A Proposed Model for Card Fraud Detection Based on CatBoost and Deep Neural Network

Nghia Nguyen
University of Economics and Law, Ho Chi Minh City
Truc Duong
University of Economics and Law, Ho Chi Minh City
Tram Chau
University of Economics and Law, Ho Chi Minh City
Van-Ho Nguyen
University of Economics and Law, Ho Chi Minh City
Trang Trinh
University of Economics and Law, Ho Chi Minh City
Duy Tran
University of Economics and Law, Ho Chi Minh City
Thanh Ho (thanhht@uel.edu.vn)
University of Economics and Law, Ho Chi Minh City

Research Article

Keywords: CatBoost, card fraud detection, Deep Neural Network, deep learning, machine learning, user separation

Posted Date: April 19th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1560840/v1

Abstract

Rapid development of technology has digitized customers' payment behavior towards a cashless society. To some extent, this has created a feast for miscreants committing fraud. According to Nilson's Report (2020), the global fraud loss is projected to reach over $35 billion in 2025. Thus, the need for a novel method to prevent this menace is undisputed. Studies in fraud detection in cards abound with various machine learning-based techniques. However, many of them were conducted on modest-sized sample with a few familiar features and mainly focused on transaction-based approach to detect fraud. These models fail to satisfy low false alarm rates when being fed with large-scale data streams. In this paper, an innovative model was proposed to improve classification accuracy based on the idea of user separation, in which we divided users into old and new people before applying CatBoost and Deep Neural Network on each category, respectively. The experimental result shows that our model achieved better as we obtained AUC score of 0.97 (CatBoost) and 0.84 (Deep Neural Network).

1 Introduction

E-commerce has flourished for the last decades. As more and more people get used to online transactions, it has contributed to the prevalence of the use of card payments. This prevailing emergence of spending behavior has, unfortunately, become an ideal condition for the increase in fraudulent activities. Oxford Dictionary has defined fraud [1] as wrongful or criminal deception intended to result in financial or personal gain. To simply put, fraud detection is a process of identifying unusual behaviors of cardholders when comparing to their prior card usage profile. Based on such differences, the alert is sent if the target transactions have a probability exceeding the threshold of being classified as fraud. Fraudulent transactions are typically performed via unauthorized access to card information like credit card number [2], email address, phone number [3], and many more to steal money.

To combat card fraud, a huge effort and finance have been put into building a fraud detection system to prevent monetary loss. In order to analyze voluminous data, a variety of machine learning algorithms have been employed, ranging from classical methods like Logistic Regression [4], Support Vector Machine [5], Decision Trees [6], Hidden Markov Models [7] to state-of-the-art ones like Gradient Boosting Tree [8] and Deep Learning [9]. Among them, Gradient Boosting Tree and Deep Learning, in particular, CatBoost and Deep Neural Network (DNN) come out to be the most promising solution given their reputation for remarkable fraud detection performance. Due to the fact that time-based DNN architectures as such cannot incorporate the user's transaction history, which conversely is the advantage of CatBoost based models, we take CatBoost for granted in handling both new users and users with historic transactions at the same time while DNN is employed for detecting fraud based on data of unknown users. To make the most of their strengths, we come up with the idea of combining CatBoost and DNN, all fitted on a monthly cross-validation setup to optimally exploit historical customer data along with real-time transaction details.

The rest of the paper is structured as follows: Section 2 introduces a brief comparative review of previous relevant studies in card fraud detection. Section 3 forms the core of the paper, providing details on our proposed model. The final section concludes our research with an evaluation of the results obtained and a light touch on ideas for further investigation.

2 Literature Review

2.1. Machine learning – based approach

The approach described by Shiyang Xuan et al., in the study [10] was a combination of two types of Random Forest model: Random-tree-based random forest and CART-based random forest. Their method was to use historical transactions data based on behavior features of normal and fraud transactions. The dataset belongs to the Chinese company specializing in e-commerce with 62 attributes and more than 30.000.000 transactions, 82.000 of which are fraudulent events. They used the ratio of normal and abnormal transactions of 5:1. The result of this research was 98.67% accuracy, 32.68% precision, and 59.62% recall. They obtained the result with high accuracy, but the false positive rate is also high, raising a concern that this detection system has a high likelihood of annoying legitimate customers. Although the Random Forest model provides highly accurate results, it is only applied to a small dataset and is unsuitable for large enterprises and financial institutions. Although we can apply Logistic Regression and the SAE method to big data, they yield low accurate results and are unsuitable for practical use. That has also been confirmed in the study [11] by Aya Abd El Naby et al. In the study [12], John O. Awoyemi et al. detected fraudulent activities of credit activities by using Naïve Bayes, K -Nearest Neighbor and Logistic Regression with accuracy of 97.92%, 97.69% and 54.86%, respectively. It was clear that Logistic Regression classifiers method were still relatively low and the risk of errors increased.

2.2. Deep learning – based approach

Hassan Najadat et al., in work [13], applied BiLSTM-MaxPooling- BiGRU-MaxPooling to predict fraud. The authors also applied Naive base, Voting, Ada Boosting, Random Forests, Decision Tree, and Logistic Regression to compare the effect of each model. The dataset used for this work is special with its highly imbalanced class. To deal with the imbalanced dataset, the authors used the Random Under Sampling, Random Over Sampling, and Synthetic Minority Oversampling Technique (SMOTE). The result was that Deep Learning models with three sampling techniques achieved significantly better accuracy than Machine Learning models. The highest accuracy of the Machine Learning based models was 81%. However, when authors concatenated BiLMST-MaxPooling with BiGRU-MaxPooling with Random Oversampling technique, they gained 91.37% accuracy. From this study, we can conclude that Machine Learning algorithms alone could not solve such a complicated big dataset.

In another study [14] by Aji Mubarek Mubalaike and Esref Adali, the authors experimented with Deep Learning to detect fraud with the aim of high accuracy. They used the Ensemble of Decision Tree model (EDT), Stacked Auto-Encoders (SAE), Restricted Boltzmann Machines (RBM) with the accuracy of 90.49%,

80.52%, and 91.53%, respectively. SAE was low compared to EDT and RBM. In the future, the authors also wanted to use the Neural Network model in Deep Learning to improve accuracy.

In 2020, Yara Alghofaili et al. [15] studied the ability of Long Short-term Memory to detect credit card fraud by comparing this algorithm with Auto-encoder and traditional machine learning models such as: Logistic Regression, Random Forest, and Support Vector Machines. The limitation of the article is that authors only compared the accuracy of the models based on the training set instead of the test set, so it lacked of concrete basis to conclude whether the LSTM really had the highest accuracy or not. Another study related to LSTM was done by Ibtissam Benchaji et al (2021) [16] using data set of 594,643 transactions from 11/2012 - 04/2013 provided by a local bank. The model compared payment data with historical information. If the data matched the pattern, then the card was definitely used by the cardholder, otherwise the possibility of fraud was very high. In the future, these authors would build a model based on another variant of Recurrent Neural Networks in order to validate its competency compared with the current model.

In general, the two types of current models on fraud detection scarcely considered the importance of the real-time approach, as a result, to bridge this gap, we introduce a live binary classification method based on the combination of machine learning and deep learning to leverage the strengths of each.

3 Proposed Card Fraud Detection Model

Figure 1 depicts our four-phase approach in detecting fraud using machine learning. The process starts with collecting the data. In our case, we used the data set called IEEE-CIS provided by Vesta Corporation, which is the forerunner in guaranteed e-commerce payment solutions [17]. A good complex dataset is a backbone for any robust machine learning model to produce a plausible reality-matched output. Therefore, this sample is chosen as it was really representative, which covered almost every challenging real-life pattern for a typical fraud detection problem, i.e., massive data volume, genuine transactions outnumber fraudulent events, and diversified card-related features ranging from time delta, transaction amount, addresses to even network connection information (IP, ISP, Proxy, etc) and digital signature (UA/browser/os/version, etc) associated with transactions.

In the second phase, we conducted a variety of methods to preprocess the data. The first thing is the minification step to reduce memory usage. This helps us save a lot of resources when building the prediction model and speed up the training process. After that, we conduct an exploratory analysis to inspect data for patterns, trends, or relationships between variables and between the target column and other variables. Then we experimented many ways to pick out the most suitable techniques for feature transformation and feature selection. The main part of the preprocessing stage is to separate users into new and old group through the process of establishing card identification based on given card-related features in the dataset. In the third phase, after processing categorical and numerical data into the suitable form, we deploy DNN on unknown users and CatBoost on the known users before combining into the final results in the last phase. Details for each process will be clarified in the following sections.

3.1 Data Source

The IEEE-CIS dataset consists of two files, namely transaction and identity joined by TransactionID, with 433 features and 590,540 instances in total. They are real-world transactions provided by Vesta Corporation, a forerunner specializing in guaranteed e-commerce payment solutions. Even though a simple glossary is provided, the meaning of each feature is quite obscured because they are all masked without a pairwise dictionary for the purpose of privacy protection agreement. To clearly manifest, a table of features in transaction and identity set based on explanations of Vesta is given in Table 1 as follows:

Table 1. Data description

Transaction	
Feature	**Description**
TransactionDT	Timedelta from a given reference DateTime (not an actual timestamp)
TransactionAMT	Transaction payment amount in USD
ProductCD	Product code, the product for each transaction
card1 - card6	Payment card information, such as card type, card category, issue bank, country, etc.
addr	Address
dist	Distance P_ and (R__)
C1-C14	Counting, such as how many addresses are found to be associated with the payment card, etc. The actual meaning is masked.
D1-D15	Timedelta, such as days between previous transactions, etc.
M1-M9	Match, such as names on card and address, etc.
Vxxx	Vesta engineered rich features, including ranking, counting, and other entity relations.
Identity	
DeviceType	Type of machine customer uses
DeviceInfo	Information of machine
id_01 - id_11	Numerical features of identity, such as device rating, ip_domain rating, proxy rating, behavioral fingerprint-like account login times/failed to login times, how long an account stayed on the page, etc.

The business logic behind binary classification, according to the owner of the dataset, is that a transaction is denoted as “isFraud=1” when there is reported chargeback on the card, and all transactions posterior to it associated with a user account, email address, etc., are labeled as fraud too. If the cardholder did not report within 120 days then those suspicious transactions are automatically

considered legitimate (isFraud=0). In other words, once a card has been reported as fraud, that account will be converted to isFraud=1. Therefore we are predicting fraudulent clients rather than fraudulent transactions.

3.2 Data Preprocessing

Exploratory Analysis. In general, transactions were recorded from November 30th, 2017 to May 31st, 2018 as depicted in Figure 2. When doing exploratory data analysis, we notice that approximately 3,5% of train transactions are fraud with more than 95% of columns have missing values, which is a normal real-world pattern for this fraud detection task.

To deal with the imbalance between the number of fraudulent transactions and the non-fraudulent transactions, we apply SMOTE method to increase the number of fraudulent transactions many times using the KNN algorithm. Specifically, a data point will be randomly selected from the pool of fraudulent transactions and determine the closest neighbors to this point, the number of fraudulent transactions is further increased between the selected point and its neighbors.

After doing multivariate analysis, we found that the number of fraudulent transactions is high in products of category W or C, paid by debit card, credit card, visa or master card. Cards like American Express, discover card have very few or even no fraudulent transactions in the case of charge cards because they are not as commonly used as other cards. In addition, fraud is associated with users with email domains of gmail.com or hotmail.com, using computers whose operating systems are Windows 7 or Windows 10 operating systems, or, if the transactions are made over the phone, the fraud occurs frequently on phones that normally use Chrome 63.0 or generic mobile safari. The probability of a fraudulent transaction when done by computer or by phone is relatively the same. For variables whose values have been encoded in the form T/F (M1 to M9 minus M4, id_35 to id_38) or New / Found / Not Found (id_15, id_16, id_28, id_29), fraudulent transactions are dominant at observation whose values are T and Found.

Feature Transformation. Most of the variables have left skewed distribution. Some variables need to undergo logarithmic transformation to return to the normal distribution, such as the TransactionAmt in Figure 3 as belows:

The data set used has a lot of null values - NaN, so any columns that have the percentage of missing values above 95% will be discarded because they do not contribute much to the model's performance. For the numerical features, they will be imputed with 0 or the mean while for the categorical features, each blank space is filled with the word "Unknown" and treated as a new separate category. Since the machine learning model only accepts numerical variables as input, categorical features will be converted to numbers through Label Encoding.

Feature Engineering. This is the major part in our process, where we will start splitting customers into known and unknown groups. Firs, the initial data set will be divided into two parts: the training set and the test set with a ratio of 7:3. Suppose one user uses multiple cards for many different transactions.

Therefore, it is necessary to define groups of cards based on the associated identifier card properties (card1, card2, card3, card4, card5, card6, productCD) with the CardID_D1 column as shown in Figure 4 below.

Next, we continue to separate the cardGroups into cardIDs based on the V307 feature, which is important in identifying cardIDs.

Table 2. First five rows of number of transactions corresponding with each card group

STT	cardGroup_name	Counts
0	15775481.0150.0mastercard102.0credit-129S	1414
1	9500321.0150.0visa226.0debit84W	480
2	7919194.0150.0mastercard166.0debit-92W	439
3	7919194.0150.0mastercard166.0debit-124W	282
4	7919194.0150.0mastercard202.0debit-34W	242

It can be seen that each color has been marked as a cardID in Figure 5 belonging to each cardGroup. And V307 is the cumulative result of the TransactionAmt value of the previous transaction. Next is the stage of identifying the customers based on the customer identification information (TransactionAmt, id_19, id_20) – assuming id_19, id_20 are information of the IP address and cardID, the result used to separate new/old users is illustrated in Figure 6 and 7 as follows:

User separation is performed for both the training set and the test set. Those identifiers presented in both data sets will be recognized as old customers, otherwise new customers. The purpose of this is to train the model to well-identify new and old users so that once that person is reported as fraudulent, subsequent transactions involving this user identifier will also be labeled “isFraud=1”.

Feature Selection. Picking the right set of features as inputs to a model is one of the key contributions to our achieved performance. In the first place, we use Principal Component Analysis (PCA) technique to reduce the number of prefix V variables from 339 down to 30 most important ones. This method is based on the observation that the data are not normally distributed randomly in space but are often distributed near certain special lines or planes. PCA considers a special case where such special planes have linear form as subspaces.

For DNN, the input variables include: categorical variables, namely ProductCD, card1-card6, addr1, addr2, P_emaildomain, R_emaildomain, M1-M9 (through Label Encoding process) and numerical variables not prefix V and id_ (through normalization to achieve zero mean and zero variance). For CatBoost, the input variables are not the following: TransactionID, TransactionDT, isFraud, discarded V variables after PCA.

3.3 Model

Our model is a combination of CatBoost and Neural Network as base learners. Their predictions on overlapping and non-overlapping parts respectively will be combined into a single output. We use CatBoost to see if we can improve the prediction rate for overlapping users. CatBoost is a powerful gradient boosted decision tree (GBDT) in classification tasks involving big data. Two innovative qualities of CatBoost are the automatic handling of categorical values, and strong performance relative to other GBDT implementations. CatBoost uses the ordering principle called Target-Based where the values for each example rely only on the observed history [18]. Thus for a set of data with plentiful categorical features as IEEE-CIS dataset, we can improve our training results without spending time and effort turning categories into numbers. CatBoost is robust as it does not require extensive hyper-parameter tuning [19]. Due to these advantages, CatBoost whose parameters were described as Table 3 outperforms most other machine learning algorithms in both speed and accuracy.

Table 3. CatBoost parameters

Parameters	Parameter description	Value
learning_rate	Used for reducing the gradient step	0.07
loss_function	The metric to use in training	Log-loss
depth	Depth of the tree	8
n_estimators	The number of of trees to build before taking the maximum voting or averages of predictions	5000

For non-overlapping users, our Neural Network architecture, as given in Figure 8, consists of an input layer with the size of the number of selected features, 3 hidden layers (the respective neurons are 512 - 256 - 1), and an output layer of one neuron. The optimal parameters for our model are shown in Table 4.

Table 4. Neural Network parameters

Parameters	Parameter description	Value
learning_rate	Used for reducing the gradient step	0.0001
loss_function	The metric to use in training	binary cross-entropy
optimizer	Adam with Nesterov momentum	Nadam

4 Experimental Results And Discussion

4.1. Experimental Results

In this section, we will propose the theory to combine new and old users' predicted models to have a final general model. We used the AUC-ROC score, which stands for “Area under the ROC curve”, and accuracy

to evaluate the performance of our model. The ROC curve plots the true positive rate against the false-positive rate at different threshold settings. ROC-AUC is our primary metric when it comes to the fraud domain as it is robust to variable fraud rates and does not capture the effect of the overly large number of legitimate events in this dataset. In fact, not a single metric could best evaluate a model. As a result, besides ROC-AUC, we also used other metrics to evaluate performance close models, such as accuracy.

Accuracy (1) is the metrics indicating the proportion of right predictions among the total of examined cases.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

1

The accuracy result is quite high for both types of customer group and there is no significant difference between the two models. However, accuracy alone is proved to be less effective for severe imbalanced classes because the model's prediction results will be biased towards the majority class, which is the number of legitimate credit card transactions, affecting the predictive power of the model, and might lead to the circumstance where no fraud is figured out by the model. Therefore, to determine a good predictive model to put into practice should not rely solely on accuracy criteria.

Figure 9 shows that if the user has a higher probability of being in the range [0,2; 1], the more likely they committed fraud. However, since the prediction result is in probabilistic form with the range of values in the range [0; 1], in order to trigger a card fraud alert, the output needs to be converted to isFraud = 1 or isFraud = 0, by defining a threshold at which the transaction is labeled as fraudulent. In this case, the defined threshold should be a number greater than or equal to 0.2 and as close to 0.2 as possible for the result in each criterion. In order to determine the threshold, we evaluate the model's effectiveness with two metrics as follows:

Metric 1

ROC – AUC curve and AUC score

In principle, the closer the curve is to the point (0, 1), the more efficient the model is. Therefore, in Fig. 10, CatBoost generates an almost perfect prediction result, which is confirmed by AUC = 0.974. Meanwhile, the DNN model has a smaller area under the ROC curve but not too much difference (AUC = 0.84).

Metric 2

Precision - Recall curve and AUC score

The Precision and Recall curves intersect at the threshold of 0.2 as depicted in Fig. 11, confirming a great degree of accuracy in predicting fraud for transactions with probability greater than or equal to this threshold. This is consistent with the results presented in the distribution chart in Fig. 9. For the DNN

model, the Recall result is quite low although the Precision is relatively high, which shows that for users who are found to be commit fraud, the accuracy of the model is quite high compared to the actual results. However, the model still hasn't found all actual fraudulent users, resulting in low Recall results.

4.2. Live fraud detection architecture

Once the model passed the evaluation gate, to bring it to a higher practical level, we designed a pipeline showing how the detection result will be used when coming into deploying. The implementation is visually described through Figure 12 as follows:

Step 1: The user makes a payment for their transactions by credit card, which will be recorded and fed into a real-time processing system using Apache Flink, which is a large-scale data processing platform that can process the generated data at very high speed with low latency.

Step 2: Our proposed credit card fraud detection model will be implemented as an API and Apache Flink will call this API to process and output the results received from the model.

Step 3: If the transaction is detected to be fraudulent, the system will send the user a warning alert at the time of payment by asking whether the user who initiated the payments was the cardholder or not. If the user does not make the transaction, the user's account will be blocked, otherwise transaction is regarded as legitimate. In the event that a signal is not received from the user, the account will be temporarily blocked until the user agrees that the transaction has just been paid by the cardholder himself.

Step 4: The prediction results will be saved to the database and presented as a dashboard for analysis.

5 Conclusion

Financial fraud is a serious issue that has far-reaching consequences for the financial system and its stakeholders. In recent years, the issue has been exacerbated by an increased dependence on developing technologies. In the age of big data, traditional tactics are inadequate. This paper outlines the recent significant damage of fraud transactions for the financial industry and presents our CatBoost and Neural Network-based approach to effectively tackle this problem and improve detection efficiency. By using this method we rejected many redundant and high capacity features to bias our model. It can be seen that these algorithms are better compared to other traditional machine learning techniques. In the future, we plan to proceed with our work to make their utilization increasingly appropriate in practical real-time situations.

Declarations

Acknowledgements

This research is funded by University of Economics and Law, Vietnam National University Ho Chi Minh City.

Funding

No funding was received for conducting this study.

Authors' contribution statements

All authors contributed to proposing the model as well as the study conception and design. Material preparation, data collection, analysis, and evaluation were conducted by all authors. The first draft of the manuscript was written and commented on versions of the manuscript by all authors. All authors read and approved the final manuscript.

References

1. Oxford Learner's Dictionaries, https://www.oxfordlearnersdictionaries.com/definition/english/fraud, accessed on 16/09/2021
2. Zareapoor, M., Yang, J. (2017): A Novel Strategy for Mining Highly Imbalanced Data in Credit Card Transactions. Intelligent Automation & Soft Computing, pp. 1–7.
3. Óskarsdóttir, M., Bravo, C., Sarraute, C., Vanthienen, J., Baesens, B. (2019): The Value of Big Data for Credit Scoring: Enhancing Financial Inclusion Using Mobile Phone Data and Social Network Analytics. Applied Soft Computing, vol. 74, pp. 26–39.
4. Sahin, Yusuf, and Ekrem Duman. "Detecting credit card fraud by ANN and logistic regression." In 2011 international symposium on innovations in intelligent systems and applications, pp. 315–319. IEEE, (2011).
5. Gyamfi, Nana Kwame, and Jamal-Deen Abdulai. "Bank fraud detection using support vector machine." In 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 37–41. IEEE, (2018).
6. Gaikwad, Jyoti R., Amruta B. Deshmane, Harshada V. Somavanshi, Snehal V. Patil, and Rinku A. Badgujar. "Credit card fraud detection using decision tree induction algorithm." International Journal of Innovative Technology and Exploring Engineering (IJITEE) 4, no. 6: 66–69, (2014).
7. Robinson, William N., and Andrea Aria. "Sequential fraud detection for prepaid cards using hidden Markov model divergence." Expert Systems with Applications 91: 235–251, (2018).
8. Taha, Altyeb Altaher, and Sharaf Jameel Malebary. "An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine." IEEE Access 8: 25579–25587, (2020).
9. Raghavan, Pradheepan, and Neamat El Gayar. "Fraud detection using machine learning and deep learning." In 2019 international conference on computational intelligence and knowledge economy (ICCIKE), pp. 334–339. IEEE, (2019).
10. Shiyang Xuan, Guanjun Liu, Zhenchuan Li, Lutao Zheng, Shuo Wang, Changjun Jiang: Random Forest for Credit Card Fraud Detection. In: 15th IEEE International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China (2018).

11. Aya Abd El Naby, Ezz El-Din Hemdan & Ayman El-Sayed (2021): Deep Learning Approach for Credit Card Fraud Detection. In: 2021 International Conference on Electronic Engineering (ICEEM).
12. John O. Awoyemi, Adebayo O Adetunmbi & Samuel Oluwadare (2017): Credit Card Fraud Detection Using Machine Learning Techniques: A Comparative Analysis. In: 2017 International Conference on Computing Networking and Informatics (ICCNI).
13. Hassan Najadat, Ola Altiti, Ayah Abu Aqouleh, Mutaz Younes: Credit Card Fraud Detection Based on Machine and Deep Learning. In: 11th IEEE International Conference on Information and Communication Systems (ICICS), Irbid, Jordan (2020).
14. Aji Mubarek Mubalaike, Esref Adali (2018): Deep Learning Approach for Intelligent Financial Fraud Detection System, 3rd IEEE International Conference on Computer Science and Engineering (UBMK).
15. Yara Alghofaili, Albatul Albattah & Murad A. Rassam (2020), 'A Financial Fraud Detection Model Based on LSTM Deep Learning Technique', Journal of Applied Security Research, 1–19.
16. Ibtissam Benchaji, Samira Douzi & Bouabid El Ouahidi (2021), 'Credit Card Fraud Detection Model Based on LSTM Recurrent Neural Networks', Journal of Advances in Information Technology *Vol. 12, No. 2*, 113–118.
17. Dataset, https://www.kaggle.com/c/ieee-fraud-detection/data, accessed on 16/09/2021
18. John T. Hancock, Taghi M. Khoshgoftaar: CatBoost for Big Data: An Interdisciplinary Review. Journal of Big Data 7, Article number: 94 (2020).
19. Prathima Gamini, Sai Tejasri Yerramsetti, Gayathri Devi Darapu, Vamsi Kaladhar Pentakoti, Vegesna Prudhvi Raju: Detection of Credit Card Fraudulent Transactions using Boosting Algorithms, Journal of Emerging Technologies and Innovative Research (JETIR), Volume 8, Issue 2 (2021).

Figures

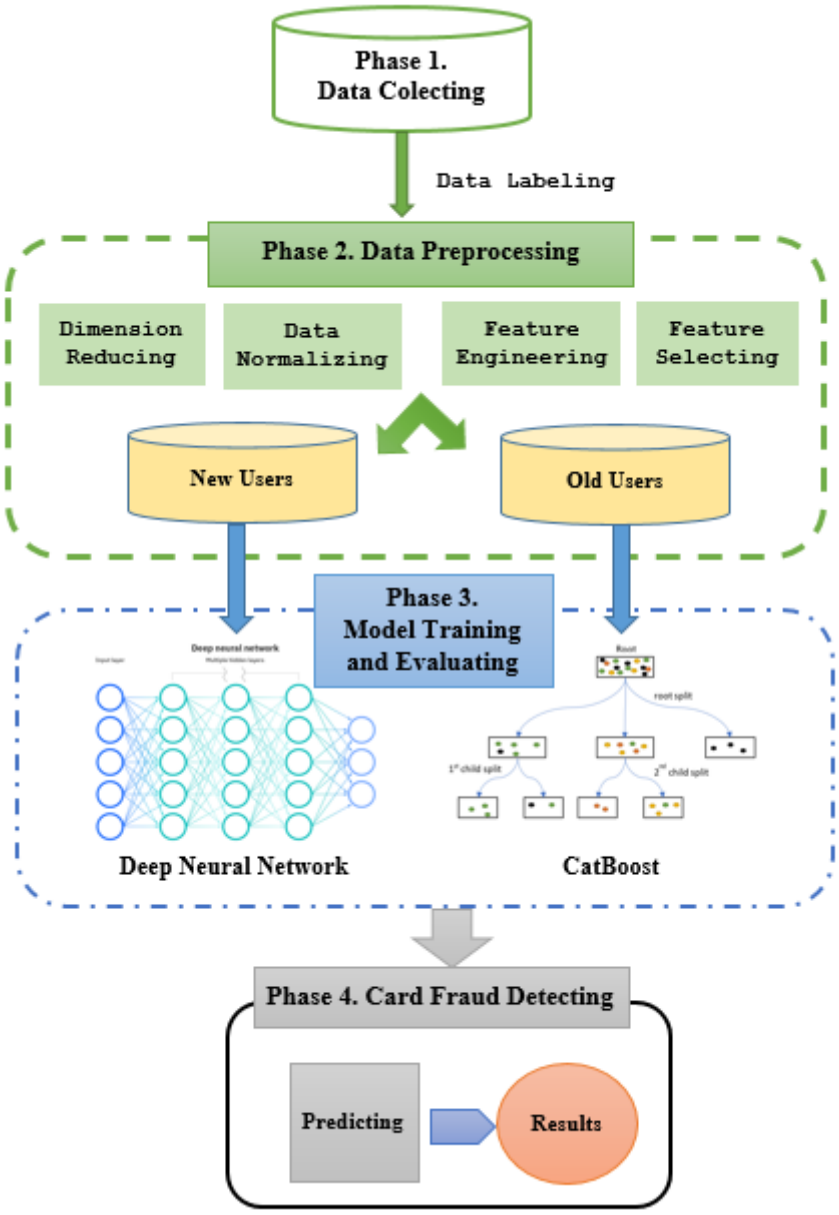

Figure 1

Proposed Card Fraud Dectection Model (Source: Authors)

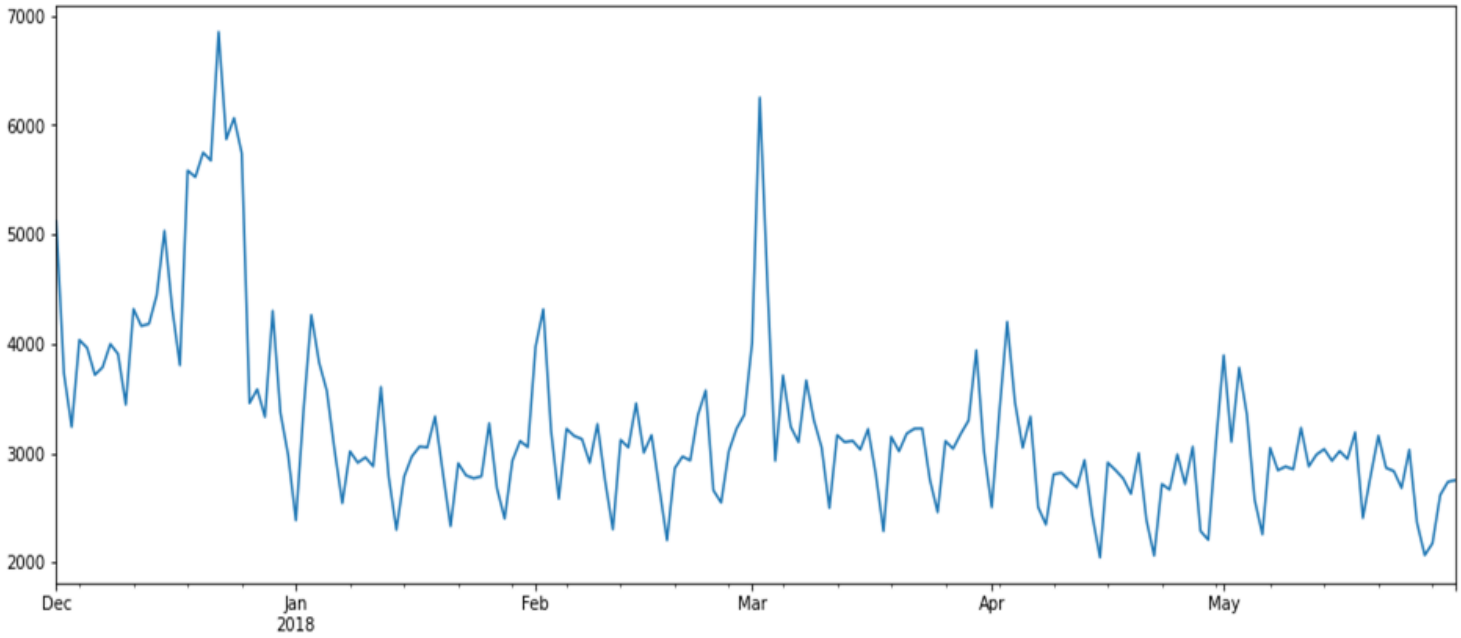

Figure 2

Monthly number of transactions

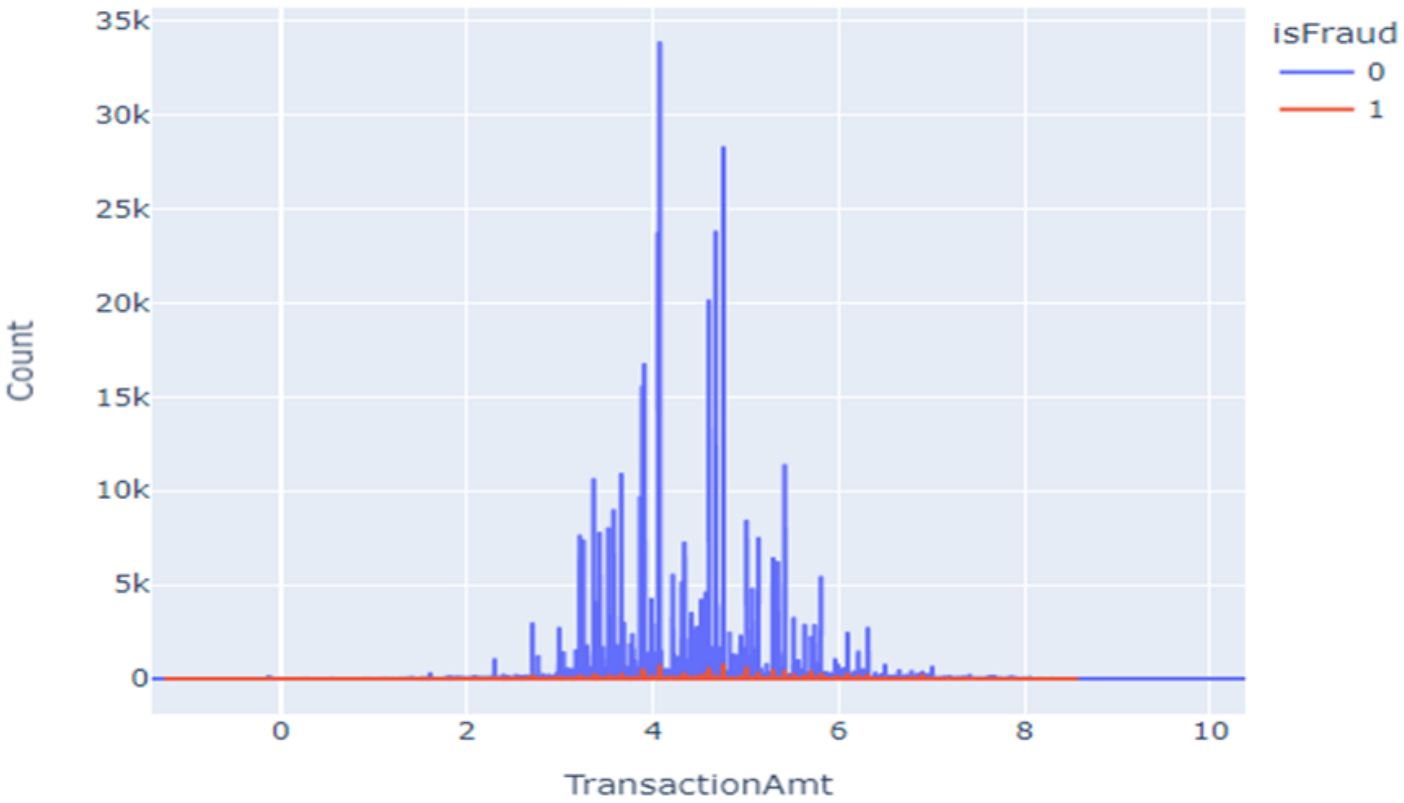

Figure 3

Distribution of TransactionAmt after log transformation

TransactionI	isFra	TransactionD	TransactionAr	ProductC	card	card	card3	card4	card5	card6	cardID_D1
3020292	0	836998	25	H	17188	321	150	visa	226	debit	17188321.0150.0visa226.0debit-9H
3019162	0	786374	35	H	17188	321	150	visa	226	debit	17188321.0150.0visa226.0debit-9H
3019999	0	832296	50	H	17188	321	150	visa	226	debit	17188321.0150.0visa226.0debit-9H
3022231	0	860364	50	H	17188	321	150	visa	226	debit	17188321.0150.0visa226.0debit-9H
3022435	0	863048	50	H	17188	321	150	visa	226	debit	17188321.0150.0visa226.0debit-9H
3529687	0	14320080	36.94	W	4090	490	150	visa	226	debit	4090.0150.0visa226.0debit-165W
3529698	0	14320218	36.94	W	4090	490	150	visa	226	debit	4090.0150.0visa226.0debit-165W
3529712	0	14320385	36.94	W	4090	490	150	visa	226	debit	4090.0150.0visa226.0debit-165W
3543113	0	14689387	58.94	W	4090	490	150	visa	226	debit	4090.0150.0visa226.0debit-165W

Figure 4

The number of transactions made by two separate customers (indicated by colors)

Transactio	Transaction	cardID_D1	V307	V307plu	V307tru	V307rou	V307plusroi	V307plusroundtr	V307trui	V307plustru	TransactionAmtti	V307plusrou	cardID
3085059	46.75	15885545.0185.0visa138.0debit-22C	51.4206	98.1706	51.42	51.421	51.421	98.171	51.42	98.17	46.75	98.17	group0_0
3085085	46.75	15885545.0185.0visa138.0debit-22C	98.1666	144.9166	98.166	98.167	98.167	144.917	98.16	144.91	46.75	144.92	group0_0
3085104	46.75	15885545.0185.0visa138.0debit-22C	144.9126	191.6626	144.912	144.912	144.913	191.663	144.91	191.66	46.75	191.66	group0_0
3139566	34.97	15885545.0185.0visa138.0debit-22C	49.7882	84.75695	49.788	49.788	84.757	84.756	49.78	84.75	34.969	84.76	group0_11
3139908	56.56	15885545.0185.0visa138.0debit-22C	84.7682	141.3307	84.768	84.768	141.331	141.33	84.76	141.33	56.562	141.33	group0_11
3139929	56.56	15885545.0185.0visa138.0debit-22C	141.3192	197.8817	141.319	141.319	197.882	197.881	141.31	197.88	56.562	197.88	group0_11
3139935	56.56	15885545.0185.0visa138.0debit-22C	197.8702	254.4327	197.87	197.87	254.433	254.432	197.87	254.43	56.562	254.43	group0_11
3139990	56.56	15885545.0185.0visa138.0debit-22C	254.4212	310.9837	254.421	254.421	310.984	310.983	254.42	310.98	56.562	310.98	group0_11
3139996	56.56	15885545.0185.0visa138.0debit-22C	310.9722	367.5347	310.972	310.972	367.535	367.534	310.97	367.53	56.562	367.53	group0_11
3140025	56.56	15885545.0185.0visa138.0debit-22C	367.5232	424.0857	367.523	367.523	424.086	424.085	367.52	424.08	56.562	424.09	group0_11
3508531	13.77	3901176.0185.0mastercard224.0credit-4	0	13.773438	0	0	13.773	13.773	0	13.77	13.773	13.77	group18491_0
3547898	78.7	3901176.0185.0mastercard224.0credit-4	13.7721	92.4596	13.772	13.772	92.46	92.459	13.77	92.45	78.688	92.46	group18491_0
3548011	78.7	3901176.0185.0mastercard224.0credit-4	92.4665	171.15399	92.466	92.466	171.154	171.154	92.46	171.15	78.688	171.15	group18491_0

Figure 5

CardID

Transaction	TransactionAr	Product	isFrau	DeviceInfo	id_	id_	id_31	cardID	groupUse	cardIDcount	UserIDcount	UserID_fraud_su	CardID_fraud_su
2987240	37.1	137785	1	Redmi Note 4 Build/I	266	325	chrome 54.0	group32724_0	group35099	3	3	3	3
2987243	37.1	137785	1	Redmi Note 4 Build/I	266	325	chrome 54.0	group32724_0	group35099	3	3	3	3
2987245	37.1	137785	1	Redmi Note 4 Build/I	266	325	chrome 54.0	group32724_0	group35099	3	3	3	3
2987779	10	23046	1	KFFOWI Build/LVY48	397	161	chrome generic	group134049_0	group15900	1	9	1	1
2987780	10	23046	1	KFFOWI Build/LVY48	397	161	chrome generic	group16817_0	group15900	8	9	8	8
2987781	10	23046	1	KFFOWI Build/LVY48	397	161	chrome generic	group16817_0	group15900	8	9	8	8

Figure 6

Customer used one card

Transaction	TransactionAr	Product	isFrau	DeviceInfo	id_	id_	id_31	cardID	groupUse	cardIDcount	UserIDcount	UserID_fraud_su	CardID_fraud_su
2987240	37.1	137785	1	Redmi Note 4 Build/I	266	325	chrome 54.0	group32724_0	group35099	3	3	3	3
2987243	37.1	137785	1	Redmi Note 4 Build/I	266	325	chrome 54.0	group32724_0	group35099	3	3	3	3
2987245	37.1	137785	1	Redmi Note 4 Build/I	266	325	chrome 54.0	group32724_0	group35099	3	3	3	3
2987779	10	23046	1	KFFOWI Build/LVY48	397	161	chrome generic	group134049_0	group15900	1	9	1	1
2987780	10	23046	1	KFFOWI Build/LVY48	397	161	chrome generic	group16817_0	group15900	8	9	8	8
2987781	10	23046	1	KFFOWI Build/LVY48	397	161	chrome generic	group16817_0	group15900	8	9	8	8

Figure 7

Customer used two or more card

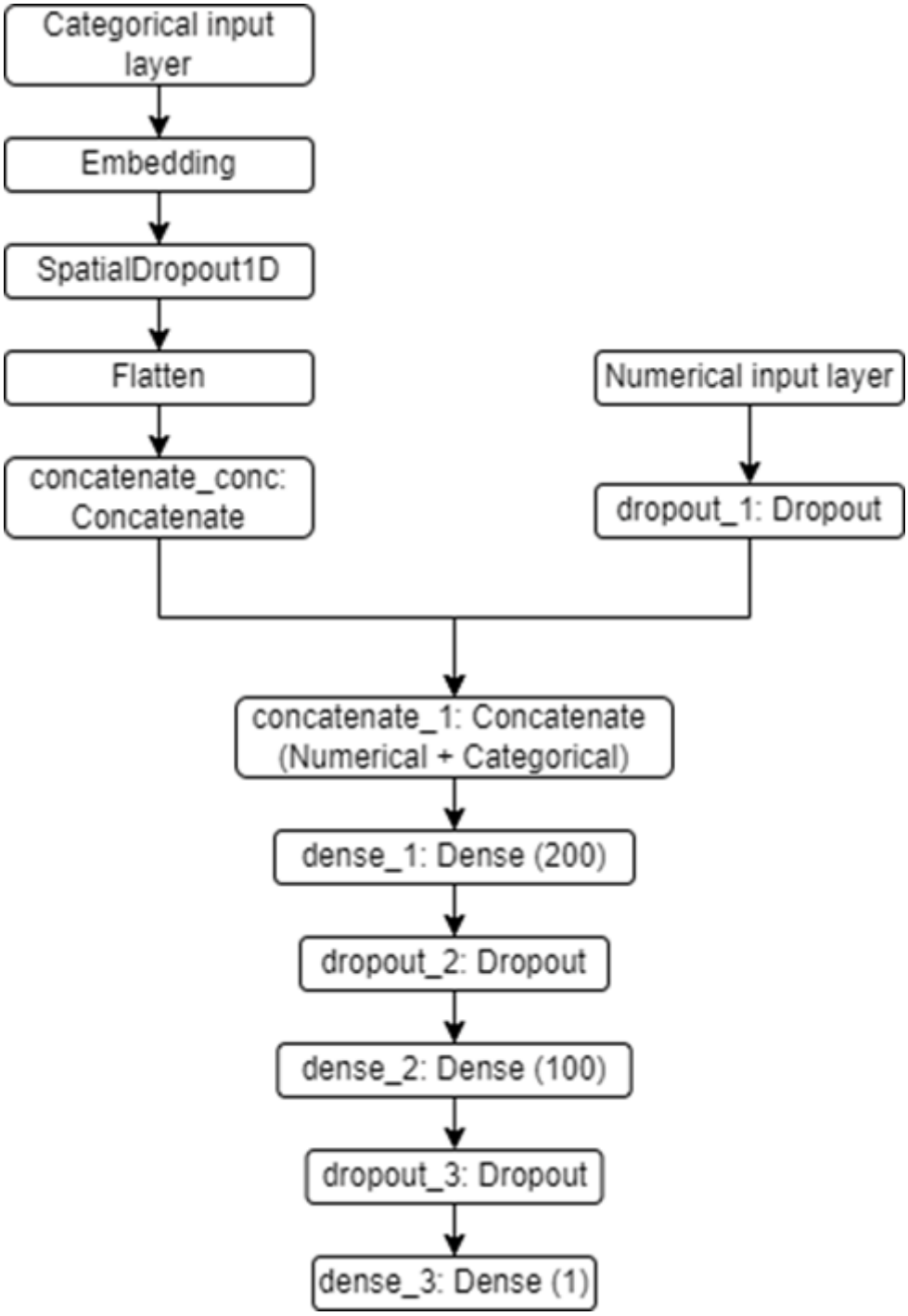

Figure 8

Neural Network architecture (Source: Authors)

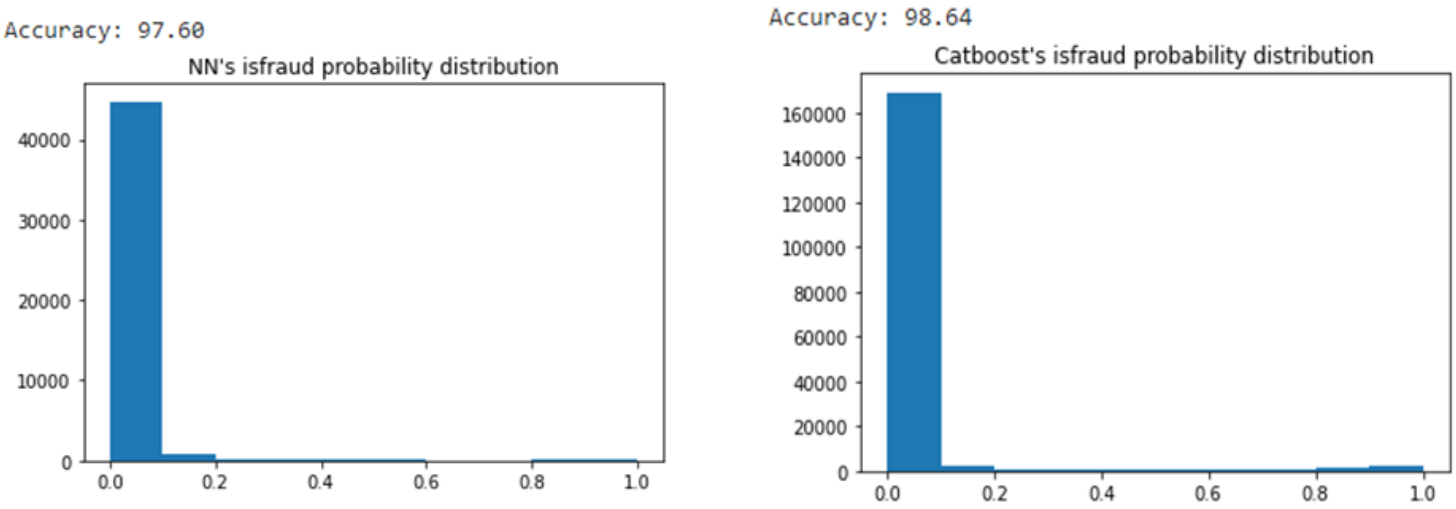

Figure 9

Distribution of the frequency of fraud and Accuracy score

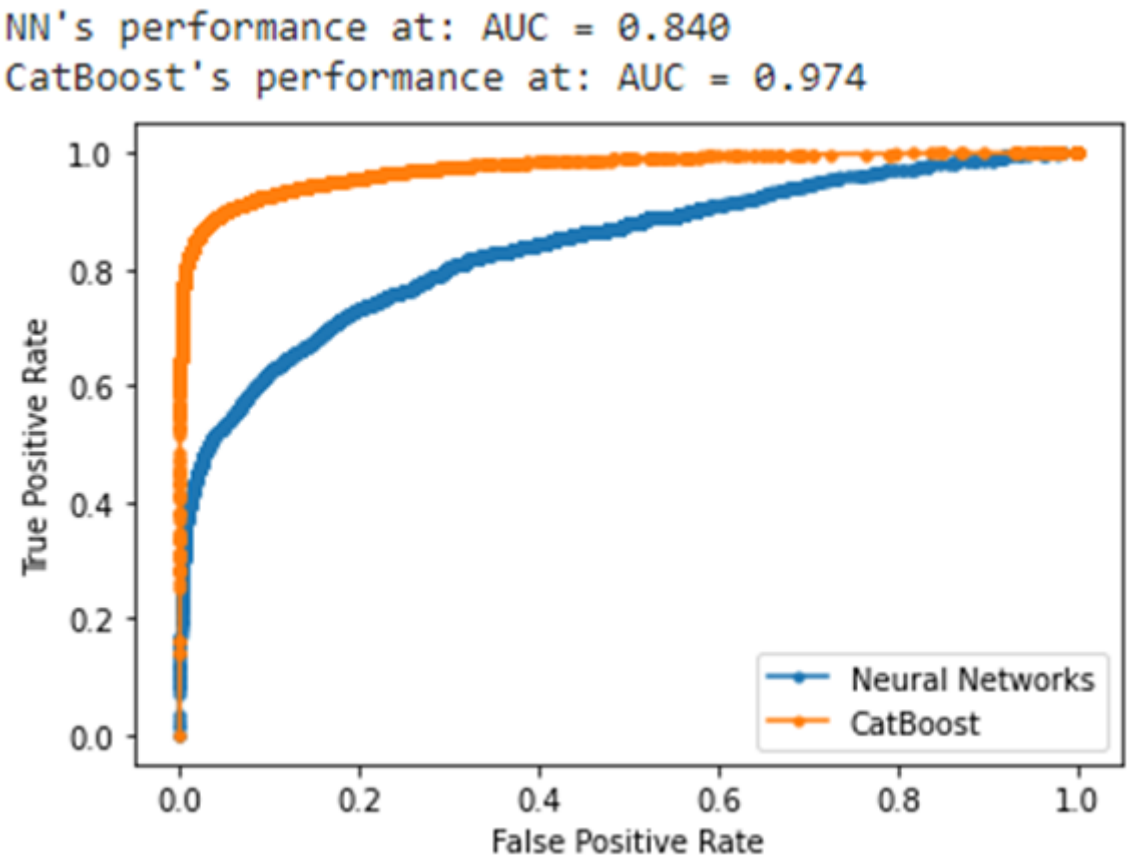

Figure 10

ROC – AUC score

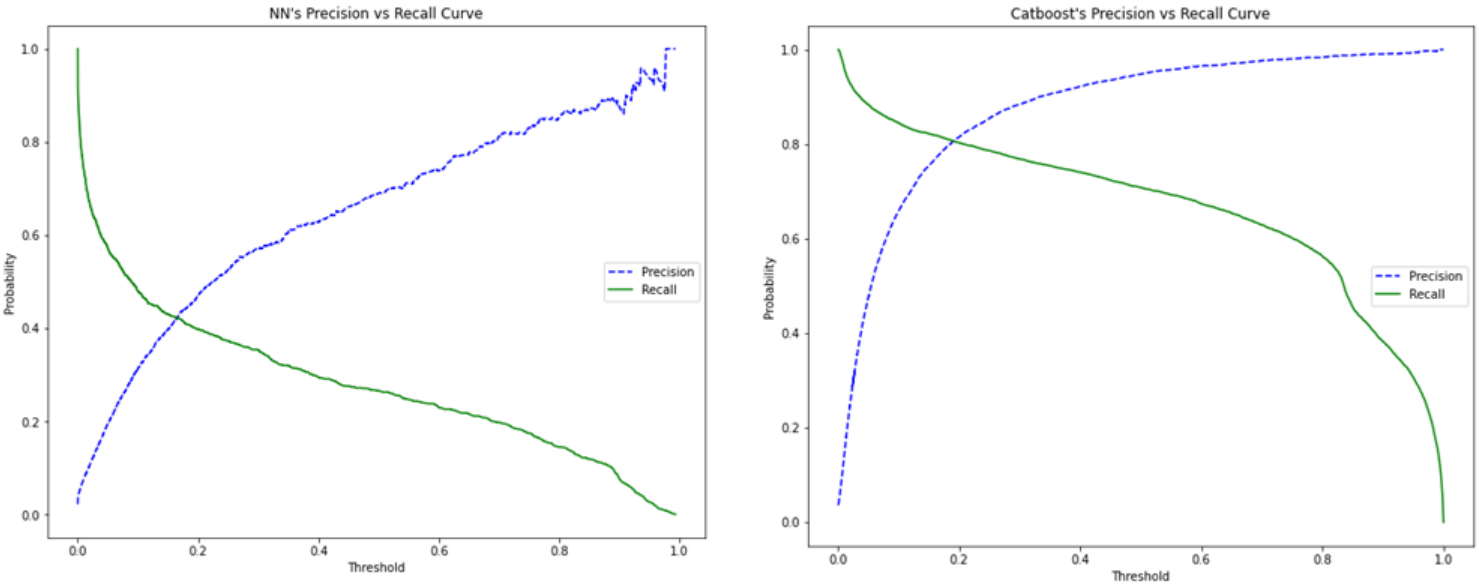

Figure 11

Precision – Recall curve (Source: Authors)

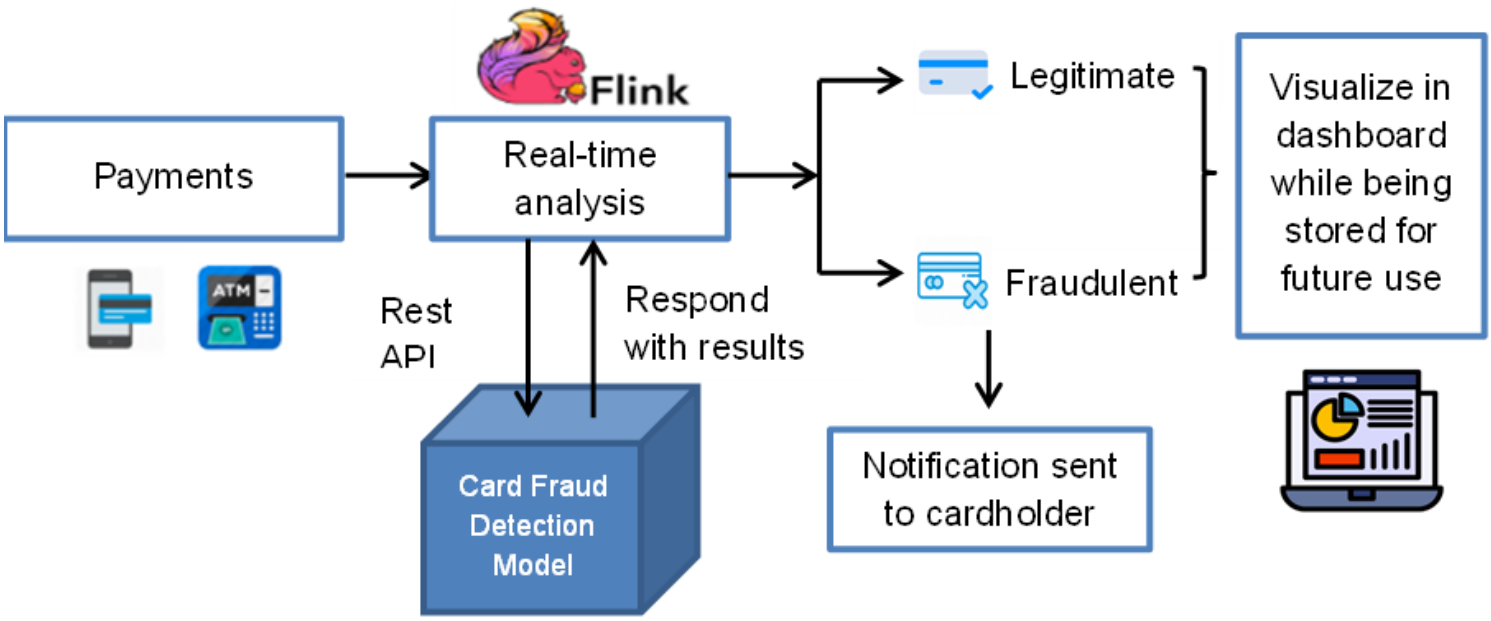

Figure 12

Live fraud detection architecture (Source: Authors)