An Anomaly Detection Model in Credit Card Transactions Using Machine Learning Technique

[1]James O.O., [2]Abdulsalami B.A. & [3]Kolawole A.A.
[1, 2, 3] Department of Mathematical & Computer Sciences
Fountain University
Osogbo, Osun State, Nigeria
E-mail: [1]seun.james@vdtcomms.com[2]abdullateefhamzah@ymail.com, [3]kolawoleayishat@yahoo.com

ABSTRACT

Credit card fraud activities have rapidly increased all over the world and are still very evolving with different techniques used by the fraudsters. As a result, organizations and individual user suffers if the information of credit card gets stolen. This work investigates the use of Feed Forward Back Propagation Neural Network algorithm for modeling fraud detection in online transaction. This was simulated using MATLAB. The System considered only credit card online transaction among other online transactions. The credit card holder's transactions details (284,807 in number), which consist of demographic and transaction variables were acquired online at www.kaggle.com. Eighty percent (70%) of the transaction dataset was used in training and the twenty percentages (30%) was used in validating the models via testing. The Feed Forward Back Propagation Neural Network (BPNN) has classifier accuracy of 99.9%, AUPRC of 59.3% and Prediction Accuracy of 79.9%. Thus, this work has helped proven that Feed Forward BPNN based system can detect fraud in online transactions with 79.9% prediction accuracy.

Keywords*:* Credit card, Machine learning, Fraud detection, Classification, Back propagation, neural networks

iSTEAMS Multidisciplinary Conference Proceedings Reference Format
James O.O., Abdulsalami B.A. & Kolawole A.A. (2019): An Anomaly Detection Model in Credit Card Transactions Using Machine Learning Technique. Proceedings of the 22nd iSTEAMS Multidisciplinary SPRING Conference. Aurora Conference centre, Osogbo, Nigeria. 17th – 19th December, 2019. Pp 123-142. www.isteams.net/spring2019. DOI - https://doi.org/ 10.22624/AIMS/iSTEAMS-2019/V22N1P11

1. INTRODUCTION

In recent times, the internet has become the main medium for conducting electronic commerce. Many products, tangible and intangible, are browsed through and sold over the Internet. Due to the increasing popularity of e-commerce in our daily lives, credit card usages have dramatically increased over the years. As credit card being the primary method of payment in online transactions, credit card frauds have also been observed to surge as the number of online transactions have increased (Lee, Ham and Jiang, 2014). Credit card fraud is increasing considerably with the development of modern technology and global super highways of communications which cost hundreds of millions of dollars annually (Akshata and Sheetal, 2015). Credit card fraud affects the organization by financial losses and individual user is also affected if the credit card gets stolen. Different credit card fraud activities have rapidly increased all over the world and are still very evolving with different techniques used by the fraudsters to perpetrate fraud.

For many years, the credit card industry has studied computing models for automated detection systems. Recently, these models have been the subjects of academic research, especially with respect to e-commerce (Chan, Fan, Prodromidis, and Stolfo, 1999).

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior. These nonconforming patterns are often referred to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities, or contaminants in different application domains. Anomaly detection finds extensive use in a wide variety of applications such as fraud detection for credit cards, insurance, or health care, intrusion detection for cyber-security, fault detection in safety critical systems, and military surveillance for enemy activities (Varun, Arindam and Vipin, 2009). Anomalies in credit card transaction data could indicate credit card or identity theft (Neda, Leila and Ebrahim, 2012). Many techniques based on Machine Learning such as Data mining, Fuzzy logic, Sequence Alignment, Clustering Algorithms, Genetic Programming, etc., has evolved in detecting various credit card fraudulent transactions.

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that provides systems the ability to learn automatically and improve from experience without being explicitly programmed or with no human intervention. AI is an aspect of computer science that emphasizes on the creation of intelligence machine that can act and work as Human. This is intelligence demonstrated by machine in which a computer system has the ability to think and make predictions on its own without human intervention.

Artificial Neural Network (ANN) is a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. It interprets sensory data through a kind of machine perception, labeling or clustering raw input. Being a non-linear statistical data modeling tool, ANN can be used to model complex relationships between inputs and outputs and can be trained to learn from observation and generalize by abstraction (Yasumoto and Jia, 2005).

In this work, we implemented a Feed forward Back Propagation Neural Network (BPNN), a variety of ANN to model a credit card fraud detection system using machine learning approach. We investigated the usefulness of applying ANN; a supervised learning technique in fraud detection in an online transaction by evaluating the performance of the model.

The rest of the paper is organized as follows: Section 2 discusses the statements of the problem we are trying to solve, followed by a section elaborating on the subject matter and existing related works. Section 4 explains the methodology adopted outlining the phases of the methodology, followed by section 5, which discusses the implementation and result. Finally, the paper concludes with some final remarks as well as directions for future work.

1.1 Problem Statement

Credit Card fraud is an evolving problem. The growing number of credit card transactions provides more opportunity for thieves to steal credit card numbers and subsequently commit fraud. Despite significant efforts by merchants, card issuers and law enforcement to curb fraud, online fraud continues to plague electronic commerce web sites (Alese, Adewale, Aderounmu, Ismaila and Omidiora, 2012). Fraud detection is a continuously evolving discipline and ever-changing tactics to commit fraud. So, it needs special methods of intelligent data analysis to detect and prevent it (Razak and Ahmed, 2014). To address this problem, Back propagation Neural Network (BPNN) for fraud detection in online transaction is developed. This work investigated the extent to which BPNN detects anomalies in credit card transactions.

2. LITERATURE REVIEW

2.1 Anomaly Detection

Detection of anomalies has been one of the major focusing areas of researches in important topic in data mining and machine learning. Many real-world applications such as intrusion or CCFD require an effective and efficient framework to identity deviated data instances. However, most anomaly detection methods are typically implemented in batch mode, and thus cannot be easily extended to large-scale problems without sacrificing computation and memory requirements. (Balakrishna and Genesh, 2014).

Anomaly detection is defined as the process of using models to identify behavior that is different from the normal behavior of a system. The importance of on-time detection of anomalies in a reliable, efficient and robust manner had been seen to be highly imperative. In 2011, Guardian Analytics stated that anomaly detection is the process of detecting something unusual relative to something expected. In the realm of online banking or online transactions, this can be termed as suspicious (unusual) behavior in order to identify account takeover and fraudulent transactions.

2.2 How Anomaly detection works

The most effective anomaly detection approach focuses on the individual account holder. Different users quite naturally have different online banking behaviour from each other. Said differently, each account holder has a unique online banking fingerprint. Anomaly detection takes advantage of this fact combined with knowledge of online banking fraud attacks and general online behaviour to determine if a specific online session is legitimate or has high risk of being fraudulent (Guardian Analytics, 2011).

Guardian Analytics (2011) presented a simple breakdown of the process anomaly detection solutions use to detect suspicious activity for each individual account holder:

i. Create and continually update a model of expected behaviour for each individual account holder.
ii. Monitor all online banking for each individual account holder.
iii. Analyze all individual account behaviour during an online banking session from login to logout – how they access their account, how they manage their accounts, the types of transactions they engage on, the frequency of activities, what kinds of activities take place during the same session, the type and amounts of payments, who the payees are, and much more.
iv. By comparing individual or groups of activities in this online session to demonstrated patterns of normal behaviour, determine if the session is legitimate or unusual, unexpected, or suspicious.

2.3 Credit Card in Online Transaction

At the current state of the world, credit cards are used for purchasing goods and services using online transaction or physical card for offline transaction, even street vendors, are now accepting cashless payments so that usage of credit card is increase. This development could be nailed to globalization and increased use of the internet for online shopping has resulted in a considerable increase of credit card transactions throughout the world (Rinky D. Patel and Dheeraj Kumar Singh, 2013), along with the rapid advances of e-commerce, the use of credit card has become a convenience and necessary part of financial life. (Ong Shu Yee *et al*, 2018). Credit card is easy to carry and easy for payments while on the move and for online purchase (Antara Dey and Kavitha Sudha, 2018). The credit card might be physical or virtual. In some physical-card, the cardholder shows his card physically to few merchants for generating payment (Xu Wei and Liu Yuan, 2012).

Consumers' demand for electronic transactions due to its convenience and ease of use, and the rise in e-commerce has opened up new opportunities for criminals to steal credit card numbers and consequently commit fraud (Royal Canadian Mounted Police 2010). Despite all the benefits credit card serves, the rapid growth in the number of credit card transactions has led to a substantial rise in fraudulent activities. According to Global Payments Report 2015, credit card is the highest used payment method globally in 2014 compared to other methods such as e-wallet and Bank Transfer. Also, in the past couple of the years, credit card breaches have been trending alarmingly. In addition, Nilson Report also reported that the global credit card fraud losses reached $16.31 billion in 2014 and it is estimated that it will exceed $35 billion in 2020.

The development of efficient methods which can distinguish rare fraud activities from billions of legitimate transactions seems essential. Although, CCFD has gained attention and extensive study especially in recent years and there are lots of surveys about this kind of fraud (Samaneh Sorournejad *et al*, 2016).

2.4 Credit Card Fraud Detection Process

Credit card fraud is one of the major problems in the financial institutions such as banks, credit card industry etc. The goal of a detection system is to be able to detect fraud in the dataset in a real time manner, so as to reduce fraudulent transactions which cost hundreds of millions of dollars annually (Akshata and Sheetal, 2015). The main idea when detecting fraud is to firstly understand and identify the type of credit card fraud. There are various types of credit card fraud (both online and offline). Depending on the type of fraud faced by banks or credit card companies, various measures has been adopted and implemented curb these activities, but however, there is still a need for a more robust system in order to detect frauds more accurately.

A transaction is fraudulent if the transaction pattern and other properties (e.g. location, amount, etc.) do not conform or follow the regular pattern of that card dataset. A fraudulent transaction can be detected if the regular way in which the card owner uses his/her card isn't followed. The credit card dataset is used in training the Machine Learning (ML) algorithms. These algorithms are capable of learning and predicting without any human intervention. Consequently, the system will be able to recognize the patterns of every particular card holder and if any future transaction is fraudulent, the algorithm or model will be able to detect such fraudulent transactions.

2.5 Related Works

A good amount of literatures has investigated previous works vis-à-vis – the methods, advantages and demerits. In 2018, Navanshu *et.al* proposed a new collative comparison measure that reasonably represents the gains and losses due to fraud detection is proposed. A cost sensitive method which is based on Bayes minimum risk is presented using the proposed cost measure. An improvement up to 23% is obtained when this method and other state of art algorithms are compared. The data set for this paper is based on real life transactional data by a large European company and personal details in data is kept confidential; accuracy of an algorithm is around 50%. The significance of this paper was to find an algorithm and to reduce the cost measure. The result obtained was by 23% and the algorithm they find was Bayes minimum risk.

In 2016, S.Fashoto, O.Adeleye and J.Wandera used a hybrid of K-means clustering with Multilayer Perceptron (MLP) and the HMM. They used K-means clustering to group together the suspected fraudulent transactions into a similar cluster. The output of this is used to train the HMM and the MLP, which then classify the incoming transactions. In their results, it was found that the detection accuracy of "MLP with K-means Clustering" is higher than the "HMM with K-means clustering" but the result is reversed for 10-fold cross-validation. In 2015, Ayushi Agrawal *et al* proposed testing a transaction using Hidden Markov Model (HMM), Behavior based technique and Genetic Algorithm, wherein they used the HMM to maintain the record of previous transactions, Behavior

based technique for grouping of datasets and lastly genetic algorithm for optimization, that is, calculating the threshold value. In the same year, Pooja Chougule *et al* proposed simple K-means and Simple Genetic Algorithm for fraud detection. In their work, they showed that how k-means algorithm grouped the transactions based on the distinct attribute values and genetic algorithm was used for optimization since with the increase in size of the input k-means algorithm produced outliers. Basically K-means algorithm produced clusters which were then optimized by the genetic algorithm.

In another work by Tanmay Kumar and Suvasini Panigrahi (2015), they proposed a hybrid approach to CCFD using Fuzzy Clustering and Neural Network. It makes use of two phases. In phase one, they used a K-means clustering algorithm to generate a suspicious score of the transaction and in next phase if a transaction is suspicious it is feed into neural network to determine whether it was really fraudulent or not. Also, J. Esmaily and R. Moradinezhad (2015) proposed a hybrid of Artificial Neural Network (ANN) and Decision Tree (DT). In their model they used a two-phase approach. In first phase the classification results of DT and Multilayer perceptron were used to generate a new dataset which in second phase is feed into Multilayer perceptron to finally classify the data. This model promises reliability by giving very low false detection rate.

In 2014, Devaki *et al* developed a CCFD using time series analysis. The fraud detection is done with data mining approaches. The parameters considered are transaction amount and transaction time. They used the periodic pattern in the spending behaviour of a cardholder to detect the anomalies in the transaction with respect to the analyses of the past history of transactions belonging to an individual cardholder. Two levels of detection methods were used. The first level detect fraud by analyzing whether the new incoming transaction is fraud or not by using distance-based method while in the second level the next transaction is predicted by means of label-prediction methodology and compared with the actual transaction, if there is deviation then it is detected to be a fraudulent transaction. If the particular transaction is considered as a fraud then the cardholder is asked to continue the transaction by asking a secret question, if the cardholder does not give correct answer then the transaction not be allowed to continue further. The approach decreased the false positive (FP) situation and hence it is ensured that genuine transaction is not rejected.

In addition, Patel and Gond (2014) developed Support Vector Machine (SVM) learning for CCFD. The SVM based method with multiple kernel involvement, which also includes several fields of user profile instead of only spending profile. The simulation result shows improvement in True Positive (TP), True Negative (TN) rate, and also decreases the FP and False Negative (FN) rate. In their work, the effectiveness of the hybridization of two techniques using both the user profile and spending profile in detecting anomaly in online transaction by improving on false rejections and prediction accuracy is looked into by comparing it with standalone units of the algorithms.

In 2013, Avinash Ingole and R. C. Thool used the HMM combined with the Clustering algorithm in CCFD. They were able to build a system that can detect fraud using spending profile. This system checks for the past transaction history of the customer and make decision from it. Its limitation is True Positive (TP) and False positive (FP) issues. Also, Falaki *et. al.,* (2012) developed a probabilistic CCFD system in online transactions. The developed probabilistic based model serves as a basis for mathematical derivation for adaptive threshold algorithm for detecting anomaly transactions. Experimental results show the performance and effectiveness of new approach system and demonstrate the usefulness of learning the spending profile of the cardholders. The optimization of parameters, Posterior-Viterbi cum new detection model performed better than Viterbi cum old detection model.

The results obtained from the evaluation showed the overall average of accuracy and precision are about 84% and about 86% respectively. In the same year, R. Dhanapal used the DT and Hunts algorithm techniques to detect credit card fraud. In this technique we simply find out the Fraudulent Customer/Merchant through Tracing Fake Mail and IP Address. Customer /merchant are suspicious if the mail is fake, they traced all information about the owner/sender through IP Address. It was called the Tracing Email and IP fraud detection.

In 2011, EkremDuman and M. HamdiOzcelik proposed a method that would improve the then existing CCFD systems in banks. They devised a system to credit each transaction certain score and based on that score the transaction was judged and for this they combined Neural Networks with scatter search. Also, in 2011, Alireza Pouramirarsalani1 *et al* proposed a new method of fraud detection which used a hybrid of feature selection and Genetic Algorithm (GA). They observed the salient features of the transactions and used the same while detecting any unusual feature and flagging it to be the fraud one. The GA was used in the optimization and search problems.

In addition, Raj and Portia (2011), proposed a paper that represents a research about a case study involving CCFD, where data normalization is applied before Cluster Analysis and with results obtained from the use of Cluster Analysis and ANN on fraud detection has shown that by clustering attributes neuronal inputs can be minimized. And promising results can be obtained by using normalized data and data should be MLP trained. This research was based on unsupervised learning. Significance of this paper was to find new methods for fraud detection and to increase the accuracy of results. Furthermore, Raghavendra Patidar and Lokesh Sharma (2011) proposed a hybrid of ANN and GA in their work. They used NN to classify the transactions and GA to optimize the solution and to not over train the system.

The use of ANN to detect credit card fraud was also deployed by Raghavedra Patidar and Lokesh Sharma. (2011), they used the GA in order to derive the optimal parameters of ANN. Like many other data mining techniques, ANNs make use of a number of parameters which need to be specified by software developers. Although the values of theses parameters can seriously affect the predicting accuracy of ANN models. A standard practice for specifying these parameters has never been established. The disadvantage of this was the time taken to train the algorithm with data and the optimization process.

More so, Brabazon*et.al* (2011) proposed an Artificial Immune System based model for online CCFD. Three (3) AIS algorithms were implemented and their performance was standardized against a logistic regression model. Their chosen algorithms were the unmodified Negative selection algorithm, the modified Negative selection algorithm and the Clonal selection algorithm. They proposed the Distance Value Metric for calculating distance between records. This metric is based on the probability of data occurrence in the training set. Where the detection rate increased, but the number of false alarms and missed frauds remained. Implementation of the Artificial Immune system (AIS) was also carried out by Arunabha Mukhopadhyay *et al*. in 2011, to detect fraud by Matching Binary Strings Using detector and response. This system works like the human immune system by fighting fraud before it can occur. The limitation is the complexity of the solution.

3. RESEARCH METHODOLOGY

Discussed in this section is the methodology adopted for this work in implementing the proposed CCFDS. CCFD in online transaction dataset using Feed forward BPNN as researched in this work involved Four (4) phases; the Data Acquisition, Data Preprocessing, which involves feature extraction, Training and testing. The Data Acquisition phase comprises of where and how the dataset used was collected, Data preprocessing phase includes how the dataset was preprocessed, and features extracted, the training and testing stage.

a. Data Acquisition

The acquisition of dataset to test the method was a very difficult task mainly because financial institutions do not generally agree to share their data with researchers for security reasons. All efforts prove abortive, therefore, a real credit card transaction of Europeans cardholders that was provided for CCFD competition on *Kaggle.com* was used to train the models and also test the model. A total number of Two Hundred and Eighty-Four Thousand, Eighty Hundred and Seven (284,807) real credit card transactions was used to train and validate the model to detect if the transaction was fraudulent or valid. Eighty percent (80%) of the transaction dataset was used in training while the remaining percentage (20%) was used in testing the models.

b. Data Preprocessing

The obtained dataset was already preprocessed in such a way that the variables were acceptable to be used in training the models. The dataset provided was already labeled i.e. classified into Fraudulent and Valid transaction. The total number of valid transactions was Two Hundred and Eighty-Four Thousand, Three Hundred and Fifteen (284,315) and the total number of fraudulent transaction was Four Hundred and Ninety-Two (492). Eighty percent (80%) of the transaction dataset was used in training and the twenty percentages (20%) was used testing the models.

c. Feature Extraction

In this work, the credit card features (information) used for training and testing were already extracted using Principal Component Analysis (PCA) except for the “Time” and “Amount” column. The main idea of the PCA is to reduce the dimensionality of a dataset consisting of many variables correlated with each other. It is an effective way to discover or hide important features of a dataset. Therefore, the transaction variables were reduced into the PCA variables. The PCA variables are the sensitive information of the dataset or cardholder such as the name, address, sex, location, marital status, etc. This sensitive information was transformed using the PCA for confidentiality issues. Figure 3 shows the features of the dataset used, where V1 to V28 are some important features of the dataset. “Amount” and “Time” is the amount of money withdrawn and the time at which it is withdrawn.

The Class represents whether a transaction is Fraud (illegal) or Normal (legal), 1 and 0 respectively.

The data set has 31 columns (V1, …, V28, Amount, Time, Class)

```
data = pd.read_csv('C:/Users/opeyemi/creditcard.csv')
print(data.columns)
data.info()
```

```
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
       'Class'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
Time      284807 non-null float64
V1        284807 non-null float64
V2        284807 non-null float64
V3        284807 non-null float64
V4        284807 non-null float64
```

Fig. 1: Feature Extraction

d. Classification

This is the last stage of the credit card detection process. This stage comprises of the training and testing stage. The training stage comprises of 7% (Two Hundred and Twenty-Seven Thousand, Eight Hundred and Forty-Six) of the dataset and the other 30% (Fifty-Six Thousand, Nine Hundred and Sixty-One) was used in testing and validation. The features extracted (PCA variables) were fed into the ANN algorithm. The implementation method used here is a supervised learning, which is like finding the correct solution to already known correct answer. Supervised learning is a learning rule that will train the NN based on already known correct output.

In NN, we first initialize the weights with adequate values then we take the input from training data. The input is formatted as input, correct output. The output from the NN is then compared with correct output and the error is calculated. Then we adjust the weights to reduce the error. And we keep re-calculating the error and adjusting the weight for all training data to reduce the error until desire output is achieved. NN stores information in terms of weights, that means, to train a NN with new information, we have to modify the weights. The difference between the output and correct output of the NN is the error. Below is the flow diagram of the detection model and the pseudocode, showing how the model works.

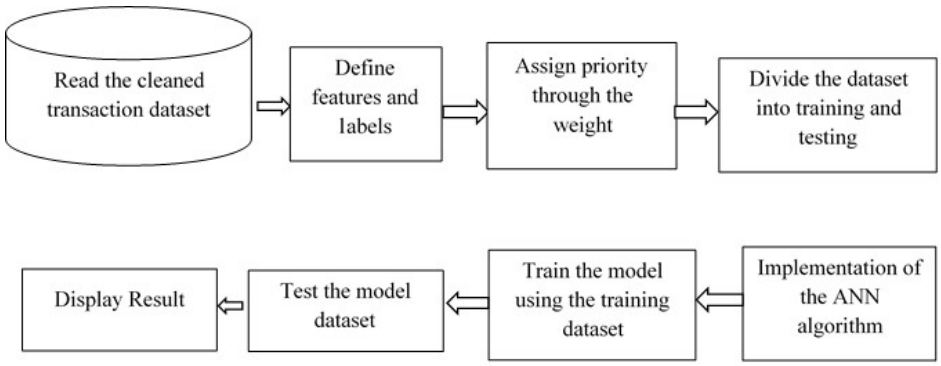

Figure 2: Flow diagram of the detection model

Pseudocode 1: Description of the Feed forward BPNN algorithm

// The following describes the how the Feed Forward Back-propagation algorithm works.
Assign all network inputs and output
Initialize all weights with small random numbers
repeat
for every pattern in the training set
Present the patter to the network
// Propagated the input forward through the network:
for each layer in the network
for every node in the layer
1. *Calculate the weight sum of the inputs to the node*
2. *Add the threshold to the sum*
3. *Calculate the activation for the node*

End
End

// Propagate the errors backward through the network
for every node in the output layer
calculate the error signal
end
for all hidden layers
for every node in the layer
1. *Calculate the node's signal error*
2. *Update each node's weight in the network*

End
End
// Calculate the Error Function
End
while ((maximum number of iterations < = specified)

3.1 Dataset for Implementation

From the dataset (Figure 2), the class column contains our already known correct output which is 0 for non-fraudulent and 1 for fraudulent activities.

	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
2	-0.5516	-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791	0.403993	0.251412	-0.01831	0.277838	-0.11047	0.066928	0.128539	-0.18911	0.133558	-0.02105	149.62	0
3	1.612727	1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336	-0.14578	-0.06908	-0.22578	-0.63867	0.101288	-0.33985	0.16717	0.125895	-0.00898	0.014724	2.69	0
4	0.624501	0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136	-2.26186	0.52498	0.247998	0.771679	0.909412	-0.68928	-0.32764	-0.1391	-0.05535	-0.05975	378.66	0
5	-0.22649	0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775	-1.23262	-0.20804	-0.1083	0.005274	-0.19032	-1.17558	0.647376	-0.22193	0.062723	0.061458	123.5	0
6	-0.82284	0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819	0.803487	0.408542	-0.00943	0.798278	-0.13746	0.141267	-0.20601	0.502292	0.219422	0.215153	69.99	0
7	1.341262	0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653	-0.03319	0.084968	-0.20825	-0.55982	-0.0264	-0.37143	-0.23279	0.105915	0.253844	0.08108	3.67	0
8	-1.41691	-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199	-0.04558	-0.21963	-0.16772	-0.27071	-0.1541	-0.78006	0.750137	-0.25724	0.034507	0.005168	4.99	0
9	-0.61947	0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822	0.324505	-0.15674	1.943465	-1.01545	0.057504	-0.64971	-0.41527	-0.05163	-1.20692	-1.08534	40.8	0
10	-0.70512	-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765	0.570328	0.052736	-0.07343	-0.26809	-0.20423	1.011592	0.373205	-0.38416	0.011747	0.142404	93.2	0
11	1.017614	0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677	0.451773	0.203711	-0.24691	-0.63375	-0.12079	-0.38505	-0.06973	0.094199	0.246219	0.083076	3.68	0
12	1.199644	-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344	-0.22137	-0.38723	-0.0093	0.313894	0.02774	0.500512	0.251367	-0.12948	0.04285	0.016253	7.8	0
13	-0.25912	-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985	0.707664	0.125992	0.049924	0.238422	0.00913	0.99671	-0.76731	-0.49221	0.042472	-0.05434	9.99	0
14	0.227666	-0.24268	1.205417	-0.31763	0.725575	-0.81561	0.873936	-0.84779	-0.68319	-0.10276	-0.23181	-0.48329	0.084668	0.392831	0.161135	-0.35499	0.026416	0.042422	121.5	0
15	-0.77366	0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805	-0.98292	-0.1532	-0.03688	0.074412	-0.07141	0.104744	0.548265	0.104094	0.021491	0.021293	27.5	0
16	0.844555	0.792944	0.370448	-0.79498	0.406796	-0.30306	-0.15587	0.778265	2.221868	-1.58212	1.151663	0.222182	1.020586	0.028317	-0.23275	-0.23556	-0.16478	-0.03015	58.8	0
17	-0.79398	-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999	0.432535	0.263451	0.499625	1.35365	-0.25657	-0.06508	-0.03912	-0.08709	-0.181	0.129394	15.99	0
18	-0.45031	0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591	-0.57568	-0.11391	-0.02461	0.196002	0.013802	0.103758	0.364298	-0.38226	0.092809	0.037051	12.99	0
19	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.025436	-0.04702	-0.1948	-0.67264	-0.15686	-0.88839	-0.34241	-0.04903	0.079692	0.131024	0.89	0
20	0.91723	0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-0.40687	-2.19685	-0.5036	0.98446	2.458589	0.042119	-0.48163	-0.62127	0.392053	0.949594	46.8	0
21	1.077542	-0.63205	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432	0.05423	-0.38791	-0.17765	-0.17507	0.040002	0.295814	0.332921	-0.22038	0.022298	0.007602	5	0
22	1.019151	1.298329	0.42048	-0.37265	-0.80798	-2.04456	0.515663	0.625847	-1.30041	-0.13833	-0.29558	-0.57196	-0.05088	-0.30421	0.072001	-0.42223	0.086553	0.063499	231.71	0
23	1.69033	0.406774	-0.93642	0.983739	0.710911	-0.60223	0.402484	-1.73716	-2.02761	-0.26932	0.143997	0.402492	-0.04851	-1.37187	0.390814	0.199964	0.016371	-0.01461	34.09	0

Figure 3: Flow diagram of the detection model

4. IMPLEMENTATION, RESULTS AND FINDINGS

The implementation method used here is a supervised learning, which is like finding the correct solution to already known correct answer. Supervised learning is a learning rule that will train the neural network based on already known correct output.

4.1 Framework Implementation

MATLAB tool was the software framework adopted for the implementation of this work. MATLAB stand for Matrices Laboratory. It's a machine learning programming tool with interactive graphical user interface (GUI). It is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. It's used for mathematical computation, Algorithm development, Modelling, Simulation, Prototyping, Data analysis, exploration, and visualization, Scientific and engineering graphics and lots more. The version of MATLAB 2018a was obtained and utilised for implementation and the extraction of results.

The first step is to gather or collect the required transaction dataset and load into the simulator environment, as shown in Figure 3. Once the collected transaction datasets are imported into the workspace, input data and known output data are separated. What follow next is to create the training network/algorithm based on define type, structure and parameters. This provides privilege to decide on; various network type MATLAB provided like feed forward, radial basis, activation function, internal structure of NN used like the number of nodes in the input layer, number of hidden layer and the number of nodes into the output layer, parameters values like weights, bias, delay.

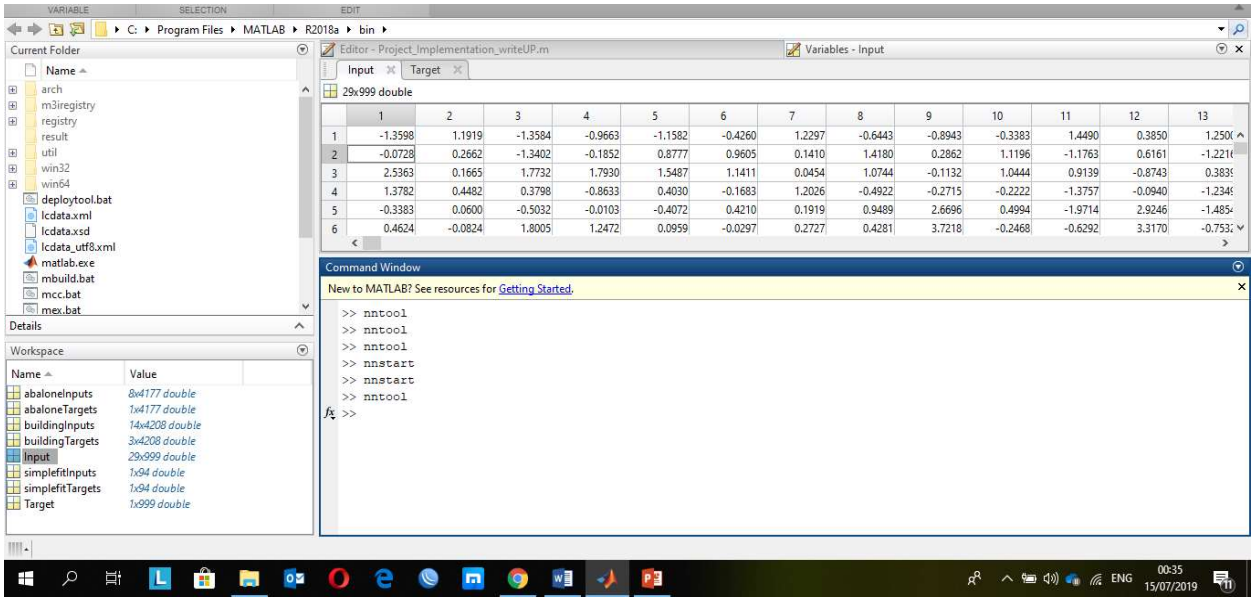

Figure 4: Creating the network

The tool offers creation of a variety of NN types like Perceptron's, Feed-forward NN, Recurrent NN, Probabilistic NN, Radial basis NN, Self-organizing, Time-delay NN, as shown in Figure 5. Each has a creation function with a set of input arguments to define its structure: inputs, number of nodes, hidden layers and number of layers, etc. For our work, Feed forward BPNN was used because transaction dataset is mapped to output. Feed forward networks consist of a series of layers. The first layer has a connection from the network input. Each subsequent layer has a connection from the previous layer. The final layer produces the network's output.

To ensure that the network is compatible with the problem we want to solve, we configure the network by arranging it. Usually the network is created with default values for its parameters, but one can change this either by reassigning. In our work, the default value of number of nodes, layers and hidden layer were used. 70% of dataset is used for training while 15% for validation and testing respectively. After the network has been configured, we initialized the weight and biases; these are adjustable network parameters which need to be tuned so that the network performance is optimized.

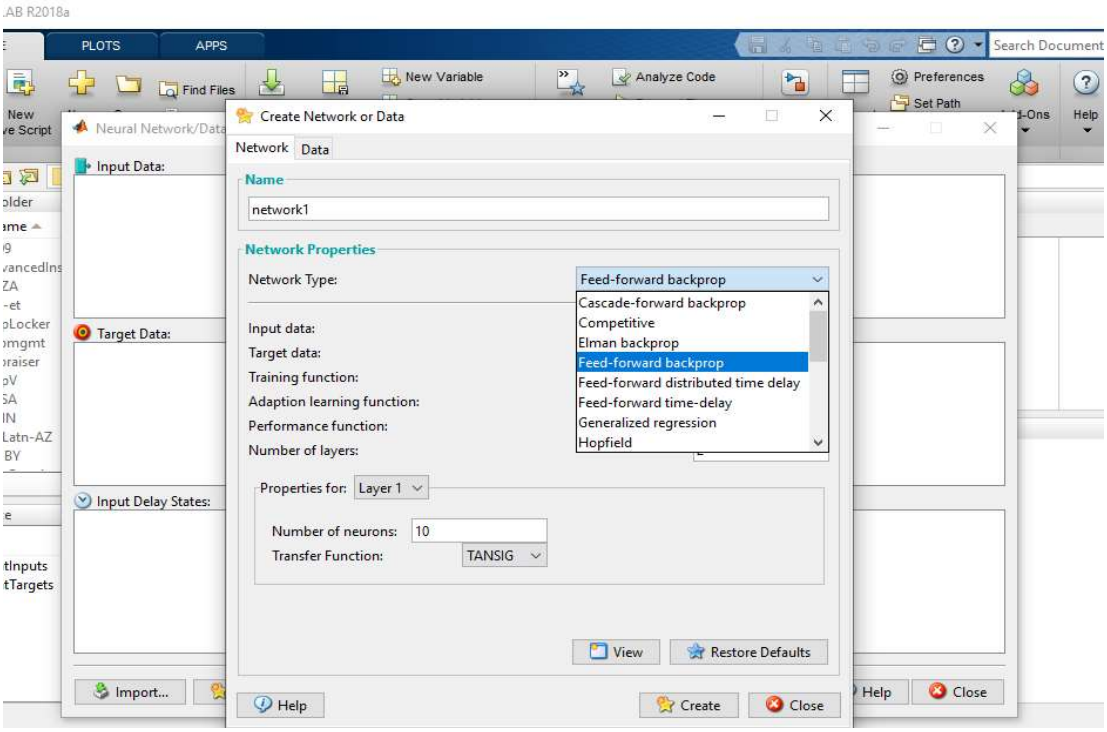

Figure 4: Creating the network

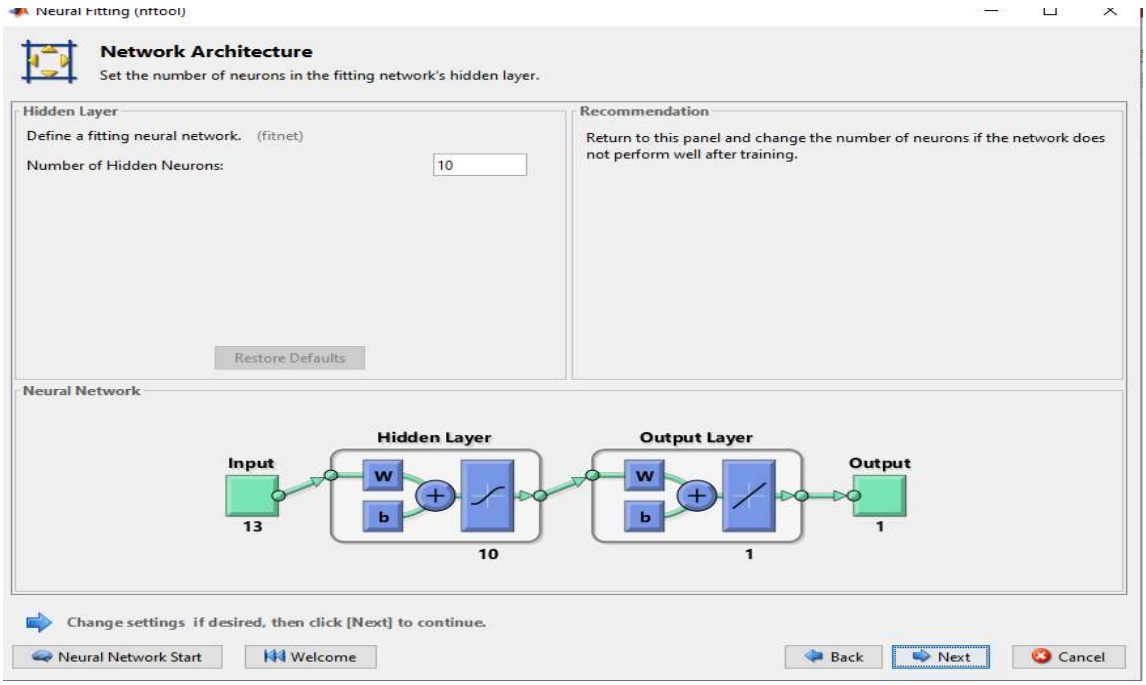

Figure 5: Showing the number of Hidden layers

Training the network, the pairs of input, output dataset is trained with the network created. Here, we created a two-layer feed-forward network, as shown in Figure 5 above. The network has ten (10) hidden layers with ten neurons.net = feedforwardnet(10); The network is trained for up to 1000 epochs to an error goal of 0.1 and then re-simulated.

net.trainParam.epochs = 1000;
net.trainParam.goal = 0.1;

To know when the training had converged, we set the parameter "**show**" before call the training function net.trainParam.show = 7;

In this case, the error value appeared on work space every "7" iterations like this:
TRAINB, Epoch 0/1000, MSE 0.5/0.1.
TRAINB, Epoch 7/1000, MSE 0.181122/0.1.
TRAINB, Epoch 14/1000, MSE 0.111233/0.1.
TRAINB, Epoch 21/1000, MSE 0.5189606/0.1.
TRAINB, Performance goal me

After training the network, we tested the performance on a test set. Figure 6 shows the training and testing section in the MATLAB environment.

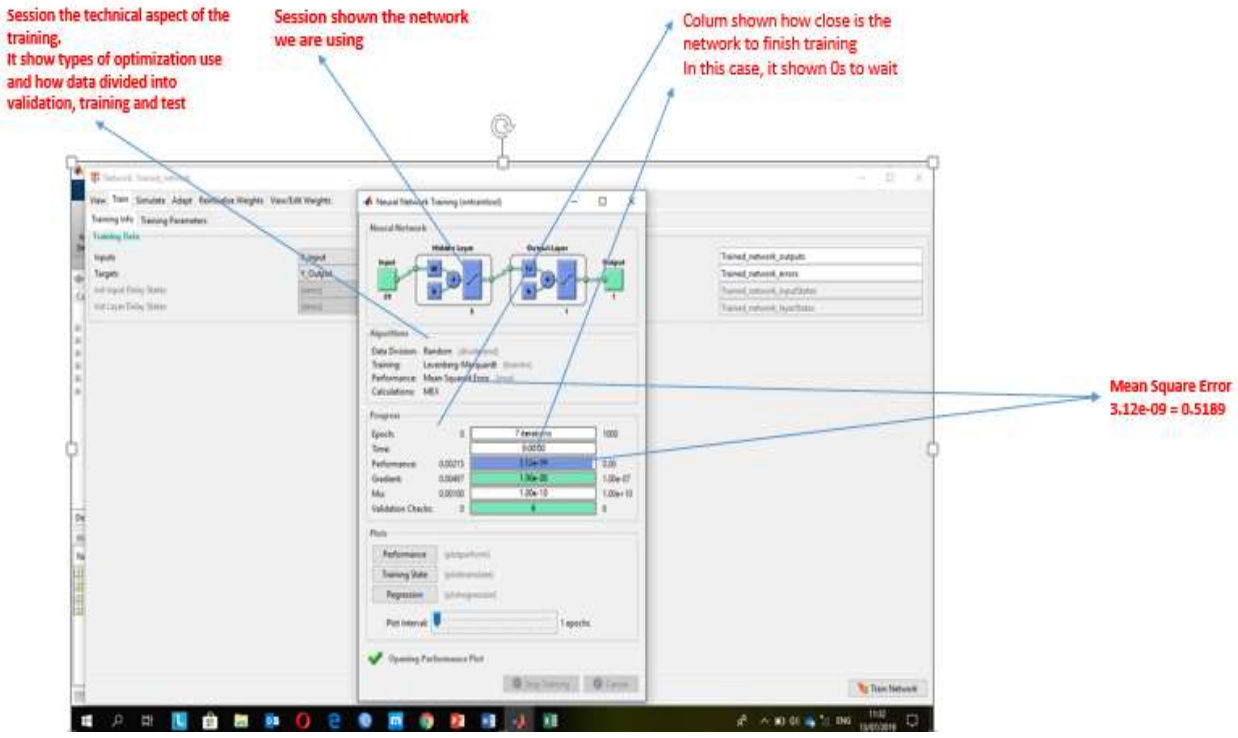

Figure 6: The Train and Test section

5. RESULTS AND DISCUSSION

The system was able to predict sample that are correctly classified and misclassified. The testing sample that are predicted YES and the actual output was also YES (true positive) is 8528e04 (92727) and the sample that are predicted NO and the output was NO (i.e. true negative) is 103. Figure 7 shows classification table where the diagonal element represents the testing sample that are correctly classified while the others diagonal element represents the testing sample that are misclassified. Since we had highly imbalance classes with less than 0.2% of fraudulent activities, the classification accuracy is extremely high, which is 0.9993 (99.9%). Figure 8 shows the result.

The Area under precision recall curve (AUPRC) of this model, as shown in Figure 9 is 0.5937, which indicates the need for improvement because a model with higher AUPRC indicate better performance, i.e. if AUPRC is equal to 1, it means the classification is perfect with 100% true positive rate and no false positive or true negative. The overall performance of system is 79% while the validation is 99.9% with training accuracy of 100% with target output, when the neural system hidden layer was adjusted to 10. The graphical result of the training and testing of the model is depicted in Figure 10, showing the training accuracy, validation and performance respectively.

The average Mean Squared Error (MSE) that is used as the loss function, that's the average squared difference between the estimated values and target is 0.378. Figure 11 indicates the MSE of the model. The best performance path is achieved when the validation is at 0.0031179 at first epoch, as shown in Figure 12.

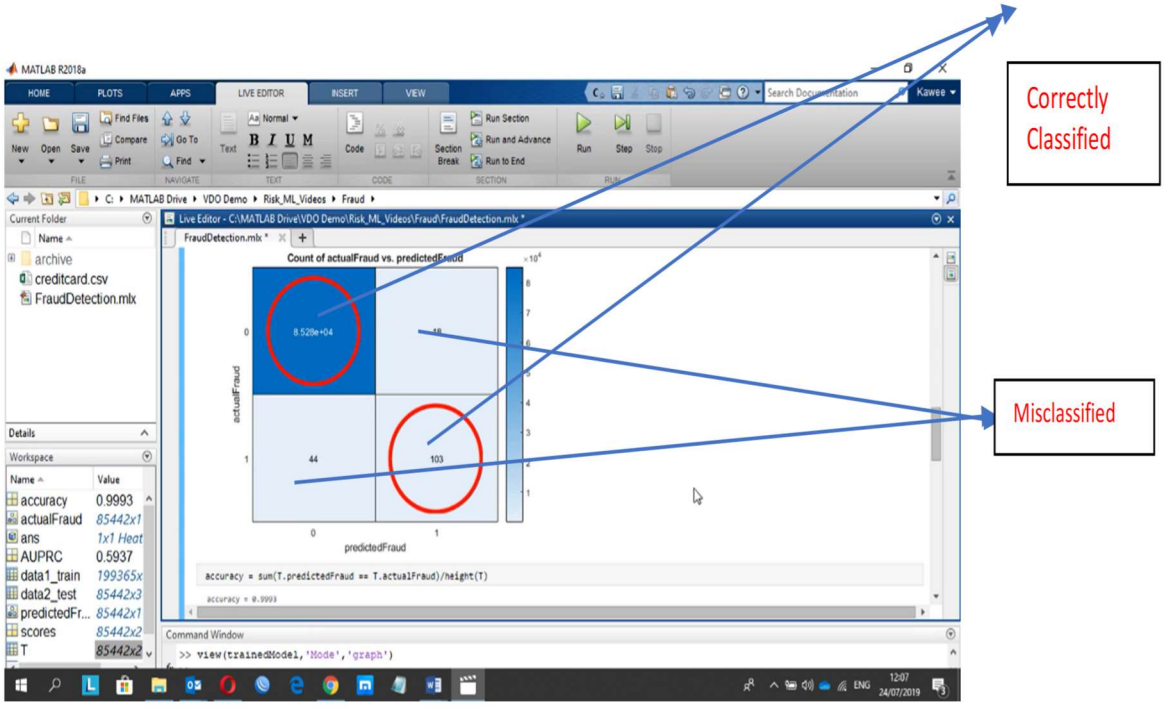

Figure 7: Classification table

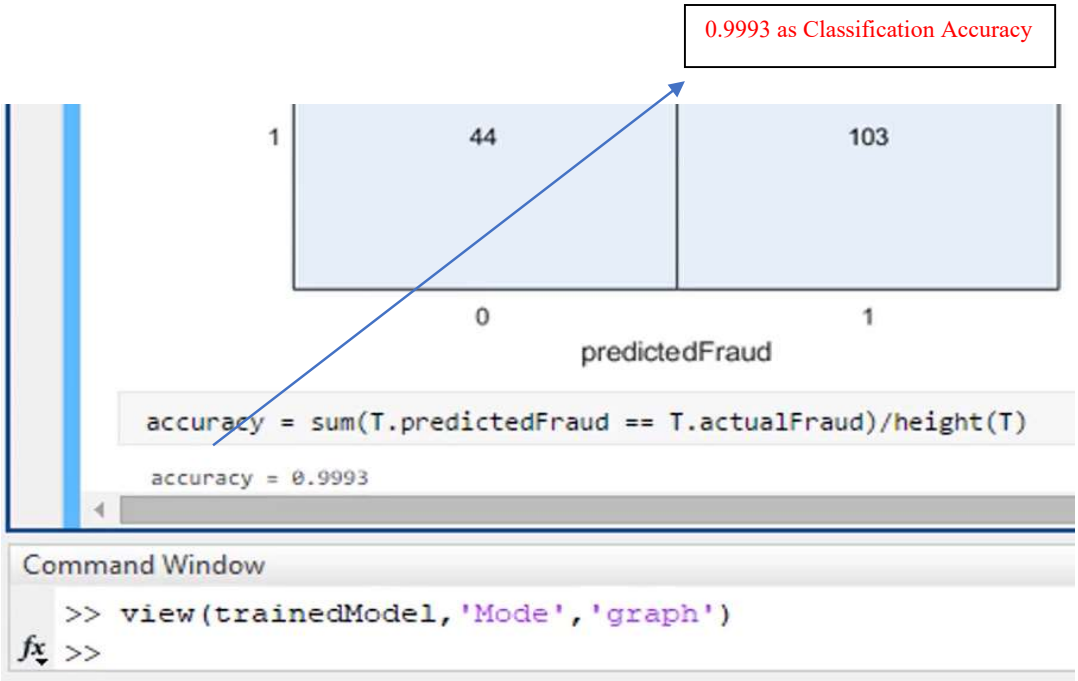

Figure 8: Classification Accuracy.

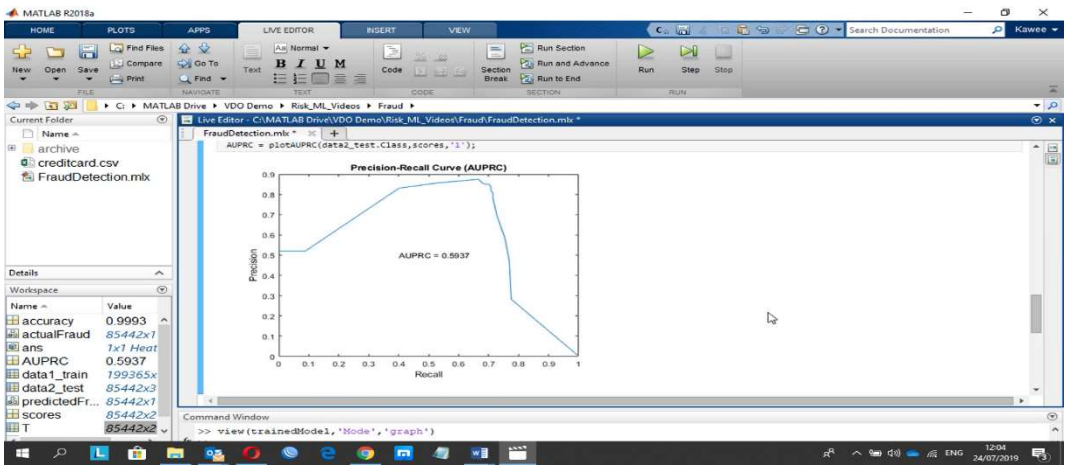

Figure 9: AUPRC Graph

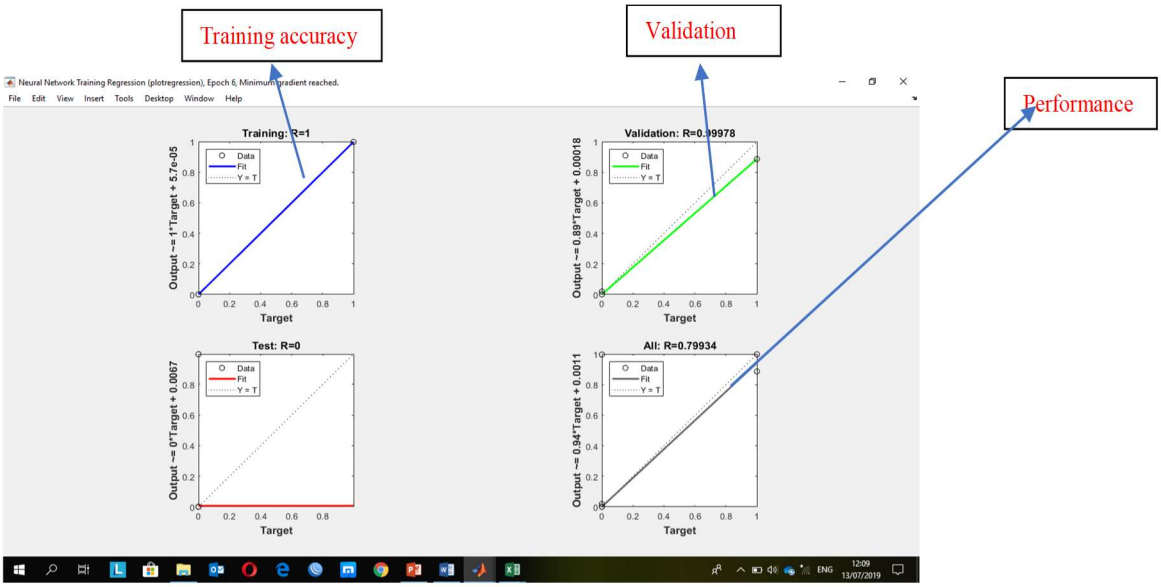

Figure 10: Graph Results of Train and Test

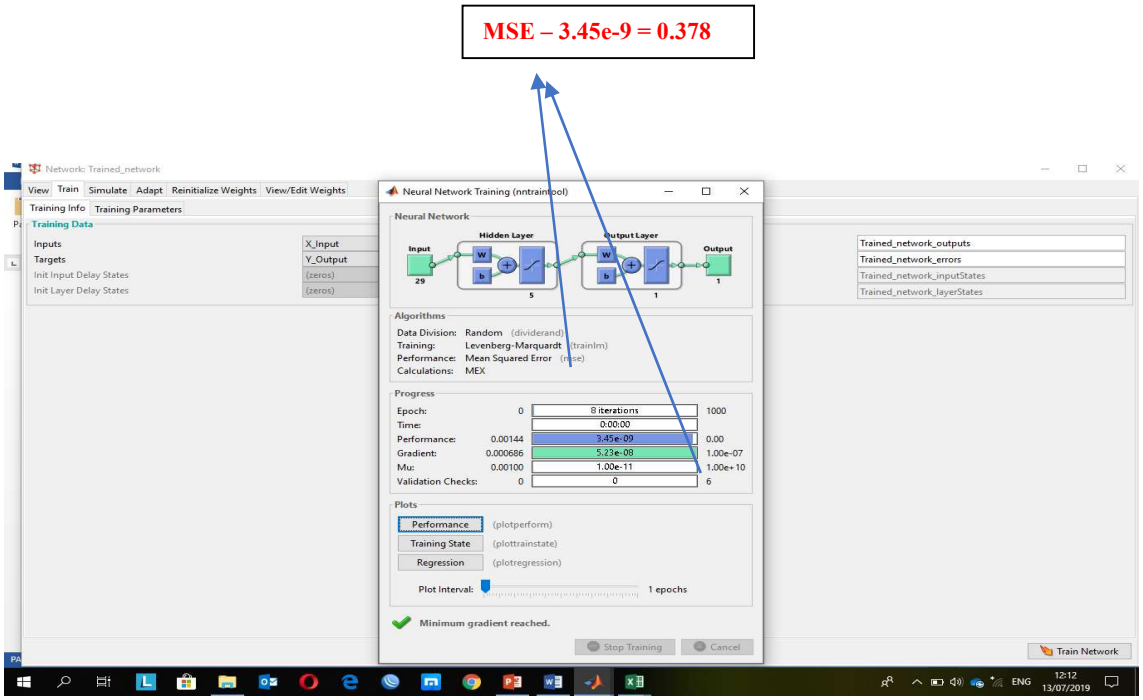

Figure 11: The Train and Test section

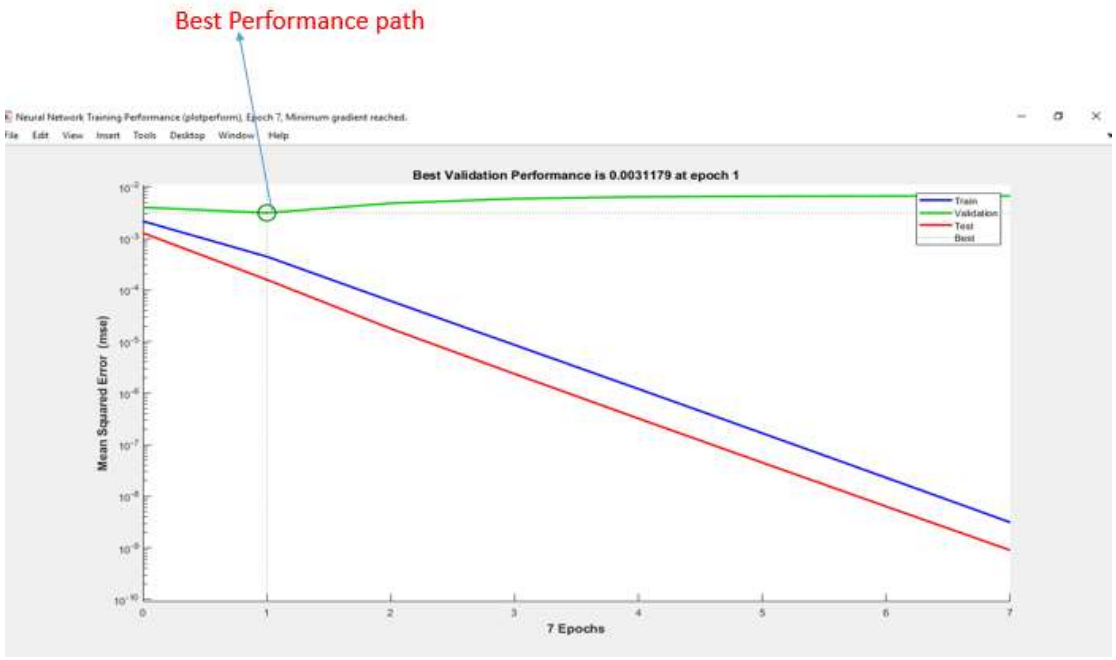

Figure 12: NN Training plot performance

Table 1: Performance Summary of the model

Evaluation Metrics	VALUE	Benchmark	Comments
AUPRC	0.5937	0.4	Need for improvement
Classifier Accuracy	0.9993	0.8	Best State because of less fraudulent activities
Prediction Accuracy	0.7993	0.8	Expected value because of less fraudulent activities
Mean Square Error	0.378	0.5	Need for improvement.
Training Accuracy	1	0.7	Effective Training model
Validation	0.999	0.7	Effective Training model
Correctly |Classified	27 363		
Incorrectly Classified	62		
Prediction Accuracy	79%	50%	Still Need more improvement

5.1 Statistical Summary

From the above evaluation (Table 1), the percentage accuracy of Feed forward BPNN model is 79.9%, and the AUPRC is 0.59, with MSE of 0.378. The number of correctly classified transactions and incorrectly classified transactions are 27363 and 62 respectively.

5.2 Findings

From the above Statistical analysis, we can conclude that feed forward BPNN is effective in the prediction or detection of fraudulent transactions. However, the value of the AUPRC indicates the need for improvement because a model with higher AUPRC indicate better performance.

6. CONCLUSION

This work has contributed to the body of knowledge by successfully demonstrating the effectiveness of BPNN as a machine learning technique for fraud detection in online credit card transactions through implementation with MATLAB. It can be concluded that the model developed can detect fraudulent transaction from any datasets it is subjected to. The model is of greater accuracy and has least tolerant for raising false alarms when compared to some existing work on other models. However, future work can be carried out using real datasets or by comparing the effect of combing the model with another optimization algorithm, as to optimize the system accuracy.

REFERENCES

1. Amir Hassan Monadjemi, Reza Ebrahimi Atani, Samaneh Sorournejad, Zahra Zojaji (2016). A survey of credit card fraud detection techniques: Data and technique oriented perspective. *https://www.researchgate.net/publication/310610856*
2. Aderowumu G. A., Adewale O.S., Alese B.K., Ismaila W.O., and Omidiora E.O. (2012), Investigating the effects of Threshold in Credit Card Fraud Detection System, *International Journal of Engineering and Technology. 2 (7), PP/328-1332*
3. Akshata Hadkar and Sheetal Yewale (2015), Online Credit Card Fraud Detection, *International Journal for Research in Engineering Application and Management (IJREAM), Vol.1, Issue 2.*
4. Antara Dey and R. Kavitha sudha (2018). Credit Card Fraud Detection Based on the Transaction by using Hidden Markov Model and PHP Software. *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 25 Issue 5.*
5. Avinash Ingole, Dr. R. C. Thool (2013), Credit Card Fraud Detection Using Hidden Markov Model and Its Performance, *International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 6, June 2013 ISSN: 2277 128X*
6. Arunabha Mukhopadhyay, Sayali Mukherjee, Ambuj Mahanti (2011), Artificial immune system for detecting online credit card frauds, *CSI communications*
7. Ayon Dey (2016), Machine Learning Algorithms: A Review, *International Journal of Computer Science and Information Technologies, Vol. 7 (3), 1174-1179.*
8. Dermala N. and Agrawal A.N. (2016), Credit card fraud detection using SVM and Reduction of false alarms, *International Journal of Innovations in Engineering and Technology (IJIET) 7(2)*.
9. Devaki R., Kathiresan V. and Gunasekaran S. (2014). Credit Card Fraud Detection using Time Series Analysis. *International Journal of Computer Applications (IJCA),* pp 8-10.
10. Dhanapal,R. and Gayathiri (2012). Credit Card Fraud Detection Using Decision Tree for Tracing Email and IP, *International Journal of Computer Science Issues (IJCSI) Vol. 9, Issue 5*
11. Falaki S.O., Alese B.K. and Ismaila W.O. (2010). An Update Research on Credit Card Online *Transactions. International Journal of Economic Development Research and Investment, 1(2,3), Pp 181-190.*
12. Khan M.Z., Pathan J.D., Ahmed A. H. E. (2014), Credit Card Fraud Detection System Using Hidden Markov Model and K-Clustering, *International Journal of Advanced Research in Computer and Communication Engineering,* 3(2), 5458-5461.
13. Krishna Kumar Tripathi and Lata Ragha, (2013), Hybrid Approach for Credit Card Fraud Detection, *International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-4.*
14. Krishna Kumar Tripathi, Mahesh A. Pavaskar (2012), Survey on Credit Card Fraud Detection Methods, *International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 11)*
15. Masoumeh Zareapoor, Seeja.K.R, and M.Afshar.Alam (2012), Analysis of Credit Card Fraud Detection Techniques: based on Certain Design Criteria, *International Journal of Computer Applications (0975 – 8887) Volume 52– No.3.*
16. Maira Anis (2015). A Comparative Study of Decision Tree Algorithms for Class Imbalanced Learning in Credit Card Fraud Detection. *International Journal of Economics, Commerce and Management Vol. III, Issue 12.*

17. Mehak Kamboj and Shankey Gupta (2016). Credit Card Fraud Detection and False Alarms Reduction using Support Vector Machines. *International Journal of Advance Research, Ideas and Innovations in Technology, Volume 2, Issue 4.*
18. Nabha Kshirsagar, Neha Pandey, Shraddha Kotkar, Suja S. Panicker and Amol Pate (2015). Credit Card Fraud Detection System using Hidden Markov Model and Adaptive Communal Detection. *International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 6 (2), 1795-1797.*
19. Navanshu Khare and Saad Yunus Sait (2018), Credit Card Fraud Detection using Machine Learning models and collating Machine Learning models. *International Journal of Pure and Applied Mathematics, ISSN: 1314-3395, pp. 825-838*
20. Neha Sethi and Anju Gera (2014), A Revived Survey of Various Credit Card Fraud Detection Technique, *International Journal of Computer Science and Mobile Computing, Vol.3 Issue.4, pg. 780-791*
21. Patel S. and Gond S. (2014), Supervised Machine (SVM) Learning for Credit Card Fraud Detection. *International Journal of Engineering Trends and Technology (IJETT) 8(3), pp 137-139.*
22. Pooja Chougule, A.D. Thakare, Prajakta Kale, Madhura Gole, Priyanka Nanekar (2015), Genetic Kmeans Algorithm for Credit Card Fraud Detection, *International Journal of Computer Sciemce and Informaion Technologies, Vol. 6 (2), 1724-1727.*
23. Rajamani R. and M.Rathika (2015) Credit Card Fraud Detection using Hidden Morkov Model and Neural Networks. *International Journal of Advanced Networking and Applications (IJANA).*
24. Raghavendra Patidar and Lokesh Sharma (2011), "Credit Card Fraud Detection Using Neural Network, *International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, 2011.*
25. Rinky D. Patel and Dheeraj Kumar Singh (2013). Credit Card Fraud Detection & Prevention of Fraud Using Genetic Algorithm. *International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6.*
26. S.Fashoto, O.Adeleye and J.Wandera (2016), Hybrid Methods for credit card fraud detection,*Available:http://www.journalrepository.org/media/journals/BJAST_5/2015/Dec/Fashoto1352015BJAST21603.pdf.*
27. Shailesh S. Dhok and Dr. G. R. Bamnote (2012). Credit Card Fraud Detection Using Hidden Markov Model. *International Journal of Advanced Research in Computer Science, Vol. 3, No. 3.*
28. Shipra Rathore and Megha Jain (2016). A Hybrid Technique for Credit Card Fraud Detection. *Communications on Applied Electronics (CAE) – ISSN: 2394-4714 Foundation of Computer Science FCS, New York, USA Volume 5 – No.5.*
29. Snehal Patil, Harshada Somavanshi, Jyoti Gaikwad, Amruta Deshmane, Rinku Badgujar (2015). Credit Card Fraud Detection Using Decision Tree Induction Algorithm. *International Journal of Computer Science and Mobile Computing, Vol.4 Issue.4.*
30. Suvasini Panigrahi and Tanmay kumar Behera (2015), Credit Card Fraud Detection: A hybrid approach using Fuzzy Clustering and Neural Network," *international Conference on advances in Ccomputing and Communication Engineering*.