IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

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

NGHIA NGUYEN[1,2], TRUC DUONG[1,2], TRAM CHAU[1,2], VAN-HO NGUYEN[1,2], TRANG TRINH[1,2], DUY TRAN[1,2] AND THANH HO[1,2]

[1] University of Economics and Law, Ho Chi Minh City, 700000, Vietnam
[2] Vietnam National University, Ho Chi Minh City, 700000, Vietnam
Corresponding author: Thanh Ho (e-mail: thanhht@uel.edu.vn).
This research is funded by University of Economics and Law, Vietnam National University Ho Chi Minh City.

ABSTRACT The 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. As a result, the need for a novel method to prevent this menace is undisputed. The research work was conducted on the IEEE-CIS Fraud Detection Dataset provided by Vesta Corporation. Based on the logic of labeling of converting the entire account to "Fraud=1" once the credit card has fraud, we navigated the research process towards predicting fraudulent credit cards rather than fraudulent transactions. The key idea behind the proposed model is the user separation, in which we divided users into old and new people before applying CatBoost and Deep Neural Network to each category, respectively. Besides, a variety of techniques to improve detection accuracy, namely handling heavily imbalanced dataset, feature transformation and feature engineering, will also be presented in detail in this paper. The experimental result shows that our model well performed as we obtained AUC scores of 0.97 (CatBoost) and 0.84 (Deep Neural Network).

INDEX TERMS CatBoost, card fraud detection, Deep Neural Network, deep learning, machine learning

I. 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 a wrongful or criminal deception that results in financial or personal gain. To simply put, fraud detection is a process of identifying cardholders' unusual behaviors when compared 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 such as credit card number [2], email address, phone number [3], and many more to steal money, according to Federal Trade Commission [4], the number of credit card fraud cases accounted for 459,297 cases, of which the cases of identity theft rose by 44.6% from 271,927 in 2019 to 393,207 in 2020.

To combat card fraud, considerable 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 [5], Support Vector Machine [6], Decision Trees [7], Hidden Markov Models [8] to state-of-the-art ones like Gradient Boosting Tree [9] and Deep Learning [10]. Among them, Gradient Boosting Tree and Deep Learning, in particular, CatBoost and Deep Neural Network (DNN) are 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 came up with combining CatBoost and DNN, all fitted on a monthly cross-validation setup to optimally exploit historical customer data and real-time transaction details.

The main contribution of this research is a hybrid of deep learning-based approach and CatBoost. This model is expected to help prevent losses when being deployed into production by more accurately detecting suspicious financial transactions and timely notifying the authorities so that necessary action could be taken.

The rest of the paper is structured as follows: Section II and III introduce theoretical foundation and a brief comparative review of previous relevant studies in card fraud detection. Section IV forms the core of the paper, providing details on our proposed model. The experimental results were presented in Section V. The final section concludes our research with an evaluation of the results obtained and a light touch on ideas for further investigation.

II. THEORETICAL FOUNDATION

In this section, we present the theoretical foundation of models and metrics employed in this study, including CatBoost, DNN, and evaluation metrics.

A. CATBOOST

Rooted in the family of gradient boosted decision trees (GBDTs), CatBoost soon enters into the list of top first-choice algorithms for the supervised classification [11] for successfully handling statistical issues that other existing state-of-the-art implementations of GBDTs face. Discovered by L. Prokhorenkova *et.al.* [12], who developed CatBoost, a prediction model F obtained after several steps of boosting is likely to suffer a phenomenon called "prediction shift", which is the shift of the distribution of $F(x_k) \mid x_k$ for a training example x_k from the distribution of $F(x) \mid x$ for a test example x. The author discovered the issue based on the hypothesis that there is a data set $\boldsymbol{D} = \{(\boldsymbol{x_k}, \boldsymbol{y_k})\}_{\boldsymbol{k=1..n,}}$, where $\boldsymbol{x_k} = (\boldsymbol{x_k^1}, \ldots, \boldsymbol{x_k^m})$ is a random vector of m features and $y_k \in R$ is a binary target variable. Samples (x_k, y_k) are independently and identically distributed according to some distribution $P(\cdot,\cdot)$. The goal of the learning task is to train a function $H: R_m \rightarrow R$ which minimizes the expected loss: $\boldsymbol{L(F) := EL(y, F(x))}$ where $\boldsymbol{L(\cdot,\cdot)}$ is a smooth loss function and (x, y) is a testing data sampled from the training data D. The procedure for gradient boosting [13] constructs iteratively a sequence of approximations $\boldsymbol{Ft: R_m \rightarrow R, t = 0, 1, \ldots}$ in a greedy fashion. From the previous approximation F^{t-1}, F^t is obtained in an additive process, such that $\boldsymbol{F_t = F_{t-1} + \alpha h^t}$ where $\boldsymbol{\alpha}$ is a step size and $\boldsymbol{function\ h^t: R_m \rightarrow R}$ (base predictor) to minimize the loss function:

$$\boldsymbol{h^t = arg\,min_{h \in H}\, L(F_{t-1} + h)}$$
$$\boldsymbol{= arg\,min_{h \in H}\, EL(y, F_{t-1}(x) + h(x))}$$

Further, distribution shift can also occur when preprocessing categorical features by converting them to their target statistics. A target statistic is a simple statistical model itself, and it can also cause target leakage and a prediction shift [12]. The authors created a novel boosting algorithm called ordered boosting that resembles the ordered target statistics method. Besides, CatBoost also has other boosting mode called "plain", which is the standard GBDT algorithm with inbuilt ordered target statistics. The procedure of building a tree in CatBoost is described in the pseudocode in reference to [12].

ALGORITHM OF BUILDING A TREE IN CATBOOST

Input: $\boldsymbol{M, \{(x_i, y_i)\}_{i=1}^n, \alpha, L, \{\sigma_i\}_i^s, Mode}$
$\boldsymbol{grad \leftarrow CalcGradient(L, M, y)}$;
$\boldsymbol{r \leftarrow random(1, s)}$;
$\boldsymbol{if\ Mode == Plain\ then}$
  $\boldsymbol{G \leftarrow (grad_\Gamma(i)\ for\ i = 1..n)}$;
$\boldsymbol{if\ Mode == Ordered\ then}$
  $\boldsymbol{G \leftarrow (grad_{\Gamma,\sigma_\Gamma(i)-1}(i) for\ i = 1..n)}$;
$\boldsymbol{T \leftarrow emty\ tree}$;
foreach $\boldsymbol{step\ of\ top - down\ procedure}$ **do**
  $\boldsymbol{foreach\ candidate\ split\ c\ do}$
    $\boldsymbol{if}$ $Mode == Plain$ $\boldsymbol{then}$
      $\Delta(i) \leftarrow \boldsymbol{avg(grad_\Gamma(p)\ for\ p: leaf_\Gamma(p) = leaf_\Gamma(i))}$
      $\boldsymbol{for\ i = 1..n}$;
    $\boldsymbol{if}$ $Mode == Ordered$ $\boldsymbol{then}$
      $\Delta(i) \leftarrow \boldsymbol{avg\left(grad_{\Gamma,\sigma_\Gamma-1}(p)\ for\ p: leaf_\Gamma(p) = leaf_\Gamma(i), \sigma_\Gamma(p) <\right.}$
      $\boldsymbol{for\ i = 1..n}$;
    $loss(T_c) \leftarrow \cos(\Delta, G)$
  $T \leftarrow \arg min_{T_c}(loss(T_c))$
$\boldsymbol{if}$ $Mode == Plain$ $\boldsymbol{then}$
  $M_{\Gamma'}(i) \leftarrow M_{\Gamma'}(i) - \alpha\ avg(grad_{\Gamma'}(p)\ for\ p: leaf_{\Gamma'}(p) = leaf_{\Gamma'}(i))$
  $for\ \Gamma' = 1..s, i = ..n;$
$\boldsymbol{if}$ $Mode == Ordered$ $\boldsymbol{then}$
  $M_{\Gamma',j}(i) \leftarrow M_{\Gamma',j}(i) - \alpha\ avg(grad_{\Gamma',j}(p)\ for\ p: leaf_{\Gamma'}(p) = leaf_{\Gamma'}(i), c$
  $for\ \Gamma' = 1..s, i = ..n, j \geq \sigma_{\Gamma'}(p) - 1;$
$\boldsymbol{return}$ T, M

In the ordered boosting mode, during the learning process, we maintain the supporting models $M_{r,j}$, where $M_{r,j}$ (i) is the current prediction for the i-th example based on the first j examples in the permutation σ_r. At each iteration t of the algorithm, we sample a random permutation σ_r from $\{\sigma_1, \ldots, \sigma_s\}$ and construct a tree T_t on the basis of it. First, for categorical features, all target statistics are computed according to this permutation. Second, the permutation affects the tree learning procedure. In the plain mode, if categorical features are present, it maintains s supporting models M_r corresponding to target statistics based on $\sigma_1, \ldots, \sigma_s$.

In CatBoost, base predictors are oblivious decision trees [14], which are trees splitted with consistent criterion across entire level. Such trees are balanced, less prone to overfitting, and allow speeding up execution at testing time significantly [12]. To prove the efficiency of CatBoost, the authors compared with other GBDTs, including XGBoost and LightGBM. The results showed that shows that on similar sizes of ensembles CatBoost can be scored around 25 times faster than XGBoost and around 60 times faster than LightGBM.

B. DEEP NEURAL NETWORK

DNN is a subtype of artificial neural network besides the shallow neural network – like models. The criterion behind

this categorization is the number of hidden layers between the input and the output layers. Like other typical artificial neural network, a signal obtained by the product of the input and its corresponding weight will be carried from the input layer to the hidden layers powered by activation function, such as sigmoid function, tangent hyperbolic function, linear function, step function, ramp function, and Gaussian function and many more [15]. The DNN parameters are estimated by minimizing the sum-of-squares error function calculated from DNN outputs. Starting from an initialization stage where the model parameters are set to an initial set of values, a stochastic gradient descent algorithm is continuously run to reduce the error function until it converges to a specified lowest value [16]. The DNN training involves two passes based on the error backpropagation algorithm, which are the forward pass and the backward pass. In the former one, the affine transformation and nonlinear activation are calculated layer by layer from the input to the output layer. In the later one, the derivatives of the error function with respect to individual weights are calculated in a reverse order, that is, from the output layer to the input layer [16].

C. EVALUATION METRICS

A typical classification task will be evaluated via metrics such as confusion matrix, accuracy, the area under the Receiver Operating Characteristics Curve (ROC-AUC), Precision – Recall, F1-score.

A confusion matrix contains information about actual and predicted classifications from a classifier [17]. The confusion matrix can be interpreted as: the True Negative (TN) and True Positive (TP) are the correctly classified classes while False Negative (FN) and False Positive (FP) are the misclassified classes.

TP: The classifier predicted a true event and the event is actually true.

TN: The classifier predicted that an event is not true and the event is actually not true.

FP: The classifier predicted that an event is true but the event is actually not true.

FN: The classifier predicted that an event is not true but the event is actually true.

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 accuracy to evaluate performance of the model, which indicates the proportion of right predictions among the total of examined cases.

III. RELATED WORKS

This section focuses on exploring related research in the field of card fraud detection. The machine learning and deep learning approaches in some research are also explored and studied to form the basis of this research.

A. MACHINE LEARNING – BASED APPROACH

The approach described by Shiyang Xuan et al., in the study [18] 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 [19] by Aya Abd El Naby et al. In the study [20], 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.

B. DEEP LEARNING – BASED APPROACH

Hassan Najadat et al., in work [21], 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 [22] 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. [23] 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) [24] 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.

IV. 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 [24]. A good complex dataset is a backbone for any robust machine learning model to produce a plausible reality-matched output.

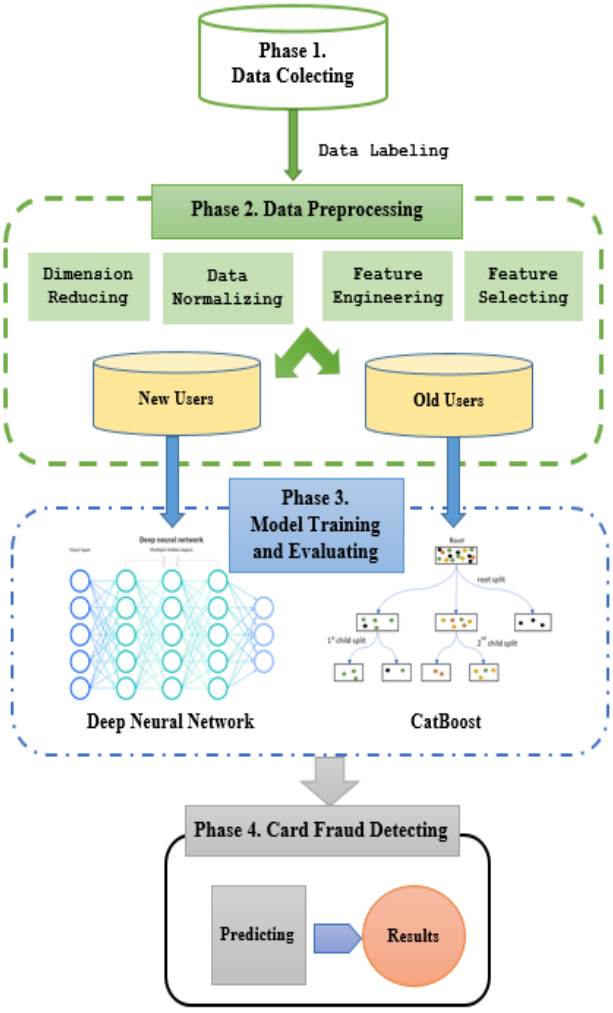


FIGURE 1. Proposed card fraud detection model (Source: Authors)

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 We chose this dataset because 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 divers ified 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 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.

A. DATA SOURCES

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 I as follows:

TABLE I
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.
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.
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.

B. DATA PREPROCESSING

1) 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.

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 K-Nearest Neighbors (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 True and Found.

2) FEATURE TRANSFORMATION.

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.

3) FEATURE ENGINEERING.

This is the major part in our process, where we will start splitting customers into known and unknown groups. First, 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), and the result is represented first five card groups in Table II as follows:

TABLE II
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

The Figure 2 illustrates the transactions are conducted by each cardGroup separately.

TransactionID	ProductCD	card1	card2	card3	card4	card5	card6	cardGroup
3019496	H	17188	321	150	visa	226	debit	17188321.0150.0visa226.0debit-9H
3020819	H	17188	321	150	visa	226	debit	17188321.0150.0visa226.0debit-9H
3049599	H	17188	321	150	visa	226	debit	17188321.0150.0visa226.0debit-9H
3019481	H	17188	321	150	visa	226	debit	17188321.0150.0visa226.0debit-9H
3020292	H	17188	321	150	visa	226	debit	17188321.0150.0visa226.0debit-9H
3019162	H	17188	321	150	visa	226	debit	17188321.0150.0visa226.0debit-9H
3529687	W	4090	490	150	visa	226	debit	4090490.0150.0visa226.0debit-165W
3529698	W	4090	490	150	visa	226	debit	4090490.0150.0visa226.0debit-165W
3529712	W	4090	490	150	visa	226	debit	4090490.0150.0visa226.0debit-165W
3543113	W	4090	490	150	visa	226	debit	4090490.0150.0visa226.0debit-165W
3543119	W	4090	490	150	visa	226	debit	4090490.0150.0visa226.0debit-165W
3543129	W	4090	490	150	visa	226	debit	4090490.0150.0visa226.0debit-165W
3543133	W	4090	490	150	visa	226	debit	4090490.0150.0visa226.0debit-165W
3543135	W	4090	490	150	visa	226	debit	4090490.0150.0visa226.0debit-165W

FIGURE 2. The number of transactions made by two separate cardGroup (indicated by colors)

We continue to separate the cardGroups into cardIDs based on the V307 feature, which is important in identifying cardIDs. Each color in Figure 3 has been marked as a cardID 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 as illustrated in Figure 4.

TransactionID	TransactionAmt	cardGroup	TransactionAmt-trunc	V307	V307-trunc	V307-round	V307-round	V307-trunc2	V307-plus	V307-plus-round	V307-plus-round-trunc	V307-plus-trun-c2	V307-plus-round2	cardID
3085059	46.75	15885545.0185.0visa138.0debit-22C	46.75	51.4206	51.42	51.421	51.42	51.42	98.1706	98.171	98.17	98.17	98.17	group0_0
3085085	46.75	15885545.0185.0visa138.0debit-22C	46.75	98.1666	98.166	98.167	98.17	98.16	144.9166	144.917	144.916	144.91	144.92	group0_0
3085104	46.75	15885545.0185.0visa138.0debit-22C	46.75	144.9126	144.912	144.913	144.91	144.91	191.6626	191.663	191.662	191.66	191.66	group0_0
3081192	49.78	15885545.0185.0visa138.0debit-22C	49.781	0	0	0	0	0	49.78125	49.781	49.781	49.78	49.78	group0_11
3139566	34.97	15885545.0185.0visa138.0debit-22C	34.969	49.7882	49.788	49.788	49.79	49.78	84.75695	84.757	84.757	84.75	84.76	group0_11
3139908	56.56	15885545.0185.0visa138.0debit-22C	56.562	84.7682	84.768	84.768	84.77	84.76	141.3307	141.331	141.33	141.33	141.33	group0_11
3139929	56.56	15885545.0185.0visa138.0debit-22C	56.562	141.3192	141.319	141.319	141.32	141.31	197.8817	197.882	197.881	197.88	197.88	group0_11
3139935	56.56	15885545.0185.0visa138.0debit-22C	56.562	197.8702	197.87	197.87	197.87	197.87	254.4327	254.433	254.432	254.43	254.43	group0_11
3139990	56.56	15885545.0185.0visa138.0debit-22C	56.562	254.4212	254.421	254.421	254.42	254.42	310.9837	310.984	310.983	310.98	310.98	group0_11
3139996	56.56	15885545.0185.0visa138.0debit-22C	56.562	310.9722	310.972	310.972	310.97	310.97	367.5347	367.535	367.534	367.53	367.53	group0_11
3508531	13.77	3901176.0185.0mastercard224.0credit-4C	13.773	0	0	0	0	0	13.7734375	13.773	13.773	13.77	13.77	group18491_0
3547898	78.7	3901176.0185.0mastercard224.0credit-4C	78.688	13.7721	13.772	13.772	13.77	13.77	92.4596	92.46	92.459	92.45	92.46	group18491_0
3548011	78.7	3901176.0185.0mastercard224.0credit-4C	78.688	92.4665	92.466	92.466	92.47	92.46	171.15399	171.154	171.154	171.15	171.15	group18491_0

FIGURE 3. Splitted CardID based on V307 feature

TransactionID	TransactionDT	TransactionAmt	ProductCD	id-19	id-20	id-31	DeviceInfo	cardID	groupsUser	CardIDcount	UserIDcount	UserFraudSum	CardFraudSum
2987240	90193	37.1	137785	266	325	chrome 54.0 for android	Redmi Note 4 Build/MMB29M	group32724_0	group35099	3	3	3	3
2987243	90246	37.1	137785	266	325	chrome 54.0 for android	Redmi Note 4 Build/MMB29M	group32724_0	group35099	3	3	3	3
2987245	90295	37.1	137785	266	325	chrome 54.0 for android	Redmi Note 4 Build/MMB29M	group32724_0	group35099	3	3	3	3
2987779	102154	10	23046	397	161	chrome generic	KFFOWI Build/LVY48F	group134049_0	group15900	1	9	1	1
2987780	102188	10	23046	397	161	chrome generic	KFFOWI Build/LVY48F	group16817_0	group15900	8	9	8	8
2987781	102193	10	23046	397	161	chrome generic	KFFOWI Build/LVY48F	group16817_0	group15900	8	9	8	8

FIGURE 4. Customer used one or more card

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".

4) 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.

V. EXPERIMENTAL MODEL

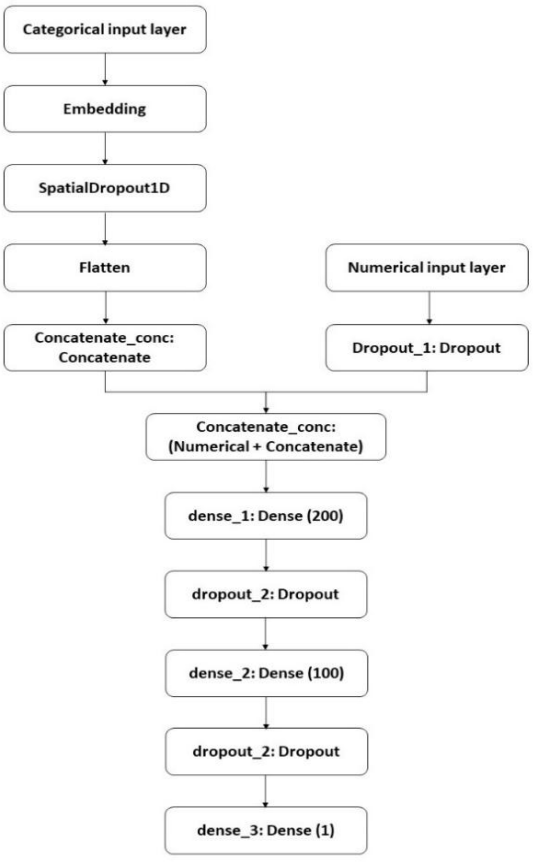


FIGURE 5. Neural Network architecture (Source: Authors)

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. For non-overlapping users, our Neural Network architecture, as given in Figure 5, 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 III.

TABLE III
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

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 [26]. 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 [27] to be able to outperform most other machine learning algorithms in both speed and accuracy. We use K-fold cross-validation with 10 folds to tune parameter. The final set is described in Table IV as follows:

TABLE IV
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 trees to build before taking the maximum voting or averages of predictions	5000

VI. EXPERIMENTAL RESULTS AND DISCUSSION

A. EXPERIMENTAL RESULTS

In this section, we will propose a 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.

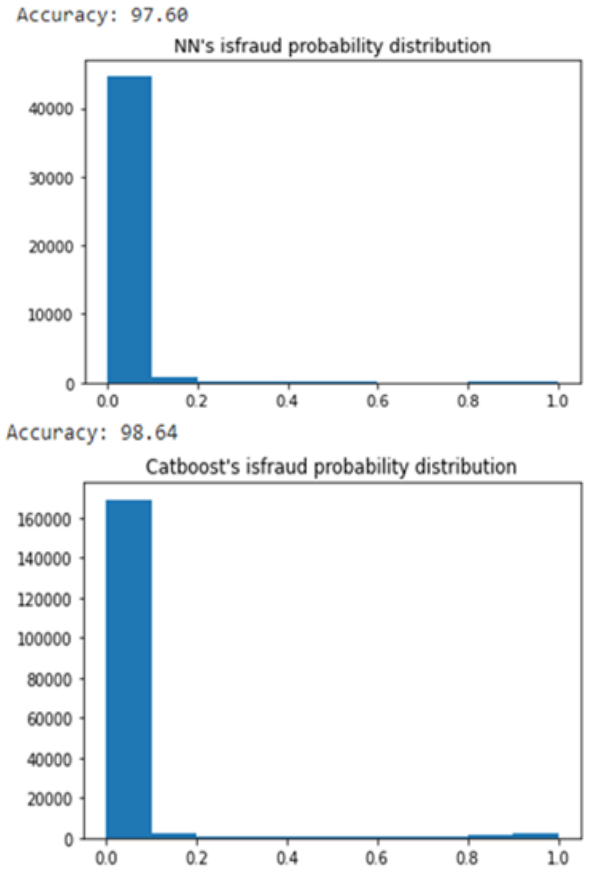


FIGURE 6. **Distribution of the frequency of fraud and accuracy score (Source: Authors)**

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 6 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 Figure 7, 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).

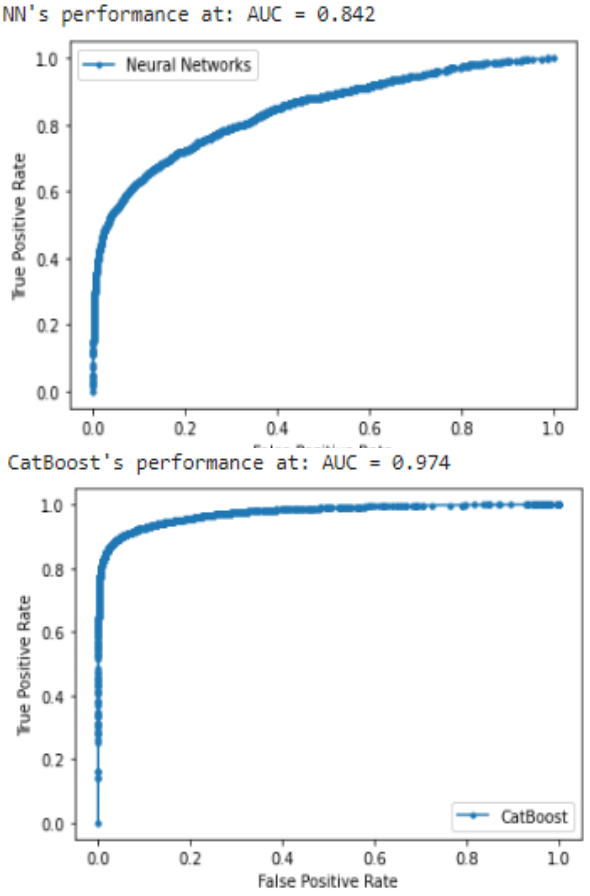
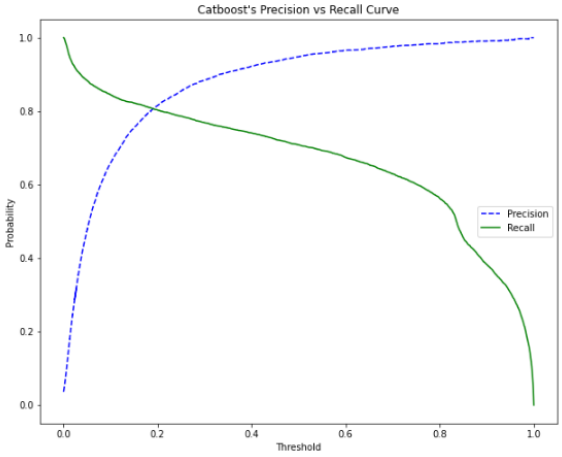


FIGURE 7. **ROC – AUC score (Source: Authors)**

Metric 2: Precision - Recall curve and AUC score



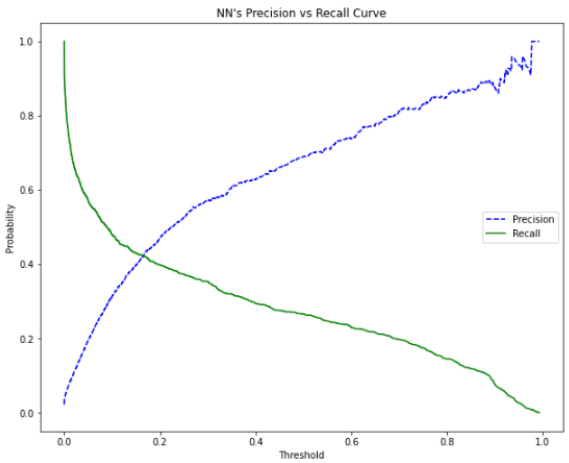


FIGURE 8. **Precision – Recall curve (Source: Authors)**

The Precision and Recall curves intersect at the threshold of 0.2 as depicted in Figure 8, 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 Figure 6. 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 has not found all actual fraudulent users, resulting in low Recall results.

B. LIVE FRAUD DETECTION ARCHITECTURE

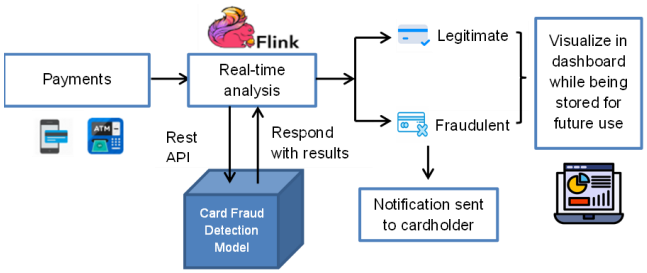


FIGURE 9. **Live fraud detection architecture (Source: Authors)**

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 in Figure 9.

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 locked; otherwise transaction is regarded as legitimate. In the event that a signal is not received from the user, the account will be temporarily locked 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.

VII. CONCLUSION

A. CONTRIBUTION

The research team proposed a model that combines two methods including CatBoost and Deep Neural Network to build a model, then evaluate and comment. The model evaluation results show that the model is highly accurate, the model can be fully integrated in software applications to detect card fraud in units and organizations.

B. LIMITATIONS AND FUTURE WORKS

The model takes long time to produce results due to the limited capacity in hardware. Besides, in case the fraud happens because the user lost the card, when it detects that the user has been classified as a fraud, if they use the new card, the model will still recognize them as the old identifier. would classify the transaction as fraudulent instead of legitimate. 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. In the future, we plan to proceed with our work to make their utilization increasingly appropriate in practical real-time situations.

REFERENCES

[1]. Oxford Learner's Dictionaries [Online]. Available: https://www.oxfordlearnersdictionaries.com/definition/english/fraud

[2]. M. Zareapoor, and J. Yang, "A Novel Strategy for Mining Highly Imbalanced Data in Credit Card Transactions", *Intelligent Automation & Soft Computing*, pp. 1-7, 2017.

[3]. M. Óskarsdóttir *et al.*, "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, 2019.

[4]. Federal Trade Commission, "25 Credit Card Fraud Statistics To Know in 2021". Available: intuit.com

[5]. Y. Sahin, and E. Duman, "Detecting credit card fraud by ANN and logistic regression", *International symposium on innovations in intelligent systems and applications*, pp. 315-319, 2011.

[6]. 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.

[7]. 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.

[8]. 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.

[9]. 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.

[10]. 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.

[11]. V. A. Dev & M. R. Eden, "Gradient Boosted Decision Trees for Lithology Classification". *Computer Aided Chemical Engineering*, Vol. 47 (2019), pp. 113 – 118.

[12]. L. Prokhorenkova et.al., "CatBoost: Unbiased Boosting with Categorical Features". *In* 32nd *Conference and Workshop on Neural Information Processing Systems*, 12/2018. Available: https://proceedings.neurips.cc/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf

[13]. J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine", *Annals of statistics*, 1189-1232, 2001.

[14]. M. Ferov and M. Modry. "Enhancing Lambdamart Using Oblivious Trees" , 2016. Available: arXiv preprint arXiv:1609.05610.

[15]. Shiruru, Kuldeep, "An Introduction to Artificial Neural Network", *International Journal of Advance Research and Innovative Ideas in Education*, Vol 1, 27 – 30, 2016.

[16]. D. Zheng et. al., D."Short-term renewable generation and load forecasting in microgrids", *Microgrid protection and control*, 2021.

[17]. A. A. Ibrahim et. al., "Comparison of the CatBoost Classifier with other Machine Learning Methods", *International Journal of Advanced Computer Science and Applications*, Vol. 11, No. 11, 2020.

[18]. Shiyang Xuan, Guanjun Liu, Zhenchuan Li, Lutao Zheng, Shuo Wang, and Changjun Jiang. "Random Forest for Credit Card Fraud Detection" in 15th IEEE International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, 2018.

[19]. Aya Abd El Naby, Ezz El-Din Hemdan, and Ayman El-Sayed, "Deep Learning Approach for Credit Card Fraud Detection", presented Int. Conf. on Electronic Engineering (ICEEM), 2021.

[20]. John O. Awoyemi, Adebayo O Adetunmbi, Samuel Oluwadare, "Credit Card Fraud Detection Using Machine Learning Techniques: A Comparative Analysis", presented at Int. Conf. on Computing Networking and Informatics (ICCNI), 2017.

[21]. Hassan Najadat, Ola Altiti, Ayah Abu Aqouleh, and Mutaz Younes, "Credit Card Fraud Detection Based on Machine and Deep Learning" presented at 11th IEEE Int. Conf. on Information and Communication Systems (ICICS), Irbid, Jordan, 2020.

[22]. Aji Mubarek Mubalaike, and Esref Adali, "Deep Learning Approach for Intelligent Financial Fraud Detection System", 3rd IEEE Int. Conf. on Computer Science and Engineering (UBMK), 2018.

[23]. Yara Alghofaili, Albatul Albattah, and Murad A. Rassam, "A Financial Fraud Detection Model Based on LSTM Deep Learning Technique", *Journal of Applied Security Research*, pp. 1-19, 2020

[24]. Ibtissam Benchaji, Samira Douzi & Bouabid El Ouahidi, "Credit Card Fraud Detection Model Based on LSTM Recurrent Neural Networks", *Journal of Advances in Information Technology*, Vol. 12, No. 2, pp. 113-118, 2021.

[25]. Dataset [Online]. Available: https://www.kaggle.com/c/ieee-fraud-detection/data. Accessed on: Sept. 16, 2021.

[26]. John T. Hancock and Taghi M. Khoshgoftaar, "CatBoost for Big Data: An Interdisciplinary Review", Journal of Big Data 7,no. 94, 2020.

[27]. Prathima Gamini, Sai Tejasri Yerramsetti, Gayathri Devi Darapu, Vamsi Kaladhar Pentakoti, and 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.

NGHIA NGUYEN was born in Binh Duong province, Viet Nam in 2001. A student majoring in Management of Information System at the University of Economics and Law have engaged in many research in machine learning with professors. He also took charge of organizing a national data analytics competition named Business Intelligence. Recently, he was a co-author of the paper published in the National Science Conference on Information Systems in Business and Management (2021).

TRUC DUONG was born in Ho Chi Minh City, Vietnam. A specialized mathematics student has a great passionate interest and talent in the application of machine learning and science of algorithms that make sense of data. Recently, she was a co-author of the paper published in National Science Conference on Information Systems in Business and Management (2021).

TRAM CHAU was born in Ho Chi Minh City, Vietnam in 2001. She is pursuing a Bachelor of MIs at University of Economics and Law, Vietnam National University in Ho Chi Minh City. She was co-author of the paper published in National Science Conference on Information Systems in Business and Management (2021).

VAN-HO NGUYEN received a B.S degree in Management Information System (MIS) from the Faculty of Information Systems, University of Economics and Law (VNU–HCM), Vietnam in 2015, and a Master degree in MIS from University of Economics Ho Chi Minh City, Vietnam in 2020, respectively. He is currently a lecturer in the Faculty of Information Systems, University of Economics and Law, VNU-HCM, Vietnam. His current research interests include Business Analytics, Business Intelligence, Data Analytics, and Machine Learning. His researches were published in international journals such as Journal of Information Processing Systems, Business Research Systems.

TRANG TRINH was born in Vietnam. She is a final-year student majoring in Management of Information System, Faculty of Information Systems, University of Economics and Law, VNU-HCM. She is a student with excellent academic performance and active participation in extracurricular activities. She has received many encouraging scholarships; award-winning the champion of the "Business Intelligence" surpassing approximately 300 teams. She was also a delegate representative for Vietnam to participate in "The 7th Asian Future Leaders Summit" in Malaysia. She was a core organizer of the project "Data Analytics and Data Privacy" funded by the American Government for helping Vietnamese citizens get the data skills.

DUY TRAN was born in Ho Chi Minh City, Vietnam in 2000. He earned a Bachelor of information system at University of Economics and Law, Vietnam National University in Ho Chi Minh City.

THANH HO received an M.S degree in Computer Science from University of Information Technology, VNU-HCM, Vietnam in 2009 and a Ph.D. degree in Computer Science from University of Information Technology, VNU-HCM, Vietnam in 2018. He is currently a Senior Lecturer in Faculty of Information Systems, University of Economics and Law, VNU-HCM, Vietnam. His research interests are Data mining, Data Analytics, Business Intelligence, Social Network Analysis, and Big Data. His researches were published in many international journals. He works as Reviewer for many journals indexed in SCOPUS/ISI. Currently, he is member of Vietnam Association of Information Systems (VAIS).