International Journal of Scientific Research and Engineering Development-– Volume 4 Issue 5, Sep- Oct 2021

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Exploring Deep Learning Based CNN-LSTM Technique for Fraud Transaction Detection

K Vasudeva Deekshith[1], Dr. Seshachalam D[2]

[1]MTech Student, Department of Electronics and Communication Engineering,
BMS College of Engineering, Bull TempleRoad,Basavanagudi,Bengaluru-560019,Karnataka, India.
vasudevak.lel19@bmsce.ac.in

[2]Professor, M.E., Ph. D., Department of Electronics and Communication Engineering,
BMS College of Engineering, Bull TempleRoad,Basavanagudi,Bengaluru-560019,Karnataka, India.
dschalam@bmsce.ac.in

----------**********----------

Abstract:

With illicit access to web services increasing, online transactions are more common in today's world. So, it's become necessary to detect and avoid them. Diverse approaches of detecting fraud, such as logistic regression, data mining, decision tree, rule-based mining, neural networks and machine learning, have also been employed in the corporate and public sectors. The aim of this paper is to offer a method to detect such fraud transactions during a pandemic which is uncontrollable.This study proposes provides an overview of a deep learning (DL) to monitor and detect fraudulent activity. The effectiveness of leveraging the Kaggle data set to increase the prediction accuracy of fraudulent transactions using a Convolution Neural Network (CNN) and long-term memory recurrent neural network (LSTM RNN) is investigated. The proposed method detects unsatisfactory transactions with 98.6 percent accuracy, 0.985 Precision, 0.97 recall and 0.98 F1 score, 0.01 MSE, AUROC score of 0.986 and 0.973 AUPRC.

Keywords **—DL-Deep Learning, CNN-Convolution Neural Network, LSTM-Long-Short Term Memory Neural Network, Fraud Transaction Detection, AUROC, AUPRC.**

----------**********----------

I. INTRODUCTION

Because of the increased volume of online transactions and the lack of robust cybersecurity safeguards, fraudsters have more room to operate. Technology may be found practically anywhere. Shopping, supermarket orders, bill payments, and other essential tasks have all moved to the digital realm. As a result, the Indian economy is fast shifting from cash to digital transactions, exposing the country to cyber-attacks. Banks have moved to the integration of digital platforms to support digital payments in order to make transactions more flexible for clients, which has resulted in an increase in the frequency of cyber scams in the banking sector.

The prevalence of online fraud is at an all-time high. Fraudsters have never had it easier to take advantage of those who use the internet in our increasingly digital environment. To make matters worse, the dark web has passwords, credit card numbers, and other critical information for many people. Every day, accounts are taken over and new fraudulent lines of credit are issued as a result of this. Many go totally unnoticed. Online transactions frauds are more popularin present world. Credit card details, PINs, and security codes can all be easily stolen and used to commit fraud. Merchants and consumers may suffer significant financial losses as a result of this. This must be identified and need to be voided.

Machine Learning (ML) is a branch of Artificial Intelligence that tries to detect patterns in data and extract specific information. ML is primarily used to detect online transaction fraud. It allows for the detection of potentially fraudulent online money transactions. Many techniques, such as Decision Trees, logistic regression, Random Forests, Artificial Neural Networks, and others, have been applied. However, machine learning has limitations, such as the lengthy time it takes to train a classifier when the data set is large. Furthermore, these algorithms fail to recognise skewed patterns in the training data. Feature reduction is also important for machine learning.This study presents Deep Learning (DL) as a solution.

A. Deep Learning

The field of data classification and prediction is known as DL. It can handle large amounts of raw data, up to millions of records, and perform activities on it directly. It achieves its goal through teaching through a hierarchical system of concepts. As a result, computers will be able to construct simple notions after they have grasped the more complicated ones. Deep Learning (DL) is implemented using a variety of algorithms, including Deep Boltzmann Machines (DBM), Deep Feedforward Networks (DFN), Deep Neural Network-based Hidden Markov Models (HMM), and Deep Convolutional Networks, among others. DL approaches are used in a variety of applications, including speech recognition, natural language processing and object detection. These methods are being used to detect onlinetransaction fraud, such as Deep Belief Networks, Deep Autoencoders, Recurrent Neural Networks, Convolutional Neural Networks, and others.

The design of a Convolution Neural Network (ConvNet/CNN) is inspired by the Visual Cortex and is similar to the connection pattern of neurons in the human brain. Each neuron could only reacts to the stimuli in the Receptive Field, a limited portion of the visual field. If a collection of comparable fields overlaps, the entire visual region is covered.

II. LITERATURE SURVEY

A plethora of studies have been undertaken in order to detect fraud. For detecting frauds, the telecoms industry has proposed a rule-based fraud detection system. The proposed model works well because it has a lesspercentage of incorrect triggering rates. To explain the overall process of detecting fraud payment throughsmart phones, supervised and unsupervised ways to identify fraud and handle massive amounts of financial data are proposed. Unsupervised machine learning approaches for financial data include Naive Bayes, SVM (Support Vector Machine), 1R, Decision Trees, C4.5, Random Forests, Logistic Regression and other methods are used to detect online financial fraud.

Among above techniques, the existing fraud detection approaches and their drawbacks were discussed [1]. A thorough trial with the answers to the problem of imbalance classification was carried out. These solutions, as well as the machine learning techniques utilised for fraud detection, were investigated.

Reshma R S [2] studied about an existing Deep learning model. The goal of the research is to use deep learning to detect fraud in credit card transactions. Neural networks with multiple hidden layers make up deep learning. The widespread usage of the Internet had a significant impact on the growth of online card transactions, particularly at the start of the previous decade [3]. With the rise in online transactions, the global banking industry has been obliged to cope with or meet an unanticipated number of fraudulent operations. As a result, rule-based systems were created to identify high-risk transactions and allow specialists to confirm whether or not they were fraudulent.Alex Sherstinsky [4] explains the essential RNN and LSTM fundamentals drawing from concepts in signal processing, Author derived the canonical RNN formulation from differential equations.

To investigate the performance of deep learning models that are being trained several metrics were considered [5].

Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan and Stephen Marshall [6] describe the importance and need of the Activation functions. The DL architectures use activation functions (AFs) to execute various computations between the hidden layers and the output layers of any given DL architecture to accomplish these state-of-the-art performances. A model, using Backpropagation and the ANN (Artificial Neural Network) method was created by Saurabh C. Dubey, Ketan S. Mundhe, Adiyta A. Kadam [7]. When compared to other algorithms, it behaves like a human brain and produces accurate and quick answers.

Dani Yogatama, Chris Dyer, Wang Ling, and Phill Blunsom [8] describe the performance of discriminative and generative LSTM models for text categorization using empirical data. RNN-based generative models exhibit greater asymptotic error rates than discriminatively trained RNN models, despite being more powerful than their bag-of-words forebears. It was also discovered that generative models approach their asymptotic error rate faster than discriminative models.Posterior probabilities of a machine learning model would be biased by under sampling [9].Although under sampling bias has no effect on the posterior probability's ranking order, it has a considerable impact on classification accuracy.

First, the customer's online transaction data is gathered from Kaggle, which includes numerous parameters such as name, time, last purchase, transaction history, and so on. The remaining 25% of data will be divided into test and validation data, which will be processed without result, and the remaining 75% of data will be trained from the dataset with the result. As a result, it performs better than other algorithms.

This will be incredibly valuable in the future because our model detects and predicts customer transactions in real time. Backpropagation is used in this, which is still under investigation in the field of AI. Real-time transaction detection is possible with this paradigm.

III. CNN-LSTM ARCHITECTURE

The CNN-Long Short-Term Memory Network, or CNN-LSTM for short, is an LSTM architecture specifically designed for sequence prediction problems with spatial inputs like text patterns, images, or videos.

Convolutional Neural Network (CNN) layers used to feature extraction on input data are paired with LSTMs to facilitate sequence prediction in the CNN-LSTM architecture.

CNN-LSTMs were created to solve visual time series prediction problems and to generate textual descriptions from image sequences. In particular, the issues of:

- ***Activity Recognition:*** Using a sequence of photos to generate a written description of an activity.
- ***Image Explanation:*** This is the process of creating a written description for a single image.
- ***Video Explanation:*** Creating a textual description of a picture sequence.

CNN-LSTMs are a type of model that is both spatially and temporally deep, and can be used to solve a wide range of vision problems with sequential inputs and outputs. Although it refers to LSTMs that employ a CNN as a fore front as CNN-LSTM in this course, this architecture was initially referred to as a Long-term Recurrent Convolutional Network or LRCN model. The duty of generating textual descriptions of photographs is handled by this architecture. The usage of a CNN that has been pre-trained on a difficult picture classification assignment and then repurposed as a feature extractor for the caption producing challenge is crucial.

It's natural to utilise a CNN to encode image/text by firstlygiving training it for an image/text classification job and then feeding the final hidden layer to the LSTM RNN decoder that creates sentences. CNNs have been utilised as feature extractors for LSTMs on textualinput data and audio data in this architecture, which has been applied to voice recognition and natural language processing issues.

This design is well suited to the following issues:

- Input with spatial organisation, such as 2D pixels in a photograph or 1D words in a phrase, paragraph, or page.
- Which require to produce temporal structure in their output, such as words in a textual description, or they have temporal structure in their input, such as the order of visuals in a video or words in text.

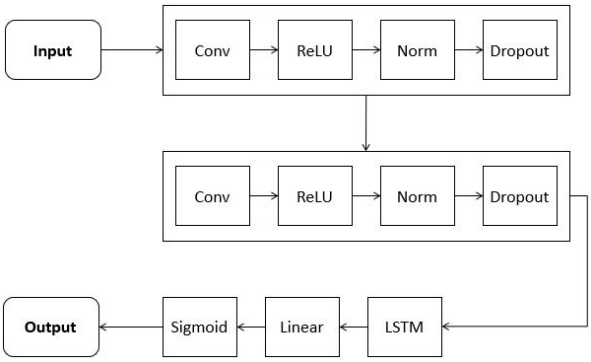

Fig. 1 CNN-LSTM Framework

A CNN-LSTM is made up of CNN layers on the front end, LSTM layers on the back end, and a Dense/Drop layer on the output.

There are two sub-models in this architecture:the CNN model extracts the local patterns in the data and the LSTM model acts to extract the global patterns in the data such that it is feasible for the projections of fraud or non-fraud.

Byretaining the sequential time series information, CNN will take the raw features and produce discriminative features that is used to determine whether the data is fraudulent.

The LSTM then works as a time series network, taking the sequence of discriminative characteristics and processing them. Linear layer, it will give the hidden dimensions of the network to detect fraud/non-fraud.

We employ the linear layer to obtain the projection layer, which converts other dimensions features into two dimensional features, in order to obtain the posterior probability for fraud/non-fraud detection.

IV. MODEL IMPLEMENTATION

The neural network models are first setup, and then the training process begins.The training process is divided into epochs, which are cycles. The dataset is partitioned into smaller pieces at this time. Finally, for epoch execution, an iterative procedure is run across a few batch sizes as subsections of the training dataset. This fraud transaction detection technique is designed to tackle the binary classification problem. The method determines whether or not the transaction is fraudulent. As a result, the Binary Cross Entropy function is employed as a criterion for training. The distance between the real value (it could be 0 or 1) and the estimation for every class is measured using Binary Cross Entropy, then the final loss is calculated by averaging these class-wise errors.

B. Input Channel

A Sequential() CNN model is appropriate for a basic stack of layers with precisely one input and output tensor in each layer.

Since the number of time steps involved in fraud detection is 10, it is considered to define the input dimension for the model. So, here, the input dimension of the neural network is 10. And the input channel is set to 1 and so is the output as the result will be either fraud or non-fraud transaction.

The Filter/Kernel is smaller than the input data, and a filter-sized patch of the input is multiplied with the filter using the dot product.

The number of kernels/filters would be the same as the number of features described in the neural network. So, here 10 filters are being used in the neural network.

Usually, the shape of the filter/kernel or kernel_size is less than the given input dimension to the network. Since it is a hyperparameter, the default value given is 2. It will perform 1-D convolutions in the neural network.

C. Convolution Layer

Conv1D() is a 1D Convolution Layer that's especially effective for extracting features from a fixed-length segment of a wider range of datasets when the feature's placement isn't as significant.

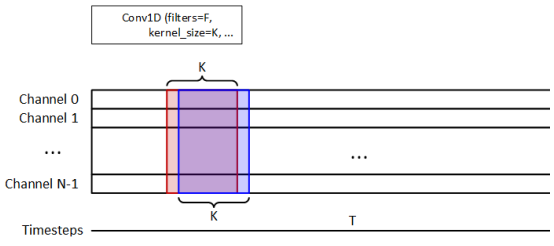

Fig. 21D Convolution Layer

The neural network is learning a total of 10 filters in the first Conv1D() layer, with a convolutional window size of 2 and an input of 1. The input shape parameter determines the input's shape. In any neural network, it is a required parameter for the first layer.

In the second Conv1D() layer, the neural network is fed the output of the first Conv1D() and the Batch Norm1D() output.

D. Batch Normalisation

To increase the stability of a neural network, batch normalisation normalises the output of a prior activation layer by subtracting the batch mean and dividing by the batch standard deviation.It uses a transformation to maintain the average output close to 0 and the standard deviation close to 1.

Dropout() is a regularisation technique that randomly deactivates or drops out a few neurons in the neural network to minimise overfitting.

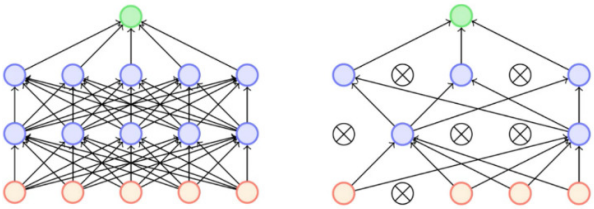

Fig. 3Network before and after Dropout()

Here, BatchNormalization1() will use the same dimension features as the above Conv1D(). As Conv1D() has 10D features, BatchNorm1D() also will have 10D features. And second Conv1D() has 10D features so, second BatchNorm1D() will also have 10D features.

Dropout(0.1) and Dropout(0.2) are added to the network after each of the Conv1D() and BatchNorm1D() layers.

E. Long-Short Term Memory

LSTM, an auto regressive component takes a 10D input features and outputs features that are projected into 100D. Here, the LSTM is 3 stacks of layers concatenated. It is useful to be projected in in higher dimensions to avoid information loss or corrupt if projected in low level. Also, it will be more discriminative and secured in this sequential manner.

Linear neural networks predict the output as a linear function of the inputs. The module nn.Linear(n,m) builds a single layer feed forward network with n inputs and m outputs.Mathematically, this module is designed to calculate the linear equation Ax = b where x is input, b is output, A is weight. Previous layer output i.e., LSTM output is in higher dimension but since a single output is needed to detect the result, a Linear layer is necessary to make it into a 1D layer. It will convert 100-dimension LSTM output to 1D, to minimize the information losses of features.

F. Activation Functions

1) ReLU

The rectified linear activation function, or ReLU for short, is a piecewise linear function that outputs the input directly if the input is positive and 0 otherwise.

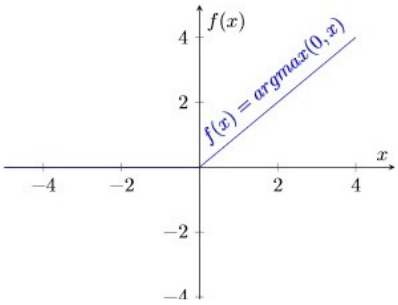

Fig. 4 ReLU Activation Function

It is a non-linear/non-negative component which allows only positive activations of previous layer i.e., f(x) = argmax (0, x).

$$\text{ReLU}(x) = \begin{cases} 0, & for\, x < 0 \\ 1, & for\, x \geq 0 \end{cases}$$

ReLU function is used in the neural network to activate convoluted samples from first and second Conv1D() operations.

2) Sigmoid

The sigmoid function is a type of logistic function that is generally symbolised by the letters (x) or sig (x). It is provided by:

$$\sigma(x) = 1/(1+\exp(-x))$$

Linear layer 1D output values might be negative or positive. But sigmoid will clip the negative values and transform them into positive values in the 0 to 1 probability range. It is to find out the probability of a sample being positive or negative. The probability of a non-fraud transaction is given by 1-(fraud).

The Sigmoid layer determines which active data should be updated and which should be disregarded. The logits are converted to posterior probabilities via the sigmoid layer.

G. SGD Optimizer

Stochastic gradient descent (SGD) is a variant of gradient descent that approximates the loss function's true gradient.

SGD is determined by taking into consideration all of the training instances and calculating an estimated gradient by iteratively taking one training example at a time until all of the training examples have been processed. The learning rate parameter of the SGD employed in the model is 0.01.

H. BCE Loss

Initially, the network will have random weights. If loss is derived w.r.t weights, some gradients will be obtained. Gradients will give the direction in which the loss will be minimized. If given a negative gradient, then the direction is to move to the opposite side, i.e., the positive side.

If there is a change in loss w.r.t weights, then the gradient has the same dimension as the weights. After updating for several time steps, an optimal solution will be obtained. Then it will give a random value. Then the values will be labelled as fraud or non-fraud. Then the labels and already predicted values will be passed as input into the BCE loss criterion. And the end of the training for a 10D feature input it will give the probability of fraud.

V. EXPERIMENTAL RESULTS

The CNN-LSTMtechnique is modelled and implemented using PyTorch open-source library packageand simulation results are observed using Google Colab software tool. Model is trained and tested with different proportion of dataset.

A loss is incurred for each epoch during the training of this model, as shown in Figure 5. As the number of epochs grows, the loss declines until it reaches a minimum.The reduced loss and greater accuracy imply that the model is working better, which is computed for 100 epochs.

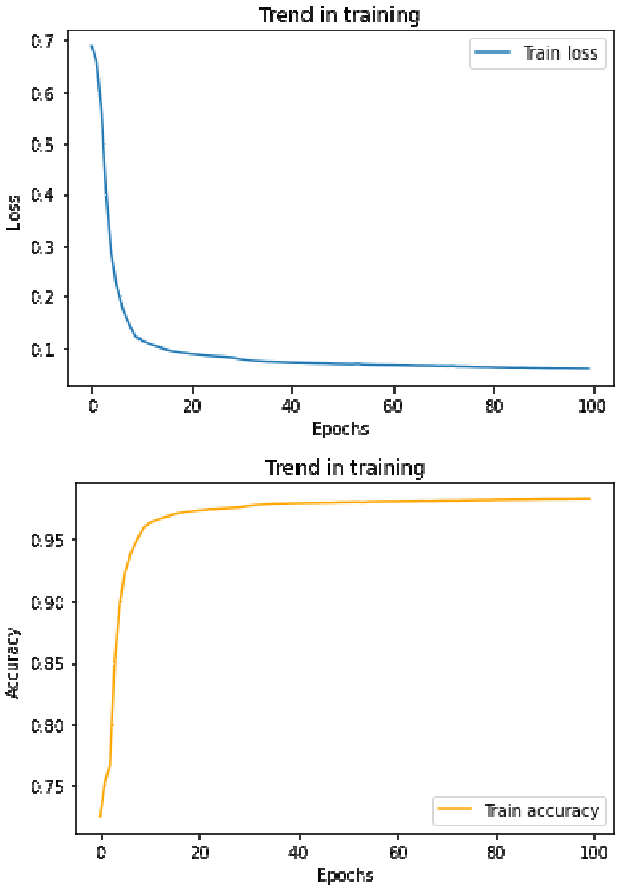

Fig. 5 ModelLoss and Accuracy during Training

The above training results were also reflected in the test data as shown in Figure 6. The model's loss and accuracy had been continuously reduced and increased respectively. The test data is used to see the performance of the model on new transactions.

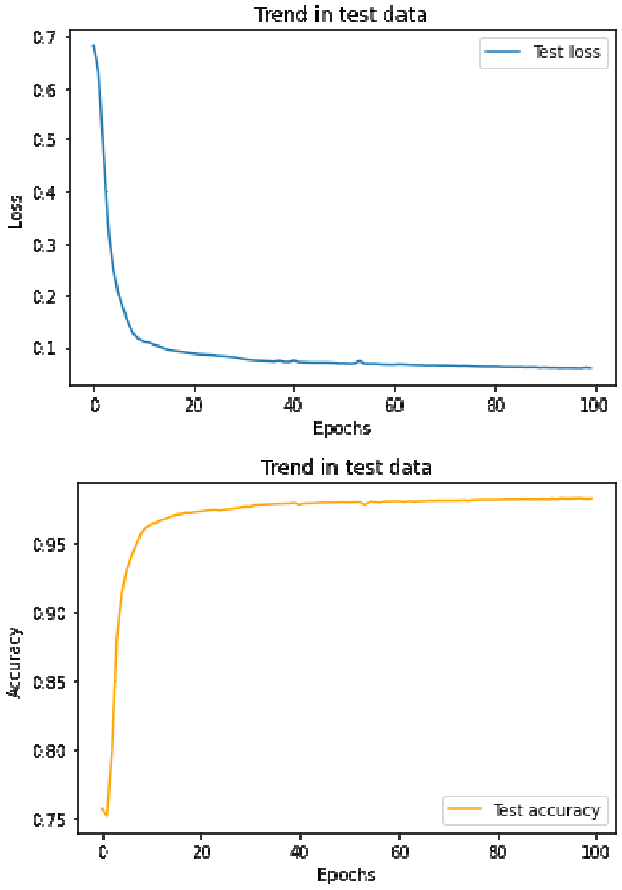

Fig. 6 ModelLoss and Accuracy during Testing

Figure 7 shows a confusion matrix that depicts both normal and fraudulent behaviour. A confusion matrix gives the number of successful and unsuccessful predictions is totalled and broken down by class using count values.

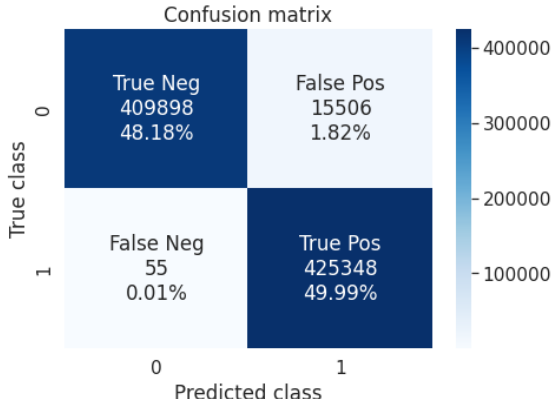

Fig. 7 Confusion Matrix of CNN-LSTM Model

The AUROC (Area Under the Receiver Operating Characteristics) curve is a performance assessment for classification issues at various threshold settings. The AUC is a measure of separability, whereas the ROC is a probability curve. The AUC indicates how well the model predicts 0 classes as 0 and 1 courses as 1. The AUROC score for the CNN-LSTM model is 0.986.

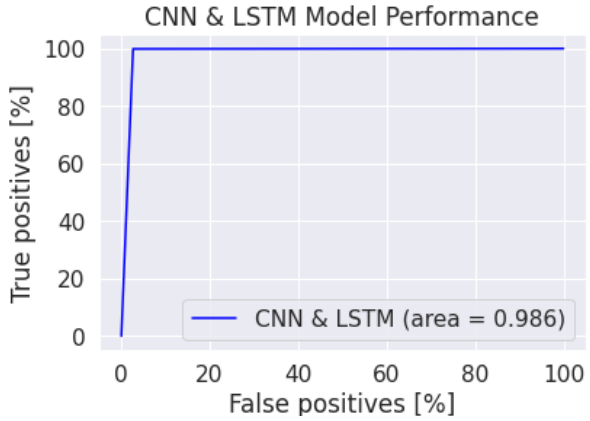

Fig. 8 AUROC plot of CNN-LSTM Model

The area under the PR curve is computed as the AUPRC (Area Under the Precision-Recall Curve). Across varying decision thresholds, a PR curve depicts the trade-off between accuracy and recall. The AUPRC value of the suggested model is 0.973.

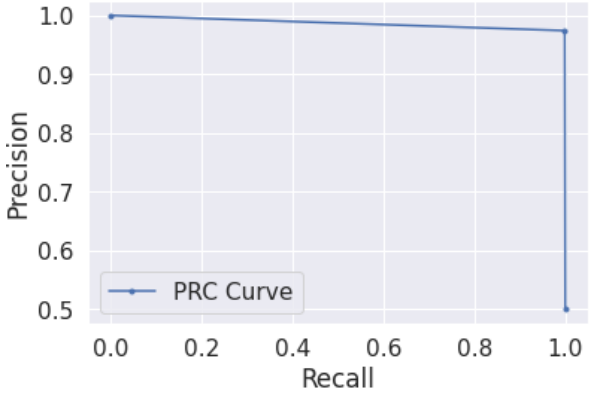

Fig. 9 AUPRC plot of CNN-LSTM Model

The Stacked CNN-LSTM model is measured in terms of the aforesaid evaluating metrics. The parameters used and the classification report are shown below. The CNN-LSTM model achieved an

Accuracy value of 98.6 percent and Mean Square Loss (MSE) of 0.01.

```
(2552419, 11, 1)
----------------------------------------------------------------
      Layer (type)         Input Shape         Param #     Tr. Param #
================================================================
         Conv1d-1         [1, 1, 11]              30              30
           ReLU-2        [1, 10, 10]               0               0
      BatchNorm1d-3        [1, 10, 10]              20              20
        Dropout-4        [1, 10, 10]               0               0
         Conv1d-5        [1, 10, 10]             210             210
      BatchNorm1d-6         [1, 10, 9]              20              20
        Dropout-7         [1, 10, 9]               0               0
           LSTM-8         [1, 9, 10]         206,400         206,400
         Linear-9           [1, 100]             101             101
       Sigmoid-10             [1, 1]               0               0
================================================================
Total params: 206,781
Trainable params: 206,781
Non-trainable params: 0
----------------------------------------------------------------
```

Fig. 10 Summary of CNN-LSTM Model

```
Training Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      0.97      0.99   1276209
         1.0       0.97      1.00      0.99   1276210

    accuracy                           0.99   2552419
   macro avg       0.99      0.99      0.99   2552419
weighted avg       0.99      0.99      0.99   2552419

Testing Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      0.97      0.99    425404
         1.0       0.97      1.00      0.99    425403

    accuracy                           0.99    850807
   macro avg       0.99      0.99      0.99    850807
weighted avg       0.99      0.99      0.99    850807
```

Fig. 11 Classification report of CNN-LSTM Model

VI. CONCLUSIONS

A Fraud Transaction Detection model using the CNN-LSTM Deep Learning technique has been implemented. The training and testing of the proposed model are programmed and implemented on Google Colab Notebook and verified through Python code using machine learning libraries. As the model learns from data rather than labels, it can be used with a variety of data sets. AUROC and AUPRC curves were plotted and AUC was taken as an accuracy criterion since the data is highly imbalanced. The CNN-LSTM model improved the AUC and obtained the value of 0.986. If the model is used with real-time data, the AUC value could be improved.

From experimental results, the suggested model is capable of detecting suspicious transactions with a high accuracy of 98.6% and precision of 98.5 percent. This approach is advantageous since it can be used to big financial datasets. The proposed neural network is not just an LSTM or feedforward network, but a combination of LSTM and CNN where CNN extracts the local patterns in the data and LSTM is used to extract the global patterns in the data such that it is feasible for projections of fraud or non-fraud. The CNN-LSTM model performance can be further improved by using the Transformers concept, which adopts a mechanism of attention so as to reduce false positives in a larger number of transactions.

REFERENCES

[1] Sara Makki; Zainab Assaghir, Yehia Taher, Rafiqul Haque, Mohand-Saïd Hacid, Hassan Zeineddine, "An Experimental Study with Imbalanced Classification Approaches for Credit Card Fraud Detection", IEEE Access on Advanced Software and Data Engineering for Secure Societies, vol.7, 2019.

[2] Reshma, R.S., “Deep learning enabled fraud detection in credit card transactions”, International Journal of Research & Scientific Innovation (IJRSI), V(VII), 111–115 (2018).

[3] Baris Can, A. Gokhan Yavuz, M. Elif Karsligil, M. Amac Guvensan, “A Closer Look into the Characteristics of Fraudulent Card Transactions”, IEEE Access, vol.8, 2020.

[4] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network”, Phys. D Nonlinear Phenom., vol. 404, no. March, pp. 1–43, 2020, doi: 10.1016/j.physd.2019.132306.

[5] J. Liu, J. Liu, W. Du, and D. Li, “Performance analysis and characterization of training deep learning models on mobile device”, 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), vol. 2019-Decem, pp. 506–515, 2019, doi: 10.1109/ICPADS47876.2019.00077.

[6] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning”, 2nd International Conference on Computational Sciences and Technologies, 17-19 Dec 2020 (INCCST 20).

[7] Saurabh C. Dubey, Ketan S. Mundhe and Aditya A. Kadam, “Credit Card Fraud Detection using Artificial Neural Network and BackPropagation”, 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020.

[8] Yogatama, D., Dyer, C., Ling, W., Blunsom, P., “Generative and discriminative text classification with recurrent neural networks”, arXiv preprint arXiv:1703.01898 (2017).

[9] Andrea Dal Pozzolo, A.D., Caelen, O., Johnson, R.A., Bontempi, G., “Calibrating probability with undersampling for unbalanced classification”, IEEE Symposium Series on Computational Intelligence, pp. 159–166. IEEE (2015).

[10] J. Padhye, V. Firoiu, and D. Towsley, “A stochastic model of TCP Renocongestion avoidance and control,” Univ. of Massachusetts, Amherst,MA, CMPSCI Tech. Rep. 99-02, 1999.

[11] Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Information Science and Statistics.