex patterns within higher layers [5-6]. When we need to extract relevant features from shorter parts of a larger data set and the position of the feature within the segment isn't critical, a 1D CNN is quite useful.

4.3.1 Convolutional Block

The Convolution block consists of three layers, the first is the 1-D convolutional layer, the second is the batch normalization layer and the third one is the Max pooling layer.[7]

The convolutional layer mainly does feature extraction and pattern detection in the original dataset (input). Similar to classic neural networks, we perform linear operations involving the multiplication of a collection of weights with the input signal represented by metrics. A filter or kernel is a collection of weights[6]. Batch normalization enables each layer of the network to learn in a more independent manner. It normalizes the previous layer's output. It is done in mini-batches instead of the full data set. Basically, it makes the model stable and faster.

The Max pooling layer basically reduces the size of the dataset and thus reduces the cost of computation. We have used Relu as our activation function. The activation function is used to add non-linearity to the neuron's output. It basically tells whether a neuron should be activated or not.

4.3.2 Output Block

The output block is mainly responsible for the classification part. One Flatten layer, two dense layers, and one output layer are included. The data obtained from the convolutional block is flattened by the flatten layer. Relu is used as an activation function in the dense layers, whereas softmax is used in the output layer.

4.3.3 Training

The model learns and gets trained with an adaptation of weights during each epoch. We have used Adam as our optimizer. The optimizers are algorithms or methods used to change the learning parameters such as weights and bias to reduce the losses. We also used binary cross-entropy as our loss function.[8]

4.4 Training, Testing, and Validation

Training and Testing split is kept at a 70:30 ratio. So,

Number of samples in train set: 12880 samples

Number of samples in test set: 5520 samples

We now split the training dataset in half, creating a training and validation dataset with an 80:20 split. so,

Number of samples in train set: 10304 samples

Number of samples in validation set: 2576 samples

4.5 Implementation of 1D-CNN for Binary Classification

For database labels are:

- Class 0 (Normal): [1,0]
- Class 1 (Abnormal): [0,1]

The model consists of three convolution blocks and an output block following it. The first and second convolution block contains a convolution layer with 80 filters each. The kernel size is 4 and the Relu activation function is used. A batch normalization layer with a pool size of three and a MaxPooling Layer with a pool size of three. The third convolution layer has a kernel size of 4 and contains 120 filters. The output block includes a Flatten layer, two Dense layers with 30 and 15 units each and Relu as an activation function, followed by a Dense Layer with 2 units and SoftMax as an activation function, resulting in predicted class labels as the final output. The Adam optimizer is used in conjunction with a binary cross entropy loss function during training. The proposed architecture is shown in Fig-4 and the detailed parameters of the model are listed in Table-1.

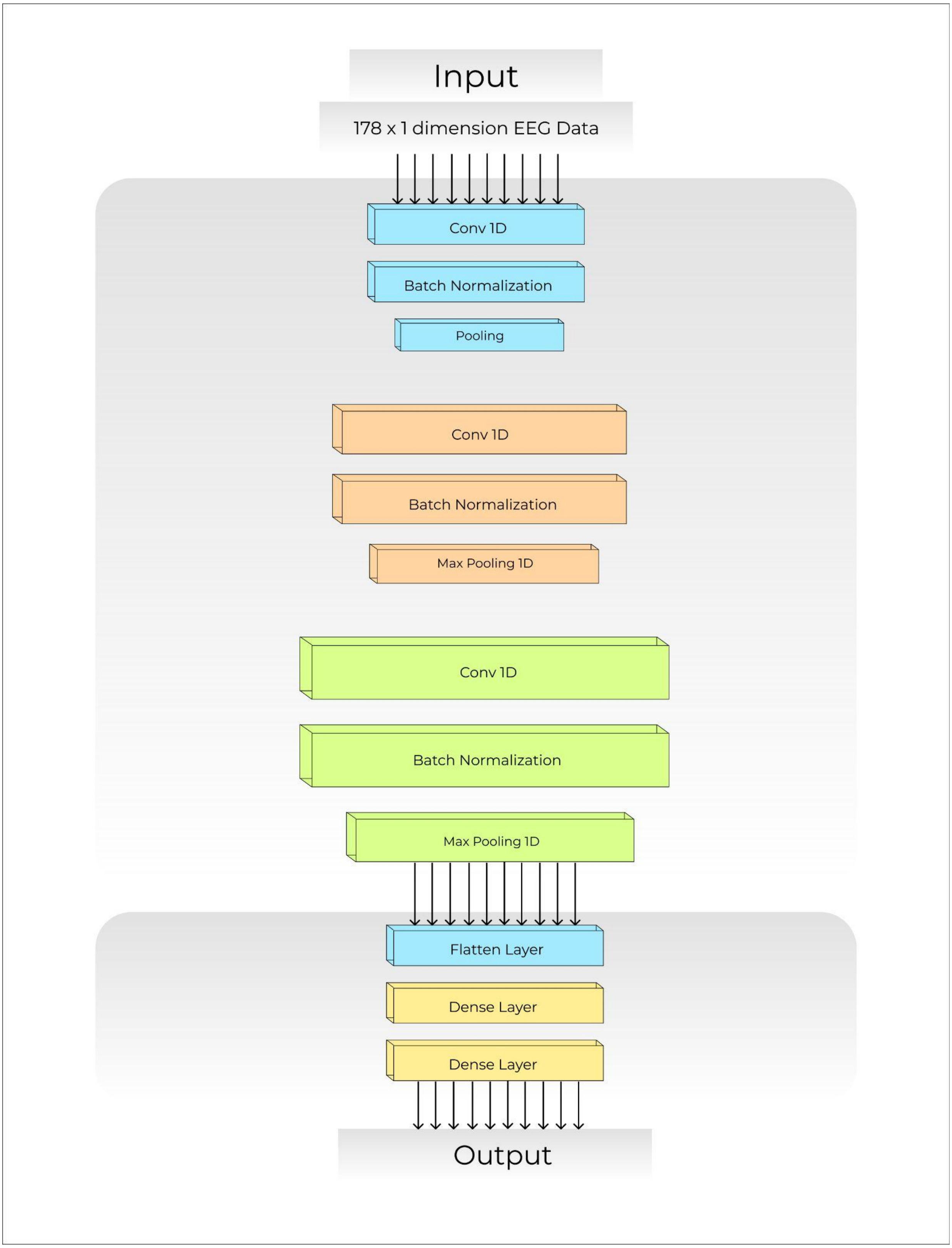

Fig-4: CNN model architecture

Table-1: Detailed parameters of the proposed 1D-CNN model

Type of Layer	Output Shape	Other Parameters of each layer
Conv1D Batch Normalization Max Pooling 1D Activation	(175;80) (175;80) (58;80) (58;80)	Filters - 80, Kernel Size - 4, Pool Size – 3, Activation - Relu
Conv1D Batch Normalization Max Pooling 1D Activation	(55;80) (55;80) (18;80) (18;80)	Filters - 80, Kernel Size - 4, Pool Size – 3, Activation - Relu
Conv1D Batch Normalization Max Pooling 1D Activation	(15;120) (15;120) (5;120) (5;120)	Filters - 120, Kernel Size - 4, Pool Size – 3, Activation - Relu
Flatten Dense Dense Output Layer	(600) (30) (15) (2)	- Units - 30, Activation – Relu Units - 15, Activation – Relu Units - 2, Activation – SoftMax

4.6 Overview of LSTM

RNN or Recurrent Neural Networks are basically neural networks that helps in finding the pattern in sequential data like time series data but it has certain shortcomings like vanishing gradient descent and short term memory.

LSTM stands for long short-term memory networks, It’s basically an extended form of RNN (Recurrent Neural Network) that overcomes the problems posed by RNN and It is capable of learning long-term dependencies, especially in sequence prediction problems.The LSTM has three layers; an input Layer, a single hidden Layer followed by a feedforward output layer.

4.7 Implementation of CNN + LSTM

We know that CNN can be used for feature extraction and LSTM for analysing the patterns across time steps so we clubbed both CNN and LSTM architectures by adding CNN layers on the front end followed by LSTM layers, with a dense layer on the output.Fig-5 describes our proposed architecture.

The model consists of three convolution blocks followed by an LSTM layer and and an output block. The first and second convolution block contains a convolution layer with 16 and 32 filters respectively. The kernel size is 3 and the Relu activation function is used. A batch normalization layer and a MaxPooling Layer having a pool size of 2. The third convolution layer consists of 64 filters and has a kernel size of 3. The fourth layer is an LSTM layer in which we have used tanh activation function. The output block contains a Flatten layer, 2 Dense layers with 30 and 15 units respectively, and Relu as an activation function, followed by a Dense Layer having 2 units with a SoftMax activation function which gives the final output as predicted class labels. For training, the Adam optimizer is used with a binary cross entropy loss function.The proposed architecture is shown in Fig-5 and the detailed parameters of the model are listed in Table-2.

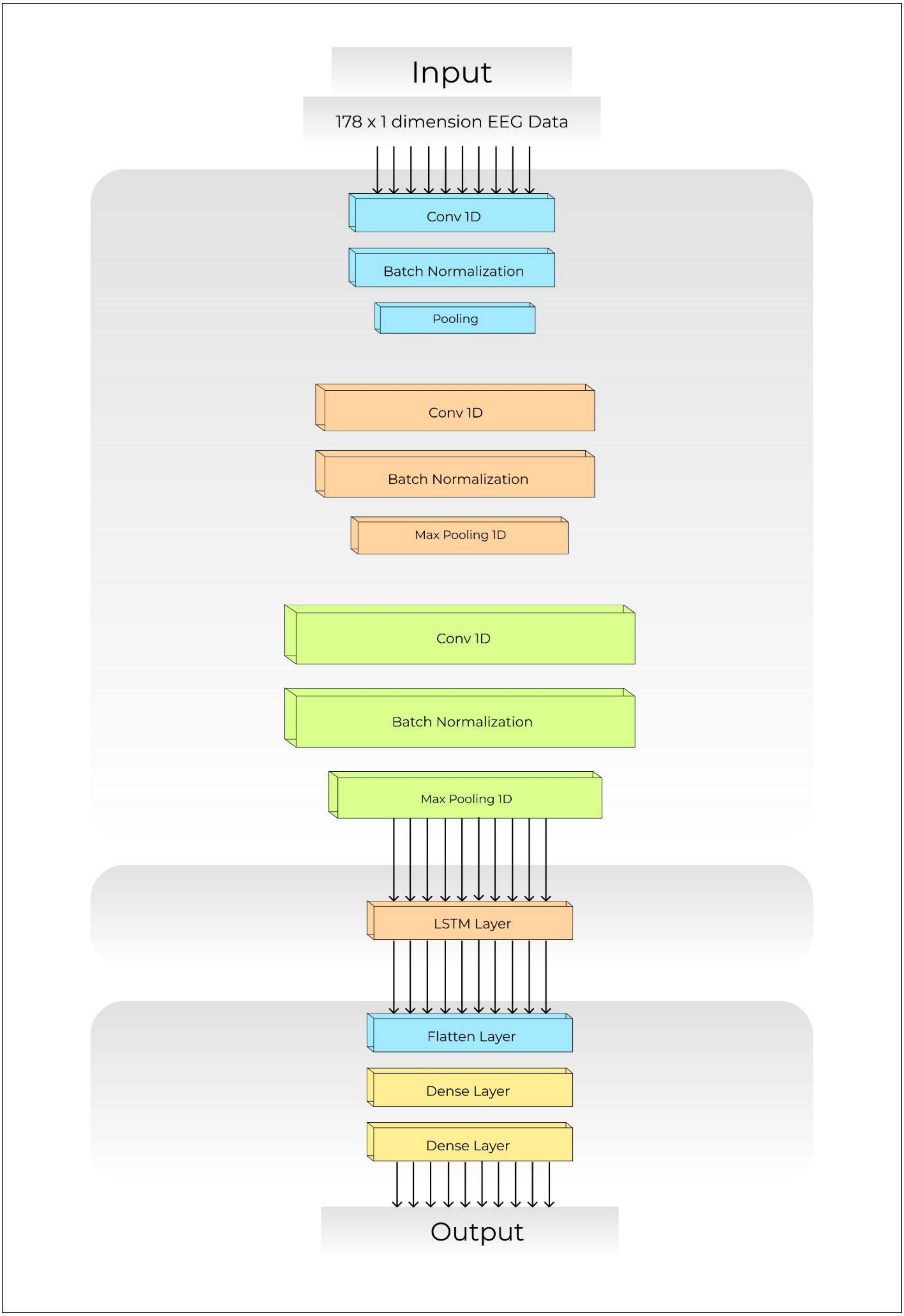

Fig-5: CNN+LSTM model architecture

Table-2: Detailed parameters of the proposed CNN + LSTM model for 5 classes

Type of Layer	**Output Shape**	**Other Parameters of each layer**
Conv1D Batch Normalization Max Pooling 1D Activation	(176;16) (176;16) (88;16) (88;16)	Filters - 16, Kernel Size - 3, Pool Size – 2, Activation - Relu
Conv1D Batch Normalization Max Pooling 1D Activation	(86;32) (86;32) (43;32) (43;32)	Filters - 32, Kernel Size - 3, Pool Size – 2, Activation - Relu
Conv1D Batch Normalization Max Pooling 1D Activation	(41;64) (41;64) (20;64) (20;64)	Filters - 64, Kernel Size - 3, Pool Size – 2, Activation - Relu
LSTM	(50)	Activation - tanh
Flatten Dense Dense Output Layer	(50) (30) (15) (5)	- Units - 30, Activation – Relu Units - 15, Activation – Relu Units - 2, Activation – SoftMax

5. Results and Discussions

Table-3: Results of different models for different number of classes

Classes	Model	Data Balancing Type	Training Accuracy (%)	Testing Accuracy (%)
5	CNN	-	98.52	75.50
	CNN+LSTM	-	98.70	80.75
3	CNN	-	99.57	88.69
	CNN+LSTM	-	99.18	96.05
2	CNN	-	99.81	98.81
	CNN	Undersampling	99.73	97.39
	CNN	Oversampling	99.73	99.52
	CNN	SMOTE	99.92	99.31
	CNN	ADASYN	99.64	98.19
	CNN	Random Oversampler	99.73	99.38
	CNN	SVM SMOTE	99.63	99.29
	CNN	Borderline SMOTE	99.91	99.11

We previously trained a CNN model for 5 classes with a training accuracy of 98.52 and a testing accuracy of 75.50, and then we trained a CNN+LSTM model for the same 5 classes with a training accuracy of 98.70 and a testing accuracy of 80.75.

Amongst these 5 classes, 2 classes comprised of subjects having a brain tumor, the other 2 comprised healthy patients, while the last one had an epileptic seizure, So we aggregated them to make three classes, trained a CNN model for these three classes, and obtained training and testing accuracy of 99.57 and 88.69, respectively. We also trained the CNN+LSTM model, which achieved training and testing accuracy of 99.18 and 96.05 respectively. The detailed results for all models and classes are listed in Table-3. Then we observed the F1 score for these 3 classes and found that for the first two classes where the subjects had no epileptic seizure the F1 score was pretty bad while the F1 score was good for class comprising of subjects having an epileptic seizure, due to this reason our accuracy was hampered so we clubbed the first two classes into one class of subjects having no epileptic seizure. Basically, we performed a binary classification, but due to this our data got imbalanced so we performed several data balancing techniques, and we got the best results from the smote method, in which we got 99.92 as training accuracy and 99.31 as testing accuracy.

After implementation of the 1-D CNN model in our dataset, we got the following training and validation accuracy graph over the epochs shown in Fig-6.

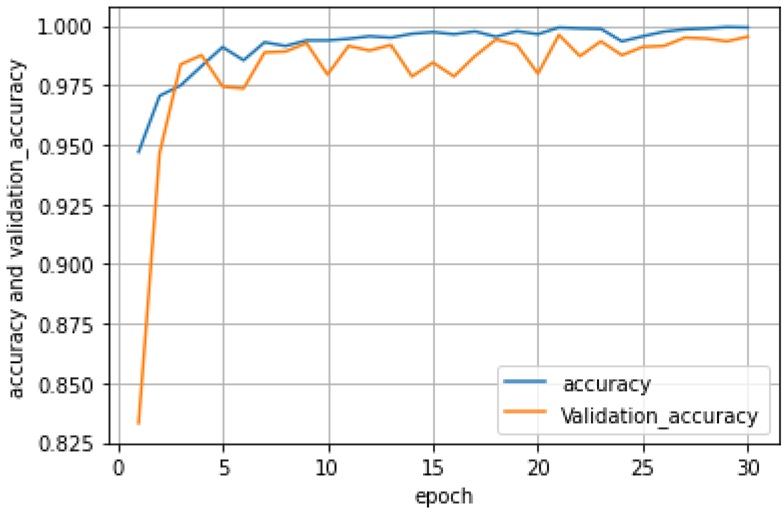

Fig-6: Training and validation accuracy

We also get the following training and validation loss graph over the epochs shown in Fig-7.

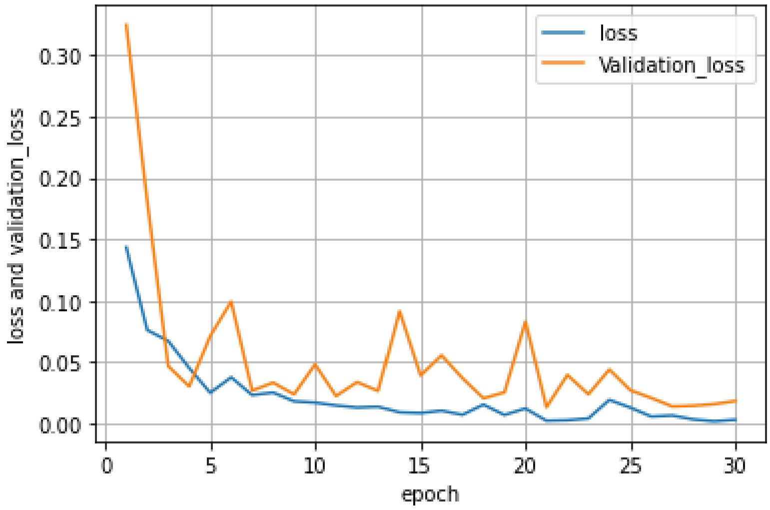

Fig-7: Training and validation loss

Finally at the 15th epoch, we got the following accuracy:

Training accuracy - 99.92 percent

Validation accuracy - 99.53 percent

Testing accuracy - 99.31 percent

```
# Training and validation accuracy
Epoch 15/15
322/322 [==============================] - 8s 24ms/step - loss:
0.0008 - accuracy: 0.9992 - val_loss: 0.0047 - val_accuracy: 0.9953

# Testing accuracy
173/173 [==============================] - 1s 5ms/step - loss:
0.0069 - accuracy: 0.9931
```

Performance of CNN models has been evaluated through several metrics/parameters listed below. For calculation of these parameters, the confusion matrix needs to be defined as listed in Table-4 below.

The six parameters used are as follows :-

- **Accuracy (ACC)**: This is the ratio of correct predictions and total number of predictions,accuracy is obtained by eq.(1),

$$ACC = (TP + TN) / (TP + TN + FP + FN) \tag{1}$$

- **Mis-Classification(MC)**: This is just opposite of ACC, it is simply calculated by eq.(2),

$$MC = 1 - ACC \tag{2}$$

- **Sensitivity or Recall (RC)**: This is the ratio of true positive and total predictions in actual positive, it is simply calculated by eq.(3),

$$RC=TP / (FN+TP) \tag{3}$$

- **Specificity (SP) / (Selectivity or True Negative Rate (TNR))**: This is the ratio of true negative and total predictions in actual negative , it is calculated by eq. (4),

$$SP = TN / (TN + FP) \tag{4}$$

- **Precision (PR) / (Positive Predictive Value(PPV))**: This is the ratio of true positive and total predictions in predicted positive , it is simply calculated by eq. (5)

$$PR = TP / (TP + FP) \tag{5}$$

- **F1 Score**: This is parameter based on previous above parameters, it is calculated as eq. (6),

$$F1=2*((PR * RC) / (PR + PC)) \tag{6}$$

Table-4: Confusion matrix

Binary Classification	**Predicted Negative**	**Predicted Positive**
Actual Negative	True Negative(TN)	False Positive(FP)
Actual Positive	False Negative(FN)	True Positive(TP)

5.1 Confusion matrix

The values of Confusion matrix of our model are listed below in Table-5 and Confusion matrix of our model is shown in Fig-8.

Table-5: Confusion matrix of our model

	Normal(predicted value)	**Abnormal(predicted value)**
Normal (true value)	2686	25
Abnormal(true value)	13	2796

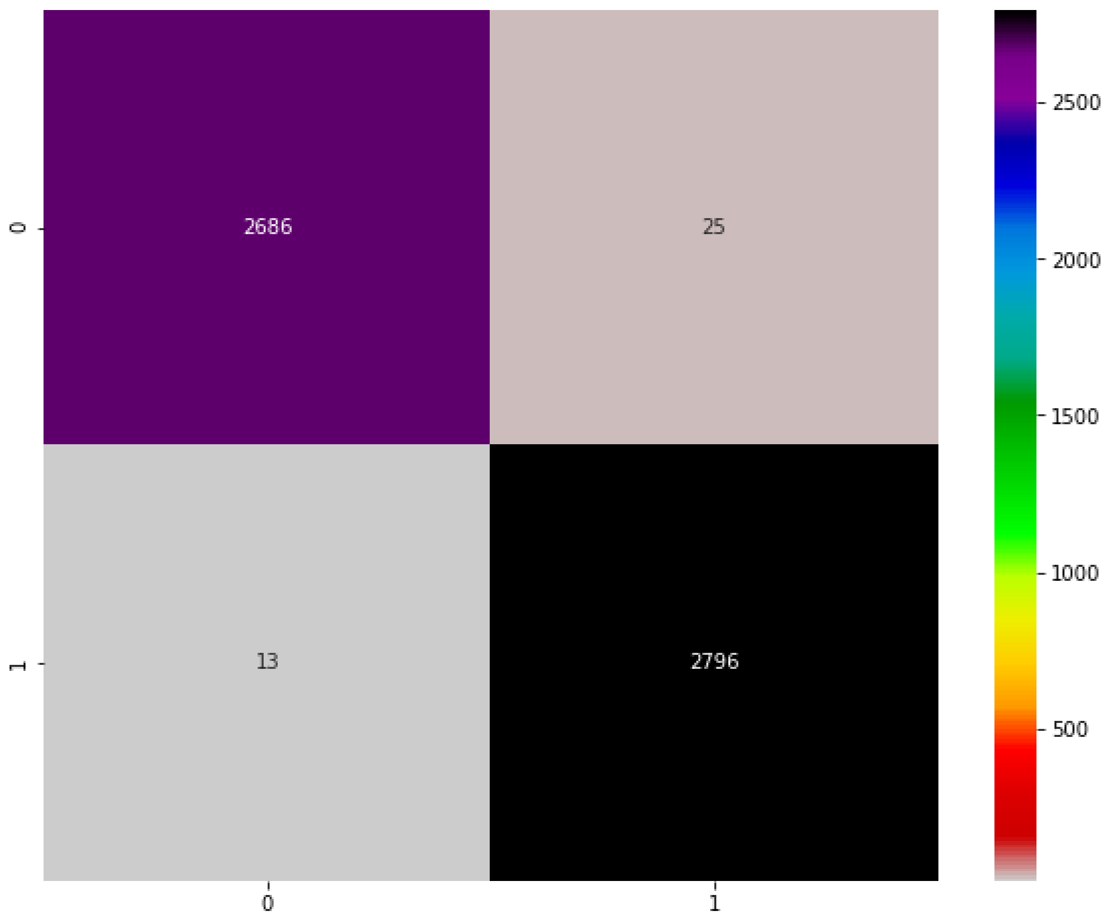

Fig-8: Confusion matrix

5.2 Classification report

The values of Classification report of our model are listed below in Table-5.

Table-6: Classification Report

	precision	**recall**	**f1-score**	**support**
0	1.00	0.99	0.99	2711
1	0.99	1.00	0.99	2809
accuracy	--	--	0.99	5520
macro avg	0.99	0.99	0.99	5520
weighted avg	0.99	0.99	0.99	5520

6. Conclusion

Epilepsy remains incurable and there are measures taken to protect the patient from brain damage or physical injury in case of seizures. Traditional methods used earlier for detecting epilepsy were time-consuming and sometimes inefficient but with the introduction of deep learning algorithms, detection is much easier, more accurate, and faster. Deep learning methods allow the use of large datasets and improved detection accuracy than human counterparts. With deep learning architecture based on 1-D CNN, we got an accuracy of 99.31 percent for detecting epileptic seizures.

7. Reference

1. Abiyev, Rahib, et al. "Identification of epileptic EEG signals using convolutional neural networks." Applied Sciences 10.12 (2020): 4089.
2. Ullah, Ihsan, Muhammad Hussain, and Hatim Aboalsamh. "An automated system for epilepsy detection using EEG brain signals based on a deep learning approach." Expert Systems with Applications 107 (2018): 61-71.

3. Fürbass, Franz. EEG monitoring based on automatic detection of seizures and repetitive discharges. Diss. Wien, 2018.
4. Shoeibi, Afshin, et al. "Epileptic Seizures Detection Using Deep Learning Techniques: A Review." International Journal of Environmental Research and Public Health 18.11 (2021): 5780.
5. Andrzejak, Ralph G., et al. "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state." Physical Review E 64.6 (2001): 061907.
6. Kiranyaz, Serkan, et al. "1D convolutional neural networks and applications: A survey." Mechanical systems and signal processing 151 (2021): 107398.
7. Sarkar, Avik, Ankita Singh, and Rakhi Chakraborty. "A Deep Learning-Based Comparative Study to Track Mental Depression from EEG Data." Neuroscience Informatics (2022): 100039.
8. Khessiba, Souhir, et al. "Innovative deep learning models for EEG-based vigilance detection." Neural Computing and Applications 33.12 (2021): 6921-6937.