Detecting Financial Fraud Using Highly Imbalanced Data With the Logistic Regression and Deep Learning Artificial Intelligence Algorithms

Zacheriah Valis
Department of Computer Science
University of South Carolina Upstate
Spartanburg, South Carolina, USA
zvalis@email.uscupstate.edu

ABSTRACT

Financial fraud has been an issue among the financial world for as long as it's existed. With the systemic expansion of technology, the doors to fraud have burst wide open with new ways to steal one's money. Despite the widespread use of CHIP&PIN technologies, the most abundant type of fraud, credit card fraud via virtual Point of Sale terminals on the internet, does not protect against this [6]. With losses expected to be in the tens of billions in 2020, it is critical for financial institutions to work on developing a novel way to combat fraud. In response, these institutions have set their eyes on artificial intelligence. But with so many different methodologies, which one will prove to be the most effective for the given dataset? This research intends to answer that very question by employing two different, well known artificial intelligence/machine learning algorithms along with using the synthetic data creation tool SMOTE in order to overcome the challenge of having highly imbalanced data. The two methods chosen for this study are Logistic Regression and Deep Learning otherwise known as a Deep Neural Network.

Keywords

AI, Artificial Intelligence, ML, Machine Learning, LR, Logistic Regression, Neural Network, NN Deep Learning, DNN, Deep Neural Network, SMOTE, Synthetic Minority Oversampling Technique

Conference'04, Month 1–2, 2004, City, State, Country.

1. INTRODUCTION

Financial fraud is best described as financial or personal gain through means of criminal deception. There are two primary ways that an institution would use to fight financial fraud. The first being fraud prevention, and the second being fraud detection. Fraud prevention aims to prevent any instance of fraud from happening, whereas the goal with fraud detection is to accurately recognize fraudulent activity when it has occurred and when it has not occurred. A fraud detection system does this by checking every transaction made regardless of any prevention systems, in an effort to stop the fraudulent transaction from processing.

Fraud in 2020 is expected to reach tens of billions in losses. While traditional methods of financial fraud can be done through mail, wire, or phone, the most concerning type of fraud is done using the Internet. Due to the ease of anonymity along with global access to the Internet, fraud is easier than ever to commit.

When it comes to financial fraud, there exists an inherent difficulty which relates to the limited amount of data. When a transaction is processed, a record of the transaction amount, date and time, address, merchant category code (MCC), and acquirer number of the merchant are stored. Not only is this a limited amount of data, it becomes extremely difficult to find patterns when there exists countless numbers of e-commerce websites. Additional challenges arise when the landscape of what fraud is and isn't changes periodically. Due to privacy and security limitations, sharing fraud detection ideas and creating new implementations proves to be difficult. Datasets that consist of real transaction data can be hard to come by and many are often censored. Many studies are done using synthetically created data [1].

With fraud being such a large issue, it is extremely important to find a reliable solution. The solution in the end will need to determine if a transaction is either one of two scenarios: fraud or not fraud. There are multiple different implementations of fraud detection systems including, but not limited to: Machine Learning-based systems, Artificial Intelligence-based systems, rule-based systems, and location-based systems. Many fraud detection techniques can be used in unison for an overall better system. This paper will focus on two different implementations.

The first being Logistic Regression, and the second being Deep Learning otherwise known as a Deep Neural Network (DNN).

An important part of detecting fraud lies in finding a pattern within a person's transaction history and determining whether or not the new transaction is fraud or not. Everyone who has used a credit card will have a data profile that consists of behavioral characteristics. Characteristics could include, but are not limited to: the average date and time they buy an item, a frequent geographical location, and even a frequent MCC [6]. By building such a profile, it allows the fraud detection system to find deviations from the profile.

The primary research question in regards to this study is as follows: Which artificial intelligence implementation is best for identifying fraudulent transactions? Using Logistic Regression, Artificial Neural Networks, and Deep Neural Networks will allow for a plethora of metrics to be analyzed.

The significance of this study is extremely relevant to today's world. Financial fraud has affected countless people and results in tens of billions of dollars lost each year with it only getting worse. Many people's credit scores are ruined by other people opening credit cards in their name, which in turn makes it extremely difficult for the affected party to acquire loans, mortgages, etc. and could take them a lifetime to improve their credit score. There's not a single doubt that improving financial fraud detection systems will go a long way in protecting the wellbeing of countless individuals.

2. LITERATURE REVIEW

The primary goal of this study is to determine, out of two different types of artificial intelligence, which is best at detecting fraudulent transactions. While in the past, ways of determining financial fraud were mostly rule-based or location-based, due to the advent of the Internet, e-commerce businesses have made detecting fraud an extraordinary issue, resulting in billions of dollars lost each year to fraud. Financial institutions around the world are trying to develop new ways of combating fraud, one of which is by the use of artificial intelligence. Many researchers have done studies using data mining, machine learning, artificial neural networks, but it doesn't seem common to compare multiple methodologies in order to determine which method is the most reliable.

In a previous study, when comparing Logistic Regression to an Artificial Neural Network, it was shown that the ANN was ultimately more accurate and had more fraudulent transactions caught than LR. Fraudulent transactions caught remained high as long as the data remained unbiased. The more the data became biased, the less accurate both LR and ANN became. This is due to the increase in training data. The more training data used, the more accurate the models became, but overall the algorithms caught less fraud and when determining fraud detection, catching a fraudulent transaction is more important than accuracy. Additionally, this study's metrics only included accuracy and number of frauds detected.

Additionally, when reviewing past research, researchers had implemented a real-time fraud detection system in which it used an Artificial Neural Network. They began by identifying four different types of fraud. These include the following: bankruptcy fraud, application fraud, behavioral fraud, and lastly, theft/counterfeit fraud. When using Machine Learning, there is an important characteristic of the algorithm, which will affect how it is created. This characteristic relates to whether the algorithm is supervised or unsupervised. Both characteristics use training data, but a key difference lies in the class labels. A supervised approach will include class labels within the training data. In regards to financial fraud, the training data will already include whether or not the transaction is fraud. With the unsupervised approach, the training data will be classless. In other words, the algorithm will not know if the transaction is fraud and will have to learn whether it is or not. The algorithm for this study was supervised. What makes this research unique, is that a simulated annealing technique was used. Traditionally, annealing is the rapid heating and cooling of metal, which in turn, makes the metal stronger. The simulated annealing technique is adapted from the Metropolis-Hastings algorithm, which was invented by M.N. Rosenbluth in 1953 [1]. Simulated annealing works by "heating" the system at temperature T in order to generate a random solution. While the algorithm is running, T decreases with each iteration and forms a model. As the system "cools" slowly, until the minimum T value is reached, a model is created at each iteration until the system reaches a global minima [1]. Training the algorithm using simulated annealing until the training reached a 1% error took days, but ultimately pays off. They were able to achieve a 92% correct fraud detection while evaluating transactions in real-time [1].

Ultimately, this technique, using simulated annealing with an Artificial Neural Network, seems to be proven worthwhile, but a shortcoming to this research lies in that there wasn't any other algorithm compared. The results gathered from this research are promising, but it is unclear whether this approach is truly superior to other algorithms.

The research done in this paper may not use advanced training techniques such as simulated annealing, but this study's biggest strength will lie in the metrics gathered by comparing two very different algorithms. In this way, we will truly be able to compare multiple approaches in order to achieve an understanding of which algorithm is best.

3. METHODOLOGY

The proposed system will be designed to take in data and provide metrics in determining which algorithm proves to be the most effective in identifying fraud. The system design consists of five sections: Exploratory Data Analysis (EDA), data visualization, data cleaning, feature selection, and lastly, data model. All code written for this study was done with Python in JupyterLab using the Anaconda Navigator.

3.1 Linear Regression Methodology

This study uses the Scikit-Learn library to implement the Linear Regression algorithm along with Numpy, Pandas, Matplotlib, Seaborn, and Imbalanced-Learn for data processing and visualization.

3.2 Deep Learning Methodology

This study uses the Keras library with the TensorFlow backend to implement the Deep Learning algorithm along with Numpy, Pandas, Matplotlib, Seaborn, Scikit-Learn, and Imbalanced-Learn for data processing and visualization.

3.3 Exploratory Data Analysis

The purpose of exploratory data analysis is to find any characteristics within the given data set. This can be done by essentially querying the data to find any connections between the different data points. Another method of EDA is to plot data in order to visually identify correlations amongst data points. While this study uses both methods of EDA, it will focus more on data visualization rather than querying. EDA is a crucial step in dealing with large amounts of data due to it allowing new discoveries to be made. This can lead to new data being added and unrelated data being deleted for a more accurate model. Upon completing EDA, new hypotheses may arise.

3.4 Data Visualization

Data visualization is an important step in finding relationships between data points. By using bar plots, histograms, scatter plots, and much more, it will be possible to determine which data points are more useful than others as well as potentially find new connections which lead to new, more useful data points being added.

3.5 Data Cleaning

Data cleaning is an important step in making sure the data set does not include incomplete, incorrect, inaccurate or irrelevant parts of data. After modifying, replacing, or deleting bad data, the new data set will be ready to undergo feature selection.

3.6 Feature Selection

It's important to pick the data points that will result in the best possible outcome. One may try to manually pick data points to test in order to determine which is the most useful, but there are also automated ways to do this. This study attempts to use Recursive Feature Elimination (RFE). What this does is RFE continuously creates a model and chooses either the best or worst performing features. It then sets that feature aside and creates a new model with one less feature. Once RFE has exhausted all features, it will rank each feature making it much easier to determine which data points to use and which to discard.

3.7 Data Model

The data model is the artificial intelligence algorithm that is chosen for a specific task. For this study, the data models used include Logistic Regression and Deep Learning / Deep Neural Network. When implementing the model, the data set will be split into two different categories, which include training data and test data. For this study, 30% of the data is training data and 70% is test data. Once the model has been trained with the training data, the test data can be evaluated on the trained model.

4. IMBALANCED DATA

A noteworthy challenge in the "Big Data Era" has to do with data imbalance. Finding extremely large data sets online is trivial nowadays, but unfortunately many of them are imbalanced. A data set is imbalanced when a few classes are abundant while other classes have just limited representation. The data mining community has named this the "Class-imbalance problem" and is inherently in a large majority of data sets [5]. For this study, the data is highly imbalanced towards non-fraudulent transactions. This can be seen clearly below.

Table 1. Comparing Fraud vs. No Fraud Transactions

No Fraud	6,354,407
Fraud	8,213

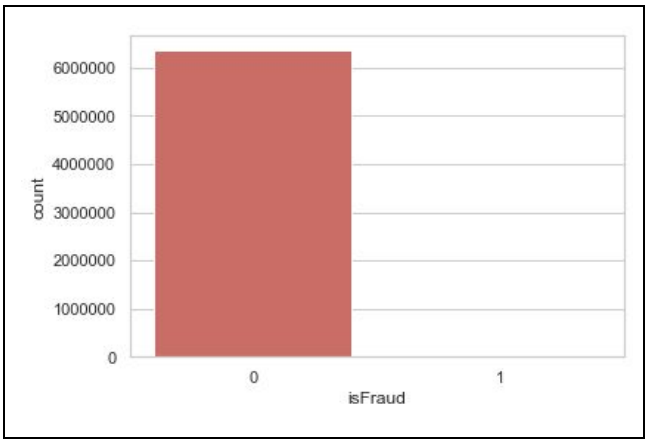

Figure 1. Chart Showing Data Imbalance

This study primarily focuses on binary classification where zero (0) is no fraud and one (1) is fraud. For a binary-class data set, it is considered imbalanced when the minority class is extremely under-represented compared to the majority class [5].

Data imbalance, if left unchecked, can lead to costly mistakes and even notable consequences throughout data analysis especially in regards to classification tasks. The classification algorithms will favor the majority class due to the skewed distribution which in turn means the concepts of the minority class will not have been learned properly [5]. If this happens, the standard classifiers, which don't consider data imbalance, will tend to misclassify the minority samples as majority samples. This will lead to poor classification performance [5]. In regard to this study, this could mean fraudulent transactions being mistaken as legitimate, which would have serious repercussions.

To make sure the Logistic Regression and Deep Learning algorithms learn from the data properly, something has to be done to the data to rebalance it. One might choose to oversample, undersample, or a combination of both. Oversampling is essentially when data is added to the minority class, whereas with undersampling, the majority class has data removed. For this study, a combination of both oversampling and undersampling is used. It is a type of synthetic data creation known as Synthetic Minority Oversampling Technique (SMOTE). This is a statistical technique for increasing the number of cases in the data set in a balanced way. This implementation of SMOTE does change the number of majority cases as can be seen below.

Table 2. New Data After SMOTE

Length of oversampled data	8,896,080
Number of no fraud	4,448,040

Number of fraud	4,448,040
Proportion of no fraud data	0.5
Proportion of fraud data	0.5

5. ALGORITHMS

Before focusing on the study of Logistic Regression and Deep Learning, it is important to compare them with other techniques to show that LR and DL are suitable for finding fraudulent transactions.

Table 3. Comparison of Fraud Detection Techniques

Technique	Advantage	Disadvantage
K-Nearest Neighbor	Easy to implement and can be used to detect anomalies in the target instance.	Memory limitations.
Hidden Markov Model	Detect fraudulent transactions at the time of transaction.	unable to detect fraud with small data sets.
Decision Tree	Can handle nonlinear transactions.	Unable to detect fraud at the real time of transaction.
Deep Learning	Analysis and learning of massive amounts of unsupervised data.	Long training times.
Logistic Regression	Easy to implement and extract useful data.	Unable to detect fraud at the real time of transaction.

5.1 Logistic Regression

Logistic Regression (LR) is a machine learning classification algorithm that, when used, can predict the probability of a categorical dependent binary variable. This variable must be binary due to this particular LR algorithm being a Binomial Logistic Regression. This algorithm can be classified as a statistical classification model and employs a logistic curve for fraud detection. The formula for a univariate logistic curve is as follows.

$$p = \frac{e^{(c_0+c_1x_1)}}{1+e^{(c_0+c_1x_1)}}$$

Figure 2. Logistic Curve Formula

This logistic curve formula will only give a value between zero (0) and one (1), so in the end it can be determined as the probability of a class association. In order to perform the regression, the logarithmic function can be applied to the logistic function below.

$$loge\left(\frac{p}{1-p}\right)$$

Figure 3. Logistic Function

In this case, *P* is the probability of a tuple being in a class while 1-*P* is the probability of a tuple not being in a class [4].

5.2 Deep Learning

In the present day, Deep Learning is a state of the art technology. It is based off of machine learning, which in turn, is based off of an artificial neural network. Deep Learning is applicable in a myriad of fields from image recognition, face recognition, natural language processing, autonomous systems, detection systems, and much more [4].

A standard neural network consists of simple, connected processors called neurons. Each of these neurons ultimately produce real-value activations. In the input layer, neurons get activated through sensors perceiving the environment, whereas neurons not activating from the environment get activated from weighted connections from previously active neurons. Some neurons can change the environment by triggering certain actions. Now, depending on the problem and how the neurons are connected, desired behavior may require long chains of computation stages. These stages have the ability to transform the aggregate activation of the network [6].

The given picture (Figure 4) provides a simple illustration of what a deep neural network looks like. Each circle in the figure represents a neuron while each connected line is a weighted connection. Each individual connection has its own weight. For clarification, this figure is not an accurate representation of the actual layers used in this study. The figure is merely to provide a visualization of what the layers might look like.

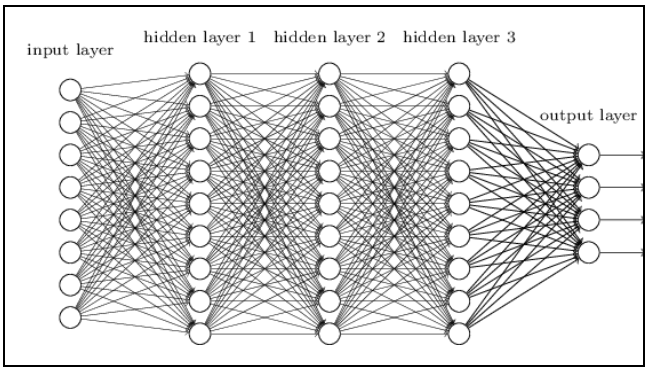

Figure 4. Deep Neural Network

6. RESULTS ANALYSIS

In this section, both algorithms will be evaluated for their effectiveness when given a data set balanced using SMOTE. For consistency amongst the two algorithms, the same exact data processing and cleaning has been performed. The data set consists of over eight million rows of data, seen in Table 2, with 9 columns for X and 1 column for Y. A snippet of code showing this can be seen below.

Table 4. Code Snippet Showing X and Y Columns

```
cols = ['amount', 'newbalanceDest', 'newbalanceOrig',
        'oldbalanceDest', 'oldbalanceOrg', 'step',
        'type_CASH_OUT','type_TRANSFER', 'HourOfDay']
X = os_data_X[cols]
Y = os_data_Y['isFraud']
```

Both Logistic Regression and Deep Learning algorithms are firstly trained using training data. The training data for both algorithms use 30% of the data. Once trained, both algorithms are tested with the test data, which contains 70% of the overall data. The following figures are representative for both algorithms since the same data processing was performed.

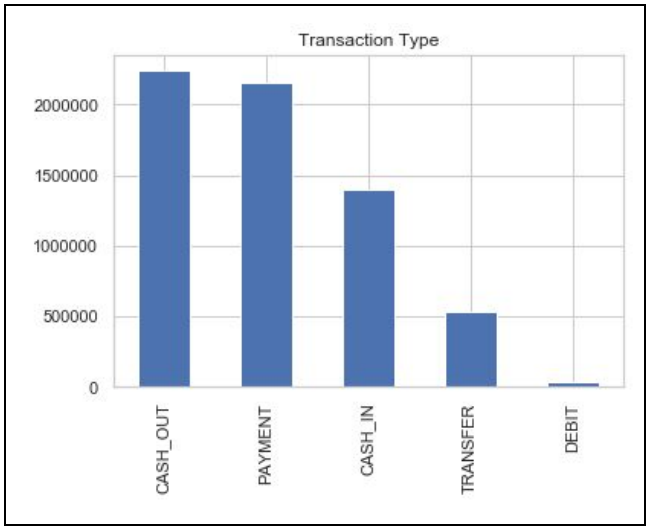

Figure 5. Number of Transaction Per Type

Table 5. Proportion of Fraud for Each Type

Type	isFraud
CASH_OUT	0.001840
PAYMENT	0.000000
CASH_IN	0.000000
TRANSFER	0.007688
DEBIT	0.000000

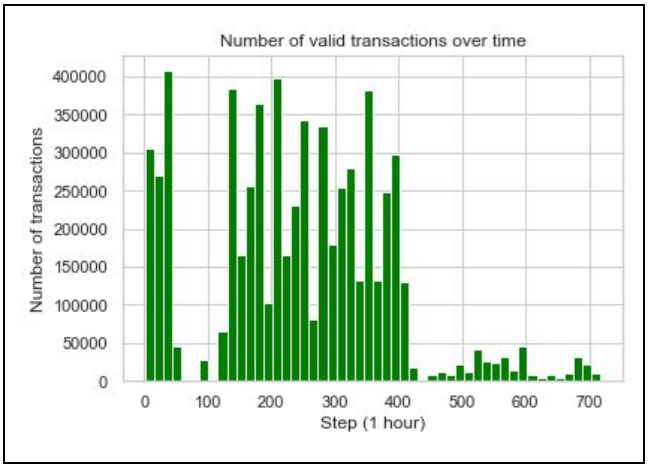

Figure 6. Number of Valid Transactions During Step

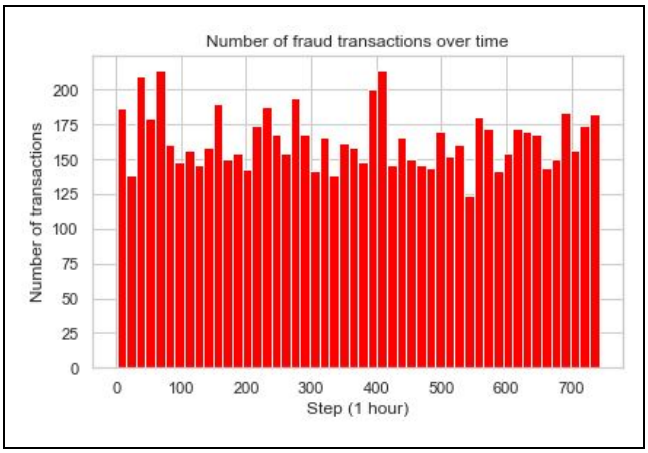

Figure 7. Number of Fraud Transactions During Step

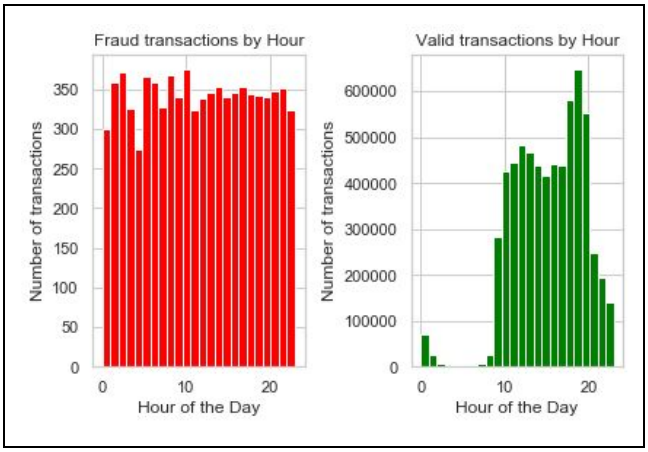

Figure 8. Fraud and Valid Transactions By Hour

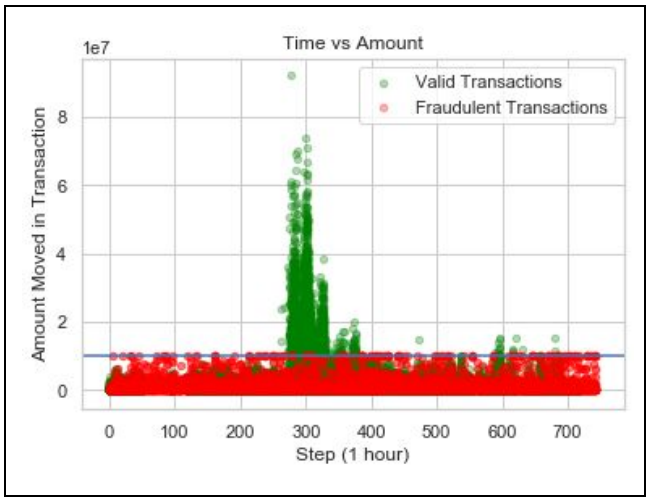

Figure 8. Time vs. Amount

What can be seen in this information is that in Table 5, the only types of payment that need to be considered are CASH_OUT and TRANSFER as they are the only two to trigger the isFraud flag. It can be seen in Figure 8 that fraudulent transactions occur fairly consistently throughout the day whereas with valid transactions there is a noticeable dip between hours 0 through 10. This could certainly be due to most people being asleep at that time and the algorithms should be able to learn that, which is why an 'HourOfDay' column is added to the data. It can also be clearly shown that in Figure 8, fraudulent transactions do not exceed 10,000,000 dollars.

6.1 Logistic Regression Results

Table 6. Logistic Regression Confusion Matrix

Actual	**No Fraud**	1,238,765	95,140
	Fraud	130,425	1,204,494
		No Fraud	**Fraud**
	Predicted		

Table 7. Logistic Regression Metrics

True Positive	1,204,494
False Positive	95,140
True Negative	1,238,765
False Negative	130,425
Accuracy	0.92
Precision	0.92
Recall	0.92
F1-score	0.92

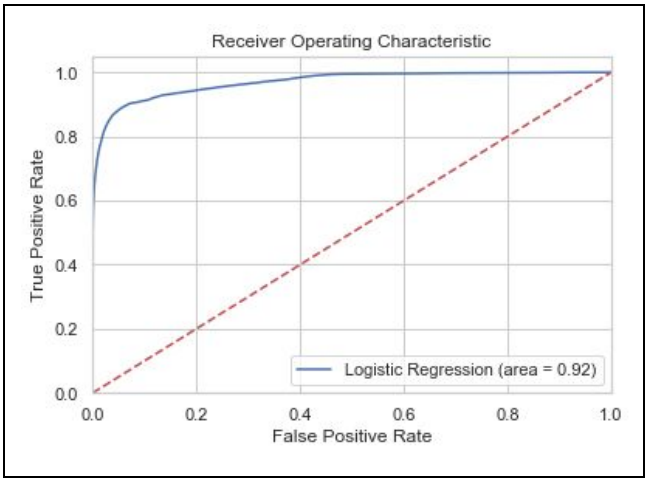

Figure 9. Logistic Regression ROC

6.2 Deep Learning Results

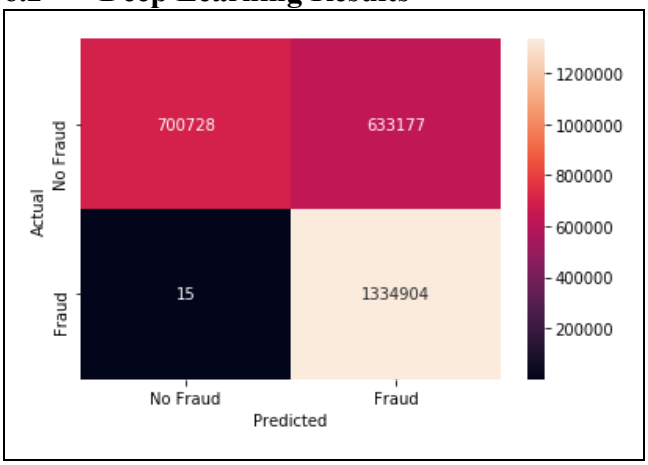

Figure 10. Deep Learning Confusion Matrix With Heatmap

Table 8. Deep Learning Metrics (100 epochs)

True Positive	1,334,904
False Positive	633,177
True Negative	700,728
False Negative	15
Accuracy	0.76
Precision	0.68
Recall	0.99

F1-score	0.81

Table 9. Deep Learning Metrics (20 epochs)

True Positive	1,334,919
False Positive	820,061
True Negative	513,844
False Negative	0
Accuracy	0.69
Precision	0.62
Recall	1.0
F1-score	0.77

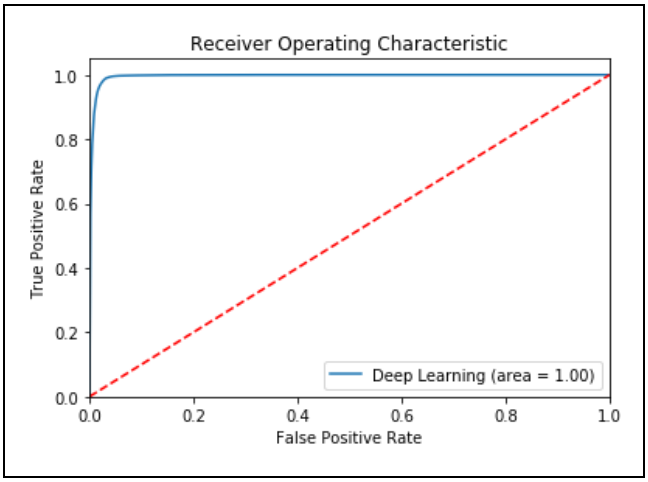

Figure 11. Deep Learning ROC (100 epochs)

7. CONCLUSION

In conclusion, based on the metrics gathered, when comparing Table 7 and Table 8, it can be seen that at face value the Logistic Regression algorithm is more accurate, more precise, and more reliable compared to the Deep Learning algorithm. The LR model is able to correctly identify over 2.4 million cases with only around 225,000 incorrect whereas the Deep Learning algorithm only correctly identifies around 2 million cases with over 600,000 incorrect.

One interesting point to note is that the Deep Learning algorithm is far superior at avoiding false negatives, but struggles with false positives. But when money is involved, it'd be more favorable to have a false positive than a false negative. A good model will produce an ROC curve that quickly goes from zero (0) to one (1). Both algorithms have produced a good ROC curve and AUC score, with Deep Learning being considerably quicker to reach 1.0. The Logistic Regression model overall can be trusted when predicting the class, but with Deep Learning's low precision and high recall, it is clear that the model can detect the class, but has trouble with other classes. This can seemingly be improved upon by increasing the number of epochs. An epoch being when the data is passed from the input layer to the output layer, then back to the input layer.

Looking at Table 8 and Table 9, it's clear that with an increase in epochs, there is an increase in accuracy, precision, and f1-score with a very slight decrease in recall. One of the biggest drawbacks to the Deep Learning algorithm is how long it takes to train the model. Where Logistic Regression only takes a minute or so, just 20 epochs in Deep Learning takes upwards of an hour, with 100 epochs taking around six (6) hours. This could be dependent on the hardware it's running on, but testing on different machines is outside the scope of this study.

Back in the Literature Review, it is mentioned that in a previous study, researchers trained their Artificial Neural Network for multiple days straight in order to achieve their desired error percentage. Considering this Deep Learning model was only trained for a maximum of around six (6) hours, one could have confidence that, if given enough time to train, the Deep Learning algorithm would eventually produce greater results than Logistic Regression.

8. REFERENCES

[1] Azeem Ush Shan Khan, Nadeem Akhtar, and Mohammad Naved Qureshi. 2014. Real-Time Credit-Card Fraud Detection using Artificial Neural Network Tuned by Simulated Annealing Algorithm . *Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing* (2014), 113–121.

[2] Pumsirirat, Apapan and Liu Yan. 2018. Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine. *(IJACSA) International Journal of Advanced Computer Science and Applications* (2018), 18-25.

[3] Schmidhuber, Jürgen. 2014. Deep Learning in Neural Networks: An Overview (October 8 2014), 1-88.[4]

[4] Suraj Patil, Varsha Nemade, and PiyushKumar Soni. 2018. Predictive Modelling For Credit Card Fraud Detection Using Data Analytics. *Procedia Computer Science 132* (2018), 385–395.

[5] Wang, Shoujin, Wei Liu, Jia Wu, Longbing Cao, Qinxue Meng, and Paul J. Kennedy. 2016. Training Deep Neural Networks on Imbalanced Data Sets. *2016 International Joint Conference on Neural Networks (IJCNN)* (2016), 4368-4374.

[6] Y. Sahin and E. Duman. 2011. Detecting credit card fraud by ANN and logistic regression. *2011 International Symposium on Innovations in Intelligent Systems and Applications* (June 2011).