International Journal of Advanced Engineering and Management Research

Vol. 10, No. 06; 2025

ISSN: 2456-3676

Credit Card Fraud Detection Model Using Deep Learning

Seun Mayowa Sunday

ADEY Innovations Limited,

Stone House, GL10 3EZ, Gloucester, United Kingdom

doi.org/10.51505/ijaemr.2025.1523 URL: http://dx.doi.org/10.51505/ijaemr.2025.1523

Received: Nov 22, 2025 Accepted: Dec 02, 2025 Online Published: Dec 15, 2025

Abstract

The ever-evolving state of technology is responsible for a wide variety of advances. In this article, we discuss online transactions that are conducted using credit cards, which might result in credit card fraud. The purpose of this research is to discuss the development of a system that can identify credit card fraud by making use of deep learning neural networks (DNN).

Because of the skewness in the dataset, even if the Neural Network (NN) is trained using a huge number of iterations, it will not be accurate enough to determine whether or not the data represents fraudulent or non-fraudulent activity. This study makes use of a variety of pre-processing methods for the data, such as the elimination of null entries, the correction of imbalanced data, and others. After that, the author proceed to the methods for feature selection, which is later followed by feature engineering in order to carry out normalisation prior to the development of the model. The results that were obtained indicate that the use of these methods improves the accuracy of the prediction process, which is then evaluated in relation to the work that has been done previously in the same field.

Keywords: DNN, credit card, NN

1.0 Introduction

1.1 Research Background

In today's society, artificial intelligence (AI) has been integrated into almost every aspect of life as its applications are widespread. It gives people control over what they see on social media newsfeeds, enables facial recognition (to unlock smartphones), and even recommends music based on past playlists. Among the subsets of AI, deep learning is becoming more integrated into everyday life.

As a branch of artificial intelligence, deep learning involves machine learning and adapting to an algorithm through experience instead of requiring extensive programming. The goal of deep learning is to develop computer programs that are capable of gaining new knowledge from a set of data, by analysing historical data and behaviours, it can predict patterns and make decisions. A

Vol. 10, No. 06; 2025

ISSN: 2456-3676

significant reason for its application in the finance industry is the enormous volume of historical financial data generated there (Shulman, 2012). Furthermore, the application excels at handling large and complex volumes of data.

The Banking sector is a major part of the finance industry, which in most countries is regulated, as it is an important factor in determining the country's financial stability. One facility provided by the Banking Regulation Act is the provision for the public to obtain a credit card for both online and offline transactions (Jency et al, 2018). The immense significance of credit cards to any individual, business or enterprise cannot be overemphasized as it plays a key role in the dynamics of any economy (Seyed & Hashemi, 2010). Credit cards offer a line of credit that can be used for transactions not only in the emerging economies but also in the developed capital markets, whether by individuals or enterprises. The economic growth of the real economy is the primary role of the financial firms and with the great importance and benefits attached to financial lending, also come some major issues and bottleneck problems (Charleonnan, 2016).

The most common and substantial issue in the domain of finance is the fair and successful issuance of credit cards to the end-user with an adequate record of all the available transactions, hence making it easy to generate the necessary dataset. In a financial transaction, the risk of a credit card fraudulent transaction may not be eliminated but can be controlled (Dunn, 2017). The use of credit cards can be profitable if credit risk management and control are done in a precise manner (CyberSource, 2017). People apply for and use credit cards daily for a variety of purposes, yet for security, almost everyone relies heavily on the issuer of the card (CyberSource, 2017). Hence, the primary objective of the bank is to provide the wealth for safer hands to avoid making huge financial losses and as such, banks only approve credit cards after verifying and validating documents provided by the customer to check if an applicant is deserving or not. Although it is no complete guarantee of a customer's credit card worthiness, it is not disputing that it reduces the chances of credit card default (Niimi, 2017).

The search for how to reduce this trend of fraudulent transactions in the financial industry has led issuers into finding ways of evaluating the transaction ability of the credit cards and the associated risks of issuance (Jency et al, 2018). Cardholders and card issuers can be assured that all transactions are safe when using their cards. Financial institutions and cardholders, on the other hand, are often tricked into believing fraudsters' illegal transactions are legitimate. It is also a fact that certain fraudulent transactions are conducted regularly without the awareness of card issuers and cardholders for the purpose of gaining financial gain. Credit card transactions are notorious for being blind to fraudulent activity by both authorised institutions and cardholders. As a result, detecting fraudulent activity among millions of actual transactions, particularly when fraudulent transactions are smaller than genuine transactions, can be challenging (Makki et al., 2022). This research attempts to apply AI to a real-world problem by using a deep learning approach as a tool to make credit card predictions providing another useful and better metric for evaluating the credibility of a credit transaction.

Vol. 10, No. 06; 2025

ISSN: 2456-3676

1.2 Statement of the Problem

With credit cards being one of the major banking products, banks are seeking ways to convince customers to use the service confidently. However, there have been several financial attacks because of this service, which leaves some customers to refrain from using this service.

1.3 Research Questions

Although credit cards are used for online transactions, which is a very dangerous domain for fraudsters.
How are financial institutions improving the security of these cards and transactions?
What is the state of credit card fraud detection?

1.4 Aims and Objectives of the Study

This project aims to design and develop a predictive financial credit-card defaulters system through the following objectives.

I. To perform careful exploratory data analysis (EDA) to understand the dataset.
II. To conduct feature engineering to tackle the imbalanced dataset.
III. To develop a credit card fraud detection model using deep learning techniques.

1.5 Scope of the Study

The scope of this research is to understand the concept of a deep learning algorithm by applying it to a credit card transaction dataset to predict fraudulent transactions.

1.6 Significance of the Study

The detection of credit card fraud is essential to good decision-making and transparency, as such a deep-learning model can provide much better insights than can be obtained from a human analyst alone. If a model can identify credit card fraudulent transactions which traditional detection does not normally recognise, then the chances of high confidence in the online transaction will increase, thereby increasing the number of people who will continue to use a credit card for digital transactions. Additionally, a lucrative niche market or micro-market would have been created, increasing the financial institution's profit margin.

1.7 Limitations of Study

Although there are several datasets, which are available for academic research in this domain, the dataset created by the Machine Learning Group (2018), which has gained important recognition in academics, is used in this research. Another limitation is that this research only develops the model. However, this paper does not cover integration into end-user devices such as mobile or web apps.

Vol. 10, No. 06; 2025

ISSN: 2456-3676

1.8 Definition of Technical Terms

Credit card: A credit card is a payment card offered to cardholders to enable them to pay for products and services depending on their accumulated debt.
Credit Fraud: Malicious activity on the credit card for financial or another gain.
Machine Learning: This is the application of algorithms and statistical models to help computer systems automatically solve specific problems.

Deep Learning: This can simply be explained as neuroscience inside the computer. The research tries to mimic how the human brain works on the computer.

2.0 Method

Deep learning has proved to be an important technique for discovering hidden patterns in different industries. However, for a model to be effective, different data pre-processing methodologies are of utmost importance. This section will begin with data pre-processing, then the feature selection, which helps to choose the best feature among the dataset, followed by the feature engineering, which attempts to normalize the dataset before the model development.

Data preparation in deep learning refers to the technique of preparing (cleaning and organising) the raw data so that it may be used to construct and train deep learning models (Kiril, 2022). At this stage, features of the dataset are checked if they contain null values, missing values, etc. This research utilizes a state-of-the-art dataset created by Pozzolo et al. (2015) and modified by the machine learning group (UCI, 2018).

The dataset contains 31 features, which are labelled *Time*, *v# (v1-v28)*, *Amount*, and *Class* (Table 2.1).

Table 2.1 Feature Description of the dataset

Feature	Meaning
Time	The number of seconds that passed between this transaction and the dataset's initial transaction.
V#	to secure user identities and sensitive features, a PCA Dimensionality reduction may be the outcome (v1-v28)
Amount	Amount of the occurred transaction
Class	Classification into fraudulent or not non-fraudulent

The phases of building a model are described as follows.

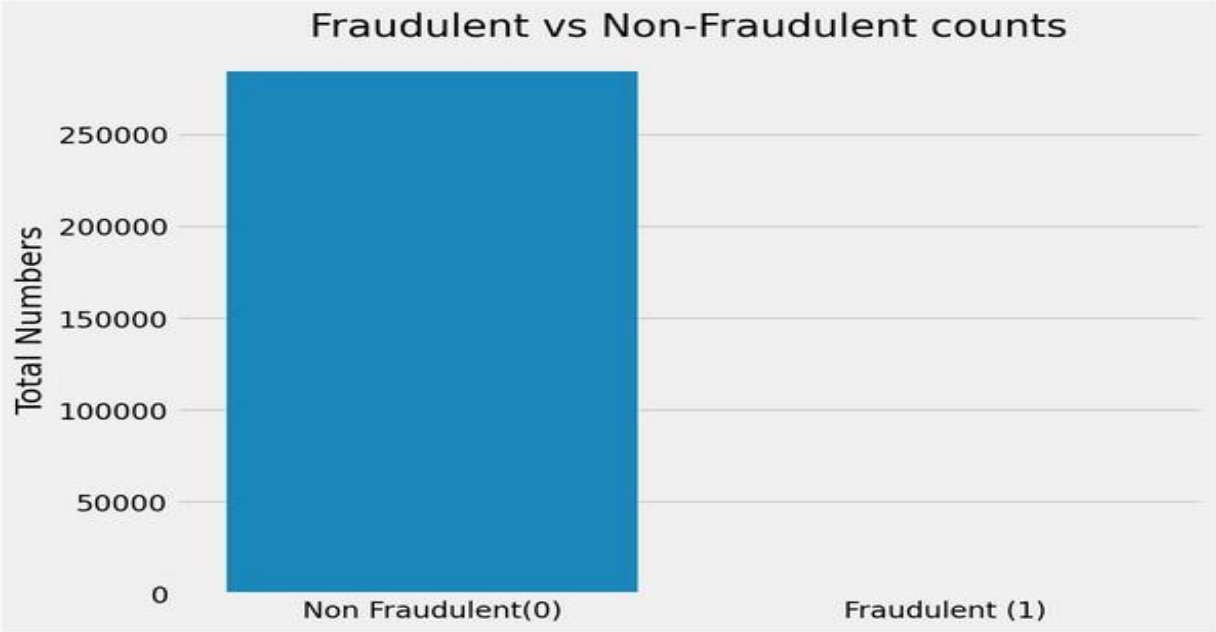
2.1 Data Pre-Processing

Although data pre-processing and exploratory data analysis (EDA) are sometimes used interchangeably, research as proved that data pre-processing and EDA are all different, though

Vol. 10, No. 06; 2025

ISSN: 2456-3676

they still have some overlapping tasks (Leah Nguyen, 2021). To prepare a deep learning model for good performance, there is the need to prepare the dataset in such a way that it can easily be used by the model. This section understands the dependent variable better.




```
Fraudulent data = 492
Non-Fraudulent data = 284315
Ratio of Fraudulent to non-Fraudulent transactions = 0.17%
```

Fig 2.1a Imbalanced dataset

Figure 2.1a shows that the variable is imbalanced, as there are 280,000 legitimate transactions and fewer than 500 fraudulent transactions. This is a normal expectation as most of the transactions are expected to be legit, and a minimal amount is fraudulent. As shown in Figure 2.1a above, there is a need to make sure that the variables are of equal size. Although there are different techniques that can be used to achieve this, the over-sampling technique (Roweida *et al.*, 2020) is used in this research.

Vol. 10, No. 06; 2025

ISSN: 2456-3676

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
```

Fig. 2.1b Entry Counts

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267

5 rows × 31 columns

Fig. 2.1c entries and features of dataset

```
<AxesSubplot:title={'center':'Dataset Features with Null Value'}>
```

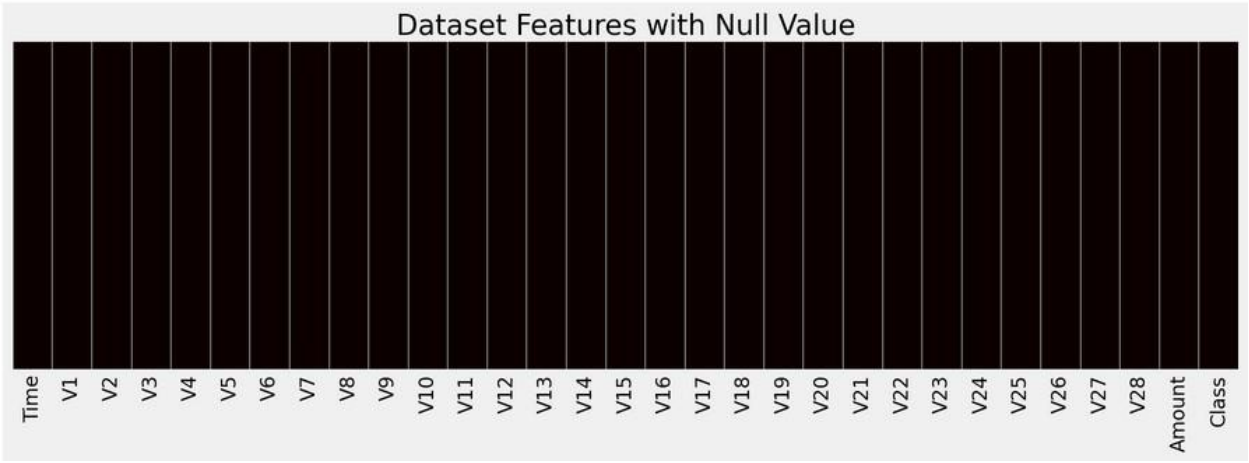


Fig 2.1d dataset null value

Vol. 10, No. 06; 2025

ISSN: 2456-3676

```
Index([], dtype='object')
```

Fig 2.1e features with categorical variable

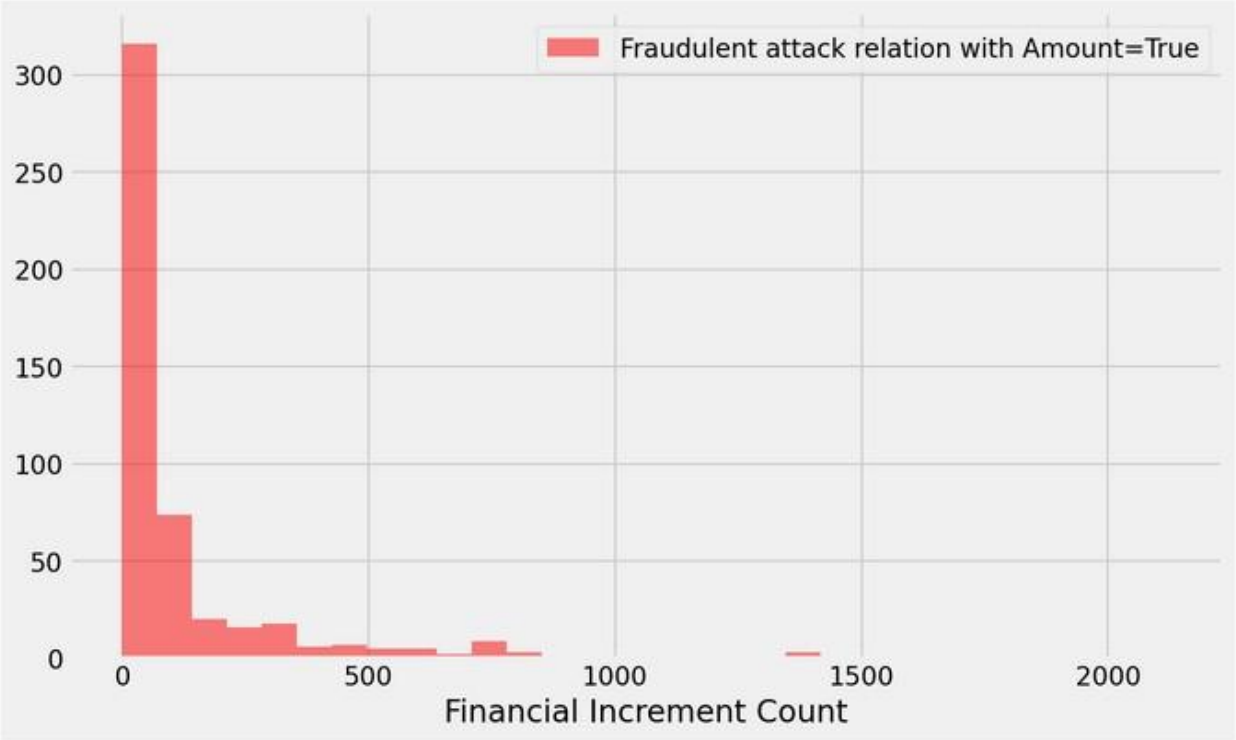


Fig 2.1f Fraudulent Transaction in relation to money

Vol. 10, No. 06; 2025

ISSN: 2456-3676

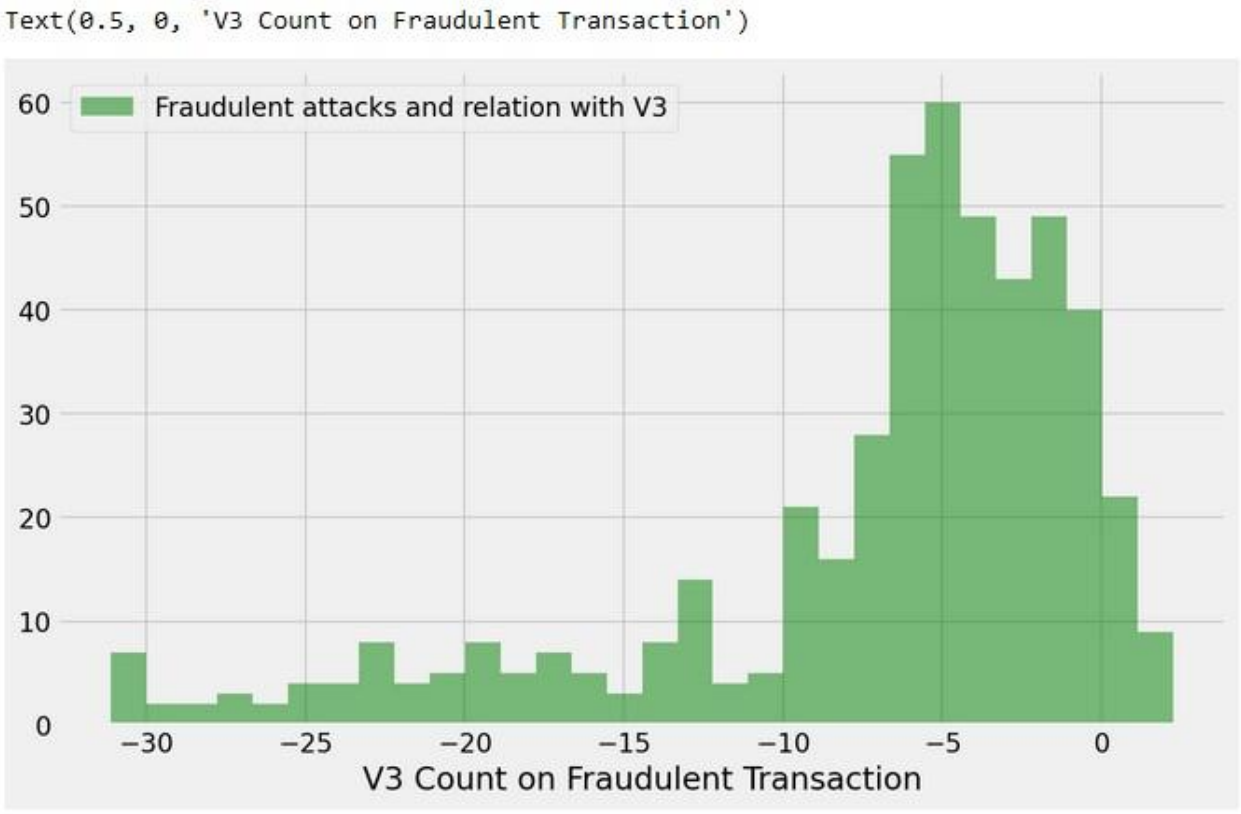


Fig 2.1g fraudulent transaction about anonymous V3

The dataset have 284807 entries and 31 features (columns) as illustrated in figure 2.1b and figure 2.1c respectively. Figure 1.1d shows that the dataset does not have any null values, whereas Figure 2.1e shows that all the variables are numerical. Figure 2.1f shows that most of the fraudulent transaction occurs by stealing very small amounts of money, which makes sense since the higher the stolen amount, the higher the chance of being noticed. Similarly, the anonymous information (V3) shows a high level of occurrence with the fraudulent transactions (Figure 2.1g).

2.2 Feature Selection

In feature selection, the features are manually or automatically selected to the ones relevant to the dependent variable. This step is conducted because it has a significant impact on the model's performance in terms of both build time and accuracy (Verónica et *al.*, 2013). Irrelevant dataset features can have a negative impact on training because they force the model to learn on data that has no influence on the prediction output. The effect of poor feature selection can be seen from the accuracy of the report, as poor features will yield unrealistic accuracy, because irrelevant data acts as noise (Verónica *et.al.*, 2013). Feature selection offers numerous benefits, including reduced overfitting, improved accuracy, and shorter training times. By utilizing feature selection,

Vol. 10, No. 06; 2025

ISSN: 2456-3676

the dataset's dimensions can be reduced (Jundong et *al*., 2018). The next sections explains the two feature selection methods used in this research.

2.2.1 Manual Feature Selection

From the provided dataset, *time* is a feature that is common to both fraudulent and non-fraudulent transactions; hence, it is removed. This makes the first feature drop to 30, as shown in Figure 2.2 below.

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...	V21	V22	V23	V24
0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	...	-0.018307	0.277838	-0.110474	0.066928
1	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	...	-0.225775	-0.638672	0.101288	-0.339846
2	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	...	0.247998	0.771679	0.909412	-0.689281
3	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	...	-0.108300	0.005274	-0.190321	-1.175575
4	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	...	-0.009431	0.798278	-0.137458	0.141267

5 rows × 30 columns

Figure 2.2 Number of used features after dropping the feature *time*.

2.2.2 Automatic Feature Selection

The mutual information classification algorithm was utilised in this analysis (Jorge & Pablo, 2013). Using this algorithm, the relationship between dependent variable and independent variables are better understood. Figure 2.1 shows the result of the mutual information techniques.

Fig. 2.1a Feature selection (mutual information technique)

From Figure 2.2a above, the small importance (0.002 below) of our dependent variable dropped. The total features used in this research was later dropped to 34 features as shown in Figure 2.2b.

Vol. 10, No. 06; 2025

ISSN: 2456-3676

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...	V12	V14	V16	V17
0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	...	-0.617801	-0.311169	-0.470401	0.207971
1	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	...	1.065235	-0.143772	0.463917	-0.114805
2	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	...	0.066084	-0.165946	-2.890083	1.109969
3	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	...	0.178228	-0.287924	-1.059647	-0.684093
4	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	...	0.538196	-1.119670	-0.451449	-0.237033

5 rows × 21 columns

Fig. 2.2b Used features after mutual information techniques

2.3 Feature Engineering

Since the dependent variables are not balanced (Figure 2.1a), the oversampling technique is used to balance the data (Mohammed *et.al*., 2020). Oversample techniques work by randomly adjusting the deficient classes (fraudulent transactions) to match the level of the leading class (non-fraudulent transactions). The next process in feature engineering is the application of normalization techniques (Ali et al., 2014). To allow fairness and a better chance of prediction equally, the normalization allows the dataset to be within a stipulated closed range within a unit, such as $[0,1], [-1,1]$ etc. This research uses $[-1,1]$ range.

2.3.0 Model Development

This section discusses the systematic development of the proposed deep neural network (DNN) model.

2.3.1 Perceptron Model

If the whole idea of deep learning is to have computers artificially mimic biological natural intelligence, then it is expected to generally build a general understanding of how biological neurons work. Figure 2.2 shows the biological structure of data processing. Information comes from different sources (dendrites) and the processed at the nucleus, which is then interpreted at the axon.

Vol. 10, No. 06; 2025

ISSN: 2456-3676

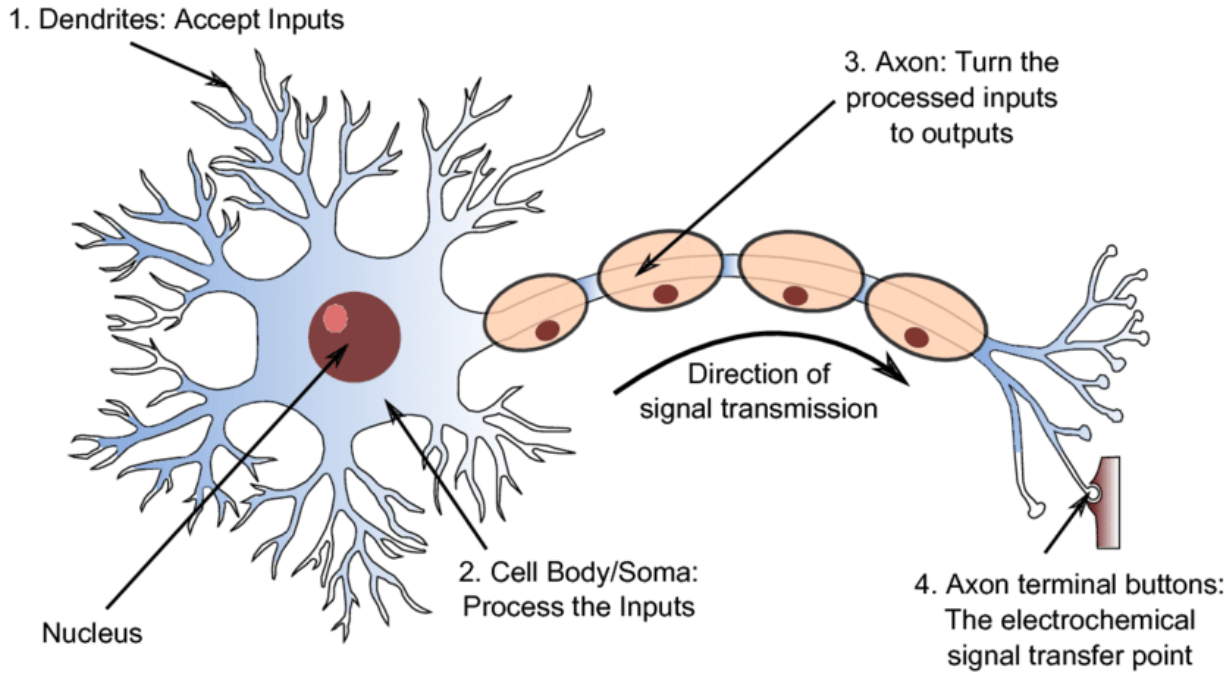


Figure 2.2 Simplified Biological Neural Model (Stevenson, 2014)

Conversion of biological neural model into the mathematical inputs is termed perceptron (Tran et al., 2020).

2.3.1.1 Conversion of Biological Model into Perceptron

Figure 2.3 shows the conversion of the biological neural network into a simple perceptron. From the Figure, it is obvious that the $f(x)$ performs some computation on the input ($x1$,$x2$) then output the result(y). If the f(x) performs simple multiplication on the input ($x1$,$x2$), then the output $(y)$is calculated to be$x1 + x2$. However, it is realistic to attempt to adjust some parameters so that the output can be improved. This is done by the weights ($w1$,$w2$). Then the total sum is $x1w1 + x2w2$. The basic summary of the perceptron model works in such a way that the perceptron model must be adjusted$w1w2$ in other to get the expected value of y.

Vol. 10, No. 06; 2025

ISSN: 2456-3676

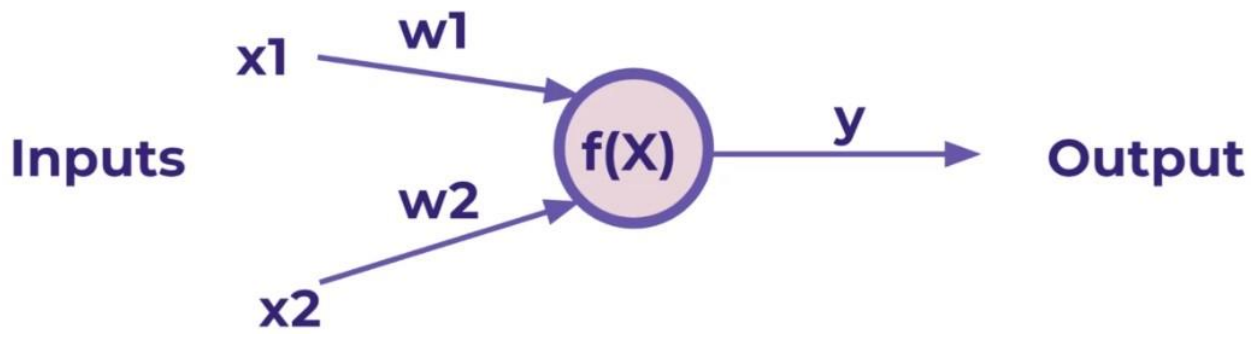


Figure 2.3 Simple Perceptron (Author)

However, from the Figure 2.3, when $X1$ or $X2$ is zero, then that means the multiplicative part will be zero. Adding bias term (*b*) to the input fixes the issue with zero value (Figure 2.4).

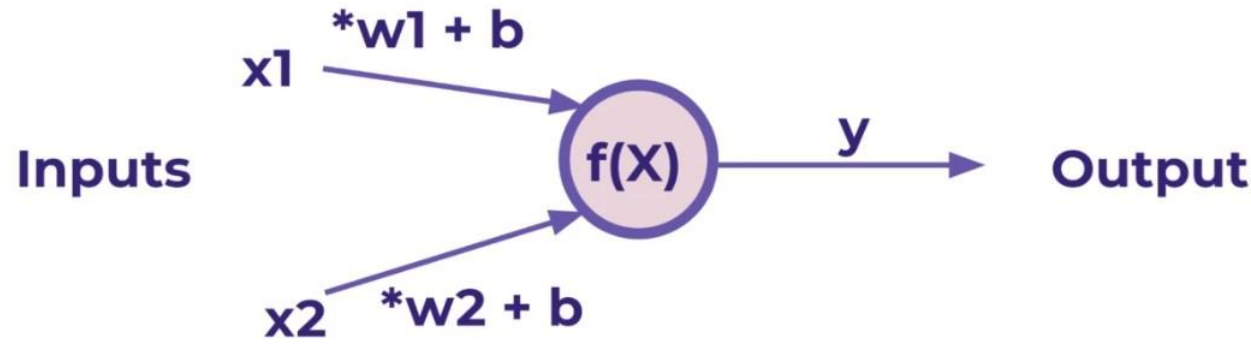


Figure 2.4 Adding biases to the weight (Author)

The bias will make the output of y to be as shown in Equation 2.1.

$$y = (x1w1 + b) + (x2w2 + b) \quad \text{Eqn. 2.1}$$

In general, the perceptron is represented mathematically by Equation 2.2 below.

$$y = \sum_{i=1}^{n} XiWi + bi \quad \text{Eqn. 2.2}$$

2.3.2 Spanning Perceptron into Neural Network

The idea that builds on the multi-layer perceptron, which is best described in Section 2.5.1, is known as a simple neural network (Cheng et al., 2019). This works by taking the vertical layer of the neuron, which is a single perceptron, and then taking its output, feeding it to the next layer of the perceptron. This makes the output of the previous layer an input of the next layer. Figure 2.5 shows that every neuron in the one layer is connected to the next layer. This is an example of a

Vol. 10, No. 06; 2025

ISSN: 2456-3676

feed-forward layer, which means that all the information originates from the input layer and is transmitted to the output layer. The first layer is the input layer, the last layer is the output layer, while the other layers in between are the hidden layer. A deep neural network is characterized by having two or more hidden layers.

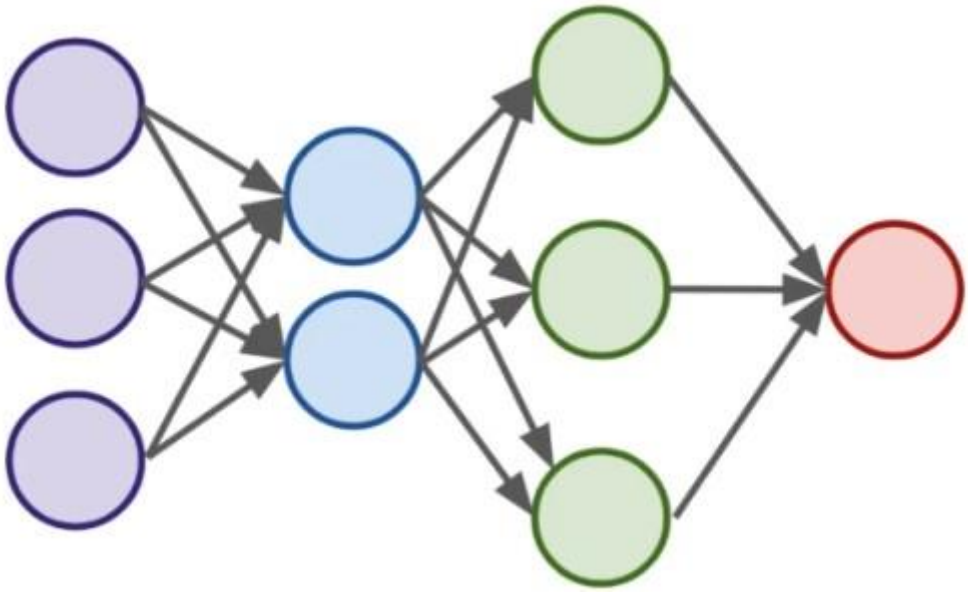


Figure 2.5 Multi-layer neuron

2.3.3 Activation Function

From section 1 above, w implies how much weight an input should have and the bias (b) implies an offset value which makes the $x * w$ reach a certain threshold before having an effect.

This is used to set output boundaries to the neuron (Mishra & Dash, 2014). Although there are several activation functions, the rectified linear unit (ReLU) function, which is graphically represented in Figure 2.6 and mathematically represented in Equation 2.3 below, is used in this research.

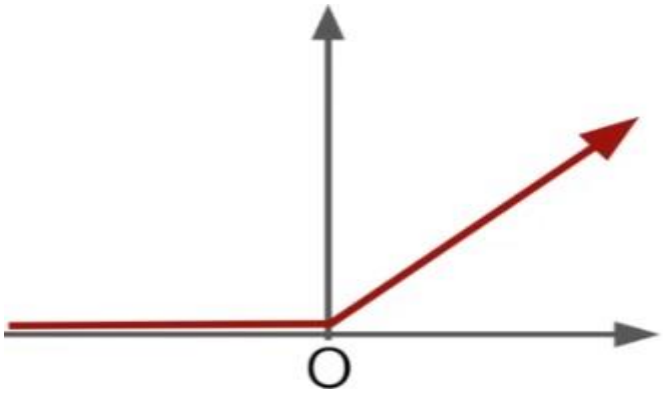


Figure 2.6 ReLU Function

Vol. 10, No. 06; 2025

ISSN: 2456-3676

$$f(z) = \begin{cases} 0 \; for \; z < 0 \\ z \; for \; z \geq 0 \end{cases}$$ equation 2.3

Where $z = wx + b$ equation 2.4

The ReLU function works in a way that anything beyond 0 is approximated to be 0 and any value above 0 is approximated to be$z$. Since the proposed research involves a binary classification, this activation function is a perfect match.

ReLU has been found to have very good performance, especially when addressing the issue of vanishing gradients (Tran et al., 2020). Hence, in this research, ReLU is used due to its overall good performance on the binary classification.

2.3.4 Cost Function and the Gradient Descent

Cost function helps to understand how close the output is to the expected value (Schmidhuber, 2015). Gradient descent simply helps to minimise the cost or error of an output. It is clear that a neural network simply takes inputs, multiplies them by weights, and adds the biases to these weighted inputs. Then, the result is passed through the activation function, which eventually yields an output (Dhankhad, Far, & Mohammed, 2018). The question of how to evaluate the network after the creation and how the evaluation network weights and biases is updated is another question. Section 2.5.5 explains how biases is updated through back propagation (Cheng et.al, 2018).

To update the weight of the network, the estimated value of the network is compared with the actual values of the label. This uses training dataset fitting (or training) of the model. However, during the test set evaluation, the update of the train set is performed. For each epoch (iteration), the cost function (or loss function) is used to monitor network performance (Portilla, 2022).
In mathematical terms,

$$\begin{cases} if \; y = true \; value \\ if \; a = neuron's \; prediction \\ then, w * x + b = z \\ so \; that \; activation \; function \; \sigma \; (z) = a \end{cases}$$ Equation 2.5

Vol. 10, No. 06; 2025

ISSN: 2456-3676

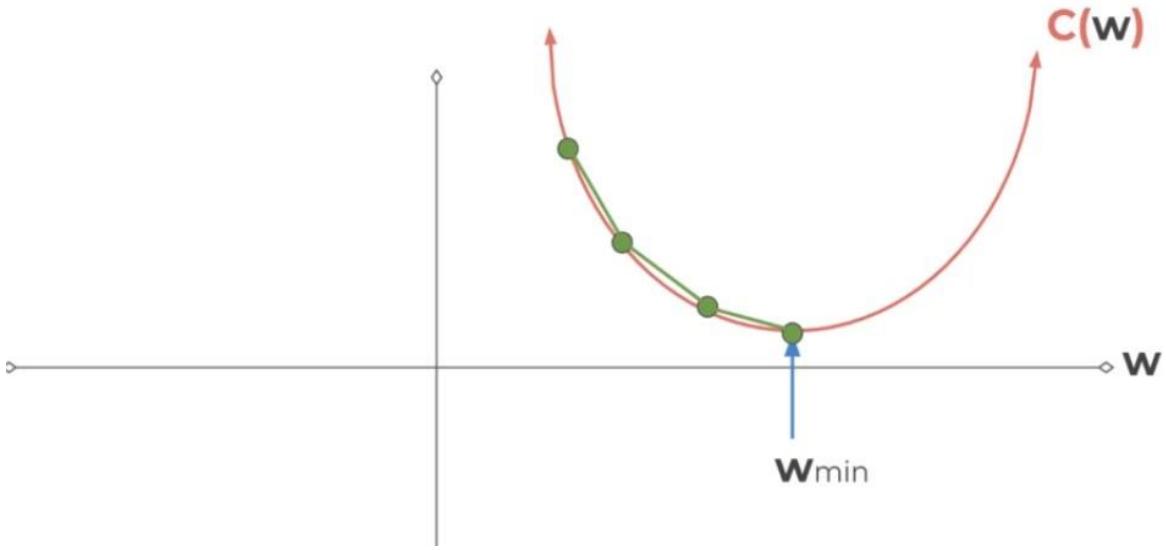


Figure 2.7 Cost Function, Step Sizes and the Learning Rate (Portilla, 2022)

The goal is to figure out what value of w minimizes the cost function, which is $Wmin$ as shown in Figure 2.7. Gradient descent helps to get the $Wmin$ As this will pick the slope at one point and then continue to move in the downward direction of the slope, as shown in Figure 2.7.

It is possible to use a constant learning rate, as shown in Figure 2.6 (green lines); however, adaptive gradient descent, which allows the learning rate to change as it approaches zero, is used in this research. The *Adam optimisation techniques, presented by Kingma & Ba (2015), are used in this research since they search for the optimal* learning rate.

It is paramount to understand that once the cost/loss value is derived, there is a need to adjust the weights and biases of the network. This is achieved using backpropagation techniques, which are discussed in Section 2.5.5.

2.3.5 Back propagation

It is fundamental to understand how the cost function results change in relation to the weights in the network; this is important for minimizing the weight of the cost function. The process of updating the weights of the network is known as backpropagation, with the goal of finding the weights that minimize the network error (Ratan, 2021). During the process for each input to output (feedforward pass), the network error is calculated, and then the error is used to slightly adjust the weight in the correct direction, thereby reducing the error slightly. This is done continually to ensure that the error is sufficiently small (Dhankhad, Far, & Mohammed, 2018; Ratan, 2021).

Vol. 10, No. 06; 2025

ISSN: 2456-3676

Depending on the nature of the weight, it can be reduced or decreased to minimize the error. For a single weight$Wnew$, which needs to be optimised depending on the previous weight$Wprevious$, with error correction$E$, then.

$$Wnew = Wprevious + \alpha\left(-\frac{\partial E}{\partial w}\right)$$ Equation 2.6

Where; α = learning rate

$\frac{\partial E}{\partial w}$ = partial derivative of error with respect to weight.

Since the error is the function of many variables, partial derivatives allow the measurement of how the error is impacted in each weight separately. Since this research is based on several perceptrons (section 2.5.2), many weights determine the network outputs, then a vector of partial derivative errors will result in handling errors from many weights, as shown in Equation 2.7.

$$Wnew = Wprevious + \alpha\nabla W(-E)$$ Equation 2.7

In summary, back propagation is simply the calculation of error (E) with respect to weight (W) in a particular location, then adjusting the weight (W) according to the calculated value of ∇W. The above calculation is done for each layer.

2.3.6 Measuring Metrics

This research utilizes Accuracy, Mean Squared Error (MSE), and Mean Absolute Error (MAE) as the measurement metrics (Frank, 2018).

To explain the above terms, the important terminologies are.
True Positive (TP): The actual value is positive, and the model predicts it to be positive
True Negative (TN): The actual value is negative, and the model predicts it to be negative
False Negative (FN): The actual value is positive, but the model predicts it to be negative
False Positive (FP): The actual value is negative, but the model predicts it to be positive.
Figure 2.8 explains the terminology properly.

Accuracy
This is one of the most popular metrics for classification. It is computed like so:

Accuracy (Xiaodan et.al., 2019) $= \frac{TP+TN}{TP+TN+FP+FN}$ Equation 2.9

Mean Squared Error (MSE)
The mathematical representation of this is shown in Equation 2.10.

Vol. 10, No. 06; 2025

ISSN: 2456-3676

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Yi - \bar{Y}i)^2$$ Equation 2.10

Where $n = number\ of\ classes$

$Y = observed\ value\ and\ \bar{Y}i = Predicted\ value$

MSE is an indicator of how closely a fitted line is to data points. The value for each data point is squared by multiplying it by the vertical distance between the point and the matching y value on the curve fit (Kiril, 2022).

Mean Absolute Error (MAE)

This is the most straightforward metric in evaluating neural network model off-line. The mathematical equation can be broken down as follows:

$$MAE = \sum_{i=1}^{n}(|Yi - Xi|)/n$$ Equation 2.11

Assuming there are n *classes* ([0, 1]) in the test set to evaluate, for each *classes*, the classification the model predicts$y$, and the actual classification$x$. Taking the absolute value of the difference between the two to measure the error for that classified prediction. This is literally just the difference between the predicted classification and the actual classification. Then, the errors across all the n classification in the test set are summed then, divided by n to get the average or mean (Frank, 2018). The lower the MAE and MSE values, the better (Chai & Draxler, 2014).

Recall

This is simply the actual true positives over how many times the classifier predicted that class.

$$Recall = \frac{TP}{TP+FN}$$

Precision

This is the number of correct predictions over how many occurrences of that class were in the test dataset.

$$Precision = \frac{TP}{TP+FP}$$

F1-score

The F1 score is the weighted harmonic mean of the test precision and recall. High f1-score means better precision.

$$F1 - score = \frac{2*Precision*recall}{precision+recall}$$

Confusion Matrix

The Confusion Matrix is a performance metric which categorise the model prediction with the expected value. Figure 2.8 below explains further.

Vol. 10, No. 06; 2025

ISSN: 2456-3676

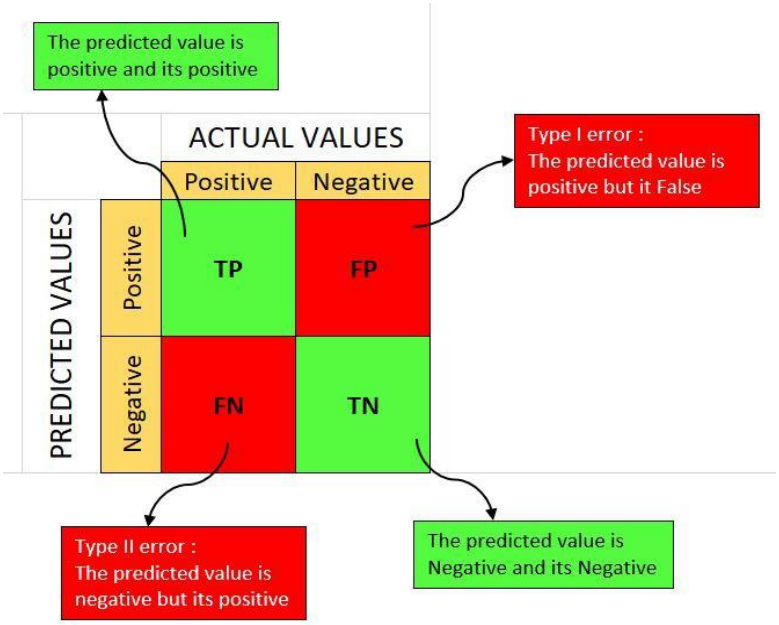


Figure 2.8 Confusion Matrix (Suresh, 2020)

3. Results

To document the result of the proposed mode, it is important to briefly explain how the measuring metrics work. This section will start by presenting the experimental results of the proposed model. Here, the result of the measuring metrics will be documented, and the graphical representation that justifies the metrics will also be discussed.

Table 3.1a shows the report of the error rate based on the mean squared error (MSE), mean absolute error (MAE) for both the train and the test data, while Table 3.1b shows the precision, recall, f1-score, and accuracy of the proposed Deep Neural Network (DNN) model created. After 19 epochs, the report shows that the proposed model could get an accuracy of 99.96%.

Table 3.1a Error Rate Report (in percentage)

MSE	Val. MSE	MAE	Val. MAE
0.036945	0.0094426	0.073716	0.11

Table 3.1b Classification Reports (in percentage)

	Precision	Recall	F1-Score	Acc.
Macro Avg.	80	90	84	
Weighted Avg.	100	100	100	99.96

Vol. 10, No. 06; 2025

ISSN: 2456-3676

The dataset has 284315 non-fraudulent transactions and 492 fraudulent transactions. This means that almost 99.9% of the dataset is non-fraudulent. After the over-sampling techniques, Table 3.1b shows that the weighted average and the macro average show that the model is performing greatly.

The deficiency of 10% accuracy in the model above can be seen from the confusion matrix (Figure 3.1). The confusion matrix shows that the model has 43 False Positives (FP) and 15 False Negatives (FN). This means the model is indicating that 43 transactions are fraudulent when, in fact, they are legitimate (FP), whereas 15 are legitimate when, in fact, they are fraudulent (FN). From the validation report (Table 3.1a), it can be seen that the proposed model is performing greatly towards detecting the classification of the unknown transaction. This proof is further validated in Figure 3.2.

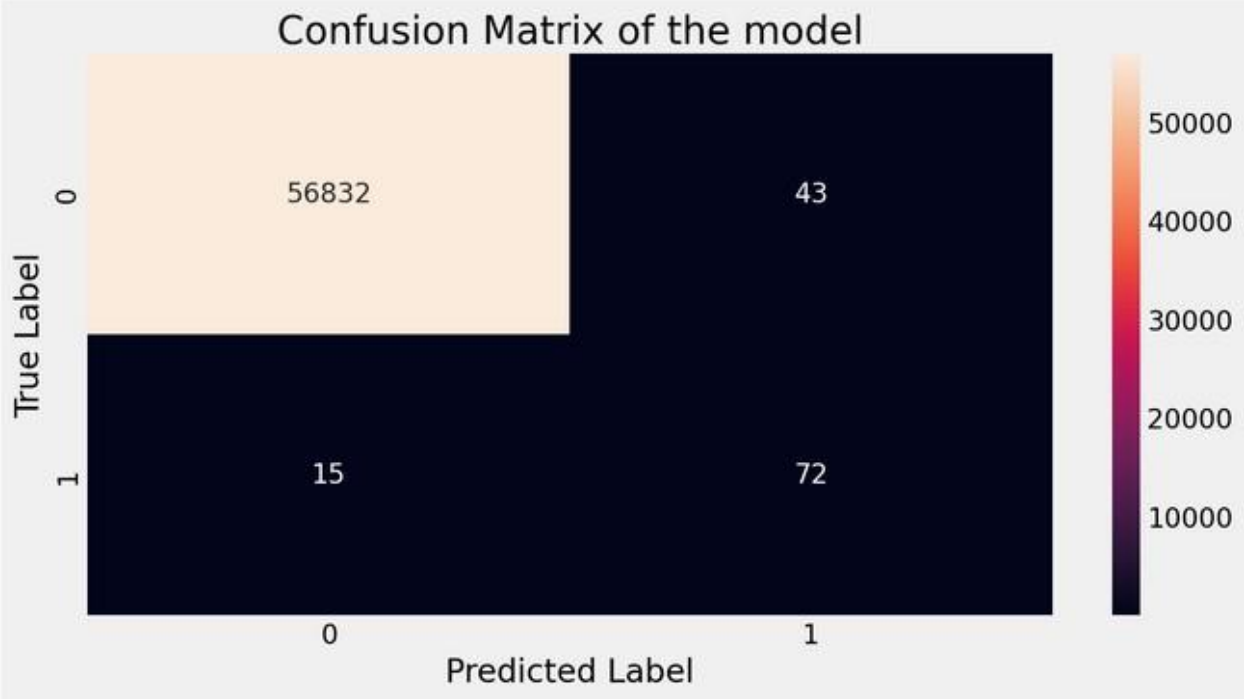


Fig 3.1 Confusion Matrix

Vol. 10, No. 06; 2025

ISSN: 2456-3676

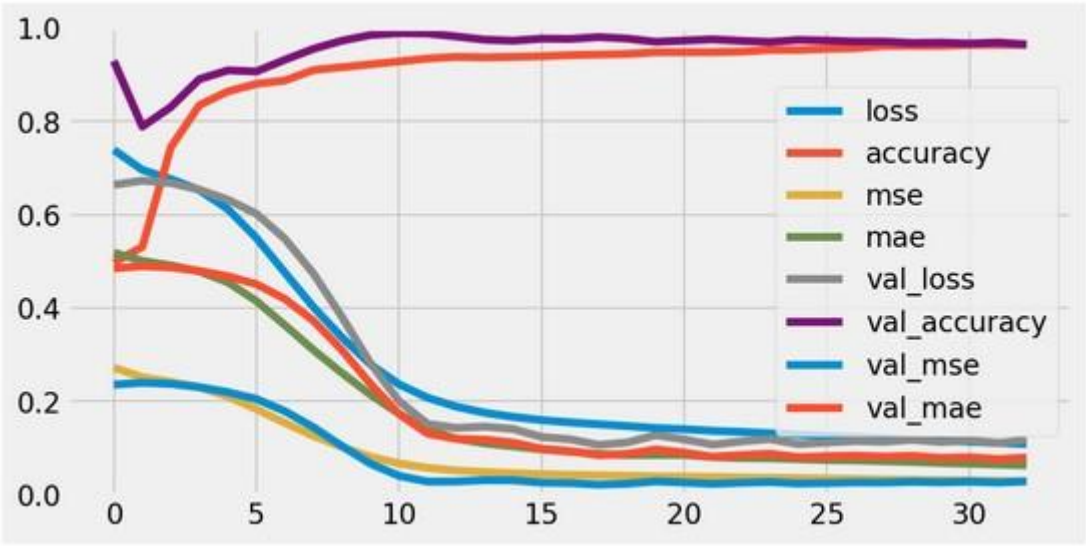


Figure 3.2 Performance of Measuring Metrics

In addition to the conventional metrics reported in Table 3.1b, further evaluation was conducted using the Receiver Operating Characteristics (ROC) and Precision-Recall (PR) curves (see Figures 3.3 and 3.4, respectively). While accuracy and F1-score show overall performance, they can be misleading when the dataset is highly imbalanced. Fraudulent transactions represent less than 0.2% of the original data; therefore, metrics that focus on rate trade-offs for the minority class provide a more objective assessment.

The proposed model achieved a ROC-AUC score of 0.9619, indicating strong discriminative ability between fraudulent and non-fraudulent transactions. However, PR-AUC, which focuses directly on the positive (fraud) class, returned a value of 0.7482, which is expected in severe class imbalance scenarios and confirms that the model performs robustly when prioritizing minority class precision and recall. These findings validate that the proposed DNN not only performs well on balanced training labels but also maintains strong sensitivity to rare fraud patterns.

Vol. 10, No. 06; 2025

ISSN: 2456-3676

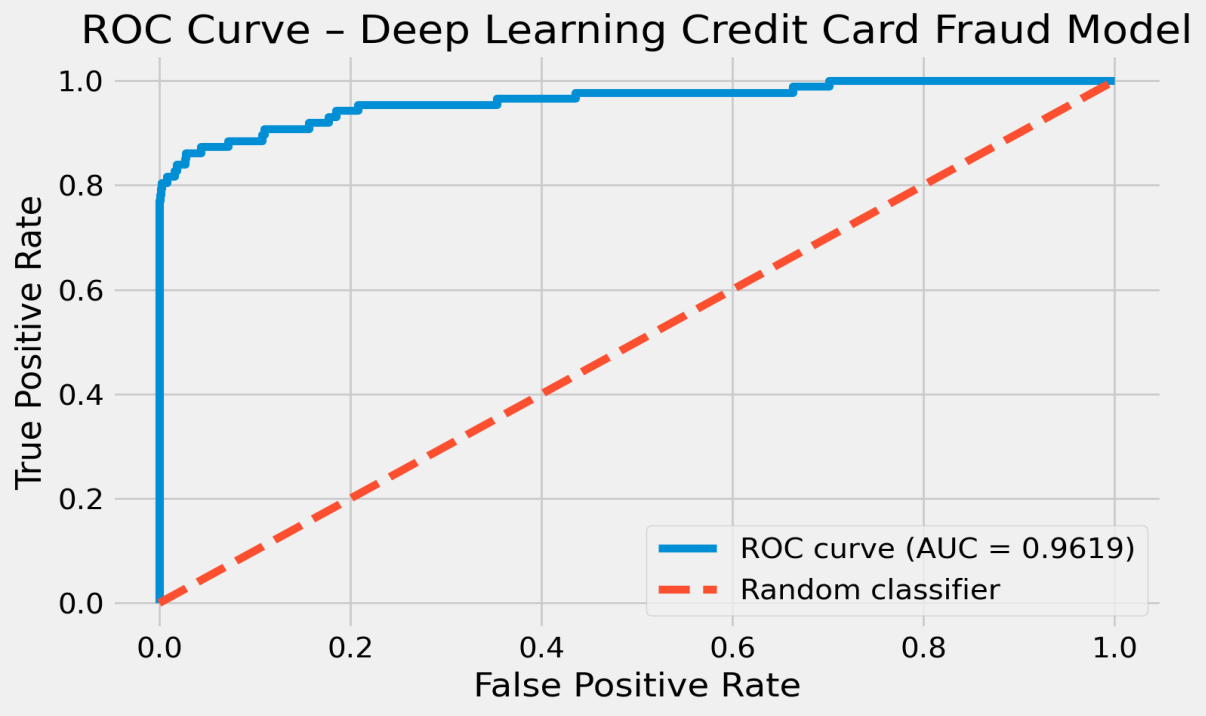


Figure 3.3. ROC curve

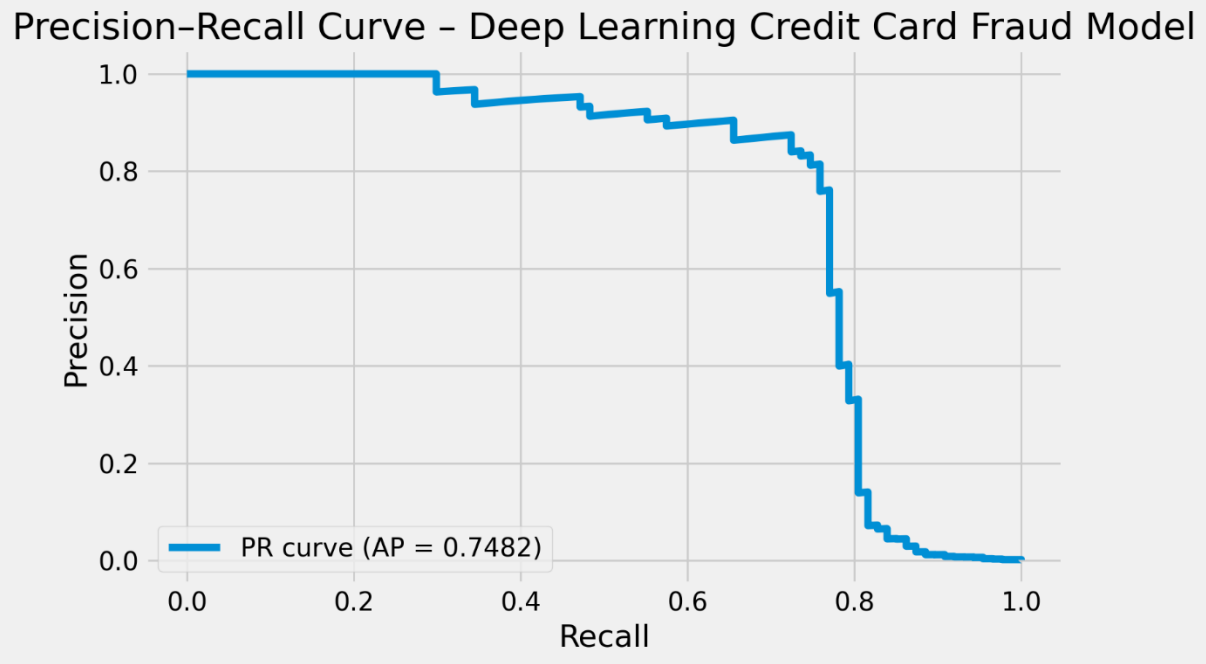


Figure 3.4 Recall Curve

Vol. 10, No. 06; 2025

ISSN: 2456-3676

Comparison with existing research

Table 3.2 shows the summary of the comparison of the proposed model with the existing ones.

Table 3.2: The comparison of the proposed research with other researchers.

Measuring Metrics (%)	Proposed Research	Oona (2021)	Pumsirirat & Yan (2018)	
			AE	RBM
Acc.	99.96	95.17	96.03	95.05
MAE	0.036945	-	-	-
MSE	0.073716	-	-	-

Acc. = Accuracy
MAE = Mean absolute error
MSE = Mean Squared Error
AE = Auto-encoder
RBM = Restricted Boltzmann Machine

The research by Oona (2021) proposed a study to identify credit card fraud, a problem that can be addressed using advanced machine learning and deep learning methods. The author identified that due to the severity of credit card theft on a global scale, the author chose to use deep neural networks to construct a model for recognising imposter frauds. The objective is to comprehend, determine, and learn the normal user behaviour, and more specifically to detect credit card fraud. The author noted that each individual has a distinct trading pattern, employs specific operating systems, has a specific time to complete the transaction, and typically spends substantial sums of money within a specific time range. The real dataset used to train the model were encoded using the one-hot technique so that the algorithm can utilise categorical data/variables. With the aid of neural networks, the author were able to get the accuracy of .95.17% (Table 3.2).

According to Pumsirirat & Yan (2018), there are no consistent patterns among frauds. They constantly alter their behaviour; hence, the author employed unsupervised learning for accurate detection. Fraudsters gain knowledge of new technologies that enable them to commit frauds via internet transactions. Fraudsters assume the typical consumer behaviour, and fraud tendencies vary rapidly. Therefore, fraud detection systems must detect online transactions by unsupervised learning, as some fraudsters conduct fraud through online channels only once before switching to other methods. In their research, they were able to develop a model of a deep autoencoder and restricted Boltzmann machine (RBM) that can reconstruct typical transactions in order to identify anomalies from normal patterns. Deep learning based on an auto-encoder (AE) is an unsupervised learning algorithm that employs backpropagation by assuming the inputs and outputs to be identical. The RBM consists of two layers: the input layer (visible) and the hidden layer. In their research, they implemented AE and RBM utilising deep learning and the Google Tensorflow library. The outcomes were shown in Table 3.2 for the same dataset used in this research.

Vol. 10, No. 06; 2025

ISSN: 2456-3676

Although Pumsirirat & Yan (2018) were able to document the MSE in their report, because the researchers utilized two different models, they did not clearly identify which of the models provided the MSE; hence, it becomes difficult to compare it with the proposed model.

According to Table 3.2, the comparison reveals that the proposed model outperforms existing research. This can be attributed to the special time allocated for the proper data preparation stage, which involves feature selection, feature engineering, and other related tasks.

However, to the best of the author's knowledge, this research will be the first to clearly document the MAE and MSE of the deep learning model development for credit card transactions; hence, there is no need to compare with previous researchers.

4. Discussion

The findings of this study strongly support the primary objective: the development of a high-precision Deep Neural Network (DNN) capable of identifying credit card fraud. With an achieved accuracy of 99.96%, the results confirm that deep learning architectures, when paired with rigorous data pre-processing, can effectively overcome the challenges posed by highly imbalanced financial datasets.

Interpretation and Contextualization The model's performance significantly exceeds benchmarks established in prior literature utilising the same dataset. Specifically, the proposed DNN outperformed the 95.17% accuracy reported by Oona (2021) and the 96.03% achieved by Pumsirirat & Yan (2018) using Auto-encoders. This superiority is largely attributed to the methodological emphasis on Feature Engineering, specifically, the oversampling technique, which effectively neutralizes the bias toward the majority class (non-fraudulent transactions). Furthermore, the use of Mutual Information for automatic feature selection allowed the model to discard noise, such as the "Time" feature, which was found to have no predictive power.

Internal Validity and Error Analysis. While the overall accuracy is high, the confusion matrix provides a more nuanced view of internal validity. The model produced 43 False Positives (FP) and 15 False Negatives (FN). In the context of financial fraud, False Negatives (undetected fraud) are costlier than False Positives (inconvenienced customers). The low count of FN (15) suggests the model is highly sensitive to fraudulent patterns, although the presence of FPs indicates a slight trade-off where valid transactions are occasionally flagged. The successful application of the ReLU activation function also validates its theoretical utility in binary classification tasks by preventing vanishing gradients during training.

In financial applications, False Negatives (fraud that the model fails to detect) are more harmful than False Positives. A missed fraudulent transaction leads directly to financial loss and possible liability claims. Therefore, even though the model recorded only 15 FN, improving FN detection remains important. Strategies such as focus-penalized loss functions, higher recall thresholds,

Vol. 10, No. 06; 2025

ISSN: 2456-3676

and anomaly exposure methods (Hendrycks et al., 2019) could be used to reduce FN further during future improvements.

Limitations: Two primary limitations constrain the generalizability of these results. First, the study relies on a static secondary dataset. While this allows for academic benchmarking, it may not fully reflect the "concept drift" of real-time fraud where attackers constantly evolve their tactics. Second, the scope was restricted to model logic and training; it did not cover the computational latency or integration challenges associated with deploying this model into live mobile or web applications.

Real-World Fraud Dynamics and Concept Drift

Although the model achieved a high accuracy of 99.96% using a benchmark dataset, real-world financial fraud changes over time due to the evolving strategies of attackers. This behaviour is known as concept drift, where patterns learned during training no longer represent new fraud activities. As a result, a model trained on a static dataset may gradually lose accuracy if it is not retrained using recent transaction data.

Real deployments, therefore, require continuous re-training, periodic data collection, and monitoring of model performance. Without these updates, the model may struggle to detect new transaction behaviours or advanced fraud attempts. Recent studies (e.g., Pourhabibi et al., 2020; Hendrycks et al., 2019) also suggest that anomaly-drift is now a major challenge in modern fraud detection systems. Therefore, while the results of this study are strong, future implementation must include dynamic model updating to maintain integrity over time.

Deployment Perspective

From a deployment perspective, an important challenge is ensuring that the model functions efficiently in real-time applications. Fraud detection systems must handle thousands of incoming transactions per second with very low latency. Model execution time, cloud integration, and streaming analysis must therefore be considered during implementation. Techniques such as mini-batch inference, GPU acceleration, and stream-based frameworks like Apache Kafka or Flink may be required to ensure real-time robustness. Future work should evaluate these operational requirements to support live banking deployment.

Conclusion and Future Implications

The practical significance of this research lies in its potential to automate security in the banking sector. By demonstrating that a DNN can achieve near-perfect accuracy through proper normalization and balancing, this study offers a viable pathway for financial institutions to reduce the manual overhead of fraud monitoring. Future work should focus on testing this architecture in a real-time streaming environment to evaluate its external validity against live, evolving fraud vectors.

Vol. 10, No. 06; 2025

ISSN: 2456-3676

References

A. Shrivastava, M. Yadav, S. Basu, S. Salunkhe and M.Shabad, " Credit card fraud detection at merchant side using neural network", 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp.667-670

Abdullah DM, Abdulazeez AM (2021). Machine learning applications based on SVM classification a review. Available at: https://pdfs.semanticscholar.org/90ea/bbfbe0d8be2ac4a53780ac1e40ad63cf6bad.pdf

Andrea Dal Pozzolo, Oliver Caelen, Reid A. Johnson and Gianluca Bontempi. "Calibrating Probability with Undersampling for Unbalanced Classification. *In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2015*

Anuganti Suresh (2020). What is confusion matrix. Available at: https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

Changjun Jiang, Jiahui Song, Guanjun Liu, Lutao Zheng, and Wenjing

Dan Hendrycks, Mantas Mazeika, Thomas Dietterich (2019). Deep Anomaly Detection with Outlier Exposure. Available at: https://arxiv.org/abs/1812.04606

Hamed Alqahtani, Iqbal H. Sarker, Asra Kalim, Syed Md. Minhaz Hossain, Sheikh Ikhlaq & Sohrab Hossain (2020). Cyber Intrusion Detection Using Machine Learning Classification Techniques. Available at: https://link.springer.com/chapter/10.1007/978-981-15-6648-6_10

Hammad, M. et al. (2020). Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data. *Applied Sciences*, 11(12), 5320.

Hendrycks, D., Mazeika, M., & Dietterich, T. (2019). Deep Anomaly Detection with Outlier Exposure. *arXiv:1812.04606*.

Hongzhi Wang; Mohamed Jaward Bah; Mohamed Hammad (2020). Progress in Outlier Detection Techniques: A Survey. Available at: https://ieeexplore.ieee.org/abstract/document/8786096

Kalyanakrishnan, Shivaram & Singh, Deepthi & Kant, Ravi.(2014), "On Building Decision Trees from Large-scale Data in Applications of On-line Advertising", CIKM 2014-Proceedings of the 2014 ACM ACM International Conference on Information and Knowledge Management.

Luan, "Credit Card Fraud Detection: A Novel Approach Using Aggregation Strategy and Feedback Mechanism", in IEEE Internet Of Things Journal, vol.5, no.5, pp.3637-3647, Oct.2018. doi: 10.1109/JIOT.2018.2816007

N. Malini and Dr M. Pushpa, "Analysis on credit card fraud identification techniques based on KNN and outlier detection" in 3rd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEEICB17).

O. Awoyemi, John & Adetunmbi, Adebayo & A. Oluwadare, Samuel. (2017). "Credit card fraud detection using Machine Learning Techniques: A comparative analysis", 1 9.10.1109/ICCNI.2017.8123782

Oona Voican (2021). Credit Card Fraud Detection using Deep Learning Techniques. Available at: http://revistaie.ase.ro/content/97/06%20-%20voican.pdf

Patil, Suraj & Nemade, Varsha & Kumar Soni, Piyush. (2018) "Predictive Modelling for Credit Card Fraud Detection Using Data Analytics", Procedia Computer Science. 132 (2018) 385-395. 10.1016/j.procs.2018.05.199.

Pourhabibi, T., Ong, K.-L., Kam, B., & Boo, Y. L. (2020). Fraud detection: a systematic review of graph-based anomaly detection approaches. *Decision Support Systems, 138*, 113526.

Pumsirirat A, Liu Y (2018). Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine. Available at: https://pdfs.semanticscholar.org/01be/7624aa0e0251182593350a984411c2e5128a.pdf

Redhwan Al-amri, Raja Kumar Murugesan, Mustafa Man, Alaa Fareed Abdulateef, Mohammed A. Al-Sharafi and Ammar Ahmed Alkahtani (2020). A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data. Available at: https://www.mdpi.com/2076-3417/11/12/5320

Renu, Suman, "Analysis on Credit Card Fraud Detection Methods", International Journal of Computer Trends and Technology (IJCTT) 8(1):45-51, February 2013. ISSN:2231-2803

Sahil Dhankhad, Behrouz Far and Emad A. Mohammed, "Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study, 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, 2018, pp.122-125. doi: 10.1109/IRI.2018.00025

Shijoe Jose, D. Malathi, Bharath Reddy and Dorathi Jayaseeli (2018). A Survey on Anomaly-Based Host Intrusion Detection Systems. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/1000/1/012049/meta

Stevenson (2014). Neural Networks (Computational Intelligence). Available at: https://sites.google.com/site/mrstevensonstechclassroom/hl-topics-only/4a-robotics-ai/neural-networks-computational-intelligence

Tahereh Pourhabibi, Kok-Leong Ong, Booi H. Kam &Yee Ling Boo (2020). Fraud detection: A systematic literature review of graph-based anomaly detection approaches. Available at: https://www.sciencedirect.com/science/article/pii/S0167923620300580

Ünal Çavuşoğlu (2019). A new hybrid approach for intrusion detection using machine learning methods. Available at: https://link.springer.com/article/10.1007/s10489-018-01408-x

Xiaodan Xu, Huawen Liu and Minghai Yao (2019). Advances in Architectures, Big Data, and Machine Learning Techniques for Complex Internet of Things Systems. Available at: https://www.hindawi.com/journals/complexity/2019/2686378/

Zhou, L., & Guo, H. (2018). *Anomaly Detection Methods for IIoT Networks*. 214-219. https://doi.org/10.1109/soli.2018.8476769