Hindawi
Mobile Information Systems
Volume 2020, Article ID 8885269, 13 pages
https://doi.org/10.1155/2020/8885269

Research Article

Facilitating User Authorization from Imbalanced Data Logs of Credit Cards Using Artificial Intelligence

Vinay Arora,[1] **Rohan Singh Leekha**,[2] **Kyungroul Lee**,[3] **and Aman Kataria**[4]

[1]*Computer Science & Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab, India*
[2]*Associate Application Support, IT-App Development/Maintenance, Concentrix, Gurugram, India*
[3]*School of Computer Software, Daegu Catholic University, Gyeongsan, Republic of Korea*
[4]*Optical Devices and Systems (Visiting Research Scholar), CSIR-CSIO, Chandigarh, India*

Correspondence should be addressed to Kyungroul Lee; lisa.sch.k@gmail.com

Received 14 July 2020; Revised 9 September 2020; Accepted 29 September 2020; Published 30 October 2020

Academic Editor: Zengpeng Li

An effective machine learning implementation means that artificial intelligence has tremendous potential to help and automate financial threat assessment for commercial firms and credit agencies. The scope of this study is to build a predictive framework to help the credit bureau by modelling/assessing the credit card delinquency risk. Machine learning enables risk assessment by predicting deception in large imbalanced data by classifying the transaction as normal or fraudster. In case of fraud transaction, an alert can be sent to the related financial organization that can suspend the release of payment for particular transaction. Of all the machine learning models such as RUSBoost, decision tree, logistic regression, multilayer perceptron, *K*-nearest neighbor, random forest, and support vector machine, the overall predictive performance of customized RUSBoost is the most impressive. The evaluation metrics used in the experimentation are sensitivity, specificity, precision, *F* scores, and area under receiver operating characteristic and precision recall curves. Datasets used for training and testing of the models have been taken from kaggle.com.

1. Introduction

For this study, the term "credit" refers to a method of e-commerce without having funds. A credit card is a thin, rectangular metal or plastic block provided by the banking institution, allowing card users to borrow cash to pay for products and services. Credit cards enforce cardholders to repay the financial leverage, interest payment, and any other fees decided from time to time. The credit card issuer often offers its customers a line of credit (LOC), allowing them to lend cash withdrawals. Issuers usually preset lending thresholds depending on specific creditworthiness [1, 2]. The use of credit cards is vital these days, and it plays a significant role in e-commerce and online funds transfer [3, 4]. The ever-increasing use of credit cards has posed many threats to the users and the companies issuing such cards. Fraudsters keep on finding new ways to commit cheating, which can cause considerable losses to card users and these companies as well [5, 6].

1.1. Credit Card Payment Processing Steps. Figure 1 illustrates how payments are transferred to the vendor's bank account, whenever the clients make purchases through the credit card [7]:

(a) A client sends a credit card purchase via Internet of Things- (IoT-) enabled swipe devices/POS/online sites.

(b) Payment gateway collects and transfers the transaction details safely to the merchant's bank computer-based controller system

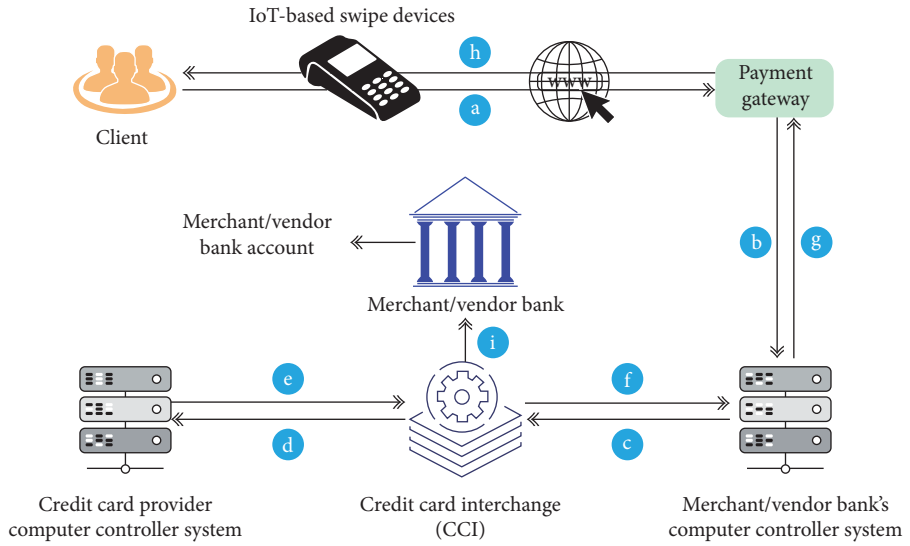

Figure 1: Payment process in the credit card system [7].

(c) The bank processor forwards the verification (i.e., processing, clearing, and settlement) process to the Credit Card Interchange (CCI)

(d) The CCI transfers the transaction to the client's credit card provider

(e) The card provider accepts or rejects the purchase based on current funds in the client's account and passes back the transaction information to the CCI

(f) The CCI transmits transaction information to the vendor's bank computer-based controller system

(g) The controller system of the vendor's bank transmits transaction details further to the payment gateway

(h) The payment gateway keeps and delivers transaction details to the vendor and/or client

(i) The CCI transfers the required funds to the vendor's bank, which further transfers funds into the merchant's account [7]

1.2. Fraud in Credit Card Transaction. Fraud and illegal behavior have various perspectives. The Association of Certified Fraud Examiners (ACFE) is a professional fraud examiner organization. Its activities include producing information, forming tools, and imparting training to avoid frauds. The ACFE has termed "fraud" as usage of one's profession for self-benefit via deliberate misapplication or misuse of assets of the organization [3]. A fraud is committed with the chief intention to acquire access by illegal means. It adversely affects the economic growth, governance, and even fundamental social values. Any technical infrastructure involving money and resources can be breached by unethical practices, e.g., auction site systems, medical insurance, vehicle insurance, credit cards, and banking. Cheating in these applications is perceived as cyber crime, potentially causing significant economic losses [3, 8].

Fraud can lower the trust in the industry, disturb the economic system, and significantly impact the overall living costs [9, 10]. IoT-enabled systems maintain the trace of their operational activities, which can be beneficial for analyzing some specific patterns. The previous methods based on manual processing such as auditing were cumbersome and ineffective due to large-size data or its attributes. Data mining techniques are considered effective in assessing small outliers in large datasets [9, 11, 12]. Frauds lead to heavy business losses. The credit card frauds contribute hundreds of millions of dollars per year for the lost revenue, and some estimates have indicated that US cumulative annual costs could surpass $400 billion [9].

1.3. Types of Credit Card-Related Frauds. The advancements in technology such as the Internet and mobile devices have contributed to increased fraudulent activities in recent times [13]. Fraudsters keep on finding new techniques, and therefore, monitoring systems are required to evolve correspondingly. Frauds related to credit cards can be broadly categorized into offline and online frauds [14]:

(i) Offline credit card fraud occurs whenever fraudsters stole the credit card and used it as the rightful owner in outlets. This is unusual as financial firms will promptly block the missing card whenever cardholders suspect the theft [3].

(ii) Online credit card frauds are more common and serious as compared with offline frauds in which credit card details are compromised by fraudsters through phishing, website cloning, and skimming and used in digital transactions [3, 15].

Global connectivity through new and advanced technology has exponentially increased the credit card frauds. Thus, the issue has acquired an alarming dimension in the present scenario, and a suitable system needs to be developed for detecting and avoiding such frauds.

1.3.1. Fraud Prevention System (FPS). FPS is the first form of defense for technological systems toward forgery. The aim of this phase is to suppress first-place fraud. The techniques in this phase prohibit, destroy, and respond to cyber attacks in computer servers (software and hardware), networks, or data, for example, encryption algorithms and firewall to decipher data and to block inner private networks from outside world, respectively [3, 16].

1.3.2. Fraud Detection System (FDS). FDS becomes the next safety measure to spot and recognize the fraudulent practices when they reach the networks and notify these to a network administrator [17]. Earlier, manual auditing methods such as discovery sampling were used to detect any such fraud [18]. This method had to tackle different environmental, political, legal, and business practices. To improve detection efficiency, computerized and automatic FDSs were developed. FDS capacities have been constrained however, as identification is primarily based on predefined rules set by the experts. Different data mining approaches are being developed to detect the frauds effectively. Oddity or outlier identification in FDS depends on behavioral profiling methods that model the pattern of behavior for every entity and assess any divergence from the normal [19]. Many authors have adopted anomaly-based FDSs in different areas of fraud detection [20–23].

1.4. Distributed Deployment of Security-Related Aspects. Financial firms have indeed acknowledged that the deployment of isolated control systems on solo delivery channels apparently no longer implements the requisite degree of vigilance toward illegal account operation. An additional layer of security, i.e., "Fraud Management," is enhancing the robustness by combining with security protocols at the level of standard channel [24]. The implemented fraud detection strategy can be distributed as reactive and proactive, depending on the point where data analysis is implemented in different transaction orders. Fraud identification approaches derived from data processing, neural networks, and/or various deep learning algorithms conduct sophisticated model processing via collected datasets in reactive fraud management to identify suspect transfers.

The newly arrived operations are evaluated "on the fly" in proactive fraud management before proper authorization and finalization, to allow the detection of unusual occurrences prior to any financial value movement. Proactive fraud detection is accomplished by relocating the inherent security which allows real-time scanning prior to completion of the transaction. Statistical analysis and data mining-related approaches have been implemented on classed post-transactional data to derive common traits correlated to suspicious occurrences in fraud strategic management.

1.5. Data Imbalance Is a Major Concern. Skewed distribution is regarded as one of the chief sensitive problems of FDS [3]. Usually, the skewed data problem is the scenario where there are far fewer instances of fraudulent cases than usual [25], making it difficult for learners to uncover trends in minority class data [26]. Moreover, class imbalance has a significant influence on the efficiency of classification models, which are normally dominated by majority class labels. Imbalanced datasets have a detrimental effect on classification performance that tends to be overshadowed by the majority class, thereby ignoring the minority class. As shown in Figure 2, the data-balancing methods can be divided into two subcategories, viz., data level methods and algorithmic level methods [27].

1.5.1. Data Level Methods. Such methods are taken as preprocessing to reorient the collected data before applying the classification algorithms. Many investigators have used the balancing methods, viz., undersampling or oversampling, in FDS-related studies [3]. In undersampling, a portion of the dataset of the dominant class is eliminated [28]. A broad range of FDS has used the undersampling technique to equalize training samples. The oversampling method duplicates minority class data samples. The oversampling technique is not frequently used because it induces overfitting of a model, especially for noisy data [29]. Synthetic minority oversampling technique (SMOTE) [30] is being used for fraud detection and considered as a superior complement to its current peers. SMOTE synthesizes new minority instances in the reported zone. Investigators, in their study [31], have conducted many simulations using various data level methods (SMOTE and EasyEnsemble) to identify the most suitable credit card FDS [3].

1.5.2. Algorithmic Level Methods. In this category, classifiers have been used to detect suspicious classes in a sample dataset. The algorithmic level approach uses cost-sensitive learning (CSL) to counter unequal class distribution. CSL places a cost variable to misinterpret the various classes by presuming that a cost matrix is present for various errors. Cost matrix structure is significantly correlated with these observations: false negative/positive and true negative/positive [32]. Another algorithmic approach followed in the FDS literature would be to use learners to manage imbalanced distribution. Such learners are either immune to class inequality by the learner's intrinsic characteristics as with Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [33] or the learners are reinforced against the issue by intrinsic alterations [3].

Falsified transactions have a narrow percentage in the overall dataset that may hinder the efficiency of FDS. In credit card systems, misclassifying legitimate transactions causes dissatisfied customers, which itself is regarded more detrimental than fraud itself. As mentioned above, two approaches, viz., algorithmic and data levels, were used to fix class imbalances. The researchers, in their works [34–38], have used undersampling techniques while dealing with the concern of class skewness in credit card FDS. However, Stolfo et al. [26] have used the oversampling method in the preprocessing stage of credit card FDS.

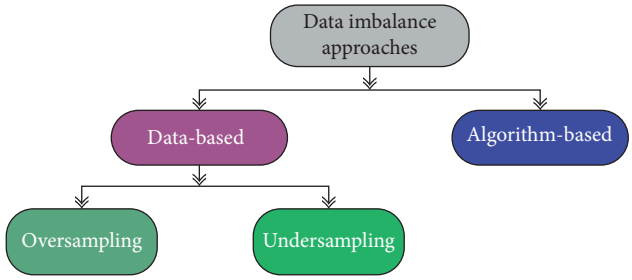

Figure 2: Various techniques of handling the concern related to data imbalance.

On the contrary, an algorithmic level approach has been followed using cost-sensitive learning techniques or by using the learner itself to manage uneven distribution. Sahin et al. [39] have used cost-sensitive classifiers to address the class imbalance. Dorronsoro et al. [21] have used nonlinear discriminant analysis (NLDA) neural models to tackle the class with imbalances. Ju and Lu [40] have used an enhanced imbalance class weighted support vector machine (ICW-SVM) to handle the skewness of the dataset. Bentley et al. [41] have given a fraud density map to enhance detection accuracy. In a study by Pozzolo et al. [42], the authors have suggested a race model to choose the right approach for an imbalance dataset. Chen [28] has used the binary support vector system (BSVS) and genetic algorithm (GA) to achieve a higher prediction accuracy from imbalance inputs. Minegishi and Niimi [43] have suggested the creation of a very fast decision tree (VFDT) learner, which could be tailored for extremely unbalanced datasets. Seeja and Zareapoor [44] have proposed FraudMiner for managing class imbalance via explicitly entering unbalanced data to the classification model. G.C. de Sá et al. have customized the bayesian network classifier (BNC) algorithm for credit card fraud detection [45]. Husejinovic has introduced a methodology to detect credit card fraud using naive bayesian and C4.5 decision tree classifiers [46]. Arya et al. have proposed deep ensemble learning to identify fraud cases in real-time data streams. The proposed model is capable of adapting to data imbalance as well as is robust to innate transaction patterns such as purchasing behavior [4].

2. Scope of the Study

This manuscript explores the concern of classifying imbalanced data by merging data level and algorithm level techniques to detect the fraudster from the log files generated for credit cards used at IoT-enabled terminals. Furthermore, an appropriate alert message can be sent to either the credit card holder or the issuer for reverting/blocking the transaction. Here, the random undersampling (RUS) approach has been deployed at the data level and boosting at the algorithmic level. The merger of these two components is RUSBoost [47]. Here, RUS is a data sampling technique that aims to mitigate class inequality by modifying the training dataset's class distribution. RUS eliminates instances from the majority class completely at random before a reasonable class distribution is reached [48, 49]. The boosting method helps in improving the classification precision of weak classifiers by combining weak hypotheses. Initially, all training dataset examples are given equal weights. Base learner forms a weak hypothesis during each iteration of adaptive boosting (AdaBoosting). Boosting is said to be adaptive since poor learners are subsequently tweaked in support of cases which are not classified by former classifiers. The inconsistency connected with the hypothesis is determined, and the weight of each instance is modified in such a manner that incorrectly classified cases raise their weights, whereas correctly classified samples decrease their weights. Thus, successive boosting steps will produce hypotheses which are able to correctly classify the previous incorrectly labeled instances. After all repetitions, a weighted vote would be used to allot a class to samples in the dataset [48]. RUSBoost is less costly than oversampling and bagging when used for classification (like SMOTEBagging).

3. Methodology

Figure 3 highlights the various phases, taking credit card transactional logs (imbalanced dataset) as input and giving an alert to the bank or the credit card holder regarding the status of the transactions performed at some IoT-based terminals.

Figure 3 shows that on the credit card transactional logs, the customized RUSBoost (CtRUSBoost) gets applied and results into showing the status of the transaction held. Here, the approach constitutes random undersampling and boosting using decision tree as per the normal RUSBoost algorithm with a further add-on/customization of having bagging process using SVM. CtRUSBoost can be deployed at the stage/step of either Credit Card Interchange or Credit Card Provider Computer Controller System (as shown in Figure 1), and from these controlling systems, an alert message can be escalated for suspending or stopping the financial transaction. The various symbolic notations used in the proposed algorithm CtRUSBoost have been defined in Table 1.

The RUSBoost given by Seiffert et al. [48, 49] has been modified by the authors here in this research work. The rounded rectangles at steps 2d, 2e, 3a, 3b, and 4 show the customization proposed by the authors here, which has resulted in comparatively better outcomes. In step 1, the weights of each sample are initialized to $(1/x)$, where x is the total of instances in the training dataset. The weak hypotheses, viz., DT and SVM, are iteratively trained in steps 2a–2i. In step 2a, random undersampling has been implemented to suppress the class labels until the required minority class proportion is reached in the current (temporary) training dataset SEG'_z. For example, if the required class proportion is 50:50, then most class instances are predictably excluded until majority and minority class instances are comparable. Therefore, SEG'_z will have a new distribution of weight as DIS'_z. Step 2b moves SEG'_z and DIS'_z to the decision tree, generating the weak hypothesis h_z (step 2c). In step 2d, support vector machine has been employed to compute the weak hypothesis h_z^{svm} in step 2e. The pseudo loss ε_t (based on SEG and DIS_z) has been determined in step 2f.

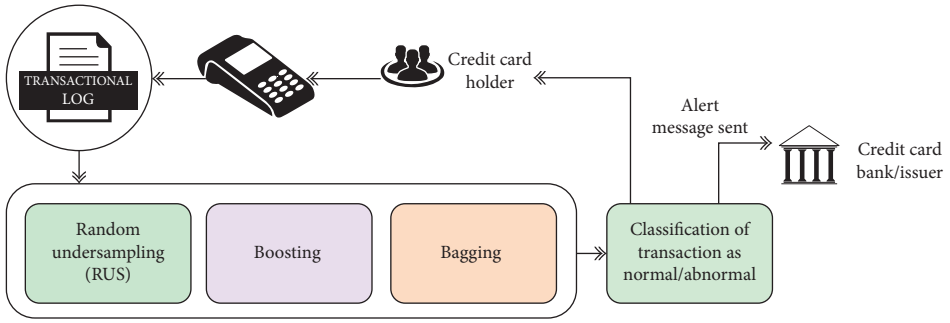

Figure 3: Steps involved in classification of imbalanced transactional logs as normal or abnormal.

Table 1: Symbolic notations used in the proposed algorithm CtRUSBoost.

SEG	Dataset segment under consideration
$h_z^{\text{svm}}(p_k)$	Hypothesis value obtained through support vector machine in z^{th} iteration for the instance p_k (this serves as a numeric confidence rating)
$h_z(p_k)$	Hypothesis value obtained through decision tree in z^{th} iteration for the instance p_k (this serves as a numeric confidence rating)
ε_z	Cumulative pseudo loss
α_z	Parameter to update the weight factor
C_z	Factor for normalizing the $(z+1)^{\text{th}}$ distribution of weights taking the full training dataset/or normalized value for the distribution
$\text{DIS}_z(k)$	Distribution of weights at z^{th} iteration taking the full training dataset for the k^{th}sample
DIS_{z+1}	Distribution of weights at $(z+1)^{\text{th}}$ iteration taking the full training dataset
DIS_z'	Distribution of weights for z^{th} temporary training dataset
SEG_z'	z^{th}temporary training dataset
p_i	i^{th} row with values of all columns except the last one (i.e., label)
q_i	A label for the i^{th} row
q^r	Minority class label
Z	Total number of iterations employed in the ML model
k or x	Total counts of samples present in the SEG
P	Rows/tuples in the dataset (excluding the last column having labeled entries)
Q	Total available labels in the dataset

In step 2f, the hypothesis values for those tuples have only been considered where there is a misclassification. Here, in the subexpression $q_k \neq q$, q_k means the original label/class of the k^{th} row/tuple in the dataset and q is the label/class obtained after employing/deploying the weak learner decision tree. Subexpression $h_z(p_k, q_k)$ is the numeric confidence value in z^{th} iteration for the instance p_k, where the label is q_k, and subexpression $h_z(p_k, q)$ is the numeric confidence value in the same z^{th} iteration for the instance p_k considered earlier, where the label is mismatched and obtained as q instead of q_k. In step 2g, the parameter α is computed as $(\varepsilon_z/(1-\varepsilon_z))$which symbolizes the weight update. In step 2h, the weight distribution gets updated DIS_{z+1}. Step 2i normalizes the value computed in the previous step. After the completion of Z iterations, in step 3a, the maximum value of h_z has been computed among the ones given by decision tree under boosting, where the knowledge/learning from the previous dataset segment has been used for getting the hypothesis value of the next dataset segment, but in the last step, all the results have not been merged to obtain the final one. Instead, the final value of the hypothesis has been obtained from the last dataset segment. In step 3b, hypothesis values as obtained by employing SVM for each dataset segment in Ziterations have been finalized by performing voting or averaging among all the values of h_z^{svm}. In step 4, the final hypothesis $H(p)$ has been computed taking the maximum of the value obtained for h_z and h_z^{svm}.

4. Results and Experiment

The results obtained after using the three different datasets, viz., (i) Abstract Dataset for Credit Card Fraud Detection [50], (ii) Default of Credit Card Client Dataset [51], and (iii) Credit Card Fraud Dataset [52] are shown in this section. Customized RUSBoost results were compared using RUSBoost, decision tree (DT), logistic regression (LR), multilayer perceptron (MLP), K-nearest neighbors (KNN), random forest (RF), AdaBoost, and support vector machine (SVM).

Three separate datasets based on the number of tuples were taken for the current work. Datasets of less than five thousand tuples were considered as small; tuples with a range of over five thousand and less than ten thousand were considered as medium; and those with a range of over ten thousand entries were considered as large. All the datasets have been divided into two partitions, i.e., 80% and 20% of the full dataset, where the bigger portion has been taken for training and the smaller one for testing of the machine learning models.

4.1. Small Dataset. The dataset called Abstract Dataset for Credit Card Fraud Detection (Dataset A) [50] has been taken from the kaggle.com database. The authors classified this as a small dataset with less than 5,000 tuples. The dataset included the usage of 3,075 clients and 11 attributes. Of the 3,075 samples, 2,627 represent nonfraudulent transactions and 448 are fraudulent transactions (about 6:1). The eleven variables taken in this dataset are described in Table 2.

4.2. Medium Dataset. The dataset called Default of Credit Card Client Dataset (Dataset B) [51] has also been taken from the kaggle.com database. This includes details on default payments, demographic factors, credit data, payment history, and credit card company bills in Taiwan from April 2005 to September 2005. Among the 30,000 observations, 23,364 are cardholders with default payment as no and 6,636 with status as yes (about 4:1). Default payment in the finance domain is known as nonrepayment of debt such as interest or principal toward credit or estate. A default can result when a purchaser could not render payments on time, slows payouts, or declines or drops payment [53].

This dataset used a binary variable default payment as the answer variable. Table 3 explains the twenty-four variables taken up in Dataset B.

4.3. Large Dataset. The dataset called Credit Card Fraud Detection (Dataset C) [52] was taken again from the kaggle.com database. This dataset includes purchases by European cardholders in September 2013. This sample dataset outlined two-day activities, with 492 frauds out of 284,807 total transactions. The dataset is highly imbalanced, where the positive class (fraud) constitutes 0.172% of all transactions deemed. The details of the dataset's features are given in Table 4 and include all numeric values.

It includes only numerical variables resulting from PCA transformation. Kaggle did not provide any original features as well as additional details due to privacy concerns. Features $V_1, V_2, \ldots,$ and V_{28} are the key PCA components with untransformed attributes as "time" and "amount."

4.4. Evaluation Metrics. Assessment measures are employed to calculate statistic or machine learning model efficiency. A confusion matrix gives us the output matrix that characterizes the model's complete efficiency. Here, in the proposed model, the security context is said to be robust if the model is capable of finding/classifying fraudster transactions accurately. The metrics used for comparing ML models for their accuracy are sensitivity and specificity from the confusion matrix, precision, $F1$ score, receiver operating characteristic (ROC), and area under precision recall (AUPR).

4.4.1. Confusion Matrix. The confusion matrix is a representation of an algorithm's performance in the field related to machine learning. The term "Confusion" has appeared from the fact that if the machine learning model causes confusion between two classes, it is easy to see. Figure 4 depicts a confusion matrix providing sensitivity, specificity, recall, and fall-out information. The column in this matrix represents instances in the actual class, while each row represents instances in one expected class.

Sensitivity is an estimate of the total of truly positive instances expected to be positive. The larger sensitivity value will have a high true positive value and less false negative value. Models with high sensitivity are required for health and financial purposes. Specificity is defined as the share of actual negatives, predicted to be negative. This ratio may also be called the false positive rate. The higher specificity value will mean the higher true negative and lower false positive rate.

4.4.2. Precision and F1 Score. Precision and F-measurements are considered more suitable for estimating the performance of a classification algorithm when the dataset is imbalanced, where precision is characterized as the positive predictive value. F-measure in the confusion matrix is the weighted harmonic mean of sensitivity and precision [54]:

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ F1 &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \end{aligned} \tag{1}$$

Precision is the percentage of true positives to all positives. For our problem statement here, the precision would be the measure of fraudster transactions that we correctly identified as fraud out of all the transactions, which are actually fraud. Recall refers to the proportion of the overall predictions of the algorithm being accurately categorized. Furthermore, the value of $F1$ gives a single score that balances out both recall and the precision.

Here, decision tree, logistic regression, multilayer perceptron (MLP), K-nearest neighbor (KNN), random forest (RF), AdaBoost, and support vector machine (SVM) models have been compared w.r.t. sensitivity, specificity, precision, and $F1$ score. Decision tree is a nonparametric, supervised learning system for classification and regression tasks. The decision tree is designed using an algorithmic method that recognizes ways of splitting data based on different conditions. Logistic regression is an algorithm for machine learning that is based on the probability principle. It is an algorithm for classification used to attribute observations to a specific class set. Using the logistic sigmoid function, logistic regression transforms the output to return a probability value. A multilayer perceptron is a neural network that links different layers in a directed graph, meaning the signal path through nodes only goes one directional. In MLP, every node is having a nonlinear activation function, except the input nodes. K-nearest neighbor is a single algorithm that holds all existing cases in a similarity measure (i.e., distance function) and classifies new cases. The random forest algorithm generates decision trees on data samples and then obtains predictions from each and finally, picks the best option by voting. In AdaBoost, a sequence of weak learners is linked so that each weak classifier attempts to enhance the

(i) Input: x, SEG, $P \times Q$(with $q^r \in Q$, $|Q| = 2$)
(ii) Output: maximum of [(maximum of h_z value), (maximum of h_z^{svm}value)]
Begin
(1) Initialization of $\text{DIS}_1(k) = 1/x$ for all k
(2) Do for $z = 1, 2, 3, \ldots, Z$
(a) Create temporary training dataset SEG_z' with weight distribution DIS_z' by using random undersampling
(b) Call decision tree, considering the sample set as SEG_z' and distribution of weight DIS_z'
(c) Compute a hypothesis h_z: $P \times Q \longrightarrow [0, 1]$
(d) Call support vector machine considering the sample set as SEG_z' and distribution of weight as DIS_z'
(e) Compute a hypothesis h_z^{svm}: $P \times Q \longrightarrow [0, 1]$
(f) Compute the pseudo loss for SEG and DIS_z
$\varepsilon_z = \sum_{(k,q):\, q_k \neq q} \text{DIS}_z(k)(1 - h_z(p_k, q_k) + h_z(p_k, q))$
(g) Compute the parameter to update the weighing factor:
$\alpha_z = (\varepsilon_z / 1 - \varepsilon_z)$
(h) Update DIS_z:
$\text{DIS}_{z+1}(k) = \text{DIS}_z(k)\alpha_z^{(1/2)(1+h_z(p_k,q_k)-h_z(p_k,q:\, q_k \neq q))}$
(i) Normalize DIS_{z+1}: Let $C_z = \sum_z \text{DIS}_{z+1}(k)$
$\text{DIS}_{z+1}(k) = (\text{DIS}_{z+1}(k)/C_z)$
(3) Find the values for h_z and h_z^{svm}
(a) For each value of h_z, where $z = \{1, 2, \ldots, Z\}$, find out the maximum value of h_z
(b) For each value of h_z^{svm}, where $z = \{1, 2, \ldots, Z\}$, apply bagging either by performing voting or averaging among all the values of hypothesis obtained
(4) Compute the final hypothesis $H(p)$ as the maximum value between $h_z$and h_z^{svm}
End

Algorithm 1: CtRUSBoost (customized RUSBoost).

Table 2: Attribute number, name, and definition of Dataset A.

Attribute	Description
$X1$	Merchant ID : ID of the merchant
$X2$	Average amount/transaction/day
$X3$	Total amount of transaction
$X4$	Is declined: declining or falling transaction (yes or no)
$X5$	Total number of declines/days: total transaction numbers declined daily
$X6$	Is foreign transaction: transaction carried out is or is not a foreign transaction
$X7$	Is high-risk country: transaction is performed in countries under high risk
$X8$	Average daily chargeback amount
$X9$	Average chargeback (taken for six months)
$X10$	Frequency of chargeback (taken for six months)
$X11$	Is fraudulent: transaction is a fraud or not

classification of observations incorrectly labeled by the preceding weak classifier. Support vector machine uses a kernel trick to transform data and then determines an optimal boundary between potential outputs. The results showing comparison among customized RUSBoost, decision tree, logistic regression, multilayer perceptron (MLP), K-nearest neighbor (KNN), random forest (RF), AdaBoost, and support vector machine (SVM) models have been presented in Tables 5–7.

In Table 7, the value that has been observed for the precision and $F1$ score is NaN under SVM because the zero divided by zero is undefined as a real number, and in computing systems, it can be represented as NaN.

4.4.3. Receiver Operating Characteristic (ROC). In machine learning, measuring efficiency is an integral activity. ROC is considered the most significant measurement to test the efficiency of any classification model. It tells how much the model can differentiate between classes. The higher the AUC, the better it would be to predict 0 s as 0 s and 1 s as 1 s. The curve for ROC is plotted with TP rate vs. FP rate, taking TP and FP rates at y-axis and x-axis, respectively [55]. Figures 5–7 depict the ROC for the customized RUSBoost and its peer techniques, i.e., simple RUSBoost, DT, LR, MLP, KNN, RF AdaBoost, and SVM, indicating the optimality of the proposed customization in RUSBoost on the benchmark datasets A, B, and C, respectively.

Table 3: Attribute number, name, and definition of Dataset B (amount in New Taiwan or NT dollar).

Attribute	Description
$X1$	Credit amount
$X2$	Gender of the borrower 1 for male 2 for female
$X3$	Level of education 1 Graduate school 2 University 3 High school 4 Others 5/6 Unknown
$X4$	Marital status of the borrower 1 Married 2 Single 3 Others
$X5$	Age of the credit card holder (in years)
$X6$–$X11$	PAY_1 to PAY_6: status of payment return in September to April 2005 Paid on-time payment = −1 One-month payment delay = 1 Two-month payment delay = 2 . . . Nine or above months of payment delay = 9
$X12$–$X17$	BILL_AMT1-6: amount of bill for the months April to September 2005
$X18$–$X23$	PAY_AMT1-6: previous payment in April to September 2005
$X24$	Status as 1 for yes and 0 for no under the default payment

		True condition	
	Total population	Actual condition positive	Actual condition negative
Predicted condition	Predicted condition positive	True positive (TP) rate, sensitivity $= \frac{\Sigma\ True\ positive}{\Sigma\ Condition\ positive}$	False positive (FP) rate $= \frac{\Sigma\ False\ positive}{\Sigma\ Condition\ negative}$
	Predicted condition negative	False negative (FN) rate $= \frac{\Sigma\ False\ negative}{\Sigma\ Condition\ positive}$	True negative (TN) rate, specificity $= \frac{\Sigma\ True\ negative}{\Sigma\ Condition\ negative}$

Figure 4: Sensitivity, specificity, FP rate, and FN rate formulas in the confusion matrix.

Table 4: Attribute number, name, and definition of Dataset C.

Attribute	Description
$V_1 \ldots V_{28}$	The parameters have been anonymized with principal component analysis (PCA) to protect the user identities
Time	Time intervened between transactions (in seconds)
Amount	Amount of the transaction
Class	Final label; 1 = fraud, 0 = otherwise

Besides ROC, the precision recall (PR) curves are also considered better for evaluating the algorithmic efficiency when the sample set is highly biased. The results of the current work are also presented through an AUPR curve obtained on various machine learning models.

4.4.4. Area under Precision Recall (AUPR). The ROC curve has some drawbacks, including class skew decoupling. That is why the precision recall (PR) curve, which plots precision against recall and is equivalent to the false discovery rate curve, has gained attention in recent years. This output

Table 5: Sensitivity, specificity, precision, and *F*1 scores obtained on Dataset A executing RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

Model name	Sensitivity	Specificity	Precision	*F*1 score
RUSBoost	50.6	99.8	33.4	40.2
Customized RUSBoost	96.3	85.6	94.2	88.6
DT	76.5	97.9	72.6	75.4
LR	57.0	99.0	86.0	68.7
MLP	70.4	99.5	95.8	81.1
KNN	80.6	99.9	95.1	87.2
RF	53.2	99.0	82.3	64.5
AdaBoost	73.4	99.0	83.7	78.2
SVM	61.2	99.9	96.8	75.7

Table 6: Sensitivity, specificity, precision, and *F*1 scores obtained on Dataset B executing RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

Model name	Sensitivity	Specificity	Precision	*F*1 score
RUSBoost	34.6	98.3	85.9	59.4
Customized RUSBoost	99.6	98.7	95.7	97.6
DT	40.6	81.0	49.5	50.7
LR	23.6	97.0	69.6	35.0
MLP	38.5	93.2	61.4	47.3
KNN	37.8	89.4	50.0	43.1
RF	5.5	99.2	68.2	10.2
AdaBoost	30.8	95.8	67.3	42.3
SVM	33.2	95.2	67.8	44.5

Table 7: Sensitivity, specificity, precision, and *F*1 scores obtained on Dataset C executing RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

Model name	Sensitivity	Specificity	Precision	*F*1 score
RUSBoost	34.6	98.3	85.9	59.4
Customized RUSBoost	99.6	98.7	95.7	97.6
DT	40.6	81.0	49.5	50.7
LR	23.6	97.0	69.6	35.0
MLP	38.5	93.2	61.4	47.3
KNN	37.8	89.4	50.0	43.1
RF	5.5	99.2	68.2	10.2
AdaBoost	30.8	95.8	67.3	42.3
SVM	33.2	95.2	67.8	44.5

metric has been widely used in various fields such as computer vision, computational biology, data analysis, medicine, and natural language processing. As a single score, the AUPR summarizes the precision recall curve and can be used to easily compare different binary classification models. The AUPR ′s value for a perfect classifier is 1. The high precision and recall system will provide correctly labeled results [55]. Figures 8–10 depict the AUPR for the customized RUSBoost and its peer techniques, i.e., simple RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM, indicating the optimality of the algorithm on the benchmark datasets A, B, and C, respectively.

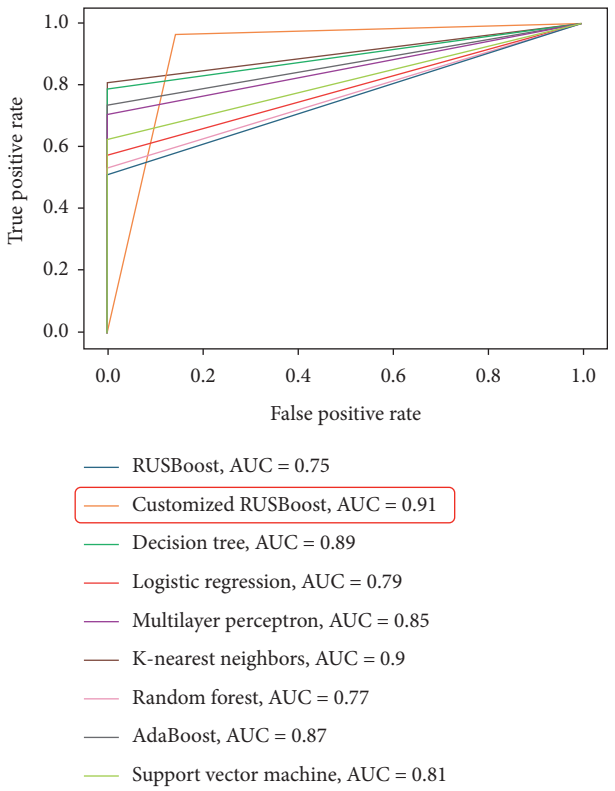

Figure 5: ROC curve obtained on the Default of Credit Card Client Dataset after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

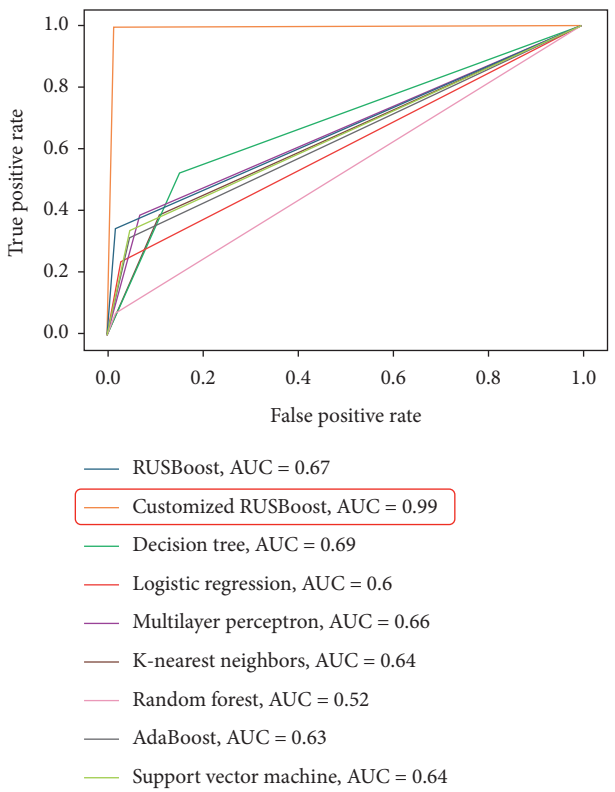

Figure 6: ROC curve obtained on the Credit Card Fraud Detection Dataset after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

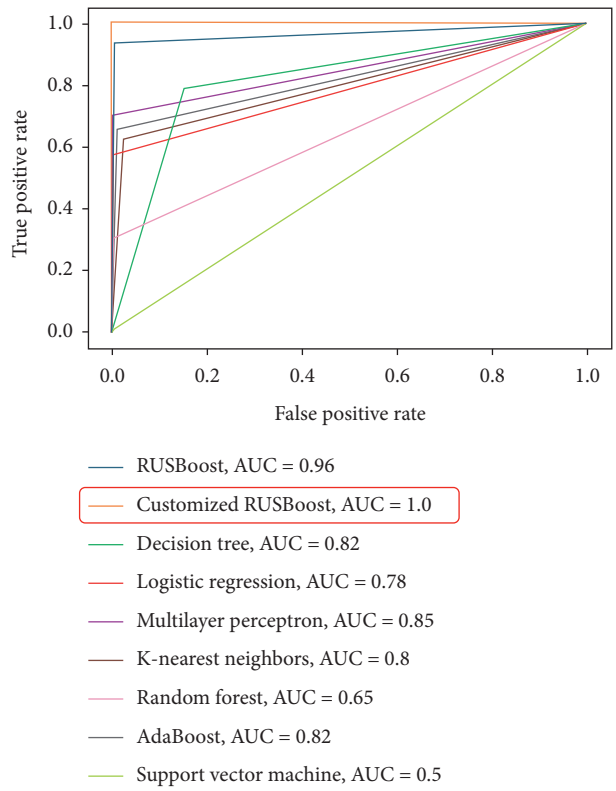

Figure 7: ROC curve obtained on the Abstract dataset for Credit Card Fraud Detection after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

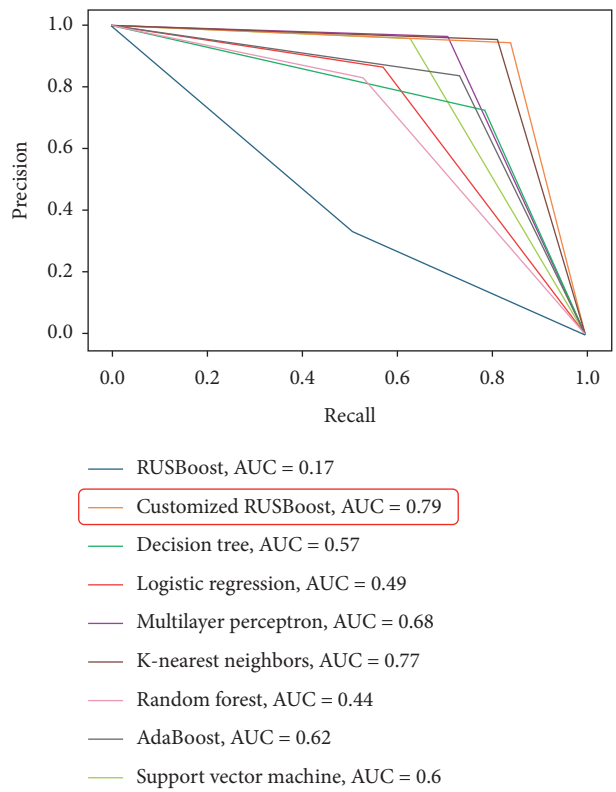

Figure 8: AUPR curve obtained on the Dataset A after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

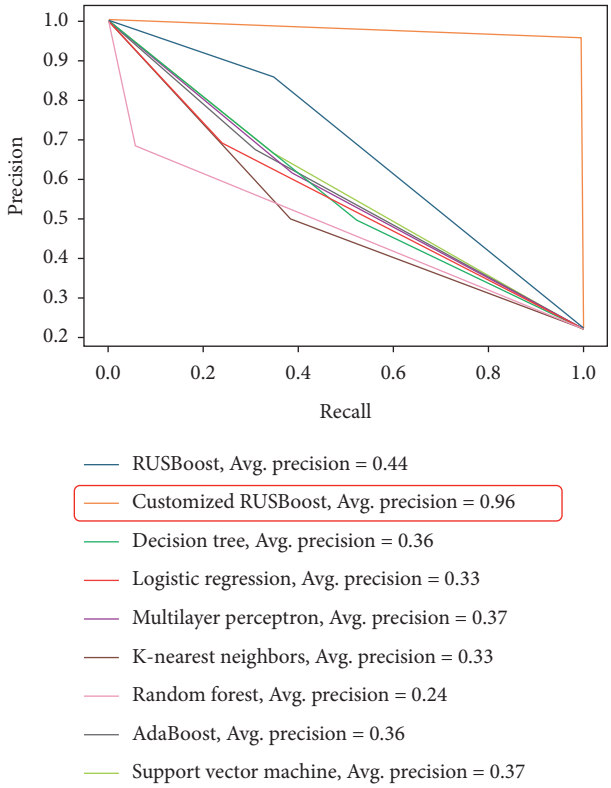

Figure 9: AUPR curve obtained on the Dataset B after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

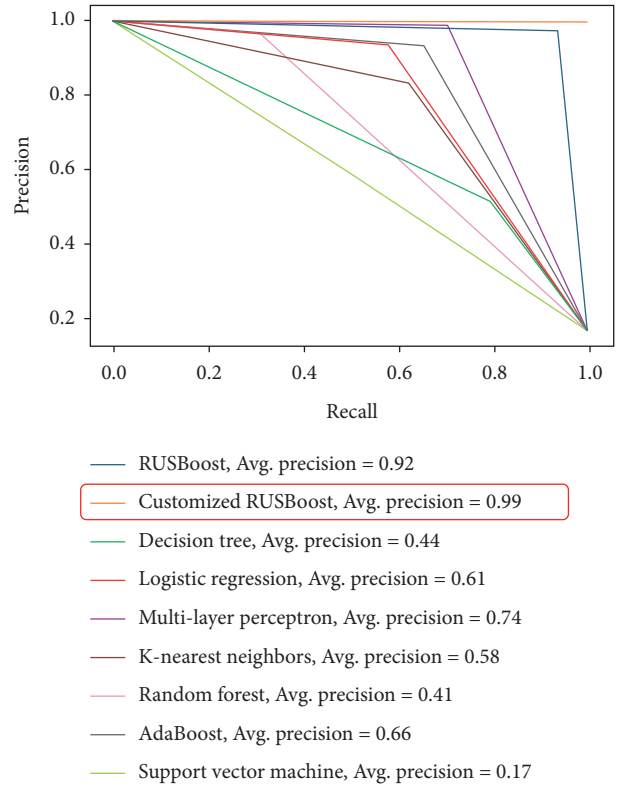

Figure 10: AUPR curve obtained on the Dataset C after deploying RUSBoost, customized RUSBoost, DT, LR, MLP, KNN, RF, AdaBoost, and SVM.

5. Conclusion

In this research work, the existing RUSBoost algorithm has been customized by using a combination of bagging and boosting. The results obtained after customizing the RUSBoost in the proposed methodology are more reliable and authentic when compared with simple/normal RUSBoost, DT, RF, AdaBoost, SVM, LR, KNN, and MLP. The scores obtained for the CtRUSBoost algorithm on three benchmark datasets A, B, and C taken from kaggle.com are 96.30, 99.60, and 100, respectively, for sensitivity; 85.60, 98.70, and 99.80, respectively, for specificity; 94.20, 95.70, and 99.30, respectively, for precision; and 88.60, 97.60, and 99.60, respectively, for $F1$ score. The results obtained from CtRUSBoost have outperformed all the peer approaches used in this study by a large margin, which means it can detect fraudster transactions more robustly. In the future, the work proposed here can be customized further by adding weak classifiers to the process such as K-nearest neighbors, linear regression, and multilayer perceptron.

Data Availability

The datasets used during the current study are available at kaggle.com, and web links to the datasets are as follows: kaggle small-sized dataset, https://www.kaggle.com/shubhamjoshi2130of/abstract-data-set-for-credit-card-fraud-detection, kaggle medium-sized dataset, https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset, and kaggle large-sized dataset, https://www.kaggle.com/mlg-ulb/creditcardfraud. The datasets used to support the findings of this study are included within the article at reference numbers [50–52].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (no. 2018R1A4A1025632).

References

[1] L. Delamaire, H. Abdou, and J. Pointon, "Credit card fraud and detection techniques: a review," *Banks and Bank Systems*, vol. 4, no. 2, pp. 57–68, 2009.

[2] S. Benson Edwin Raj and A. Annie Portia, "Analysis on credit card fraud detection methods," in *Proceedings of the 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*, pp. 152–156, Tamil Nadu, India, March 2011.

[3] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: a survey," *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.

[4] M. Arya and G. Hanumant Sastry, "DEAL–"deep ensemble algorithm" framework for credit card fraud detection in real-time data stream with Google TensorFlow," *Smart Science*, vol. 8, no. 2, pp. 71–83, 2020.

[5] K. K. Sherly and R. Nedunchezhian, "BOAT adaptive credit card fraud detection system," in *Proceedings of the 2010 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–7, Coimbatore, India, December 2010.

[6] N. Khare, P. Devan, C. Lal Chowdhary et al., "Spider monkey optimization and deep neural network hybrid classifier model for intrusion detection," *Electronics*, vol. 9, no. 4, p. 692, 2020.

[7] "InterWeave payment gateway, CreatioMarketplace," 2020, https://marketplace.creatio.com/app/interweave-payment-gateway.

[8] S. P. Mishra and P. Kumari, "Analysis of techniques for credit card fraud detection: a data mining perspective," in *New Paradigm, in Decision Science and Management*, I. A. S. Patnaik, M. Tavana, and V. Jain, Eds., vol. 1005, pp. 89–98, Springer, Singapore, Asia, 2020.

[9] J. West and M. Bhattacharya, "Intelligent financial fraud detection: a comprehensive review," *Computers & Security*, vol. 57, pp. 47–66, 2016.

[10] J. Johannes, M. Granitzer, K. Ziegler et al., "Sequence classification for credit-card fraud detection," *Expert Systems with Applications*, vol. 100, pp. 234–245, 2018.

[11] V. Sharma, R. Kumar, W.-H. Cheng, M. Atiquzzaman, K. Srinivasan, and A. Y. Zomaya, "Neuro-fuzzy based horizontal anomaly detection in online social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 11, pp. 2171–2184, 2018.

[12] D. Yue, X. Wu, Y. Wang, Li Yue, and C.-H. Chu, "A review of data mining-based financial fraud detection research," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 5519–5522, Shanghai, China, September 2007.

[13] Z. Li, V. Sharma, and S. P. Mohanty, "Preserving data privacy via federated learning: challenges and solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 3, pp. 8–16, 2020.

[14] N. Laleh and M. A. Azgomi, "A taxonomy of frauds and fraud detection techniques," in *Proceedings of the International Conference on Information Systems, Technology and Management*, pp. 256–267, Ghaziabad, India, March 2009.

[15] S. Zhang and J.-H Lee, "A group signature and authentication scheme for blockchain-based mobile-edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4557–4567, 2019.

[16] S. Singh, P. K. Sharma, B. Yoon, M. Shojafar, G. H. Cho, and I.-H. Ra, "Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city," *Sustainable Cities and Society*, vol. 63, Article ID 102364, 2020.

[17] M. Behdad, L. Barone, M. Bennamoun, and T. French, "Nature-inspired techniques in the context of fraud detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1273–1290, 2012.

[18] S. Tennyson and P. Salsas-Forn, "Claims auditing in automobile insurance: fraud detection and deterrence objectives," *Journal of Risk & Insurance*, vol. 69, no. 3, pp. 289–308, 2002.

[19] J. Veeramreddy, V. V. Rama Prasad, and K. Munivara Prasad, "A review of anomaly based intrusion detection systems," *International Journal of Computer Applications*, vol. 28, no. 7, pp. 26–35, 2011.

[20] S. Ghosh and D. L. Reilly, "Credit card fraud detection with a neural-network," in *Proceedings of the System Sciences, Proceedings of the Twenty-Seventh Hawaii International Conference*, pp. 621–630, Wailea, HI, USA, January 1994.

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

[22] M. Taniguchi, M. Haft, J. Hollmén, and V. Tresp, "Fraud detection in communication networks using neural and probabilistic methods," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, pp. 1241–1244, Seattle, WA, USA, May 1998.

[23] R. Brause, T. Langsdorf, and M. Hepp, "Neural data mining for credit card fraud detection," in *Proceedings of the 11th International Conference on Tools with Artificial Intelligence*, pp. 103–106, Chicago, IL, USA, November 1999.

[24] E. Michael and P. R. Falcone Sampaio, "The design of FFML: a rule-based policy modelling language for proactive fraud management in financial data streams," *Expert Systems with Applications*, vol. 39, no. 11, pp. 9966–9985, 2012.

[25] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using Bayesian and neural networks," in *Proceedings of the 1st International Naiso Congress on Neuro Fuzzy Technologies*, pp. 261–270, Havana, Cuba, January 2002.

[26] S. J. Stolfo, D. W. Fan, W. Lee, and A. L. Prodromidi, "Credit card fraud detection using meta-learning," in *Proceedings of the AAAI Workshop on Fraud Detection and Risk Management*, pp. 83–90, Providence, RI, USA, July 1997.

[27] V. López, A. Fernández, G. Jose, Moreno-Torres, and F. Herrera, "Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6585–6608, 2012.

[28] R.-C. Chen, T. Chen, and C.-C. Lin, "A new binary support vector system for increasing detection rate of credit card fraud," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 20, no. 2, pp. 227–239, 2006.

[29] P. Brennan, "A comprehensive survey of methods for overcoming the class imbalance problem in fraud detection," M.Sc. in Computing Thesis, Institute of Technology, Blanchardstown, Dublin, Ireland, 2012.

[30] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[31] A. Dal Pozzolo, C. Olivier, Yann-Aël Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4915–4928, 2014.

[32] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Proceedings of the 3rd International Conference on Data Mining*, pp. 435–442, Melbourne, FL, USA, November 2003.

[33] P. K. Chan, W. Fan, A. Prodromidir, and S. Stalfo, "Distributed data mining in credit card fraud detection," *IEEE Intelligent Systems and Their Applications*, vol. 14, no. 6, pp. 67–74, 1999.

[34] F. Nick, R. Tubb, and P. Krause, "Neural network rule extraction to detect credit card fraud," in *Proceedings of the Engineering Applications of Neural Networks*, pp. 101–110, Corfu, Greece, September 2011.

[35] E. Duman and Y. Sahin, "Detecting credit card fraud by decision trees and support vector machines," in *Proceedings of*

the International Multi Conference of Engineers and Computer Scientists (IMECS), vol. 1, Hong-Kong, China, March 2011.

[36] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: a comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.

[37] C. Phua, K. Smith-Miles, V. C.-S. Lee, and R. Gayler, "Resilient identity crime detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 533–546, 2010.

[38] E. Duman, A. Buyukkaya, and I. Elikucuk, "A novel and successful credit card fraud detection system implemented in a Turkish bank," in *Proceedings of the 13th International Conference on Data Mining Workshops*, pp. 162–171, Dallas, TX, USA, December 2013.

[39] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916–5923, 2013.

[40] Q. Lu and C. Ju, "Research on credit card fraud detection model based on class weighted support vector machine," *Journal of Convergence Information Technology*, vol. 6, no. 1, pp. 62–68, 2011.

[41] P. J. Bentley, J. Kim, G.-H. Jung, and J.-U. Choi, "Fuzzy darwinian detection of credit card fraud," in *Proceedings of the 14th Annual Fall Symposium of the Korean Information Processing Society*, vol. 14, Seoul, Korea, Ootober 2000.

[42] A. D. Pozzolo, O. Caelen, S. Waterschoot, and G. Bontempi, "Racing for unbalanced methods selection," in *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*, pp. 24–31, Hefei, China, October 2013.

[43] T. Minegishi and A. Niimi, "Proposal of credit card fraudulent use detection by online-type decision tree construction and verification of generality," *International Journal for Information Security Research (IJISR)*, vol. 1, no. 4, pp. 229–235, 2011.

[44] K. R. Seeja and M. Zareapoo, "FraudMiner: a novel credit card fraud detection model based on frequent itemset mining," *The Scientific World Journal*, vol. 2014, Article ID 252797, 2014.

[45] A. G. C. De S´a, A. C. M. Pereira, and G. L. Pappa, "A customized classification algorithm for credit card fraud detection," *Engineering Applications of Artificial Intelligence*, vol. 72, pp. 21–29, 2018.

[46] A. Husejinovic, "Credit card fraud detection using naive Bayesian and C4.5 decision tree classifiers," *Periodicals of Engineering and Natural Sciences*, vol. 8, no. 1, pp. 1–5, 2020.

[47] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "A novel noise-resistant boosting algorithm for class-skewed data," in *Proceedings of the 11th International Conference on Machine Learning and Applications*, pp. 551–557, Boca Raton, FL, USA, December 2012.

[48] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: a hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2009.

[49] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: improving classification performance when training data is skewed," in *Proceedings of the 19th International Conference on Pattern Recognition*, pp. 1–4, Tampa, FL, USA, December 2008.

[50] S. Joshi, "Abstract data set for credit card fraud detection," 2020, https://www.kaggle.com/shubhamjoshi2130of/abstract-data-set-for-credit-card-fraud-detection.

[51] U. M. Learning, "Default of credit card clients dataset," 2016, https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset.

[52] M. L. G. ULB, "Credit card fraud detection," 2018, https://www.kaggle.com/mlg-ulb/creditcardfraud.

[53] J. Chen, "Default," 2020, https://www.investopedia.com/terms/d/default2.asp.

[54] J. Akosa, "Predictive accuracy: a misleading performance measure for highly imbalanced data," in *Proceedings of the SAS Global Forum*, pp. 2–5, Orlando, FL, USA, April 2017.

[55] V. Arora, R. Leekha, R. Singh, and I. Chana, "Heart sound classification using machine learning and phonocardiogram," *Modern Physics Letters B*, vol. 22, no. 26, Article ID 1950321, 2019.