IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 August 2022, accepted 5 September 2022, date of publication 8 September 2022, date of current version 19 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3205416

RESEARCH ARTICLE

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

NGHIA NGUYEN[1,2], TRUC DUONG[2,3], TRAM CHAU[1,2], VAN-HO NGUYEN[1,2], TRANG TRINH[1,2], DUY TRAN[1,2], AND THANH HO[1,2]

[1]Faculty of Information Systems, University of Economics and Law, Ho Chi Minh City 700000, Vietnam
[2]Vietnam National University, Ho Chi Minh City 700000, Vietnam
[3]Faculty of Economic Law, University of Economics and Law, Ho Chi Minh City 700000, Vietnam

Corresponding author: Thanh Ho (thanhht@uel.edu.vn)

This work was supported by the University of Economics and Law, Vietnam National University Ho Chi Minh City.

ABSTRACT The rapid development of technology has digitized customer payment behavior towards a cashless society. To a certain extent, this has created a feast for miscreants to commit fraud. According to Nilson (2020), global fraud loss is projected to reach over $35 billion by 2025. Consequently, the need for a novel method to prevent this menace is undisputed. This research was conducted on the IEEE-CIS Fraud Detection Dataset provided by Vesta Corporation. Based on the logic of labeling for converting the entire account to "Fraud=1" once the credit card has fraud, we navigate the research process towards predicting fraudulent credit cards rather than fraudulent transactions. The key idea behind the proposed model is user separation, in which we divide users into old and new people before applying CatBoost and Deep Neural Network to each category, respectively. In addition, a variety of techniques to improve detection accuracy, namely handling heavily imbalanced datasets, feature transformation, and feature engineering, are also presented in detail in this paper. The experimental results showed that our model performed well, 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 in the recent decades. As an increasing number of people are accustomed to online transactions, this has contributed to the prevalence of card payments. Unfortunately, the prevailing emergence of spending behavior has become an ideal condition for the increase in fraudulent activities. The Oxford Dictionary has defined fraud [1] as wrongful or criminal deception that results in financial or personal gain. Fraud detection is the process of identifying cardholders' unusual behaviors when compared to their prior card usage profile. Based on such differences, an 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 numbers [2], email addresses, phone numbers [3], and many more to steal money. According to the Federal Trade Commission [4], the number of credit card fraud cases accounted for 459,297 cases, of which the cases of identity theft increased 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. To analyze voluminous data, a variety of machine learning algorithms have been employed, including classical methods such as logistic regression [5], support vector machine [6], decision trees [7], hidden Markov models [8], and state-of-the-art methods such as 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 solutions, given their reputation for remarkable fraud detection performance. Because time-based DNN architectures 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

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

transactions simultaneously, while DNN is employed for detecting fraud based on the data of unknown users. To make the most of their strengths, we combined 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 a deep learning-based approach and CatBoost. This model is expected to help prevent losses when deployed into production by more accurately detecting suspicious financial transactions and timely notifying authorities so that necessary action can be taken.

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

II. THEORETICAL FOUNDATION

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

A. CATBOOST

Rooted in the family of gradient boosted decision trees (GBDTs), CatBoost soon enters the list of top first-choice algorithms for supervised classification [11] to successfully handle statistical issues faced by other existing state-of-the-art implementations of GBDTs. Discovered by 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 in 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 this issue based on the hypothesis that there exists a dataset $\boldsymbol{D} = \{(\boldsymbol{x_k}, \boldsymbol{y_k})\}_{\boldsymbol{k=1..n}},$ where $\boldsymbol{x_k} = (\boldsymbol{x_k^1}, \ldots, \boldsymbol{x_k^m})$ is a random vector of m features and $y_k \epsilon$ R is a binary target variable. The samples (x_k, y_k) are independently and identically distributed according to the 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 the testing data sampled from the training data D. The procedure for gradient boosting [13] iteratively constructs 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 the step size and function $\boldsymbol{h^t{:}R_m \rightarrow R}$ (base predictor) to minimize the loss function:

$$\begin{aligned} \boldsymbol{h^t} &= \boldsymbol{argmin_{h \epsilon H} L(F_{t-1} + h)} \\ &= \boldsymbol{argmin_{h \epsilon H} EL(y, F_{t-1}(x) + h(x))} \end{aligned}$$

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

Algorithm 1 Algorithm of Building a Tree in Catboost

```
Input: $M, \{(x_i, y_i)\}_{i=1}^{n}, \alpha, L, \{\sigma_i\}_i^s, Mode$
$grad \leftarrow CalcGradient\,(L, M, y)$;
$r \leftarrow random\,(1, s)$;
if Mode == Plain then
    $G \leftarrow (grad_\Gamma\,(i)\,for i = 1..n)$;
if Mode == Ordered then
    $G \leftarrow (grad_{\Gamma,\sigma_\Gamma(i)-1}\,(i)\,for i = 1..n)$;
$T \leftarrow$ emty tree;
foreach step of top − down procedure do
    foreach candidate split c do
        if Mode == Plain then
            $\Delta\,(i) \leftarrow avg\,(grad_\Gamma\,(p)\,for p : leaf_\Gamma\,(p) = leaf_\Gamma\,(i))$
            $for i = 1..n$;
        if Mode == Ordered then
            $\Delta\,(i) \leftarrow avg\,(grad_{\Gamma,\sigma_\Gamma - 1}\,(p)\,for p : leaf_\Gamma\,(p)$
            $= leaf_\Gamma\,(i), \sigma_\Gamma\,(p) < \sigma_\Gamma\,(i))$
            $for i = 1..n$;
        $loss(T_c) \leftarrow \cos(\Delta, G)$
    $T \leftarrow \arg min_{T_c}(loss(T_c))$
if Mode == Plain then
    $M_{\Gamma'}\,(i) \leftarrow$
      $M_{\Gamma'}\,(i) - \alpha avg\,(grad_{\Gamma'}\,(p)\,for p{:}leaf_{\Gamma'}\,(p) = leaf_{\Gamma'}\,(i))$
    $for \Gamma' = 1..s, i = ..n$;
if Mode == Ordered then
    $M_{\Gamma',j}\,(i) \leftarrow M_{\Gamma',j}\,(i) -$
      $\alpha avg\,(grad_{\Gamma',j}\,(p)\,for p{:}leaf_{\Gamma'}\,(p) = leaf_{\Gamma'}\,(i), \sigma_{\Gamma'}(p) \leq j)$
    $for \Gamma' = 1..s, i = ..n, j \geq \sigma_{\Gamma'}\,(p) - 1$;
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 ith 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 based on this permutation. First, for categorical features, all target statistics are computed according to this permutation. Second, permutation affects the tree-learning procedure. In plain mode, if categorical features are present, it maintains the supporting models M_r corresponding to the target statistics based on $\sigma_1, \ldots, \sigma_s$.

In CatBoost, the base predictors are oblivious decision trees [14], which are trees split with consistent criteria across the entire level. Such trees are balanced, less prone to overfitting, and allow rapid execution at testing time [12]. To prove the efficiency of CatBoost, the authors compared it with other

GBDTs, including XGBoost and LightGBM. The results showed that for ensembles of similar sizes, CatBoost can be scored approximately 25 times faster than XGBoost and approximately 60 times faster than LightGBM.

B. DEEP NEURAL NETWORK

DNN is a subtype of artificial neural network, in addition to shallow neural network – like models. The criterion behind this categorization was the number of hidden layers between the input and output layers. Similar to 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 an activation function, such as a sigmoid function, tangent hyperbolic function, linear function, step function, ramp function, and Gaussian function [15]. The DNN parameters were estimated by minimizing the sum-of-squares error function calculated from the 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]. DNN training involves two passes based on the error backpropagation algorithm, namely, the forward pass and backward pass. In the former, the affine transformation and nonlinear activation are calculated layer-by-layer from the input to the output layer. In the latter, the derivatives of the error function with respect to the individual weights are calculated in reverse order, that is, from the output layer to the input layer [16].

C. EVALUATION METRICS

A typical classification task will be evaluated using metrics such as confusion matrix, accuracy, area under the receiver operating characteristic curve (ROC-AUC), precision-recall, and F1-score.

A confusion matrix contains information regarding the actual and predicted classifications from a classifier [17]. The confusion matrix can be interpreted as follows: true negative (TN) and true positive (TP) are correctly classified classes, while false negative (FN) and false positive (FP) are 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 that 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 in the fraud domain as it is robust to variable fraud rates and does not capture the effect of an overly large number of legitimate events in this dataset. In fact, no single metric can best evaluate a model. As a result, in addition to ROC-AUC, we also used accuracy to evaluate the performance of the model, which indicates the proportion of correct predictions among the total examined cases.

III. RELATED WORKS

This section explores the related research in the field of card fraud detection. Machine learning and deep learning approaches have also been explored and studied to form the basis of this research.

A. MACHINE LEARNING – BASED APPROACH

The approach described by Xuan *et al.* [18] was a combination of two types of random forest models: random tree-based random forest and CART-based random forest. Their method used historical transaction data based on the behavioral features of normal and fraudulent transactions. The dataset belongs to a 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 a ratio of normal-to-abnormal transactions of 5:1. The results of this research showed 98.67% accuracy, 32.68% precision, and 59.62% recall. They obtained the result with high accuracy, but the false positive rate was also high, raising the 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 small datasets and is unsuitable for large enterprises and financial institutions. Although we can apply logistic regression and the stacked auto-encoders method to big data, they yield low-accuracy results and are unsuitable for practical use. This was confirmed in a study [19] by Aya Abd El Naby *et al.* Awoyemi *et al.* [20] detected fraudulent credit activities using naive Bayes, K-Nearest Neighbor, and logistic regression with accuracy of 97.92%, 97.69%, and 54.86%, respectively. It was clear that the logistic regression classifier method was still relatively low, and the risk of errors increased.

B. DEEP LEARNING – BASED APPROACH

Najadat *et al.* [21] applied BiLSTM-MaxPooling-BiGRU-MaxPooling to predict fraud. The authors also applied a naive base, voting, AdaBoost, random forests, decision tree, and logistic regression to compare the effects of each model. The dataset used in this study is unique because of its highly imbalanced class. To deal with the imbalanced dataset, the authors used the random under-sampling, random oversampling, and synthetic minority oversampling techniques. The results showed that deep learning models with the three sampling techniques achieved significantly better accuracy than machine learning models. The highest accuracy of the machine-learning-based models was 81%. However, when the authors concatenated BiLMST-MaxPooling with BiGRU-MaxPooling with a random oversampling technique, they achieved 91.37% accuracy. From this study, we can conclude that machine learning algorithms alone cannot solve such a complicated large dataset.

Ebenezer Esenogho *et al.* [22] proposed an intricate ensemble of LSTM as the base learner in AdaBoost and the synthetic minority oversampling-edited nearest neighbor technique. The authors experimented with a dataset containing transactions within two days in September 2013 by European credit card clients. The final result achieved included a sensitivity of 0.996, specificity of 0.998, and AUC of 0.990, which was superior to other traditional algorithms such as SVM, MLP, decision tree, solely LSTM, and AdaBoost.

In another study [23] by Mubalaike and Adali, the authors experimented with deep learning to detect fraud with the aim of achieving high accuracy. They used an ensemble of a decision tree model (EDT), stacked auto-encoders (SAE), and restricted Boltzmann machines (RBM) with accuracy of 90.49%, 80.52%, and 91.53%, respectively. The SAE was lower than that of the EDT and RBM. In the future, the authors also want to use the neural network model to improve the accuracy.

In 2020, Alghofaili *et al.* [24] studied the ability of long short-term memory to detect credit card fraud by comparing this algorithm with an auto-encoder and traditional machine learning models such as logistic regression, random forest, and support vector machines. The limitation of this study is that the authors only compared the accuracy of the models based on the training set instead of the test set; therefore, it lacked a concrete basis to conclude whether the LSTM had the highest accuracy. Another study related to LSTM was conducted by Ibtissam Benchaji *et al.* [25], using a dataset of 594,643 transactions from 11/2012 to- 04/2013 provided by a local bank. The model compared payment data with historical information. If the data matched the pattern, the card was definitely used by the cardholder; otherwise, the possibility of fraud was very high. In the future, they plan to build a model based on another variant of Recurrent Neural Networks to validate its competency compared with the current model.

In general, the two types of current models for fraud detection scarcely consider 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 to detecting fraud using machine and deep learning. The process starts with data collection. In our case, we used the IEEE-CIS dataset provided by Vesta Corporation, which is the forerunner of guaranteed e-commerce payment solutions [26]. A good complex dataset is the backbone for any robust machine-learning model to produce a plausible reality-matched output In the second phase, we used a variety of methods to preprocess the data. The first step is the minification step to reduce the memory usage. This helps save a lot of resources when building the prediction model and speeding up the training process. Subsequently, we conducted an exploratory analysis to inspect data for patterns, trends, or relationships between variables and between the target column and other variables. We then experimented with many ways to select the most suitable techniques for feature transformation and selection.

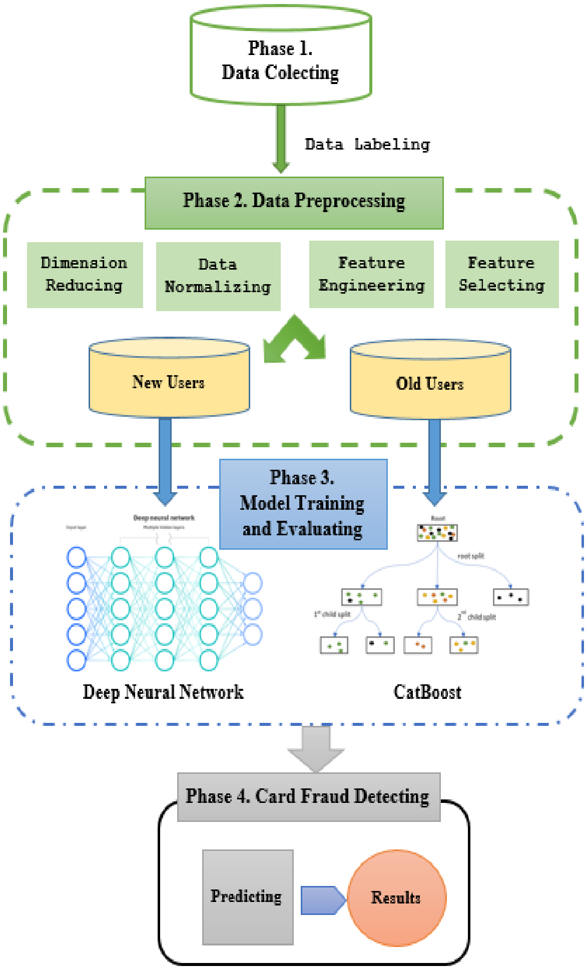
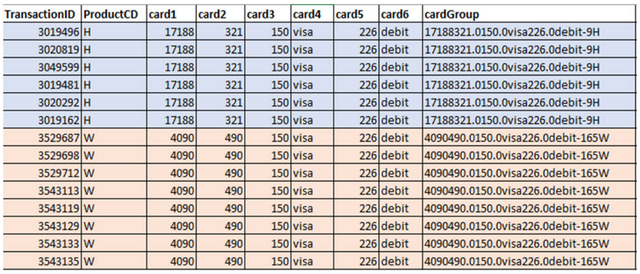


FIGURE 1. Proposed card fraud detection model (Source: Authors).

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 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 features in the dataset. In the third phase, after processing categorical and numerical data into a suitable form, we deploy DNN on unknown users and CatBoost on known users before combining them into the final results in the last phase. The details of each process are presented in the following sections.



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).

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. These 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 agreements. To clearly manifest this, a table of features in the 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.
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, those suspicious transactions were automatically considered legitimate (isFraud=0). In other words, once a card has been reported as fraudulent, that account is converted to isFraud=1. Therefore, we predict fraudulent clients, rather than fraudulent transactions.

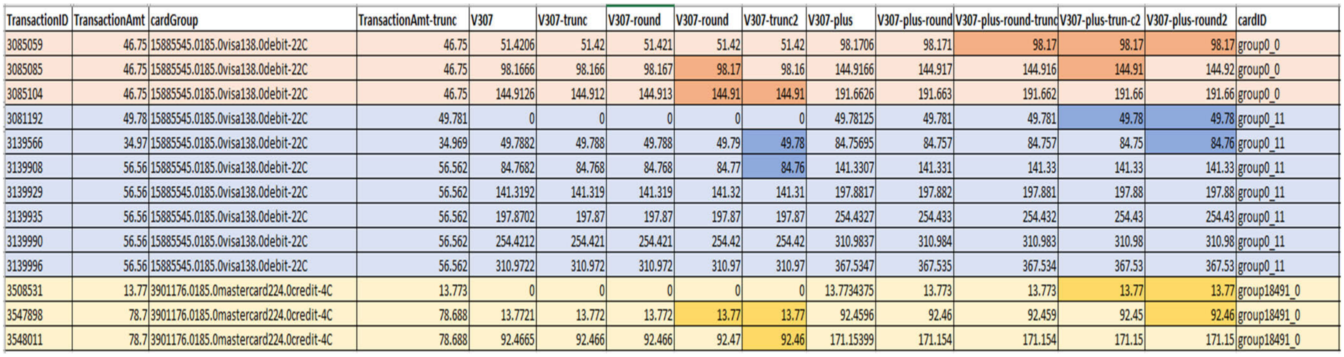
B. DATA PREPROCESSING

1) EXPLORATORY ANALYSIS

In general, transactions were recorded from November 30, 2017, to May 31, 2018, as depicted in Figure 2. When conducting exploratory data analysis, we noticed that approximately 3.5% of train transactions are fraudulent, with more than 95% of columns having missing values.

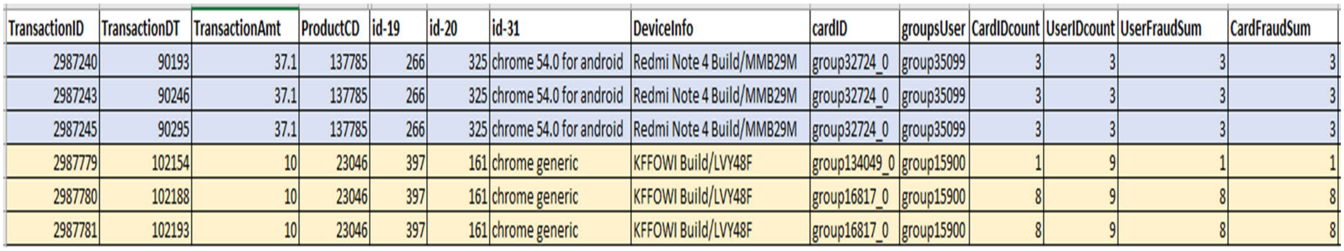
To deal with the imbalance between the number of fraudulent and non-fraudulent transactions, we apply the SMOTE method to increase the number of fraudulent transactions many times using the K-Nearest Neighbors (KNN) algorithm. Specifically, a data point is randomly selected from the pool of fraudulent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

After performing multivariate analysis, we found that the number of fraudulent transactions is high for products of category W or C, paid by debit cards, credit cards, visas, or master cards. Cards such as the American Express and Discover cards 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, fraud occurs frequently on phones that normally use Chrome 63.0 or generic mobile safari. The probability of a fraudulent transaction when performed by a computer or phone is relatively the same. For variables



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.

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 in observations whose values are true and found.

2) FEATURE TRANSFORMATION

The numerical features 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. Because the machine learning model only accepts numerical variables as inputs, categorical features are converted to numbers through label encoding.

3) FEATURE ENGINEERING

This is a major part of our process, where we start splitting customers into known and unknown groups. First, the initial dataset was divided into two parts: the training set and test set with a ratio of 7:3. Suppose that one user uses multiple cards for several 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 2 as follows:

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

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

We continue to separate cardGroups into cardIDs based on the V307 feature, which is important in identifying cardIDs. Each color in Figure 3 is marked as a cardID belonging to each cardGroup. V307 is the cumulative result of the Transaction Amt value of the previous transaction. Next, we identify the customers based on customer identification information (TransactionAmt, id_19, id_20), assuming id_19 and id_20 are information of the IP address, as illustrated in Figure 4.

User separation was performed for both training and test sets. The identifiers presented in both datasets will be recognized as old customers, otherwise new customers. The purpose of this is to train the model to 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 correct set of features as inputs to the model is a key contribution to our achieved performance. First, we used the principal component analysis (PCA) technique to reduce the number of prefix V variables from 339 to the 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 in which such planes have a linear form as subspaces. For DNN, the input variables include categorical variables, namely ProductCD,

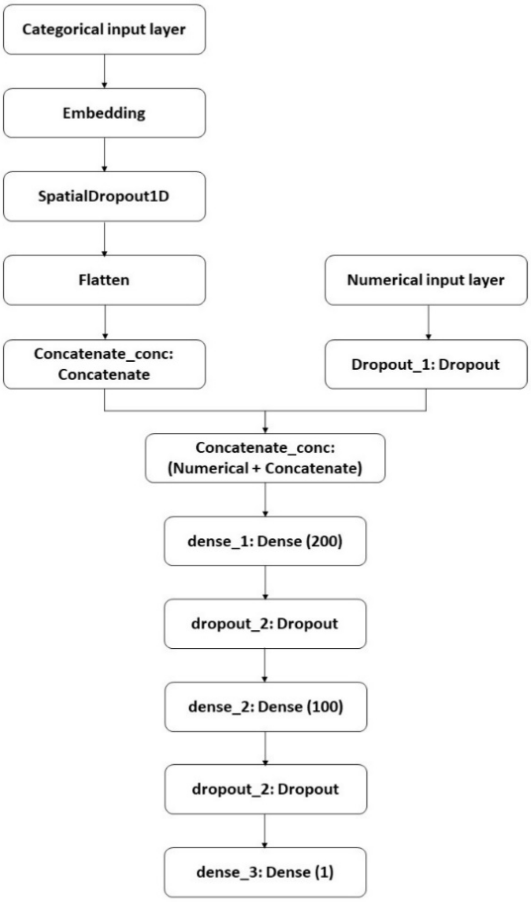


FIGURE 5. Neural network architecture (Source: Authors).

card1-card6, addr1, addr2, P_emaildomain, R_emaildomain, M1-M9 (through the 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 as follows: TransactionID, TransactionDT, isFraud, and discarded V variables after PCA.

V. EXPERIMENTAL MODEL

Our model is a combination of CatBoost and neural networks as the base learners. Their predictions on overlapping and non-overlapping parts were combined into a single output. For non-overlapping users, our neural network architecture, as shown in Figure 5, consists of an input layer with the size of the number of selected features, three hidden layers (the respective neurons are 512–256–1), and an output layer of one neuron. The optimal parameters for our model are listed in Table 3.

We used CatBoost to determine if we could 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 its strong performance relative to other GBDT implementations. CatBoost uses the target-based ordering principle, where the values for each example rely only on the observed history [27]. Thus, for a set of data with plentiful categorical features, such as the IEEE-CIS dataset, we can improve our training results without spending time and effort turning categories into numbers.

TABLE 3. 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

CatBoost is robust as it does not require extensive hyper-parameter tuning [28] to outperform most other machine learning algorithms in terms of both speed and accuracy. We use K-fold cross-validation with 10 folds to tune the parameter. The final set is shown in Table 4.

TABLE 4. 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 propose a theory to combine the new and old users' predicted models to obtain 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 accuracy result is quite high for both types of customer groups, and there is no significant difference between the two models. However, accuracy alone has proven to be less effective for severely 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 determined by the model. Therefore, determining a good predictive model for practice should not rely solely on the 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,

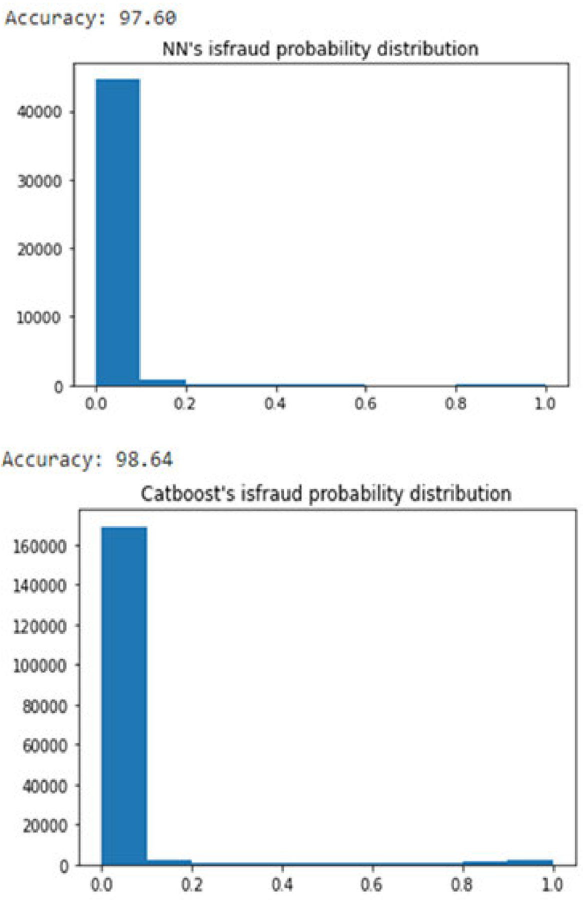


FIGURE 6. **Distribution of the frequency of fraud and accuracy score (Source: Authors).**

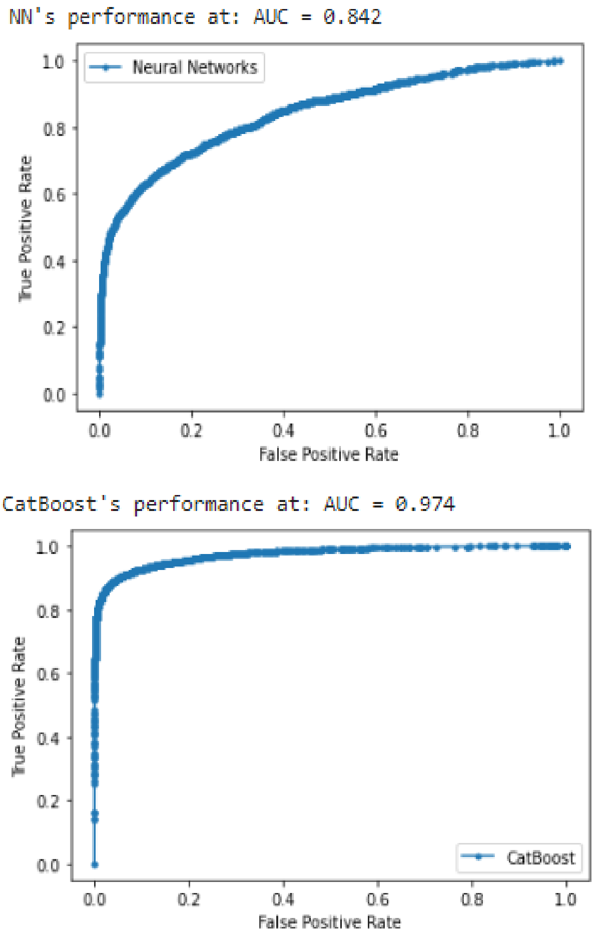


FIGURE 7. **ROC – AUC score (Source: Authors).**

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. To determine the threshold, we evaluated the model's effectiveness using two metrics as follows:

Metric 1: ROC – AUC curve and AUC score

In principle, the closer the curve is to point (0, 1), the more efficient the model is. Therefore, in Figure 7, CatBoost generated an almost perfect prediction result, which was confirmed by an AUC of 0.974. The DNN model had a smaller area under the ROC curve, but the difference was not significant (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 Figure 8, confirming a high degree of accuracy in predicting fraud for transactions with a 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 have committed fraud, the accuracy of the model is quite high compared to the actual results. However, the model did not find all actual fraudulent users, resulting in low recall results.

B. LIVE FRAUD DETECTION ARCHITECTURE

Once the model passes the evaluation gate, to bring it to a higher practical level, we designed a pipeline showing how the detection result will be used when it is deployed. The implementation is illustrated in Figure 9.

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

Step 2: The proposed credit card fraud detection model is implemented as an API, and Apache Flink calls this API to process and output the results received from the model.

Step 3: If the transaction is detected to be fraudulent, the system sends the user a warning alert at the time of payment by asking whether the user who initiated the payments was the cardholder. If the user does not make a transaction, the user's account will be locked; otherwise, the 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.

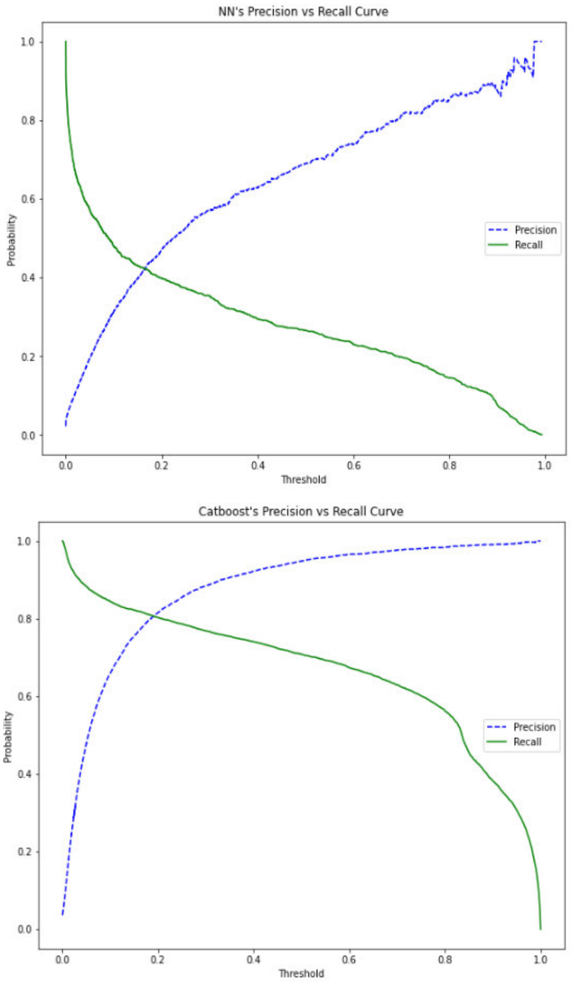


FIGURE 8. Precision – recall curve (Source: Authors).

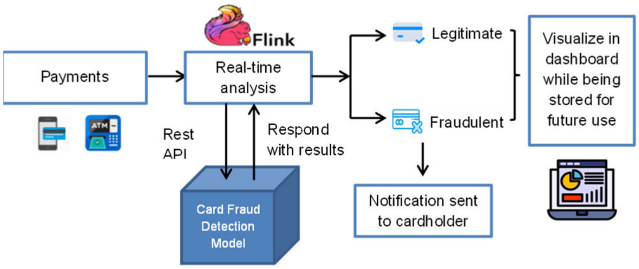

Flink
Payments
Real-time analysis
Legitimate
Fraudulent
Visualize in dashboard while being stored for future use
Rest API
Respond with results
Card Fraud Detection Model
Notification sent to cardholder

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

Step 4: The prediction results were 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, CatBoost and DNN, to build a model, and then evaluate and comment. The model evaluation results show that the model is highly accurate and can be fully integrated in software applications to detect card fraud in units and organizations.

B. LIMITATIONS AND FUTURE WORKS

The model takes a long time to produce results owing to its limited hardware capacity. In addition, if a fraud occurs because the user loses the card, when it detects that the user has been classified as a fraud, and if they use the new card, the model will still recognize them as the old identifier. classifies transactions as fraudulent rather than legitimate. This paper outlines the recent significant damage of fraudulent transactions for the financial industry and presents our CatBoost and neural network-based approach to effectively tackle this problem and improve detection efficiency. Using this method, we rejected many redundant and high-capacity features to bias the model. In the future, we plan to proceed with our work to make their utilization increasingly appropriate for practical real-time situations.

REFERENCES

[1] *Oxford Learner's Dictionaries*. Accessed: Oct. 26, 2021. [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," *Intell. Autom. Soft Comput.*, vol. 23, no. 4, pp. 1–7, May 2017, doi: 10.1080/10798587.2017.1321228.

[3] M. Óskarsdóttir, C. Bravo, C. Sarraute, J. Vanthienen, and B. Baesens, "The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics," *Appl. Soft Comput.*, vol. 74, pp. 26–39, Jan. 2019, doi: 10.1016/j.asoc.2018.10.004.

[4] Federal Trade Commission. *25 Credit Card Fraud Statistics To Know in 2021*. Accessed: Jun. 30, 2022. [Online]. Available: https://intuit.com

[5] Y. Sahin and E. Duman, "Detecting credit card fraud by ANN and logistic regression," in *Proc. Int. Symp. Innov. Intell. Syst. Appl.*, Jun. 2011, pp. 315–319, doi: 10.1109/INISTA.2011.5946108.

[6] N. K. Gyamfi and J.-D. Abdulai, "Bank fraud detection using support vector machine," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2018, pp. 37–41, doi: 10.1109/IEMCON.2018.8614994.

[7] J. R. Gaikwad, A. B. Deshmane, H. V. Somavanshi, S. V. Patil, and R. A. Badgujar, "Credit card fraud detection using decision tree induction algorithm," *Int. J. Innov. Technol. Exploring Eng. (IJITEE)*, vol. 4, no. 6, pp. 66–69, 2014.

[8] W. N. Robinson and A. Aria, "Sequential fraud detection for prepaid cards using hidden Markov model divergence," *Expert Syst. Appl.*, vol. 91, pp. 235–251, Jan. 2018.

[9] A. A. Taha and S. J. Malebary, "An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine," *IEEE Access*, vol. 8, pp. 25579–25587, 2020.

[10] P. Raghavan and N. E. Gayar, "Fraud detection using machine learning and deep learning," in *Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE)*, Dec. 2019, pp. 334–339.

[11] V. A. Dev and M. R. Eden, "Gradient boosted decision trees for lithology classification," *Comput. Aided Chem. Eng.*, vol. 47, pp. 113–118, Jan. 2019.

[12] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. 32nd Conf. Workshop Neural Inf. Process. Syst.*, Dec. 2018, pp. 1–11. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf

[13] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[14] M. Ferov and M. Modrý, "Enhancing LambdaMART using oblivious trees," 2016, *arXiv:1609.05610*.

[15] H. Kukreja, N. Bharath, C. S. Siddesh, and S. Kuldeep, "An introduction to artificial neural network," *Int. J. Advance Res. Innov. Ideas Educ.*, vol. 1, pp. 27–30, Sep. 2016.

[16] D. Zheng *et al.*, "Short-term renewable generation and load forecasting in microgrids," in *Microgrid Protection and Control*. 2021, pp. 57–96, doi: 10.1016/B978-0-12-821189-2.00005-X.

[17] A. A. Ibrahim, R. L. Ridwan, M. M. Muhammed, R. O. Abdulaziz, and G. A. Saheed, ''Comparison of the CatBoost classifier with other machine learning methods,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 11, pp. 1–11, 2020.

[18] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, ''Random forest for credit card fraud detection,'' in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Zhuhai, China, Mar. 2018, pp. 1–6.

[19] A. A. E. Naby, E. E.-D. Hemdan, and A. El-Sayed, ''Deep learning approach for credit card fraud detection,'' presented at the Int. Conf. Electron. Eng. (ICEEM), 2021.

[20] O. John Awoyemi, A. O. Adetunmbi, and S. Oluwadare, ''Credit card fraud detection using machine learning techniques: A comparative analysis,'' presented at the Int. Conf. Comput. Netw. Inform. (ICCNI), 2017.

[21] H. Najadat, O. Altiti, A. A. Aqouleh, and M. Younes, ''Credit card fraud detection based on machine and deep learning,'' presented at the 11th IEEE Int. Conf. Inf. Commun. Syst. (ICICS), Irbid, Jordan, 2020.

[22] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, ''A neural network ensemble with feature engineering for improved credit card fraud detection,'' *IEEE Access*, vol. 10, pp. 16400–16407, 2022.

[23] A. M. Mubalaike and E. Adali, ''Deep learning approach for intelligent financial fraud detection system,'' in *Proc. 3rd Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2018, pp. 598–603.

[24] Y. Alghofaili, A. Albattah, and A. Murad Rassam, ''A financial fraud detection model based on LSTM deep learning technique,'' *J. Appl. Secur. Res.*, vol. 15, no. 4, pp. 498–516, 2020.

[25] I. Benchaji, S. Douzi, and B. E. Ouahidi, ''Credit card fraud detection model based on LSTM recurrent neural networks,'' *J. Adv. Inf. Technol.*, vol. 12, no. 2, pp. 113–118, 2021.

[26] IEEE—Computational Intelligence Society and Vesta Corporation. (2019). *IEEE—CIS Fraud Detection Dataset*. Kaggle. [Online]. Available: https://www.kaggle.com/c/ieee-fraud-detection/data

[27] J. T. Hancock and T. M. Khoshgoftaar, ''CatBoost for big data: An interdisciplinary review,'' *J. Big Data*, vol. 7, no. 1, p. 94, Dec. 2020.

[28] P. Gamini, S. T. Yerramsetti, G. D. Darapu, V. K. Pentakoti, and V. P. Raju, ''Detection of credit card fraudulent transactions using boosting algorithms,'' *J. Emerg. Technol. Innov. Res.*, vol. 8, no. 2, 2021.

NGHIA NGUYEN was born in Binh Duong, Vietnam, in 2001. He is currently pursuing the degree in management of information system with the University of Economics and Law. He has 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 is the coauthor of the paper published in the National Science Conference on Information Systems in Business and Management, in 2021.

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

TRAM CHAU was born in Ho Chi Minh City, Vietnam, in 2001. She is currently pursuing the Bachelor of MIS degree with the University of Economics and Law, Vietnam National University in Ho Chi Minh City. She is the coauthor of the paper published in the National Science Conference on Information Systems in Business and Management, in 2021.

VAN-HO NGUYEN received the 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 the master's degree in MIS from the University of Economics Ho Chi Minh City, Vietnam, in 2020. He is currently a Lecturer at the Faculty of Information Systems, VNU-HCM. His current research interests include business analytics, business intelligence, data analytics, and machine learning. His research was published in international journals, such as the *Journal of Information Processing Systems* and *Business Research Systems*.

TRANG TRINH was born in Vietnam. She is currently pursuing the degree in management of information systems with the Faculty of Information Systems, University of Economics and Law, VNU-HCM. She was 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 of Vietnam who participated 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 acquire data skills.

DUY TRAN was born in Ho Chi Minh City, Vietnam, in 2000. He received the Bachelor of Information System degree from the University of Economics and Law, Vietnam National University, Ho Chi Minh City.

THANH HO received the M.S. and Ph.D. degrees in computer science from the University of Information Technology, VNU-HCM, Vietnam, in 2009 and 2018, respectively. He is currently a Senior Lecturer at the Faculty of Information Systems, University of Economics and Law, VNU-HCM. His research interests include data mining, data analytics, business intelligence, social network analysis, and big data research. His research has been published in several international journals. He works as a reviewer for many journals indexed in SCOPUS/ISI. He is a member of the Vietnam Association of Information Systems (VAIS).