Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 2531210, 13 pages
https://doi.org/10.1155/2021/2531210

Research Article

A Heterogeneous Ensemble Learning Model Based on Data Distribution for Credit Card Fraud Detection

Yalong Xie, Aiping Li, Liqun Gao, and Ziniu Liu

College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China

Correspondence should be addressed to Aiping Li; liaiping@nudt.edu.cn

Received 11 June 2021; Revised 4 July 2021; Accepted 12 July 2021; Published 22 July 2021

Academic Editor: Shan Zhong

Credit card fraud detection (CCFD) is important for protecting the cardholder's property and the reputation of banks. Class imbalance in credit card transaction data is a primary factor affecting the classification performance of current detection models. However, prior approaches are aimed at improving the prediction accuracy of the minority class samples (fraudulent transactions), but this usually leads to a significant drop in the model's predictive performance for the majority class samples (legal transactions), which greatly increases the investigation cost for banks. In this paper, we propose a heterogeneous ensemble learning model based on data distribution (HELMDD) to deal with imbalanced data in CCFD. We validate the effectiveness of HELMDD on two real credit card datasets. The experimental results demonstrate that compared with current state-of-the-art models, HELMDD has the best comprehensive performance. HELMDD not only achieves good recall rates for both the minority class and the majority class but also increases the savings rate for banks to 0.8623 and 0.6696, respectively.

1. Introduction

With the rapid development of mobile internet and e-commerce technologies, online payment tools such as credit cards are welcomed by more and more people. While credit cards bring convenience to customers, they also expose cardholders and banks to potential fraud risks [1, 2]. Credit card fraud is a global problem. The Nilson report found that by 2023, the worldwide fraud loss is expected to reach \$35.67 billion annually [3]. Fraud prevention and fraud detection are two main ways to combat credit card fraud [4]. Fraud prevention consists of a series of rules, procedures, and protocols. Commonly used technologies in fraud prevention include secure payment gateways, intrusion detection systems, and firewalls [5]. Fraud detection takes place after the fraud prevention mechanism has been breached [4], which means that fraud detection is the last line of defense to ensure the security of credit card transactions. Banks have to invest considerable money to optimize their fraud detection system [6], due to the need to protect cardholder's funds and their own business reputation.

Data mining and machine learning are widely used technologies in financial fraud detection [7–9]. As early as 1998, researchers had begun to build CCFD systems based on machine learning techniques [10]. After more than two decades of development, researchers have proposed many different methods and models [2, 11]. In machine learning terms, CCFD is a typical binary classification problem. The detection system is aimed at determining whether the current transaction is either legal (the transaction was made by the cardholder) or fraudulent (the transaction was made by an unauthorized person) based on historical transaction data [12]. Various methods have been proposed to tackle this problem, including supervised learning, unsupervised learning, and semisupervised learning. In supervised learning, the historical transaction data (training data) are labeled with known outcomes. Commonly used supervised learning models include Hidden Markov Model (HMM) [13], Logistic Regression (LR) [14], Support Vector Machine (SVM) [15], K-nearest neighbors (KNN) [16], Bayesian Networks (BN) [17], Decision Tree (DT) [18], random forest (RF) [19], and Artificial Neural Network (ANN) [20]. Conversely, the historical transaction data used in unsupervised learning models (ULMs) are unlabeled. ULMs judge whether transactions are fraudulent by observing the distribution of current and historical transaction data. Commonly used ULMs include artificial immune systems [21] and self-organizing maps [22]. Semisupervised learning models are a combination of supervised and unsupervised learning models, which use some labeled data in combination with a large amount of unlabeled data. This can help banks reduce the cost of labeling large volumes of transaction data [12, 23].

In the real world, the proportion of fraudulent transactions (minority class) is much lower than that of legitimate transactions (majority class), which means that the distribution of credit card transaction data is highly imbalanced, and this increases the difficulty of fraud detection [15, 24]. Most standard classifiers have poor performance on imbalanced data, especially for the minority class [25]. Resampling is a widely used method to address the problem of imbalanced classification data. Several resampling algorithms have been proposed to improve the recognition performance of classifiers for the minority class [26–28]. However, the disadvantage of the resampling method is that it significantly reduces the performance of classifiers for the majority class. For CCFD, this means that a large number of legal transactions are misclassified as fraudulent, which will significantly increase the investigation costs. Therefore, it is critical to build a CCFD model with strong recognition performance in both the minority and majority classes.

To address the above issues, we propose a new kind of heterogeneous ensemble learning model based on data distribution (HELMDD) for credit card fraud detection. The core idea is to incorporate a resampling method based on the distribution of data (RMDD). To reduce information loss in the majority class and improve the performance of the base classifiers, RMDD applies KNN and K-Means algorithms to obtain samples from the majority class, which retain its diversity and boundary contours. Finally, balanced subsets for training the base classifiers are obtained by pairing majority and minority class training subsets.

The main contributions of our study are as follows:

(1) We design a new undersampling method based on the distribution of majority class samples, RMDD, which can reduce information loss within the majority class

(2) We design a novel combination based on heterogeneous ensemble learning and our RMDD resampling method to obtain better prediction performance in highly imbalanced credit card transaction datasets

(3) Experimental results on two real credit card fraud datasets demonstrate that the proposed model can achieve better performance

2. Literature Review

2.1. Credit Card Fraud Detection Model. Credit card datasets contain detailed information about each transaction, such as account number, transaction amount, time, location, and merchant category. We can construct a model to determine whether a transaction is fraudulent or not by expressing the transaction-related information as vectors and calculating their similarity. Singh and Jain [29] reviewed literature on CCFD and summarized the topical issues in current research, such as datasets, evaluation matrices, and the advantages and disadvantages of different models. Armel and Zaidouni [30] compared and analyzed the effectiveness of simple anomaly detection using DT, RF, and Naive Bayes (NB) in CCFD through a series of experiments. Sohony et al. [4] found that RF enables higher accuracy in predicting legal transaction instances and a Feedforward Neural Network (FNN) achieves higher accuracy in predicting instances of fraudulent transactions. Consequently, they proposed an ensemble learning model based on RF and FNN.

Deep learning for CCFD has been discussed in several works [20, 31, 32]. Rushin et al. [20] conducted comparative experiments on deep learning, LR, and Gradient Boosted Tree (GBT) with a dataset containing approximately 80 million account level transactions. The results showed that the performance of deep learning models is better than the GBT and LR. Kim et al. [31] proposed a champion-challenger framework that includes deep learning and ensemble learning and evaluated it on a large transaction dataset taken from a major card issuing company in South Korea. Li et al. [32] proposed a deep representation learning model based on a full center loss function, which considers both distances and angles among different features.

Some studies have made improvements in feature engineering methods for credit card transaction data. Zhang et al. [24] proposed a feature engineering method based on homogeneity-oriented behavior analysis and then used a deep belief network for learning the extracted features. Lucas et al. [33] proposed an HMM-based feature engineering strategy that could incorporate sequential knowledge in the transactions in the form of HMM-based features, which enabled a nonsequential RF classifier to make use of the sequential information. Wu et al. [34] proposed a new feature engineering method to detect fraudulent cash-out of credit cards that considers both snapshot and dynamic behavioral patterns of cardholders and conducted a comparative experiment with the feature extraction method based on Whitrow's strategy. Vlasselaer et al. [35] proposed a feature engineering method based on the network structure of cardholders and merchants and then calculated a time-dependent suspiciousness score for each network object.

Many other approaches have been used recently in the identification of credit card fraud. Gianini et al. [36] proposed a method of rule pool management based on game theory in which the system distributes suspicious transactions for manual investigation while avoiding the need to isolate the individual rules. Based on generative adversarial networks, Fiore et al. [37] proposed a method to generate simulated fraudulent transaction samples to improve the effectiveness of classification models. Carcillo et al. [38] proposed a scalable real-time CCFD framework that could deal with imbalance and feedback latency based on big data tools such as Spark. Their work provides a reference for real-time detection in massive credit card transaction data.

2.2. Imbalanced Data Learning Methods. Imbalanced distribution of data (class imbalance) has a great impact on the performance of classification models, reducing the accuracy of prediction in the minority class [25]. Some effective solutions for class imbalanced data have been proposed by many researchers. These solutions can be arranged into two groups: data level and algorithm level [2].

Resampling is a simple and efficient way to address the problem of class imbalance at the data level. Current resampling strategies can be divided into those that oversample the minority class samples and those that undersample the majority class samples. Commonly used oversampling methods include Random Oversampling (ROS), Synthetic Minority Oversampling Technique (SMOTE) [39], and Borderline-SMOTE [40]. For a highly imbalanced credit card transaction dataset, oversampling generates many minority class samples (fraudulent transactions). Although this can increase the learning weight of the classification model for minority class samples, it also increases computational complexity and generates many noise samples, which will reduce the predictive performance for the majority class (legal transactions). Commonly used undersampling methods include Random Undersampling (RUS), one-sided dynamic undersampling [41], and neighborhood-based undersampling [42]. The undersampling approach involves deleting a large number of majority class samples. This improves the computational efficiency of the classification model but may result in the loss of important information from the majority class samples, which can increase the false-positive rate of the classification model and lead to additional investigation costs for the banks.

Cost-sensitive learning technology is often used to address the problem of imbalanced datasets at the algorithm level. These learning models introduce some constraints and weights through a cost matrix based on the loss function of conventional learning models, which causes models to shift to a smaller total cost. The advantage of cost-sensitive learning technology is that it does not generate or add new information, thereby avoiding the introduction of external noise into the classification model. The disadvantage of cost-sensitive learning technology is that the establishment of the cost matrix needs to be estimated by business experts and cannot be calculated accurately. Commonly used cost-sensitive learning models include cost-sensitive SVM [43], cost-sensitive LR [44], and cost-sensitive DT [18].

Akila and Reddy [45] proposed a cost-sensitive risk-induced Bayesian inference bagging model for CCFD to help card issuers reduce costs. They verified the effectiveness of this model on a dataset from a Brazilian bank. Nami and Shajari [46] proposed a two-stage detection algorithm to address class imbalance in payment card fraud detection. The first stage extracts the relevant features from the transaction data and the second stage extracts the recent transaction behavioral characteristics of cardholders. In the second stage, a cost-sensitive dynamic random forest model is used to improve classification performance.

3. Methodology

In this section, we introduce the proposed heterogeneous ensemble learning model based on data distribution (HELMDD) in details, which consists of two main components. The first is a resampling method based on data distribution (RMDD), as illustrated in Figure 1. RMDD undersamples the majority class based on the data distribution of the majority samples and creates several balanced training subsets by using KNN and *K*-Means. The second one is a framework based on a heterogeneous ensemble learning model (HELM), as illustrated in Figure 2. HELM is a framework that integrates seven kinds of heterogeneous classification models (LR, SVM, NB, DT, RF, AdaBoost, and XGBoost) in the bagging method.

3.1. KNN. KNN is a widely used unsupervised learning method. KNN can predict the category of samples by calculating the Euclidean distance between different points. The formula for calculating the Euclidean distance between points (x_i, y_j) is shown in the following equation:

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - y_{jk}\right)^2} \quad (k = 1, 2, \cdots, n). \tag{1}$$

RMDD divides the majority class samples of the training dataset into a subset of boundary samples and a subset of ordinary samples using a KNN algorithm. A selection of samples is then drawn from each of these subsets to create several new balanced datasets that contain cases from both the majority and minority classes. The advantage of this method is that the new balanced training subsets retain some of the boundary features of the majority class from the original training dataset, which can reduce information loss in these critical boundary cases.

3.2. K-Means. K-Means is a popular unsupervised clustering algorithm. Taking dataset S and the number of classes k as inputs, the K-Means algorithm is aimed at dividing S into k subsets quickly. Specifically, K-Means randomly select k samples as initial clusters. Then, for each sample s in the dataset, the Euclidean distance d_{sj} between sample s and the centroid of k different clusters is calculated. If the distance between s and the centroid of cluster i is the shortest, s is assigned to cluster i. The third step is to calculate the average value of samples in cluster i and update the centroid of cluster i. The second and third steps are repeated until the difference between the old centroid and the new centroid is less than a preset threshold. After the algorithm is executed, we can obtain the data distribution of the majority class samples.

In the task of imbalanced classification, undersampling methods can increase the learning weight of minority class samples, which helps classification models to attain a higher recall rate. RUS is the most widely used method, but a significant defect of RUS is that it discards a large number of samples from the majority class, which may increase the false-positive rate of classification models. During the undersampling process, if we do not consider the distribution of

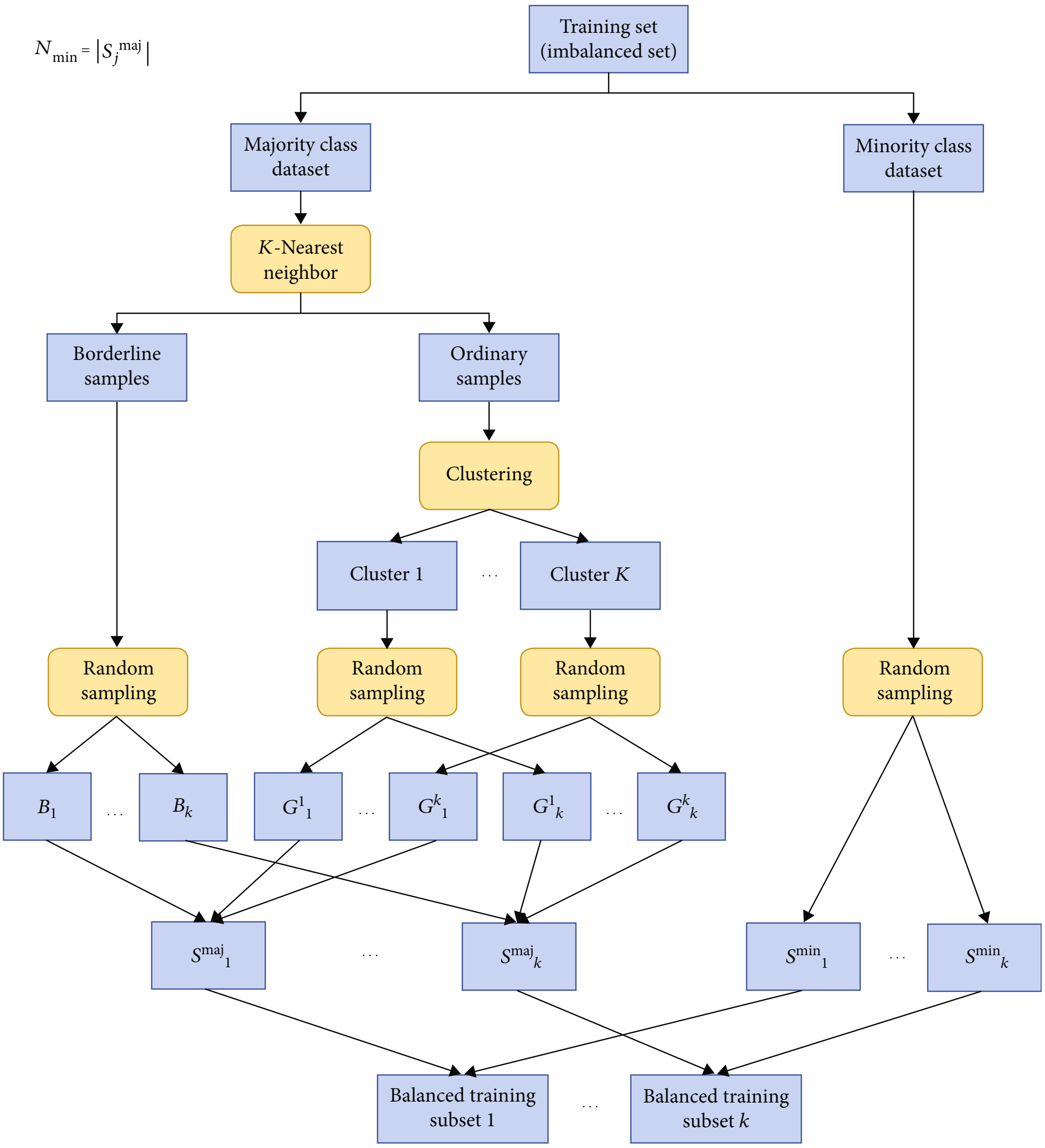

Figure 1: Flowchart of RMDD resampling algorithm, including three classic algorithms: KNN, K-Means, and RUS.

majority class samples, the selected samples cannot represent main features of the majority class and thus could decrease the performance of the base classifiers. In our method, we divide the majority class into k clusters using the K-Means algorithm and then randomly sample from each cluster in different proportions. This resampling method fully considers the distribution of majority class samples and thus can better retain the main features of these cases.

3.3. RMDD Resampling Method. RMDD is an undersampling algorithm that fully considers data distribution, which has three components. The first is to sample the minority class. Due to the highly imbalanced distribution in a CCFD dataset, we use all the minority class samples to improve recognition ability for the minority class of the base classifier. In the second part, we undersample the majority class to generate multiple subsets so that the number of majority class samples is the same as the number of minority class samples, which forms the core of the RMDD algorithm. The third part is to generate several balanced subsets to provide training data for the base classifiers by merging the minority class samples and the subset of majority class samples. The flowchart of the RMDD resampling algorithm is shown in Figure 1.

The second part above consists of the following 4 steps:

(1) We divide the majority class samples into a boundary sample set B and an ordinary sample set G by the KNN algorithm. Then, for any sample x in the majority class (labeled 0), we find K-nearest neighbor samples to x from the training set. If there are more than $k/2$ neighboring samples with a label of 1, then $B = B \cup \{x\}$, otherwise $G = G \cup \{x\}$

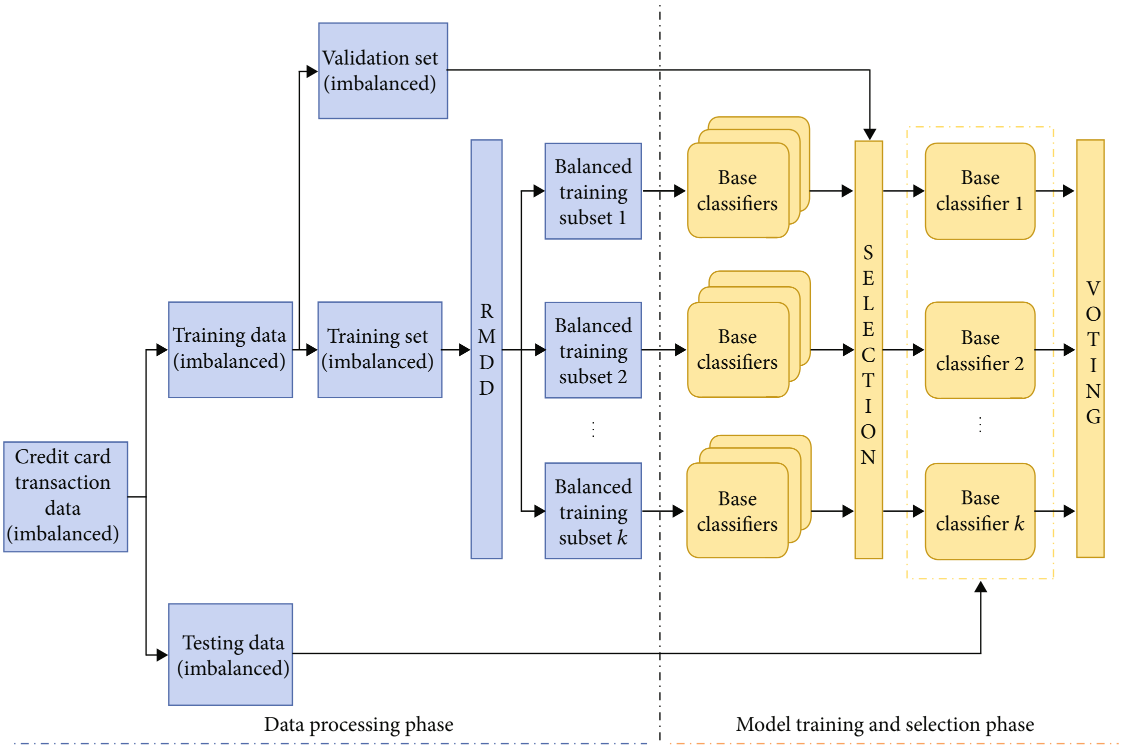
Validation set (imbalanced)
Credit card transaction data (imbalanced)
Training data (imbalanced)
Training set (imbalanced)
Testing data (imbalanced)
RMDD
Balanced training subset 1
Balanced training subset 2
Balanced training subset k
Base classifiers
SELECTION
Base classifier 1
Base classifier 2
Base classifier k
VOTING
Data processing phase
Model training and selection phase

Figure 2: Framework of HELM. Preprocessing original data by RMDD resampling algorithm and selecting best classifiers from different base classifiers by AUC score.

(2) Random sampling with replacement is used to divide sample B into k subsets. N_b represents the number of samples in set B. The formula for calculating the number of samples in each subset is shown as

$$N_{bi} = \lambda_i N_b, \quad i \in [1, 2, \cdots, k], \lambda_i \in [0, 1], \tag{2}$$

$$\sum_{i=1}^{k} \lambda_i = 1. \tag{3}$$

λ_i is a weight parameter, which is used to adjust the number of boundary samples in each subset

(3) Using the K-Means algorithm, we divide ordinary sample set G into k clusters and sample with replacement from each cluster with different sampling rate. The sampling rate in the ith cluster is calculated from Equation (4), where N_i represents the number of samples in the ith cluster, and N_g represents the number of ordinary samples in set G. $N_{\min}$ represents the number of minority samples, and N_{gi} represents the number of samples randomly selected from the ith cluster and can be calculated from Equation (5).

$$\mathrm{SR}_i = \frac{N_i}{N_g}, \quad i \in [1, 2, \cdots, k], \tag{4}$$

$$N_{gi} = (N_{\min} - N_{bi})\mathrm{SR}_i \tag{5}$$

(4) Combining boundary sample subset B_j with ordinary sample subset G_j to construct a subset S_j^{maj} of majority class samples. The corresponding calculation formula is as follows:

$$S_j^{\mathrm{maj}} = \left\{B_j, G_j^1, G_j^2, \cdots, G_j^k\right\}, \quad j \in [1, 2, \cdots, k]. \tag{6}$$

The number of samples in the majority class is equal to the number in the minority class, that is, $N_{\min} = |S_j^{\mathrm{maj}}|$

3.4. Framework of HELM. To improve classification performance, we propose a heterogeneous ensemble learning model (HELM) framework, as shown in Figure 2. The HELM framework uses data resampling and ensemble learning technologies to address the problem of imbalanced data in CCFD. Through the training and screening of multiple heterogeneous base classifiers, we improve the robustness of HELM as well as avoid reliance on a single classifier. The HELM framework consists of two phases: (1) the data processing phase and (2) the model training and selection phase.

3.4.1. Data Processing Phase. The main task of this phase is to preprocess the original credit card transaction dataset, including feature selection, data normalization, dataset division, and resampling. First, we divide the original dataset into a training set and a test set. The training set is used for estimating parameters of the classification model, and the test set is used for evaluating the trained classification model. We further divide the training set into a training subset and a validation subset and use the RMDD algorithm to resample the training subset, which divides the highly imbalanced training subset into k balanced subsets for training base classifiers. The RMDD algorithm fully considers the distribution of majority class samples. Boundary samples are distinguished from ordinary samples by applying the KNN algorithm. Ordinary samples are grouped into k classes by the K-Means algorithm. Through these two algorithms, we can build several balanced subsets and ensure that each balanced training subset contains a certain ratio of boundary samples and ordinary samples from each cluster. The advantage of this is that more feature information of samples in the majority class can be preserved while generating new balanced training subsets. In addition, the introduction of boundary samples retains some of boundary contours from the original dataset in the new balanced subset, which can help to improve classification performance.

3.4.2. Model Training and Selection Phase. When the preprocessing phase is completed, we have k balanced training subsets. For each subset, we use seven different base classifiers for training, including LR, SVM, NB, DT, RF, AdaBoost, and XGBoost. Then, we use an imbalanced validation subset to obtain Area Under the Curve (AUC) score for each base classifier and select the base classifier with the best AUC score as recommended classifier for that subset. Finally, we obtain an ensemble learning model with heterogeneous or isomorphic recommended classifiers that are trained with other subsets. For samples in the test dataset, each recommended classifier will give an initial prediction; then, the final prediction result is generated through a voting method across each recommended classifier. In the model selection phase, we use AUC score as the selection condition because AUC score takes into account both the prediction accuracies of the majority and minority classes at the same time, which gives us a good compromise between the accuracy and recall metrics for the classification model. In credit card fraud prediction, misclassification of legitimate transactions as fraudulent transactions or mis classification of fraudulent transactions as legitimate transactions willincur costs for banks and customers, such as loss of transaction amount, manual investigation costs, etc. Therefore, by comparing the AUC score of multiple base classifiers and selecting the base classifier with the best AUC score to build an ensemble model, we can effectively improve prediction performance and reduce economic losses for cardholders and banks.

The HELM framework can be deployed in a distributed manner. Base classifier training tasks on different subsets in HELM can be assigned to different cluster nodes. Each node can perform model training in parallel during periods of low credit card transactions (such as the early morning). Since the proportion of fraudulent transactions is very low, the balanced training subset space generated by the RMDD algorithm is quite small, which can significantly reduce the training time of base classifiers. Compared with other traditional methods, our HELM framework can significantly reduce model training and deployment time by reducing the training sample space and facilitating the application of parallel computing technologies.

4. Experiments

4.1. Dataset Description. In this paper, we use two real credit card transaction datasets: one from Kaggle (public dataset) and one from a bank in China (our private dataset). The detailed statistics are shown in Table 1.

(1) Kaggle dataset [47]. This dataset is composed of credit card transaction records of European cardholders in September 2013. The time span of these transactions is two days, and each transaction record contains 30 features. Due to privacy considerations, 28 features were encoded by Principal Component Analysis (PCA), except for two features: transaction time and amount. This dataset contains a total of 284,807 instances, of which 492 are minority class samples (fraudulent transactions). The fraud rate of this dataset is 0.173%, which indicates that the dataset is highly imbalanced

(2) Our private dataset. This dataset is provided by a bank in China and contains credit card transaction records of customers on a typical day in May 2017. Each instance has 23 features, including some personal information of cardholders (such as age, gender, marital status, and education level) and transaction-related features (such as transaction amount, time, and merchant number). This dataset contains 24,024 instances, including 660 fraud instances. The fraud rate of this dataset is 2.747%.

4.2. Performance Measures. Confusion matrix provides helpful information regarding the actual labels and predicted labels proposed by the classification model. The confusion matrix used in this study is shown in Table 2. Due to the highly imbalanced phenomenon of our credit card datasets, widely used evaluation indexes (such as accuracy and precision) do not fully represent the performance of classification models. For example, if we classify all samples in the Kaggle dataset as legitimate transactions, the accuracy will be close to 98%; it is clear that this prediction model is not a good classification model. Therefore, we choose Fra_Recall (fraud class recall), Leg_Recall (legal class recall), G-mean, AUC, and savings rate [48] to evaluate the model.

Fra_Recall and Leg_Recall are calculated by Equations (7) and (8), respectively. The larger the Fra_Recall value, the higher the proportion of fraudulent transactions that are identified by the classification model, and the more fraud losses that can be avoided for banks and cardholders. The larger the Leg_Recall value, the higher the proportion of legitimate transactions that is identified by the classification model, and the greater the investigation costs that can be saved for banks. The ideal model is that Fra_Recall and Leg_Recall are close to 1 at the same time. G-mean and AUC are very important measures that are widely used in

Table 1: Dataset description.

Dataset	Instances	Features	Fraud instances (minority class)	Legal instances (majority class)	Fraud ratio
Kaggle dataset	284807	30	492	284315	0.173%
Our private dataset	24024	23	660	23364	2.747%

Table 2: Confusion matrix.

	Actual fraud (positive)	Actual legal (negative)
Predicted fraud (positive)	True positive (TP)	False positive (FP)
Predicted legal (negative)	False negative (FN)	True negative (TN)

model evaluation studies in the presence of imbalanced data. The larger the *G*-mean and AUC value, the better the performance of the classification model. *G*-mean and AUC can be calculated by Equations (9) and (10).

$$\text{Fra\_Recall} = \text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{7}$$

$$\text{Leg\_Recall} = \text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \tag{8}$$

$$\text{G-mean} = \sqrt{\text{sensitivity} \times \text{specificity}}, \tag{9}$$

$$\begin{aligned} \text{AUC} = 1 - \frac{1}{|D^+| \cdot |D^-|} \sum_{x^+ \in D^+} \sum_{x^- \in D^-} \\ \cdot \left(\mathbf{I}(f(x^+) < f(x^-)) + \frac{1}{2} \mathbf{I}(f(x^+) = f(x^-)) \right). \end{aligned} \tag{10}$$

D^+ and D^- denote the collection of fraudulent transactions and legitimate transactions, respectively.

The savings rate is an indicator that banks attach great importance to, because it is always used to quantify the economic benefits that fraud detection models can create for banks. The CCFD cost matrix [48] is shown in Table 3. Among them, y_i is the actual label for transaction *i*, and y_i' is the predicted label for transaction *i* given by classifier *f*. If a transaction is predicted to be a fraudulent transaction (TP or FP), the bank needs to investigate the transaction incurring a cost of C_a. Conversely, if the transaction is predicted to be legitimate (TN or FN), there is no investigation cost, but in the case of FN, the loss of the bank is equal to the transaction amount Amt_i. If no classifier is used for CCFD, the total loss of the bank is calculated by Equation (11). The proportion of cost saved for the bank by using classifier *f* is calculated by Equation (12).

$$C_{\text{total}} = \sum_{i=1}^{n} y_i \text{Amt}_i, \tag{11}$$

$$\text{Savings}(f) = \frac{\left(\sum_{i=1}^{n} y_i y_i{}' \text{Amt}_i - y_i{}' C_a \right)}{C_{\text{total}}} \tag{12}$$

4.3. Experimental Design. To evaluate the effectiveness of the HELMDD model, we conducted experiments on two real credit card datasets and compared the proposed model with several competing approaches. Most of models can be divided into two categories: independent model and ensemble learning model. The independent model we used in our experiment includes LR, SVM, NB, and DT. The ensemble learning model we used in our experiment includes RF, AdaBoost, and XGBoost. In addition, we also combined these models with different resampling methods, such as SMOTE and RUS.

5. Experimental Results and Discussion

To directly compare with previous works, we evaluate our model using 10-fold cross-validation similar to prior approaches on the two datasets. The experimental results for each classification model on the Kaggle dataset and our private dataset are shown in Tables 4 and 5, respectively. For convenience of comparison, we have also presented the data of Tables 4 and 5 in histogram form, as shown in Figures 3 and 4. In Tables 4 and 5, numbers in italic indicate the best values of the model in the corresponding evaluation measure.

For the Kaggle dataset, we compare the proposed model with several competing approaches and show the results in Table 4. From the results, we can observe the following:

(1) In the case of the same classification model, those implementing resampling methods to preprocess the training subset achieved better performance than models with the original imbalanced training subset. Fra_Recall, AUC, and G-mean have different degrees of improvement. For example, Fra_ Recall increased from 0.0235 (DT model with SMOTE method) to 0.2353 (LR model with RUS method), AUC increased from 0.0023 (AdaBoost model with SMOTE method) to 0.0275 (NB model with RUS method), and G-mean increased from 0.0018 (DT model with RUS method) to 0.1239 (LR model with RUS). The main reason is that preprocessing the original imbalanced dataset with SMOTE or RUS method helps models to improve the learning rate of fraudulent transaction instances and therefore enhances the ability to identify fraudulent transactions

Table 3: Credit card fraud cost matrix.

	Actual fraud (positive) $y_i = 1$	Actual legal (negative) $y_i = 0$
Predicted fraud (positive) $y_i' = 1$	$C_{\mathrm{TP}_i} = C_a$	$C_{\mathrm{FP}_i} = C_a$
Predicted legal (negative) $y_i' = 0$	$C_{\mathrm{FN}_i} = \mathrm{Amt}_i$	$C_{\mathrm{TN}_i} = 0$

Table 4: Results on the Kaggle dataset.

Model	Rebalancing technique	Fra_Recall	Leg_Recall	AUC	*G*-mean	Savings
LR	Imbalanced	0.6176	0.9942	0.9500	0.7836	0.5980
	SMOTE	0.8412	0.9755	0.9571	0.9058	0.7591
	RUS	0.8529	0.9657	0.9630	0.9075	0.7399
SVM	Imbalanced	0.8059	0.9963	0.9559	0.8961	0.7893
	SMOTE	0.8824	0.9700	0.9678	0.9252	0.7829
	RUS	0.8529	0.9605	0.9697	0.9051	0.7237
NB	Imbalanced	0.7001	0.9996	0.9413	0.8365	0.6943
	SMOTE	0.8012	0.9920	0.9539	0.8908	0.7700
	RUS	0.8059	0.9783	0.9688	0.8879	0.7329
DT	Imbalanced	0.7647	0.9996	0.8821	0.8743	0.7585
	SMOTE	0.7882	0.9965	0.8924	0.8863	0.7723
	RUS	0.8471	0.9062	0.9061	0.8761	0.5477
RF	Imbalanced	0.7588	*0.9999*	0.9614	0.8711	0.7538
	SMOTE	0.8235	0.9998	0.9709	0.9074	0.8178
	RUS	0.8588	0.9703	0.9719	0.9129	0.7604
AdaBoost	Imbalanced	0.7765	0.9998	0.9601	0.8811	0.7708
	SMOTE	0.8706	0.9850	0.9624	0.9260	0.8182
	RUS	0.8824	0.9473	0.9694	0.9143	0.7117
XGBoost	Imbalanced	0.7706	*0.9999*	0.9649	0.8778	0.7656
	SMOTE	0.8765	0.9885	0.9700	0.9308	0.8194
	RUS	*0.9001*	0.9663	0.9725	0.9325	0.7886
HELMDD	RMDD	0.8882	0.9903	*0.9853*	*0.9379*	*0.8623*

(2) In the case of applying the same resampling method, AUC obtained by ensemble learning models are generally better than those from independent learning models. As shown in Table 4, the highest AUC obtained by independent learning models with three different resampling methods are 0.9559 (imbalanced data), 0.9678 (SMOTE method), and 0.9697 (RUS method), while the average AUC obtained by ensemble learning models with three different sampling methods are 0.9621 (imbalanced data), 0.9678 (SMOTE method), and 0.9713 (RUS method), so the ensemble learning models are slightly better than the independent learning models. This is because ensemble learning models are strengthened by using multiple weak classification models. Compared with independent models, ensemble models can obtain a smaller deviation and better generalization ability

(3) For the same classification model, those using RUS to preprocess the training dataset achieve better Fra_Recall and AUC than models based on SMOTE. For example, Fra_Recall increases from 0.0047 (NB model) to 0.0589 (DT model), and AUC increases from 0.0010 (RF model) to 0.0149 (NB model). However, we cannot ignore that Leg_Recall decreases by 0.0095 (SVM model) to 0.0903 (DT model). This is because the RUS method discards many legitimate transaction samples and leads to an improvement in the identification of fraudulent transactions while increasing the false prediction rate for legitimate transactions

Table 5: Results on our private dataset.

Model	Rebalancing technique	Fra_Recall	Leg_Recall	AUC	G-mean	Savings
LR	Imbalanced	0.5928	0.8145	0.7334	0.6949	0.5414
	SMOTE	0.5825	0.7431	0.7058	0.6579	0.5130
	RUS	0.6959	0.6483	0.7250	0.6717	0.6014
SVM	Imbalanced	0.5155	0.7679	0.6567	0.6291	0.4527
	SMOTE	0.6959	0.6068	0.6881	0.6498	0.5909
	RUS	0.5773	0.8751	0.7509	0.7108	0.5415
NB	Imbalanced	0.3505	0.9428	0.7076	0.5749	0.3335
	SMOTE	0.5722	0.8055	0.7245	0.6789	0.5186
	RUS	0.5722	0.8038	0.7274	0.6782	0.5182
DT	Imbalanced	0.0876	*0.9659*	0.5268	0.2909	0.0783
	SMOTE	0.1598	0.9072	0.5335	0.3807	0.1350
	RUS	0.6443	0.6017	0.6230	0.6226	0.5384
RF	Imbalanced	0.5773	0.7958	0.7252	0.6778	0.5213
	SMOTE	0.5412	0.7126	0.6564	0.6210	0.4642
	RUS	0.6186	0.7861	0.7528	0.6973	0.5598
AdaBoost	Imbalanced	0.6186	0.8047	0.7465	0.7055	0.5645
	SMOTE	0.6031	0.7360	0.7101	0.6662	0.5316
	RUS	0.6392	0.7531	0.7581	0.6938	0.5718
XGBoost	Imbalanced	0.6649	0.7387	0.7543	0.7008	0.5937
	SMOTE	0.6082	0.7398	0.7014	0.6708	0.5377
	RUS	0.6753	0.7693	0.7624	0.7208	0.6118
HELMDD	RMDD	*0.7213*	0.7985	*0.7941*	*0.7589*	*0.6696*

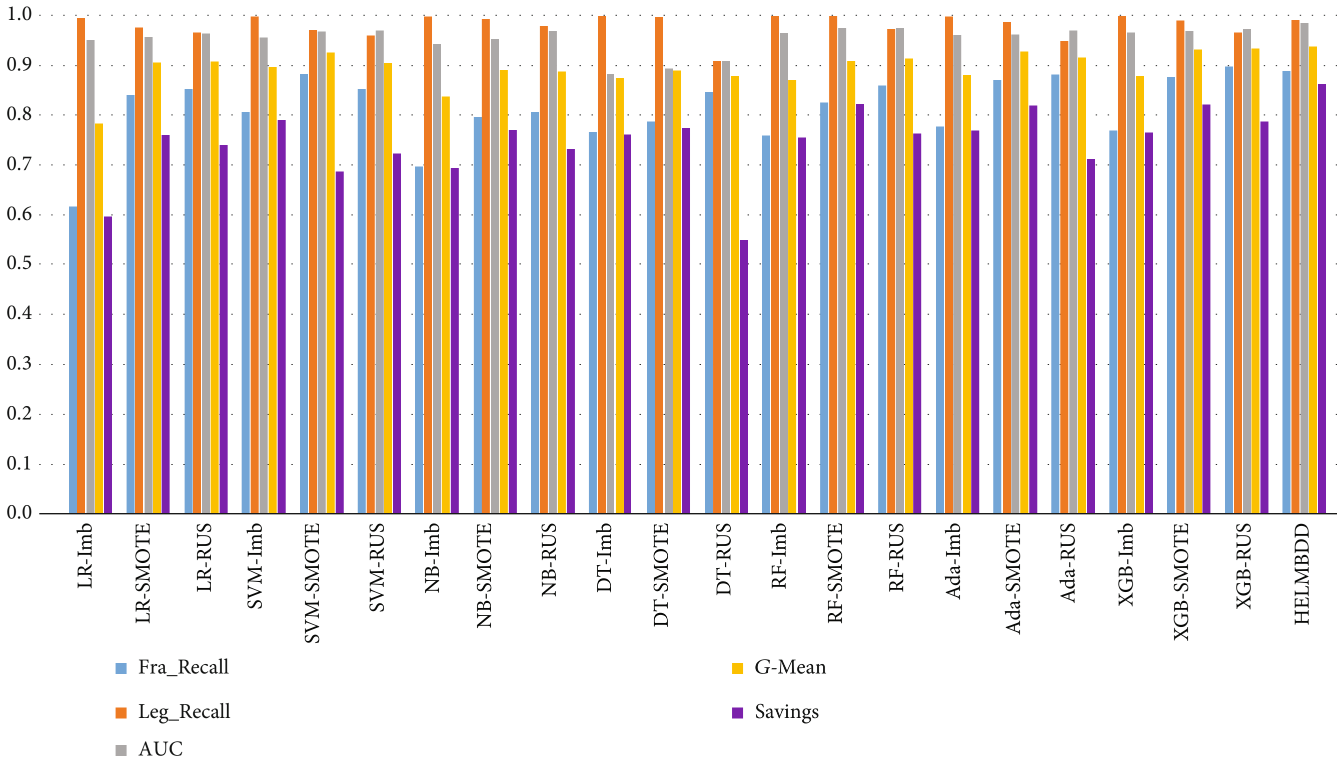

Figure 3: Histogram for the mean of model evaluation measures on the Kaggle dataset.

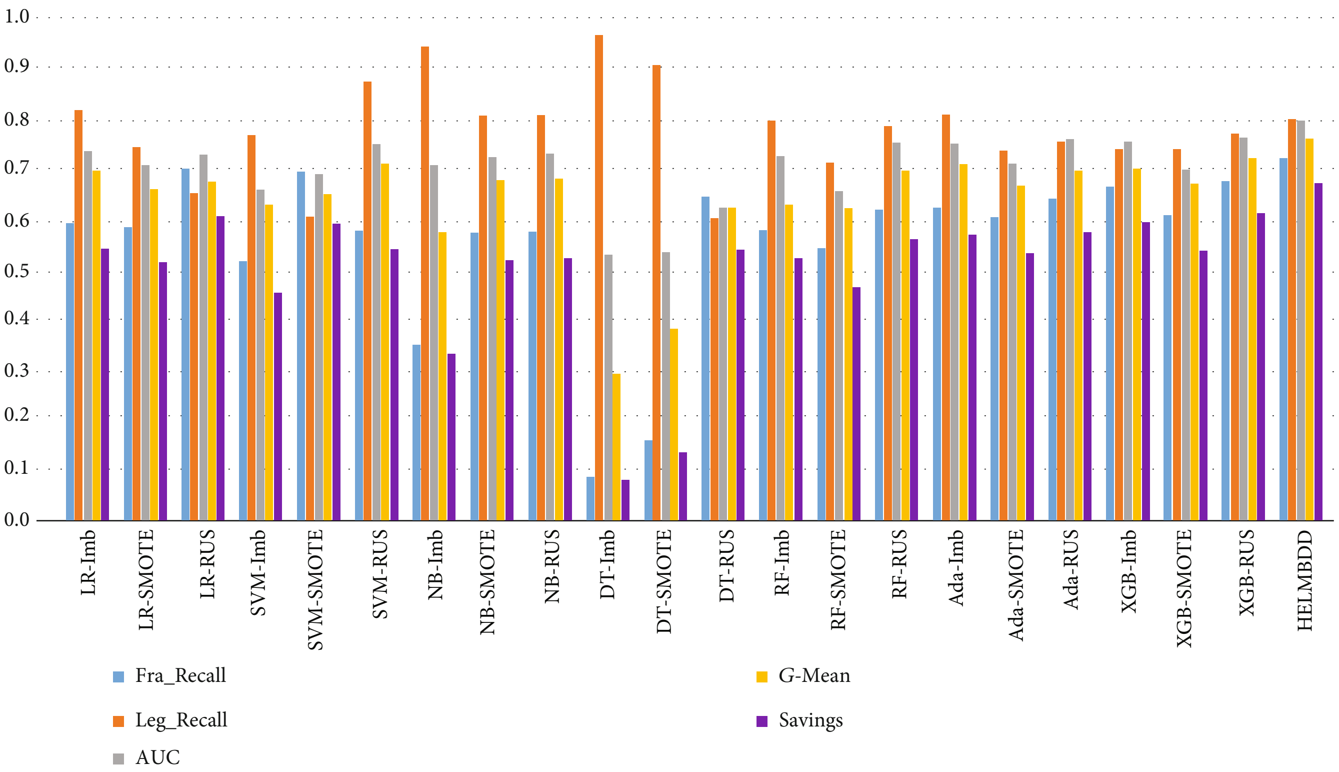

Figure 4: Histogram for the mean of model evaluation measures for our private dataset.

(4) In terms of savings score, we have two findings: first, after resampling the original imbalanced training data with SMOTE method, the savings score of six classification models has been improved to varying degrees. For example, the XGBoost model has been increased from 0.7656 to 0.8194 and the LR model has been upgraded from 0.5980 to 0.7591. Second, when we use the RUS method to resample original data, the savings score of three classification models (SVM, DT, and AdaBoost) has been reduced by different degrees, such as the SVM model which has been reduced from 0.7893 to 0.7237, while the savings score of the other four classification models have been improved by different degrees, such as the XGBoost model which has been increased from 0.7656 to 0.7886. There may be two reasons for this: first, the savings score is highly correlated with the recognition rate of fraudulent transactions; SMOTE method can help classification models increase the recall rate of fraudulent transactions and reduce fraud losses of banks. Second, the RUS method discards many legitimate transaction samples, although it strengthens the learning of fraud samples and improves the recall rate of fraud transaction for the model, but it also leads to an increase in the false prediction rate of legitimate transactions and increases investigation cost for banks

(5) Our HELMDD model proposed in this paper achieved the best AUC, *G*-mean, and savings scores, which were 0.0128, 0.0054, and 0.0429 higher than previous state-of-the-art methods, respectively. The model showed good stability. While obtaining the second highest Fra_Recall, it did not significantly reduce Leg_Recall, thus ensuring that banks can achieve greater savings. The overall performance of HELMDD is better than the ensemble learning models (such as XGBoost, AdaBoost, and RF) with different resampling methods. This is because the RMDD resampling algorithm fully considers the distribution of legitimate transaction samples. Samples extracted from the boundary subset and multiple clusters fully retain the diversity and boundary contours of legitimate transaction samples. In addition, the selection mechanism of base classification models also helps to improve the overall performance of the framework

Table 5 presents the performance comparison between our approach and other competitive methods on our private dataset. From the results, we can observe the following:

(1) Using the SMOTE method to resample the training dataset may not necessarily improve performance of the classification models and may even lead to a deterioration in classification performance. In Table 5, AUC and *G*-mean obtained by SVM, NB, and DT combined with SMOTE have been improved to varying degrees. For example, AUC increases from 0.0067 (DT model) to 0.0314 (SVM model), and *G*-mean

Table 6: Ablations on two datasets with different resampling and model ensemble methods.

Method	Kaggle dataset		Our private dataset	
	AUC	Savings	AUC	Savings
HELMDD	0.9853	0.8623	0.7941	0.6696
-RMDD	0.9764	0.8255	0.7737	0.6324
-HELM	0.9788	0.8472	0.7853	0.6582
Base	0.9725	0.7886	0.7624	0.6118

increases from 0.0207 (SVM model) to 0.1040 (NB model). However, the overall performance of LR, RF, AdaBoost, and XGBoost combined with SMOTE decreases to varying degrees, in which AUC decreases by 0.0276 (LR model) to 0.0688 (RF model), and G-mean decreases from 0.0300 (XGBoost model) to 0.0568 (RF model). This is possibly because through the SMOTE method, a large amount of minority sample noise is generated in the process of resampling for the training dataset, which decreases the performance of some classification models

(2) Using RUS to resample the training dataset, apart from AUC of the LR model that dropped from 0.7334 to 0.7250, AUC obtained by the other six classification models manifests an improvement of 0.0081 (XGBoost model) to 0.0962 (DT model). This may be because the absence of new minority samples generated in the RUS process avoids the introduction of noise samples and improves the performance of the classification models

(3) The HELMDD model proposed in this article achieved the best Fra_Recall, AUC, G-mean, and savings scores, which were 0.0254, 0.0317, 0.0381, and 0.0578 higher than the corresponding measures of the previous state-of-the-art models, respectively. The validity and stability of HELMDD model were thus verified again

6. Ablation Study

We conduct an ablation study to investigate the effectiveness of our model components.

Table 6 shows the effects of the different resampling and model ensemble methods on AUC and savings scores. Here, -RMDD denotes using RUS instead of the RMDD resampling technique in HELMDD, and -HELM denotes using XGBoost instead of the seven heterogeneous models and ensemble in HELMDD. Base is the model generated by performing the above two ablations, which is the default XGBoost model with RUS resampling. For Kaggle and our private dataset, we observe that both RMDD and HELM are beneficial for identifying fraudulent transactions and controlling the cost of investigating fraudulent transactions. The reason is that two model components can significantly improve the recognition rate of fraudulent transactions without reducing the recognition rate of legitimate transactions.

7. Conclusions

In this paper, we propose a heterogeneous ensemble learning model based on data distribution (HELMDD) for the problem of the highly imbalanced data distribution encountered in CCFD. In our HELMDD model, we first propose an undersampling method, RMDD, based on the distribution of the majority class. RMDD divides the majority class into boundary samples and ordinary samples and then generates multiple balanced subsets based on the idea of clustering to train multiple base classifiers. The RMDD algorithm can maintain the classification boundary contours of the majority class and reduce the loss of sample information. Therefore, our model can obtain a higher majority class recall rate while also improving the minority class recall rate. In terms of model selection, we chose base classifiers that obtain the best AUC score in the balanced subset to generate an ensemble model, which helped to improve classification performance. Finally, we evaluate the proposed method on the Kaggle dataset and our private dataset. The results show that HELMDD achieves new state-of-the-art performance compared to other competing approaches.

Data Availability

The labeled dataset used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Nos. 61732022, 61732004, 61672020, and 62072131) and the National Key R&D Program of China (Nos. 2017YFB0802204, 2019QY1406, and 2017YFB0803303).

References

[1] L. Zheng, G. Liu, C. Yan, and C. Jiang, “Transaction fraud detection based on total order relation and behavior diversity,” *IEEE Transactions on Computational Social Systems*, vol. 5, no. 3, pp. 796–806, 2018.

[2] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, “An experimental study with imbalanced classification approaches for credit card fraud detection,” *IEEE Access*, vol. 7, pp. 93010–93022, 2019.

[3] C. V. Priscilla and D. P. Prabha, “Influence of optimizing XGBoost to handle class imbalance in credit card fraud detection,” in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 1309–1315, Tirunelveli, TN, India, August 2020.

[4] I. Sohony, R. Pratap, and U. Nambiar, “Ensemble learning for credit card fraud detection,” in *2018 Proceedings of the ACM India Joint International Conference on Data Science and Management of Data (CoDS-COMAD)*, pp. 289–294, Panaji, Goa, India, January 2018.

[5] 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.

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

[7] N. Carneiro, G. Figueira, and M. Costa, “A data mining based system for credit-card fraud detection in e-tail,” *Decision Support Systems*, vol. 95, pp. 91–101, 2017.

[8] 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.

[9] S. Carta, G. Fenu, D. R. Recupero, and R. Saia, “Fraud detection for E-commerce transactions by employing a prudential multiple consensus model,” *Journal of Information Security and Applications*, vol. 46, pp. 13–22, 2019.

[10] P. K. Chan and S. J. Stolfo, “Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection,” in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 164–168, New York, NY, USA, August 1998.

[11] S. J. Omar, K. Fred, and K. K. Swaib, “A state-of-the-art review of machine learning techniques for fraud detection research,” in *Proceedings of the 2018 International Conference on Software Engineering in Africa (ICSE)*, pp. 11–19, Gothenburg, GOT, Sweden, May 2018.

[12] F. Carcillo, Y.-A. L. Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, “Combining unsupervised and supervised learning in credit card fraud detection,” *Information Sciences*, vol. 557, pp. 317–331, 2021.

[13] V. Bhusari and S. Patil, “Study of hidden Markov model in credit card fraudulent detection,” in *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*, pp. 1–4, Coimbatore, India, February 2016.

[14] S. Jha, M. Guillen, and J. C. Westland, “Employing transaction aggregation strategy to detect credit card fraud,” *Expert Systems with Applications*, vol. 39, no. 16, pp. 12650–12657, 2012.

[15] N. Rtayli and N. Enneya, “Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization,” *Journal of Information Security and Applications*, vol. 55, article 102596, 2020.

[16] H. Wang, P. Zhu, X. Zou, and S. Qin, “An ensemble learning framework for credit card fraud detection based on training set partitioning and clustering,” in *2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBD)*, pp. 94–98, Guangzhou, GZ, China, October 2018.

[17] S. Panigrahi, A. Kundu, S. Sural, and A. K. Majumdar, “Credit card fraud detection: a fusion approach using Dempster-Shafer theory and Bayesian learning,” *Information Fusion*, vol. 10, no. 4, pp. 354–363, 2009.

[18] 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.

[19] S. Bagga, A. Goyal, N. Gupta, and A. Goyal, “Credit card fraud detection using pipeling and ensemble learning,” *Procedia Computer Science*, vol. 173, pp. 104–112, 2020.

[20] G. Rushin, C. Stancil, M. Sun, S. Adams, and P. Beling, “Horse race analysis in credit card fraud—deep learning, logistic regression, and gradient boosted tree,” in *2017 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 117–121, Charlottesville, VA, USA, April 2017.

[21] N. S. Halvaiee and M. K. Akbari, “A novel model for credit card fraud detection using artificial immune systems,” *Applied Soft Computing*, vol. 24, pp. 40–49, 2014.

[22] J. T. S. Quah and M. Sriganesh, “Real-time credit card fraud detection using computational intelligence,” *Expert Systems with Applications*, vol. 35, no. 4, pp. 1721–1732, 2008.

[23] F. Carcillo, Y.-A. L. Borgne, O. Caelen, and G. Bontempi, “An assessment of streaming active learning strategies for real-life credit card fraud detection,” in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 631–639, Tokyo, TKY, Japan, October 2017.

[24] X. Zhang, Y. Han, W. Xu, and Q. Wang, “HOBA: a novel feature engineering methodology for credit card fraud detection with a deep learning architecture,” *Information Sciences*, vol. 557, pp. 302–316, 2021.

[25] P. Cao, X. Liu, J. Zhang, D. Zhao, M. Huang, and O. Zaiane, “$\ell_{2,1}$ norm regularized multi-kernel based joint nonlinear feature selection and over-sampling for imbalanced data classification,” *Neurocomputing*, vol. 234, pp. 38–57, 2017.

[26] F. Zhang, G. Liu, Z. Li, C. Yan, and C. Jiang, “GMM-based undersampling and its application for credit card fraud detection,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Budapest, BP, Hungary, July 2019.

[27] H. He, W. Zhang, and S. Zhang, “A novel ensemble method for credit scoring: adaption of different imbalance ratios,” *Expert Systems with Applications*, vol. 98, pp. 105–117, 2018.

[28] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, “Data imbalance in classification: experimental evaluation,” *Information Sciences*, vol. 513, pp. 429–441, 2020.

[29] A. Singh and A. Jain, “An empirical study of AML approach for credit card fraud detection-financial transactions,” *International Journal of Computers Communications & Control*, vol. 14, no. 6, pp. 670–690, 2020.

[30] A. Armel and D. Zaidouni, “Fraud detection using apache spark,” in *2019 5th International Conference on Optimization and Applications (ICOA)*, pp. 1–6, Kenitra, Morocco, April 2019.

[31] E. Kim, J. Lee, H. Shin et al., “Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning,” *Expert Systems with Applications*, vol. 128, pp. 214–224, 2019.

[32] Z. Li, G. Liu, and C. Jiang, “Deep representation learning with full center loss for credit card fraud detection,” *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 569–579, 2020.

[33] Y. Lucas, P.-E. Portier, L. Laporte et al., “Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs,” *Future Generation Computer Systems*, vol. 102, pp. 393–402, 2020.

[34] Y. Wu, Y. Xu, and J. Li, “Feature construction for fraudulent credit card cash-out detection,” *Decision Support Systems*, vol. 127, article 113155, 2019.

[35] V. V. Vlasselaer, C. Bravo, O. Caelen et al., “APATE: a novel approach for automated credit card transaction fraud detection using network-based extensions,” *Decision Support Systems*, vol. 75, pp. 38–48, 2015.

[36] G. Gianini, L. G. Fossi, C. Mio, O. Caelen, L. Brunie, and E. Damiani, “Managing a pool of rules for credit card fraud detection by a game theory based approach,” *Future Generation Computer Systems*, vol. 102, pp. 549–561, 2020.

[37] U. Fiore, A. D. Santis, F. Perla, P. Zanetti, and F. Palmieri, “Using generative adversarial networks for improving classification effectiveness in credit card fraud detection,” *Information Sciences*, vol. 479, pp. 448–455, 2019.

[38] F. Carcillo, A. D. Pozzolo, Y.-A. L. Borgne, O. Caelen, Y. Mazzer, and G. Bontempi, “*SCARFF*: a scalable framework for streaming credit card fraud detection with spark,” *Information Fusion*, vol. 41, pp. 182–194, 2018.

[39] 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, no. 1, pp. 321–357, 2002.

[40] H. Han, W.-Y. Wang, and B.-H. Mao, “Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning,” in *Advances in Intelligent Computing, International Conference on Intelligent Computing (ICIC)*, pp. 878–887, Hefei, China, August 2005.

[41] Q. Fan, Z. Wang, and D. Gao, “One-sided dynamic undersampling no-propagation neural networks for imbalance problem,” *Engineering Applications of Artificial Intelligence*, vol. 53, pp. 62–73, 2016.

[42] P. Vuttipittayamongkol and E. Elyan, “Neighbourhood-based undersampling approach for handling imbalanced and overlapped data,” *Information Sciences*, vol. 509, pp. 47–70, 2020.

[43] A. Iranmehr, H. Masnadi-Shirazi, and N. Vasconcelos, “Cost-sensitive support vector machines,” *Neurocomputing*, vol. 343, pp. 50–64, 2019.

[44] A. C. Bahnsen, D. Aouada, and B. Ottersten, “Example-dependent cost-sensitive logistic regression for credit scoring,” in *13th International Conference on Machine Learning and Applications (ICMLA)*, pp. 263–269, Detroit, MI, USA, December 2014.

[45] S. Akila and U. S. Reddy, “Cost-sensitive risk induced Bayesian inference bagging (RIBIB) for credit card fraud detection,” *Journal of Computational Science*, vol. 27, pp. 247–254, 2018.

[46] S. Nami and M. Shajari, “Cost-sensitive payment card fraud detection based on dynamic random forest and k -nearest neighbors,” *Expert Systems with Applications*, vol. 110, pp. 381–392, 2018.

[47] Kaggle, “Kaggle,” November 2020, https://www.kaggle.com/mlg-ulb/creditcardfraud.

[48] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, “Feature engineering strategies for credit card fraud detection,” *Expert Systems with Applications*, vol. 51, pp. 134–142, 2016.