Improving SMOTE Technology for Credit Card Fraud Detection Category Imbalance Issues

Ke Zhou, Chunna Zhang*, Yang Yu, Shengqiang Cong, Xiaoping Yue

Abstract—**Credit cards play an important role in today's economy, but they also provide fraud conditions for outlaws. Often, the data for fraud detection is extremely imbalanced, which seriously affects the detection effect of classification models. The KCSMOTE (Kmeans Center Synthetic Minority Oversampling Technique) model is proposed to address the problem of imbalance in credit card fraud data affecting the effectiveness of model detection. The K-means algorithm is used to cluster the samples to find safe clusters with different sparsity, and then K-means++ is used to find a few class centroids of the safe clusters, using the centroid as the base points to improve the SMOTE algorithm. XGBoost and Random Forest algorithms were used to validate the effectiveness of the KCSMOTE model. ADASYN, k-means-SMOTE, borderline-SMOTE, SMOTETomek, SMOTEEnn, and SMOTEWB as well as the original data were selected for comparison experiments, and several metrics, F1-score, Precision, Recall, and AUC (area under the curve), were chosen to determine the results. Experimental results show that the KCSMOTE model is more effective in dealing with unbalanced fraud data than other sampling algorithms.**

Index Terms—**fraud detection, imbalanced data, K-means, K-means++, SMOTE**

I. Introduction

CREDIT card fraud has become a pressing issue, exacerbated by advancements in technology. In 2022, global fraudulent transactions reached an even more alarming peak. To address this challenge, traditional algorithms have been augmented with machine learning and deep learning networks[1]. In the field of fraud detection, contemporary techniques heavily depend on machine learning to extract invaluable insights from complex and diverse data[2].Various machine learning algorithms, such as neural networks, Naive Bayes, K-nearest neighbor (KNN), Support Vector Machine (SVM), and others, are commonly used for fraud detection and have shown satisfactory results[3-5]. However, these models are not immune to the impact of data imbalance, particularly when dealing with significantly unbalanced credit card fraud data. Therefore, effectively handling imbalanced fraud data is crucial for consistent and accurate fraud detection. Currently, the most commonly used approach to address class imbalances is data resampling[6]. Resampling techniques include over-sampling algorithms and under-sampling algorithms. The former increases the number of minority classes, while the latter decreases the number of majority classes. Many scholars have further refined the sampling algorithms[7]. In 2017, Farshid et al[8] introduced an undersampling technique that employs a clustering algorithm aimed at balancing the distribution of the sample data. The approach involves initially clustering the majority class and then randomly selecting some of its samples in each cluster. This effectively balances the sample datasets of the two classes. Lin et al[9] proposed two new undersampling algorithms, based on research by Farshid et al. The first algorithm draws a majority class sample by taking the mean of samples in each cluster. The second algorithm selects the sample in each cluster with the smallest difference from the mean of the cluster samples as the majority class sample. Chawla et al[10] introduced the SMOTE algorithm (Synthetic Minority Oversampling Technique) which is widely used to perform random linear interpolation through each minority class sample and its nearest neighbor samples. The algorithm effectively avoids the overfitting phenomenon by synthesizing new minority class samples for oversampling. However, it still exhibits shortcomings. In 2022, F Sathurlam et al[11] proposed combining the novel noise detection method with the SMOTE algorithm to create the SMOTEWB model. This model used the information to ascertain the optimal number of neighbors for each observation value in the SMOTE algorithm. In 2018, Felix et al[12] introduced an oversampling algorithm that combines K-means clustering with SMOTE. This approach utilizes the SMOTE algorithm and identifies secure clusters through clustering. The number of samples is then assigned based on the minority class's sparsity before application, thus avoiding SMOTE's susceptibility to noisy data. This algorithm helps to prevent the impact of noisy data on the SMOTE algorithm and resolves the issue of small intra-class separation in the initial data[13]. However, it cannot address the issues of data marginalization and fuzzy boundaries caused by the SMOTE algorithm, and it will also change the distribution structure of the original data, causing difficulties in the classification of the detection model and the possibility of misclassification.

A minority class composition algorithm is utilized to balance the dataset and overcome the issue of minority class classification[14]. This paper proposes a KCSMOTE model based on the k-means-SMOTE algorithm with enhancements to the SMOTE over-sampling algorithm. The idea of the k-means-SMOTE algorithm is first used to cluster around the original data by the K-means algorithm and to determine the number of samples for each secure cluster, addressing the

Manuscript received May 30, 2023; revised Sep 29, 2023.

Ke Zhou is a graduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning, China. (e-mail: zhouke0229@163.com).

Chunna Zhang* is an associate professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning, China. (corresponding author, e-mail: zcn1979@163.com).

Yang Yu is a graduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning, China. (e-mail: 2267935580@qq.com).

Shengqiang Cong is a graduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning, China. (e-mail: 997674085@qq.com).

Xiaoping Yue is a graduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, Liaoning, China. (e-mail: 2469622191@qq.com).

influence of noisy samples and the problem of small intra-class separation. The minority class samples in each secure cluster are then clustered using the K-means++ algorithm to find the central sample of the minority class samples. The SMOTE oversampling algorithm is improved by using the central sample point as the base point, and the points generated by the linear interpolation of the algorithm will be centered on the cluster centroid and lie in the central region of the minority class within the secure cluster. The newly generated samples from the minority class in edge regions and classification boundaries will be closer to the centers of the minority class. This will resolve issues of data marginalization and blurred classification boundaries and will avoid the need to change the distribution structure of the original data, thus ensuring the authenticity and reliability of the data.

II. Basic Algorithm Analysis

A. K-means and K-means++

K-means is a traditional unsupervised clustering algorithm and one of the most extensively used clustering algorithms based on distance division[15]. The algorithm consists of the division of data points into K clusters, where K represents the number of clusters. The hyperparameter K and the initial cluster centroids heavily influence the algorithm, and the proper selection of K is essential. The K-means clustering algorithm includes the following basic steps:

1) K initial clustering centroids were randomly selected in the dataset.

2) The distance between each data point and the centroid of each cluster is calculated separately, usually using the Euclidean distance, before being allocated to the cluster containing the centroid with the closest distance.

3) Replacing the cluster centroids for each cluster.

4) Repeat step 2) to determine whether the categories to which all data belong before and after the change of cluster centroids have changed, and if so, repeat step 3) until the cluster centroids no longer change.

5) Determining the clustering results.

The K-means++ clustering algorithm is an improved algorithm of the K-means clustering algorithm, which improves the way the K-means algorithm selects the initial clustering centroids. The basic principle of the K-means++ algorithm for selecting the initial clustering centers is that the clustering centers should be as far away from each other as possible[16]. The K initial clustering centroids were selected as follows:

1) A randomly selected point from the dataset is used as the first cluster centre.

2) The shorter the distance between each sample and the existing centroid, the greater the distance, the greater the probability of selecting the centre of the cluster.

B. SMOTE Oversampling Algorithm

The SMOTE algorithm is an improved oversampling technique based on the random oversampling algorithm and is currently the most widely used sampling method. Chawla et al[7] proposed the SMOTE algorithm in 2002, which enhances the sampling method of the random oversampling algorithm and mitigates the risk of overfitting associated with random oversampling. This technique goes beyond mere replication of existing observations. Essentially, it involves selecting a minority class sample from an imbalanced dataset, identifying its K-nearest neighbor minority class sample, and introducing a new synthetic sample to the dataset through random linear interpolation between the minority class sample and its closest neighbor[17]. The method surpasses the mere duplication of existing observations. The fundamental steps of the SMOTE oversampling algorithm are as follows:

1) For all selected minority class samples X_i, the distance between each minority class sample and it in the dataset is calculated (Euclidean distance is generally chosen as the calculation criterion) to obtain it like K nearest neighbor samples.

2) Calculate the imbalance ratio of the data set and determine the sampling multiplier N.

3) For each minority class sample X_i, a number of K nearest-neighbor samples are randomly selected, assuming that the selection isX_n.

4) A new minority class sample X_{new} is generated based on linear interpolation of X_i and X_n according to Equation (1) until the data reach equilibrium, where $\lambda \in (0, 1)$.

$$X_{new} = X_i + \lambda * (X_n - X_i) \tag{1}$$

The SMOTE oversampling algorithm effectively mitigates the risk of overfitting associated with random oversampling and effectively addresses class-to-class imbalance. However, it encounters challenges in handling intra-class imbalance and noise points, as it tends to overlook the intra-class imbalance issue. Sparse regions remain sparse, while dense regions become even denser. Furthermore, if the selected samples contain noise, the generated synthetic data by the algorithm may still contain noise. The algorithm lacks the ability to determine the distribution area for generating new samples, which can result in new samples being generated in the classification boundary area. This blurs the classification boundary and increases the likelihood of misclassification by the detection model. When faced with two randomly selected samples, the newly generated sample from the sample located in the edge region remains in the edge region, leading to data marginalization. This alteration of the overall distribution structure of a few classes of samples in the original data makes it more challenging for the detection model to recognize patterns. Figure 1 illustrates the behavior of the SMOTE oversampling algorithm with respect to noise, boundary, and classification boundaries.

III. KCSMOTE

The imbalanced credit card fraud detection dataset is processed using the KCSMOTE model proposed in this paper. Firstly, K-means clustering is applied to the training set data without categorical labels. Each obtained cluster is then filtered to identify safe clusters, addressing the noise problem. Distinct sampling weights are assigned to clusters with varying sparsity of samples in specific classes, aiming to tackle intra-class imbalance.Next, the minority class samples within the identified safe clusters are clustered using the K-means++ clustering algorithm to determine the cluster centroids representing these minority class samples. To increase the number of minority class samples and achieve

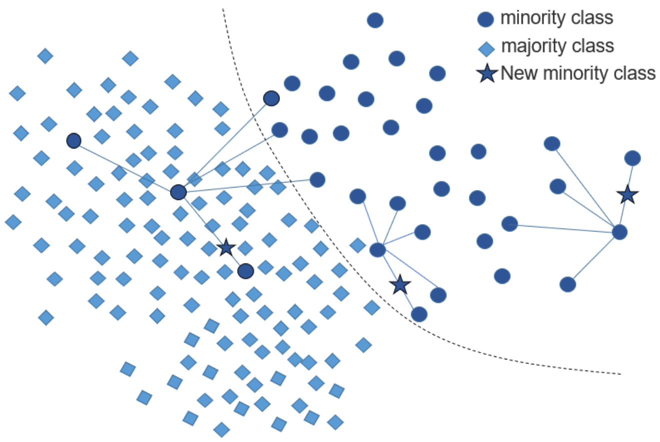

Fig. 1. Possible behaviour of the SMOTE algorithm in the presence of noise and edges and classification boundaries

data balancing on the dataset, the resulting minority class clustering centroids are randomly linearly interpolated with a randomly selected K-nearest neighbor sample of the same class.The KCSMOTE algorithm generates new minority class samples by equalizing the dataset. These new samples are centered around the cluster centroids and are generated in a more logical manner, ensuring that there are no blurred classification boundaries or data marginalization. Importantly, this process preserves the overall distribution structure of the minority class samples in the original data.

A. Construction of the KCSMOTE Model

The basic algorithmic flow of the KCSMOTE model is as follows:

1) The training set of the original data was clustered, and the K-means algorithm was used to divide the data into c clusters, $C = \{C_1, C_2, C_3, ..., C_c\}$. The selection of the hyperparametric clustering number c values was completed by the control variables' method. c values of 5, 10, 25, 50, 100, 125, 150, 200, 250, and 300 were taken for the experiments.

2) The clusters obtained by step 1) are filtered for safe clusters. For each cluster, C_i is filtered using Equation (2) and the selection criteria are based on the ratio between the number of minority classes, and the number of majority of classes unbalanced. To prevent the situation where most classes in a cluster are 0 or a few classes are 0, the numerator and denominator are each added by 1 when calculating unbalanced. The default value of unbalanced is 1. If the result is greater than or equal to 1 then it is a safe cluster $S = \{S_1, S_2, S_3, ..., S_n\}(n < c)$, otherwise, it is a dangerous cluster.

$$unbalanced = \frac{(minoritySum(C_i) + 1)}{(majoritySum(C_i) + 1)} \tag{2}$$

3) Calculate the mean distance $d(S_i)$ for each safety cluster that requires oversampling. To obtain the Euclidean distance matrix D for each security cluster, calculate the Euclidean distance between each minority class sample. Add up the remaining elements of the matrix excluding the diagonal elements, and divide by the number of non-diagonal elements to obtain the average distance $d(S_i)$.

4) The density metric is computed following Equation (3), which utilizes the number of minority classes present in the security cluster, the average distance $d(S_i)$, and the number of features m to derive the density metric density(S_i).

$$density(S_i) = \frac{minoritySum(S_i)}{d(S_i)^m} \tag{3}$$

5) Based on the density metric, the sparsity of the minority class samples in the security cluster is calculated using Equation (4). The sparsity(S_i) of each cluster is obtained by taking the reciprocal of the density measure obtained in step 4). The smaller the density measure, the sparser the minority class of samples within the safety cluster, and the larger the value of sparsity(S_i).

$$sparsity(S_i) = \frac{1}{density(S_i)} \tag{4}$$

6) The sampling weights for each security cluster are calculated as shown in Equation (5). The cluster weight is obtained by dividing the sparsity of each cluster by the sum of the sparsity of all clusters, and the sampling weight values for each security cluster are summed to 1. Larger values of sparsity(S_i), i.e. clusters with sparser samples in a few classes, will be assigned larger sampling weights.

$$r(S_i) = \frac{sparsity(S_i)}{\sum_{j=1}^{n} sparsity(S_j)} \tag{5}$$

7) The number of new minority class samples that need to be generated is calculated for each security cluster using Equation (6). The imbalance ratio N is calculated from the number of minority class samples to the number of majority of class samples in the original data, and then the total number of samples n is calculated from the imbalance ratio N. According to the sampling weight of each security cluster obtained in step 6). The number of samples for each security cluster is calculated, and the larger the weight the greater the number of samples, solving the problem of small intra-class separation.

$$number(S_i) = n * r(S_i) \tag{6}$$

8) Calculate the centroids of the minority class samples in each security cluster. The minority class samples in each of the clusters to be oversampled are clustered using the K-means++ clustering algorithm to find the cluster centroid center(S_i) of the minority class samples in the safe cluster.

9) For a small number of classes of samples in each security cluster find their like K nearest neighbor samples. A minority class sample X_i is arbitrarily selected in a safe cluster, and the Euclidean distance between X_i and each minority class sample in the same cluster is calculated to obtain the K nearest neighbor samples $X_1, X_2, ..., X_K$, and the sample X_n is randomly selected from the K nearest neighbor samples.

10) An oversampling algorithm for generating a small number of classes of new samples within a safe cluster. To rationalize the synthesis of new samples of minority classes, the newly generated minority classes were allowed to move closer to the central region. For this purpose, the original SMOTE model is improved using the safety cluster clustering

centroids obtained in step 8) as base points. A new minority class sample is synthesized using the modified Equation (7), and a new sample point is generated by random linear interpolation of the cluster center(S_i) with similar nearest neighbors X_n, where $\lambda \in (0, 1)$.

$$X_{new} = center(S_i) + \lambda * (X_n - center(S_i)) \quad (7)$$

The clusters chosen by the KCSMOTE model for clustering are designated as secure zones, which predominantly consist of minority classes. This categorization effectively mitigates the impact of noise data from the selected minority classes. By assigning varying weights to the secure clusters based on their degree of sparsity, the algorithm prioritizes sparser clusters by assigning them higher sampling weights. Consequently, a greater number of new samples are generated to address the issue of small intra-class separation observed in a few classes within the dataset. To generate new samples for minority classes, the KCSMOTE algorithm utilizes the centroids of the minority class clusters within the secure zones as reference points for random linear interpolation. The newly generated minority class samples are derived from these centroid samples and are clustered within the proximity of the respective minority class centroid region. This approach ensures that the classification boundary remains clear and avoids potential difficulties that may arise from blurring the boundary due to sample generation in the boundary region. Moreover, new samples generated from the edge regions are adjusted to move closer to the centroid of the respective minority class, effectively addressing the marginalization issue encountered with newly generated minority class samples. The use of the KCSMOTE algorithm for oversampling operations effectively avoids the problem of new samples being generated in different regions causing changes to the overall distribution structure of the few classes of samples in the original data. The schematic diagram of the KCSMOTE model is shown in Figure 2:

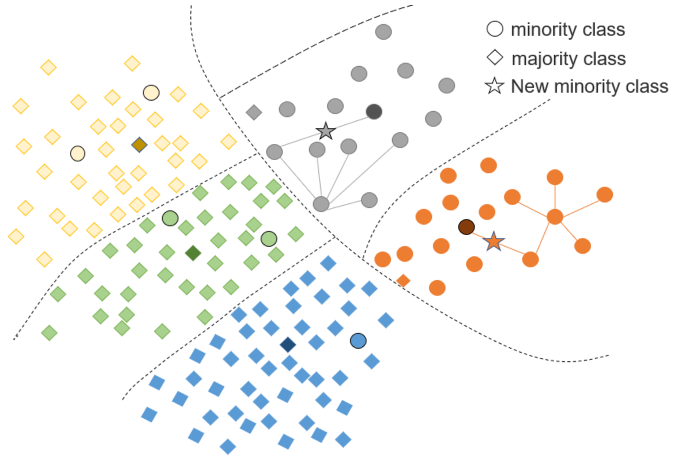

Fig. 2. Schematic diagram of the KCSMOTE model for generating minority class samples

IV. Experimental Results and Analysis

To investigate whether the KCSMOTE model can enhance the effectiveness of the detection algorithm in identifying unbalanced credit card fraud data, the XGBoost algorithm and Random Forest algorithm are chosen as classifiers. Train the classifier using the training set processed by the KCSMOTE model and apply it to predict the original test set that was not treated with KCSMOTE. To verify its validity, the ADASYN algorithm and other most commonly used SMOTE improved algorithms, including k-means- SMOTE, borderline-SMOTE, SMOTETomek, SMOTEEnn, and SMOTEWB algorithms were used to process training set data and unbalanced raw data as controls.

A. Data Set Selection and Splitting

In selecting the dataset, this paper utilizes the credit card fraud detection dataset published on the Kaggle platform[18], as manually generated fraud data lacks the information diversity, complexity, and other characteristics of genuine transaction data. This dataset is a real transaction record from a European credit card company, and it contains data on transactions made by European cardholders via credit cards over two days, guaranteeing the authenticity and validity of the dataset. The dataset comprises a total of 284,807 data points, with a mere 492 instances of fraudulent transactions, resulting in a highly imbalanced dataset. The dataset consists of 31 features. Because of the need to keep customer information confidential, $V_1, V_2, ..., V_{28}$ are anonymous features, the Time feature represents the time when the transaction occurred, the Amount feature represents the amount of the transaction, and the Class feature is the transaction attribute. A Class of 1 is a minority class of positive samples for fraudulent transactions, and a Class of 0 is a majority class of negative samples for normal transactions.

The dataset was split using the train_test_split function in sklearn to distribute the training and test sets in a ratio of 8:2, and the distribution of the obtained training and test sets is shown in Table I below. The KCSMOTE model exclusively operates on the training set data, keeping the test set unaltered.

TABLE I
Data Splitting Situation

	Normal Trading	Fraudulent Trading	Total
Training Sets	227454	391	227845
Test Sets	56861	101	56962
Total	284315	492	284807

B. Evaluation Indicator Selection

To ensure a comprehensive evaluation, this paper selects multiple indicators, including F1-score, Precision, Recall, and AUC, to assess the experimental data, thereby avoiding reliance on a single evaluation metric. Precision refers to the proportion of true positive class samples that are predicted to be positive class samples from a prediction perspective, as shown in Equation (8). Recall refers to the proportion of true positive class samples in the test set that are predicted to be positive from a true perspective, as shown in Equation (9). The model indicates good and stable detection only when both Precision and Recall exhibit high values simultaneously. F1-score is one of the important evaluation metrics to measure the classification problem of unbalanced data, taking into account both the accuracy and recall of the classification model, and can be seen as a kind of weighted average of Precision and Recall. It can be seen from Equation (10) that the value of F1 is only large when both precision and recall are large. AUC represents the area under the ROC curve, serving as a metric to evaluate the effectiveness of a dichotomous classification model by considering its classification ability for both positive and negative samples. The calculation of AUC is based on TPR (True Positive Rate) and FPR (False Positive Rate), as provided in Equations (11) and (12) respectively.

$$Precision = \frac{TP}{(TP + FP)} \tag{8}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{9}$$

$$F1 = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{10}$$

$$TPR = \frac{TP}{(TP + FN)} \tag{11}$$

$$FPR = \frac{FP}{(FP + TN)} \tag{12}$$

The evaluation indicators mentioned above are all derived from formulae that are based on a confusion matrix. This matrix serves as the foundation for the calculation and is displayed in Table II.

TABLE II
CONFUSION MATRIX

	True Positive	True Negative
Test is Positive	TP	FP
Test is Negative	FN	TN

C. Comparison of the Experimental Results

The training dataset underwent processing using the KCSMOTE technique. The clustering parameter, c, was determined using the control variable method, employing values of 5, 10, 25, 50, 100, 125, 150, 200, 250, and 300 in the experiments. The results indicate that a c value of 25 achieves optimal performance when utilizing XGBoost as the detection model, whereas a c value of 125 is most suitable for Random Forest.

The XGBoost algorithm and the Random Forest algorithm are trained using the training set that has been equalised by the KCSMOTE model data, and then the trained detection model is used to detect fraud on the test set of the original data. The training set data processed by other sampling algorithms and the original training set data are compared with the training set obtained from the KCSMOTE model. The results show a significant improvement in credit card fraud detection obtained by the detection model trained on the training set processed by the KCSMOTE model, and the experimental results are detailed in Tables III and IV:

TABLE III
EXPERIMENTAL RESULTS OF THE XGBOOST ALGORITHM AS A FRAUD DETECTION MODEL

	F1	Precision	Recall	AUC
KCSMOTE	0.871	0.871	0.871	0.936
Original Training Set	0.869	0.922	0.822	0.911
ADASYN	0.804	0.761	0.851	0.926
k-means-SMOTE	0.811	0.775	0.851	0.926
borderline-SMOTE	0.860	0.869	0.851	0.926
SMOTETomek	0.835	0.819	0.851	0.925
SMOTEEnn	0.816	0.777	0.861	0.930
SMOTEWB	0.866	0.870	0.861	0.931

The presented analysis of the experimental results indicates that employing XGBoost as a credit card fraud detection model, KCSMOTE has elevated the AUC value by approximately 1% when compared to other techniques. The KCSMOTE model resulted in an AUC value 2.5% higher than that of the original training set, 1% higher than the ADASYN, k-means-SMOTE, and borderline-SMOTE algorithms, and 1.1% higher than the SMOTETomek algorithm. This is an improvement of 0.6% compared to the SMOTEEnn and 0.5% compared to the SMOTEWB algorithms. The enhancement of the F1 result value is considerably notable, exhibiting a 0.2% improvement over the Original Training Set, a 6.7% advancement over the ADASYN algorithm, a 6% increase over the k-means-SMOTE algorithm, and a 1.1% progression over the borderline-SMOTE algorithm. Furthermore, it exemplifies a 3.6% augmentation over the SMOTETomek algorithm, a 5.5% boost over the SMOTEEnn algorithm, and a 0.5% elevation over the SMOTEWB algorithm.

TABLE IV
EXPERIMENTAL RESULTS OF THE RANDOM FOREST ALGORITHM AS A FRAUD DETECTION MODEL

	F1	Precision	Recall	AUC
KCSMOTE	0.879	0.897	0.861	0.931
Original Training Set	0.856	0.930	0.792	0.896
ADASYN	0.846	0.850	0.842	0.921
k-means-SMOTE	0.851	0.851	0.851	0.926
borderline-SMOTE	0.850	0.891	0.812	0.906
SMOTETomek	0.851	0.851	0.851	0.926
SMOTEEnn	0.851	0.851	0.851	0.926
SMOTEWB	0.869	0.922	0.822	0.911

When utilizing the Random Forest algorithm as the model for detecting fraud, it is observed that the KCSMOTE model attains the greatest F1 and AUC values with noteworthy enhancement. The AUC achieved is 3.5% higher than the Original Training Set, 1% higher than ADASYN, 2.5% higher than borderline-SMOTE, 2% higher than SMOTEWB, and 0.5% higher than the remaining methods. The F1 value obtained is 2.3% greater than that of the Original Training Set, 3.3% greater than ADASYN, 2.9% greater than borderline-SMOTE, 1% greater than SMOTEWB, and 2.8% greater than the remaining three.

The results indicate that the training set, after being subjected to the KCSMOTE treatment, displays the best and most stable fraud detection performance. Conversely, in the absence of data balancing, this model exhibits the worst fraud detection effectiveness. To summarize, the KCSMOTE model proposed in this paper offers an effective solution to address the problem of data imbalance in credit card fraud detection.

V. Conclusion

As technology and the economy developed, the number of credit card fraud cases is increasing, and fraudulent methods are emerging, so how to efficiently and accurately identify fraudulent transactions in highly unbalanced fraud data is a key area of research. Therefore, based on the k-means-SMOTE algorithm, this paper proposes a SMOTE algorithm with a few classes of clustering centers as the base point improvement to achieve data balancing. The results show that the KCSMOTE model is significantly effective in dealing with the credit card fraud data imbalance problem, and the new fraud samples generated by the KCSMOTE model are more reasonable when compared with other sampling algorithms. The fuzzy boundary and data marginalization problems of newly generated fraud sample data are greatly reduced, the risk of original data distribution structure changes is reduced, and the detection efficiency and stability of the fraud detection model are improved to a certain extent.

References

[1] N. Liu, J. Che, and Y. Ye, “Wind turbine fault diagnosis based on feature selection and stacking model fusion with small-scale data,” *Engineering Letters*, vol. 30, no. 4, pp. 1588–1595, 2022.

[2] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, “Machine learning: a review of classification and combining techniques,” *Artificial Intelligence Review*, vol. 26, no. 3, pp. 159–190, 2006.

[3] M. Eshtay, H. Faris, and N. Obeid, “Improving extreme learning machine by competitive swarm optimization and its application for medical diagnosis problems,” *Expert Systems with Applications*, vol. 104, pp. 134–152, 2018.

[4] S. V. Kovalchuk, E. Krotov, P. A. Smirnov, D. A. Nasonov, and A. N. Yakovlev, “Distributed data-driven platform for urgent decision making in cardiological ambulance control,” *Future Generation Computer Systems*, vol. 79, pp. 144–154, 2018.

[5] R. Nagarajan and M. Upreti, “An ensemble predictive modeling framework for breast cancer classification,” *Methods*, vol. 131, pp. 128–134, 2017.

[6] M. Kumari and N. Subbarao, “A hybrid resampling algorithms smote and enn based deep learning models for identification of marburg virus inhibitors,” *Future Medicinal Chemistry*, vol. 14, no. 10, pp. 701–715, 2022.

[7] D. L. Weller, T. M. Love, and M. Wiedmann, “Comparison of resampling algorithms to address class imbalance when developing machine learning models to predict foodborne pathogen presence in agricultural water,” *Frontiers in Environmental Science*, vol. 9, p. 701288, 2021.

[8] F. Rayhan, S. Ahmed, A. Mahbub, R. Jani, S. Shatabda, and D. M. Farid, “Cusboost: Cluster-based under-sampling with boosting for imbalanced classification,” in *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS).* IEEE, 2017, pp. 1–5.

[9] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, “Clustering-based undersampling in class-imbalanced data,” *Information Sciences*, vol. 409, pp. 17–26, 2017.

[10] 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.

[11] F. Sağlam and M. A. Cengiz, “A novel smote-based resampling technique trough noise detection and the boosting procedure,” *Expert Systems with Applications*, vol. 200, p. 117023, 2022.

[12] G. Douzas, F. Bacao, and F. Last, “Improving imbalanced learning through a heuristic oversampling method based on k-means and smote,” *Information Sciences*, vol. 465, pp. 1–20, 2018.

[13] S. Bakhtiari, Z. Nasiri, and J. Vahidi, “Credit card fraud detection using ensemble data mining methods,” *Multimedia Tools and Applications*, pp. 1–19, 2023.

[14] S. M. Mostafa, S. A. Salem, and S. M. Habashyis, “Predictive model for accident severity,” *IAENG International Journal of Computer Science*, vol. 49, no. 1, pp. 110–124, 2022.

[15] W. M. Dai Yueming, M. Zhang, and Y. Wang, “Optimizing initial cluster centroids by svd in k-means algorithm for chinese text clustering,” *Journal of System Simulation*, vol. 30, no. 10, p. 3835, 2018.

[16] Y. Jiang, D. Yu, M. Zhao, H. Bai, C. Wang, and L. He, “Analysis of semi-supervised text clustering algorithm on marine data,” *Computers, Materials and Continua*, vol. 64, no. 1, pp. 207–216, 2020.

[17] G. Douzas and F. Bacao, “Geometric smote a geometrically enhanced drop-in replacement for smote,” *Information Sciences*, vol. 501, pp. 118–135, 2019.

[18] S. Dan, S. Wei-li, R. Lan-xiang, M. Sha-sha, G. Xiao-ming, and L. Yi-lun, “Credit card fraud detection method based on improved smote+ enn and xgboost algorithm# br,” *Computer and Modernization*, no. 09, p. 111, 2022.