CALIFORNIA STATE UNIVERSITY SAN MARCOS

THESIS SIGNATURE PAGE

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

THESIS TITLE: CREDIT CARD FRAUD DETECTION IN REAL TIME

AUTHOR: Harshit Lamba

DATE OF SUCCESSFUL DEFENSE: 12/9/2020

THE THESIS HAS BEEN ACCEPTED BY THE THESIS COMMITTEE IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF
MASTER OF SCIENCE IN COMPUTER SCIENCE.

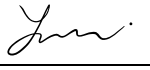
Dr. Yanyan Li		12/10/2020
THESIS COMMITTEE CHAIR	SIGNATURE	DATE
Dr. Xin Ye		12/10/2020
THESIS COMMITTEE MEMBER	SIGNATURE	DATE
Name of Committee Member		
THESIS COMMITTEE MEMBER	SIGNATURE	DATE

Table of Contents

For our training model, we have used ANN since it provides us with the best accuracy, F-1 score, and AUC (Area Under the Curve). Its structure and working are very closely similar to a human brain consisting of computational neurons. The number of neurons and the number of layers of neurons can be defined as per the needs of the data and output. The ANN model counts on Artificial Intelligence to do the automation provide the results. Further, we have combined the machine learning model with an automation tool 'Apache Airflow' which helps us streamline bigdata and further enables the author to programmatically schedule their workflows and monitor them with the help of its built-in interface. Hence, this project enables the fraud detection manager to closely monitor the transactions and take timely action to prevent fraudulent transactions.

Artificial Neural Network, usually shortened as ANN. The ANN works like a human brain hence have used ANN over other algorithms. The ANN is inspired by a biological neural network that constitutes the human brain. The main objective for using ANN is that it can work on a large dataset and also it gives better and accurate results than any other method or algorithm. The ANN technique/method relies on Artificial Intelligence to work automatically and give better results on its own and has also used backpropagation with gradient descent. In our model for the evaluation purpose, the parameter such as accuracy, recall, precision, and confusion matrix is used.

All over the world, almost every business is done digitally and is using plastic money i.e. credit cards for this purpose. In the current situation, only the card information is needed to do the fraud. While card detail is entered online, won't know how the person will use our card details online or if our card may have been lost, stolen by an unauthorized person [9]. In India, card details of around 70 million people are being sold on the dark web. So, it is easy to conduct fraud. That is why an initiative is taken to create this project to prevent credit card fraud. In this case, our project will give real-time results.

In this paper, we propose a unique credit card fraud detection system that can easily be implemented in any organization dealing with a credit card transaction. Our system shall be able to detect fraud in real-time and provide the fraud manager updates quickly to stop the fraudulent transactions. In this paper,

1. We implement an Artificial Neural Network to train and apply the model on incoming credit card transactions.
2. We implement the Apache Airflow automation which will keep performing the check on the incoming transaction file every 5 minutes and update the Fraud Manager to then take action on it as needed.
3. Finally, we implement a front-end GUI using the flask application which shall provide any user the convenience to observe the Fraud Transactions taking place.

2. LITERATURE SURVEY

In studies in the past, researchers have already tried and tested several approaches such as unsupervised learning, supervised learning, etc. With time as cybercriminals have evolved in using several different techniques, similarly, the researchers have also had to improvise with their research observations and patterns.

The authors of [9] and [14] have specified that due to the lack of a good publicly available dataset they have been come short on their findings and research observations. In two separate research studies in 2016 by Z. Zojaji, R. E. Atani, A. H. Monadjemi, and David Robertson it can be observed that they have used highly imbalanced data since the number of fraud transactions are very few as compared to the number of non-fraud transactions. In [10], the authors have used ANN and Linear Regression (LR) to create 13 alternative predicting models. It was concluded with ANN being more superior to LR since ANN provides much higher accuracy on the same dataset. Paper [8] gives an insight into the author's work, where they have created a Neural network consisting of 17 neurons in its input layer, and two hidden layers each consisting of 60 and 50 computational neurons respectively, and the output layer consisting of only 1 neuron. This method was created while keeping an eye out for the highly imbalanced dataset and with an attempt to keep the cost low.

Paper [11] proposes a system for classification models and intrusion detection by implementing a Multilayer Perceptron (MLP) algorithm and compared it with the Naïve Bayes and Decision Tree Algorithm. The ANN-MLP model produces better results as compared to the Naïve Bayes or Decision Tree learning model. The authors [6] propose a framework for credit card fraud detection using ANN and Self-Organizing Maps (SOM). They implemented clustering in SOM to capture fraud detection.

3. PROBLEM STATEMENT

The problem with current methods is that their models are so complex that it is not easy to deploy into the bank system has used SOM as a method for detecting fraud [6]. The data points must behave similarly to use SOM as a method for detecting fraud. Using SOM for detecting fraud will have a latency problem when it will be deployed and used in the real world in a banking system. Current systems will detect fraud after the transactions have occurred.

4. PROPOSED SYSTEM

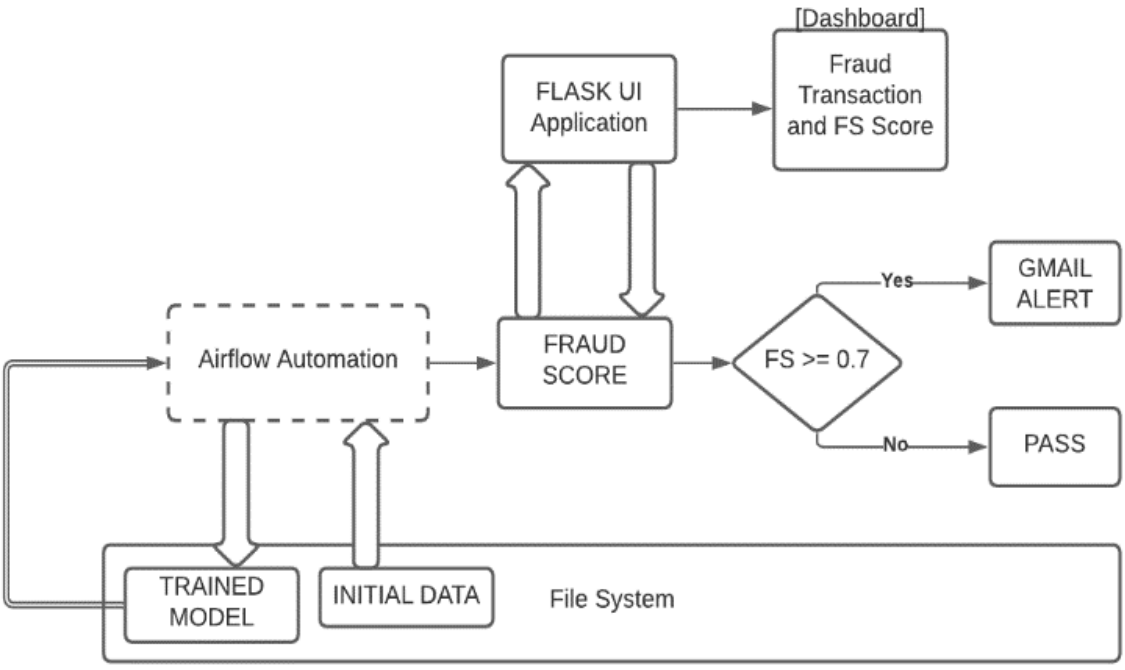


Figure 1. System Architecture

We propose an end-to-end system that can be easily be deployed in any organization dealing with real-time online credit card transactions. The proposed architecture will consist of a training model known as the Artificial Neural Network to be able to correctly classify whether the incoming transaction is fraudulent or not in real-time. The model is then further integrated with Apache Airflow to automate the prediction model without any human intervention and also added a simple functionality where if the predicted transaction in real-time has a Fraud Score ≥ 0.7 then the Fraud Manager will simply receive an alert on the dashboard or their authenticated Gmail inbox to review the transaction and act accordingly. We have also deployed a simple GUI using the FLASK application as the front end. The dataset used in our training and prediction was sourced from an open library on Kaggle. The data was collected in September 2013 over 48 hours of European Credit Card Users. The dataset consists of a total of 284,807 transactions where only 492 transactions are classified as fraud. Since the positive class only accounts for a total of 0.172% due to this the data is highly unbalanced or skewed[2][7]. For privacy reasons,

the majority of the values in our dataset are numerical since they have undergone Principal Component Analysis (PCA). Only the attributes ‘Time’, ‘Amount’ and ‘Class’ are kept as it is. The PCA technique is used as an attempt to reduce the dimensionality of the dataset and then scaling it to be used with the learning model. The data is then oversampled and undersampled to create a balanced dataset. On oversampling, we use Synthetic Minority Oversampling Technique (SMOTE) to generate more fraud transactions to match the number of non-fraud transactions. This ends up increasing the total number of transactions present in the dataset. Similarly, on undersampling, we end up losing the number of non-fraud transactions to match it with the number of fraud transactions. This ends up in creating a dataset consisting of fewer transactions as compared to that of the original dataset.

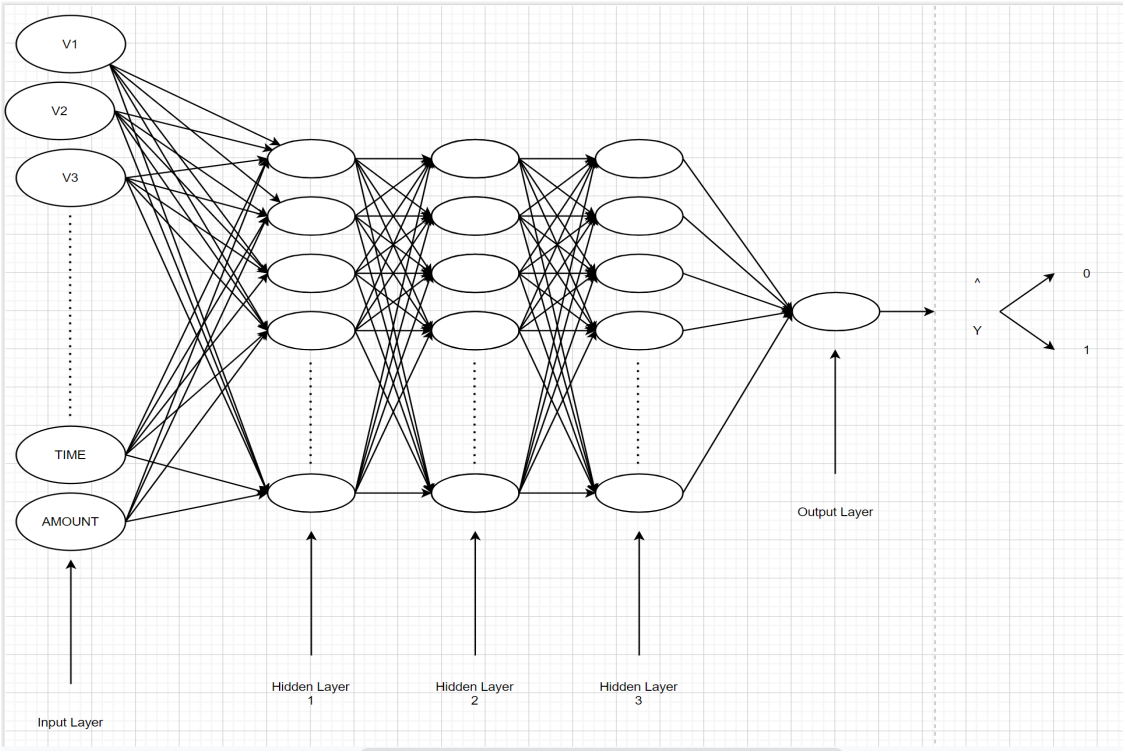


Figure 2. Artificial Neural Network Architecture

We introduce these 31 attributes as inputs to our ANN model along with one bias node. The inputs then pass into the first hidden layer consisting of 10 computational neurons. Also, in ANN with a set of given inputs, the activation function of the neuron decides what exactly the output will be from the neuron. In our model, we have chosen RELU as our activation function for the hidden layers, and for the output layer, the SOFTMAX activation function is chosen. These activation functions were chosen since they do not go These activation functions were chosen since they do not enter a vanishing gradient problem to make it harder for the model to train further on i.e. the convergence decelerates down. The RELU activation function equation is explained further below.

$$f(z) = (0, z) = \left\{ \frac{o \; if \; z \leq 0}{z \; otherwise} \right\} \qquad (1)$$

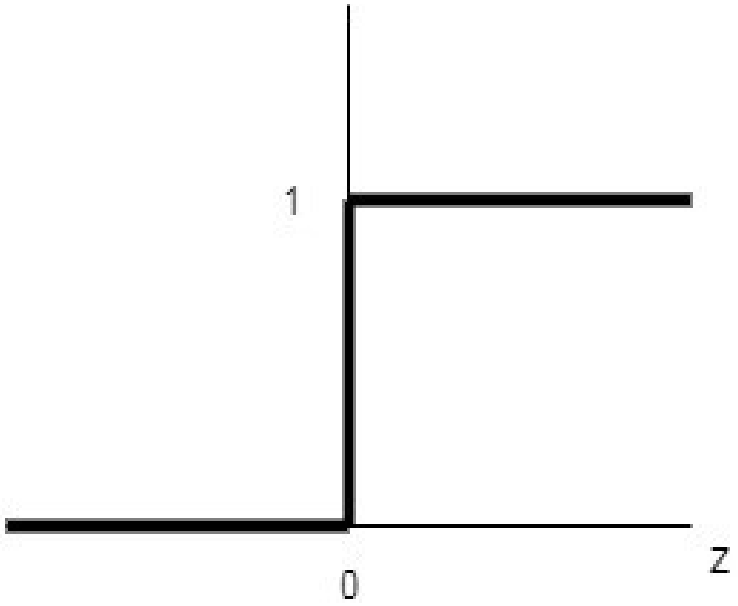


Figure 3. RELU Activation Function

From the equation (1) we can see that it gives 0 as an output if the value of z is less than 0 or else it gives back z. To calculate forward propagation we use the equation

$$z^{[L]} = w^{[L]} a^{[L-1]} + b^{[L]} \qquad (2)$$

$$a^{[L]} = g^{[L]}(z^{[L]}) \qquad (3)$$

In equation(2), we can observe that $z^{[L]}$ is obtained by obtaining the product of $w^{[L]}$, where $w^{[L]}$ is the weight of the layer, to the input layer along with the addition to the bias. In equation (3), $g^{[L]}$ is the Activation layer of Layer L. This can also be calculated by using the steps below,

For the First Hidden layer L,

$$z^{[1]} = w^{[1]}x + b^{[1]}$$

$$A^{[1]} = g^{[1]}(z^{[1]}) = \sigma(z^{[1]})$$

For the Second Hidden layer L,

$$z^{[2]} = w^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = g^{[2]}(z^{[1]} = \sigma(z^{[1]})$$

For the Fifteenth Hidden layer L,

$$z^{[15]} = w^{[15]}A^{[14]} + b^{[15]}$$

$$A^{[15]} = g^{[15]}(z^{[15]} = \sigma(z^{[15]})$$

Hence the inside of each neuron the calculation should like

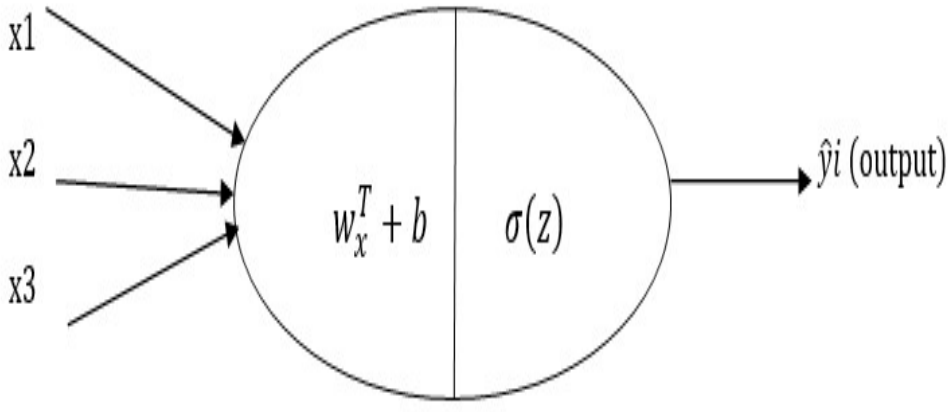


Figure 4. A single Neuron

For the output layer, the SOFTMAX activation function is applied which is a generalization of logistic function to multiple dimensions. Here, it simply takes the output of our network and normalizes it into a probability distribution over the predicted output class. Softmax is a mathematical function that simply transforms a vector of numbers into a vector of probabilities, where the probabilities are directly proportional to each unique value in the vector on the relative scale. A single standard softmax function is defined mathematically as

$\sigma: R^k \rightarrow R^k$

where,

$$\sigma(z)_i = \frac{e^{zi}}{\sum_{j=1}^{k} e^{zj}} \qquad (4)$$

for i = 1, ... K and $z = (z_1, z_2, \ldots, z_k) \in R^k$

We further combine the gradient descent and backpropagation to find the results from the Loss function and then simply updating their parameters.

$$w = w - \alpha \frac{\partial J}{\partial w} \qquad (5)$$

$$b = b - \alpha \frac{\partial J}{\partial b} \qquad (6)$$

For the First Hidden layer,

$$d(z)^{[1]} = w^{[2]}\, dz^{[2]} * g^{[1]}(z^{[1]})$$

$$dw^{[1]} = dz^{[1x]} x^T$$

$$db^{[1]} = dz^{[1]}$$

For the Second Hidden layer,

$$d(z)^{[2]} = w^{[3]}\, dz^{[3]} * \; g^{[2]}(z^{[3]})$$

$$dw^{[2]} = dz^{[2]} a^{[1]^T}$$

$$db^{[2]} = dz^{[2]}$$

And so on. Now the parameters w and b are updated using gradient descent in equations 5 and 6. Where,

w = weights

α = learning rate

$\frac{J}{\partial w}$ = cost function derivative

∂w, $\frac{\partial J}{\partial b}$ = cost function conceding ðb derivative

Hence, the steps taken in our Artificial Neural Network will look like,

(1) We at the outset set the neurons and bias with random initialization and zero initialization respectively.
(2) D = $\{x_i, y_i\}$
For every x_i in D
   (a) We initiate forward propagation by feeding x_i through the network.
   (b) Next, we calculate the Loss function,
   $L(y_i', y_i)$ where y_i' is the anticipated value and y_i is the authentic value. Here, y = {0, 1} where 0 is the non-fraud class and 1 is the fraud class.
   (c) Further we obtain the output value for all the hidden layers and execute backpropagation.

Algorithm:

(1) Step 1: Setting the neurons and bias with random initialization and zero initialization respectively.
(2) Step 2: Using Scikit-Learn library to import the needed libraries and accessing the 'creditcard.csv' dataset file.
(3) Step 3: Oversampling the dataset to create a balanced dataset.
(4) Step 4: Pass the dataset values into the designed artificial neural network.
(5) Step 5: Training the neural network with the feed-forward technique and applying the desired activation function on each layer.
(6) Step 6: Executing backpropagation after the first iteration.
(7) Step 7: Repeat steps 1 – 6 as many times as desired to train and improve the neural network prediction model. Here we do it for 20 epochs.

The final output file of the tested dataset will consist of two columns with the prediction class of the transaction (0 or 1) and floating-point values between 0 and 1 due to the use of the softmax function in the final layer.

We further use our trained model to create a real-time fraud detection system with the help of apache airflow and introducing a simple graphical user interface for the fraud manager using a simple FLASK application.

Apache Airflow is an open-source workflow management platform that is used to schedule, monitor, and organize complex workflows and data pipelines for large amounts of data. This is ideal to use in a real-life scenario since thousands of card transactions take place in real-time at one single instance of time. This tool consists of core features such as being dynamic, extensible, elegant, and scalable.

We break down our python script and create individual tasks for the apache airflow to execute them on a collection of workers while following stipulated dependencies. This process enables us to automate the backend without any human intervention.

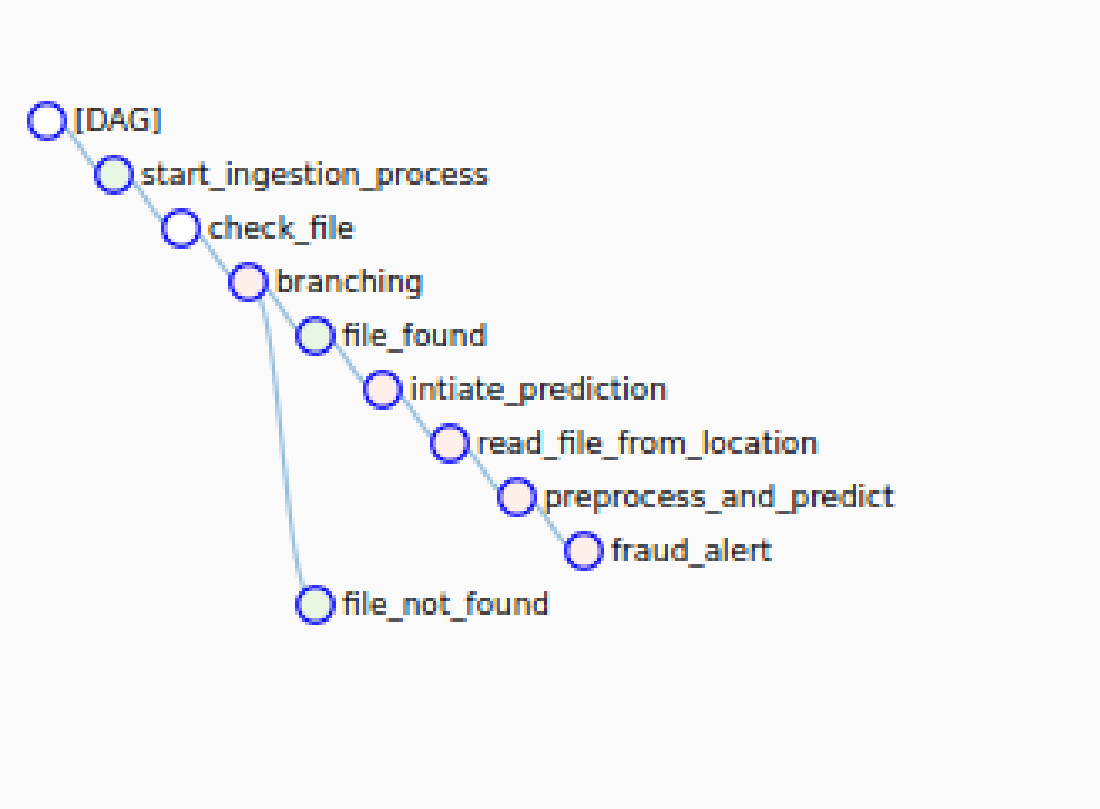


Figure 5. Apache Airflow Webserver

This proposed system uses Artificial Neural Networks and Backpropagation to find fraudulent transactions. This will give the prediction of each transaction and it will detect the fraudulent transactions in real-time. The Credit Card Customer dataset is used that has 31 attributes to our ANN model. The first 30 attributes have information related to the customer's transaction history.

5. MODEL EVALUATION

We evaluate the results of our trained architecture with the help of several tools and diagrams such as F-1 Score, Accuracy, Area Under the Curve(AUC), Precision, and Confusion Matrix.

Often when predicting a model we might think that the one with the highest accuracy should be our ideal selection. However, sometimes it is desirable to select a model with low accuracy since it provides us with greater prediction power on the problem. This is known as the Accuracy Paradox. Hence we use other metrics such as F-1 Score, AUC, Precision, etc. along with Accuracy to choose the best prediction model.

Confusion Matrix - It is a performance measurement used for machine learning classification where the resulting output could be classified into two or more individual classes. In our binary classification, we witness a 2X2 table with 4 different combinations of predicted and actual values.

(Predicted Values)

(Actual Values)		
	True Positive (TP)	False Negative (FN)
	False Positive (FP)	True Negative (TN)

Figure 6. Confusion Matrix

Where,

True Positive(TP) – Model predicts positive and it is true.

True Negative(TN) – Model predicts negative and it is true.

False Positive(FP) – Model predicts Positive and its false.

False Negative(FN) – Model predicts Negative and its false.

Accuracy – It is the ratio between the correctly predicted observations and the total observations.

Precision – It is the ratio between the correctly predicted positive observations and the total predicted positive observations.

Recall – It is the ratio between the correctly predicted positive observations and all observations in the actual class.

F-1 Score – It is the weighted average of precision and recall.

(Predicted Values)

(Actual Values)

56863	0
0	98

Figure 7. Ideal case Confusion Matrix

Figure (4) shows us what an ideal confusion matrix with 100% accuracy would like to compare our results with our undersampled model and oversampled model confusion matrices further.

For our undersampled dataset, we use 629 samples to train and 158 samples to validate our model.

The confusion matrix and the model evaluation for our undersampled model are given below.

(Predicted Values)

(Actual Values)

53569	3294
8	90

Figure 8. Undersampled Confusion Matrix

For our testbed for this model, we have used a dataset consisting of a total of 56961 transactions to check the performance of our trained model. Here, the accuracy of our model is calculated by simply substituting the appropriate values in the formula where Accuracy $=$ $\frac{\text{(True Positive + True Negative)}}{\text{(Positive + Negative)}}$

$$= \frac{(53569+90)}{(56961)} = \frac{53659}{56961} = 0.94203051 \ (94.2\%)$$

For calculating Recall, we substitute the appropriate values in the formula where Recall $= \frac{(\text{True Positive})}{(\text{True Positive} + \text{False Negative})}$

$$= \frac{(53569)}{(53569 + 3294)} = \frac{53569}{(56863)} = 0.94207129 = (94.2\%)$$

For calculating Precision, we substitute the appropriate values in the formula where Precision $= \frac{(\text{True Positive})}{(\text{True Positive} + \text{False Positive})}$

$$= \frac{53569}{53569 + 8} = \frac{53569}{53577} = 0.99985068 = (99.98)$$

Further for calculating the F-1 score, we substitute the appropriate values in the formula where,

$$\text{F-1 score} = 2\left(\frac{Precision * Recall}{Precision + Recall}\right) = 2\left(\frac{99.98 * 94.2}{99.98 + 94.2}\right)$$

$$= 2\left(\frac{9418.116}{194.18}\right) = 97.0039757$$

Further, we also obtain the AUC score of 0.9733

For the oversampled dataset we use 291,138 samples to train and 72,785 samples to validate our model.

(Predicted Values)

(Actual Values)		
	56830	33
	17	81

Figure 9. Oversampled Confusion Matrix

For our testbed for this model, we have used a dataset consisting of a total of 56961 transactions to check the performance of our trained model. Here, the accuracy of our model is calculated by simply substituting the appropriate values in the formula where Accuracy $= \frac{(\text{True Positive} + \text{True Negative})}{(\text{Positive} + \text{Negative})}$

$$= \frac{(56830 + 81)}{(56961)} = \frac{56911}{56961} = 0.99912221 \ (99.91\%)$$

For calculating Recall, we substitute the appropriate values in the formula where Recall $= \frac{(\text{True Positive})}{(\text{True Positive} + \text{False Negative})}$

$$= \frac{(56830)}{(56830 + 33)} = \frac{56830}{(56863)} = 0.99941966 = (99.94\%)$$

For calculating Precision, we substitute the appropriate values in the formula where Precision $= \frac{(\text{True Positive})}{(\text{True Positive} + \text{False Positive})}$

$$= \frac{56830}{56830 + 17} = \frac{56830}{56847} = 0.99970095 = (99.97)$$

Further for calculating the F-1 score, we substitute the appropriate values in the formula where,

$$\text{F-1 score} = 2\left(\frac{Precision*Recall}{Precision+Recall}\right) = 2\left(\frac{99.97 * 99.94}{99.97 + 99.94}\right)$$

$$= 2\left(\frac{9991.0018}{199.91}\right) = 99.954998$$

Further, we also obtain the AUC score of 0.7641

On comparing the AUC and F-1 score values from both the undersampled and oversampled datasets we can observe that the oversampled dataset is producing more accurate results with fewer errors. Hence, the oversampled model becomes our model of choice by default.

To conclude ANN will be our choice of training model we also tried and tested our results with Logistic Regression, K-nearest, Support Vector Classifier, Decision Trees, and ANN. We can see the results in the table below.

Training model	Precision	Recall	F-1 Score
Logistic Regression	0.90	0.99	0.94
K-nearest Neighbour	0.87	1.00	0.93
Support Vector Classifier	0.88	0.99	0.93
Decision Trees	0.87	0.99	0.93
Artificial Neural Networks(oversampled data)	0.99	0.99	0.99

Fig 10. Model Evaluation for different machine learning models.

[3] M. Ummul Safa and R. M. Ganga. “Credit Card Fraud Detection Using Machine Learning.” 2019 International Journal of Research in Engineering, November-2019. https://www.ijresm.com/Vol.2_2019/Vol2_Iss11_November19/IJRESM_V2_I11_80.pdf

[4] Saurabh C. Dubey, Ketan S. Mundhe, and Aditya A. Kadam. “Credit Card Fraud Detection using Artificial Neural Network and BackPropogation”.F. Ghobadi and M. Rohani, "Cost-sensitive modeling of credit card fraud using neural network strategy," 2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS), Tehran, 2016, pp. 1-5.

[5] M. Syeda, Yan-Qing Zhang, and Yi Pan, "Parallel granular neural networks for fast credit card fraud detection," 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No.02CH37291), Honolulu, HI, USA, 2002, pp. 572-577 vol.1.

[6] J. R. Dorronsoro, F. Ginel, C. Sanchez, and C. S. Cruz, "Neural fraud detection in credit card operations," in IEEE Transactions on Neural Networks, vol. 8, no. 4, pp. 827-834, July 1997.

[7] Y. Sahin and E. Duman, "Detecting credit card fraud by ANN and logistic regression," 2011 International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, 2011, pp. 315-319.

[8] M. Mubarek and E. Adalı, "Multilayer perceptron neural network technique for fraud detection," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 383-387.

[9] E. Saraswathi, P. Kulkarni, M. N. Khalil, and S. Chandra Nigam, "Credit Card Fraud Prediction And Detection using Artificial Neural Network And Self-Organizing Maps," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 1124-1128.

[10] Y. Lee, Y. Yeh and Y. F. Wang, "Anomaly Detection via Online Oversampling Principal Component Analysis," in IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 7, pp. 1460-1470, July 2013.

[11] K. Alrawashdeh and C. Purdy, "Fast Activation Function Approach for Deep Learning-Based Online Anomaly Intrusion Detection," 2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), Omaha, NE, 2018, pp. 5-13.

[12] J. Amrutha and A. S. Remya Ajai, "Performance analysis of Backpropagation Algorithm of Artificial Neural Networks in Verilog," 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication

Technology (RTEICT), Bangalore, India, 2018, pp. 1547-1550.

[13] S. Karn, S. Sangole, A. Gawde, and J. Joshi, "Prediction and Classification Of Vector-Borne and Communicable Diseases through Artificial Neural Networks," *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, 2019, pp. 1011-1015.

[14] V. Ceronmani Sharmila, K. K. R., S. R., S. D., and H. R., "Credit Card Fraud Detection Using Anomaly Techniques," 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), CHENNAI, India, 2019, pp. 1-6, DOI: 10.1109/ICIICT1.2019.8741421.

[15] N. D. Marom, L. Rokach, and A. Shmilovici, "Using the confusion matrix for improving ensemble classifiers," 2010 IEEE 26-the Convention of Electrical and Electronics Engineers in Israel, Eliat, 2010, pp. 000555-000559

[16] "Beyond Accuracy: Precision and Recall," [Online], Available: https://towardsdatascience.com/beyond-accuracy-precision-and- recall-3da06bea9f6c.