RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt25512

Enhancing Credit Scoring through a Multi-Stage Hybrid Ensemble Classification Model with Feature Engineering and Data Balancing

P. Ramila Rajaleximi [a,*], A. Saravanan [b], B. SivaSakthi [c], Leena Jaganathan [d], S. Sathya Bama [e]

[a] Department of Computer Science, Sanpada College of Commerce and Technology, Sanpada, Navi Mumbai - 400705, Maharashtra, India

[b] Department of Computing, Coimbatore Institute of Technology, Coimbatore - 641014, Tamil Nadu, India.

[c] Department of Data Science and Computer Science and Design, MVJ College of Engineering, Bengaluru - 560067, Karnataka, India.

[d] Department of Computer Science, University of Zambia, Lusaka - 10101, Zambia.

[e] Independent Researcher, Lawley Road, Coimbatore - 641003, Tamil Nadu, India.

* Corresponding Author Email: pslexiimi@gmail.com

DOI: https://doi.org/10.54392/irjmt25512

Received: 23-05-2025; Revised: 02-09-2025; Accepted: 16-09-2025; Published: 29-09-2025

Abstract: Financial institutions face significant credit risk when evaluating credit applications, making fraud detection and prevention a primary task for all financial institutions and service companies. After decades of research and development, the credit scoring model has been improved by artificial intelligence and machine learning. This research proposes a new multi-stage hybrid ensemble classification model for improving the efficiency of credit scoring applications by leveraging artificial intelligence and machine learning advancements. A new weighted ensemble filter and an improved borderline SMOTE for minority oversampling are proposed to enhance the dataset quality by identifying the most suitable set of features for analysis, addressing the bias caused by an imbalanced class distribution, and improving predictive accuracy. Moreover, to reduce the variance and enhance the accuracy, a new nested ensemble classification model was introduced to enhance the predictive accuracy of credit scoring performance. The proposed model outperformed four credit scoring datasets, achieving an improved accuracy and AuC of 92.43% and 0.968 for the Australian credit dataset, 83.16% and 0.901 for the German credit dataset, and an improved AUC of 0.938 for the Japanese credit dataset. The experimental findings validate the practical efficiency of the proposed model, which outperforms competing models owing to its unique combination of feature engineering, balancing techniques, and use of a novel nested ensemble classifier. The study offers financial institutions not only a more robust tool for credit scoring to make more reliable credit decisions but also reduces operational costs associated with manual evaluation processes, risk management, and fraud mitigation.

Keywords: Credit Scoring, Hybrid Ensemble Classification, Feature Importance, Feature Selection, Minority Oversampling.

1. Introduction

The provision of credit is fundamental to the financial system, and credit risk has a substantial influence on the profitability of financial corporations and institutions. Financial defaults and bad debts at epidemic levels threaten the stability of the financial system and, by extension, the national economy. Furthermore, businesses and institutions may go bankrupt because of excessive credit risks [1]. Therefore, building effective and reliable models for predicting creditworthiness is of great practical importance for preventing losses and maintaining competitiveness in the industry [2]. While traditional bankruptcy prediction models have contributed to financial stability assessments, recent advancements in artificial intelligence (AI) and machine learning (ML) have revealed new opportunities to enhance predictive accuracy in credit scoring [3, 4]. However, existing models lack a comprehensive approach that combines feature engineering, class balancing, and ensemble learning to address these challenges [5]. Credit classification is typically a binary classification task in which customers are labelled as having good or bad credit based on their default risk [6]. However, the quality of the datasets used to train the model significantly influences the success of these models, as irrelevant or duplicated features and class imbalance negatively impact prediction accuracy [7-10]. To improve credit scoring models, researchers have focused on enhanced feature selection approaches to address class distributions. Feature selection

techniques are potential strategies in ML for identifying key characteristics to expedite classification model training and improve predictive abilities [11-12]. Credit scoring models require effective feature processing to eliminate redundant and irrelevant features, improve prediction performance, and reduce computational complexity, while also revealing implicit knowledge [2, 13]. Similarly, the credit scoring model's prediction performance is also impacted by an unbalanced class distribution in customer datasets, with a smaller sample size for customers with bad credit [14, 15]. The classification model's bias towards the majority class leads to high accuracy, but poor accuracy for the minority class having customers with bad credit, resulting in greater financial losses [16, 6]. Resampling the training set to ensure an even class distribution is common, but traditional methods may overfit by adding the same samples, requiring further exploration [17].

Researchers have emphasized the classification process using single and ensemble classification methods to boost the predictive ability of classification models [18]. Some standard classification models include random forest (RF), Naïve Bayes (NB), logistic regression (LR), linear logistic regression (LL), k-nearest neighbor (KNN), artificial neural network (ANN), AdaBoost (AB), decision tree (DT), and support vector machine (SVM) [16]. Many studies indicate that the classification error rate may be lowered, and the model's resilience can be increased by combining various base classifiers. Since then, ensemble classification models such as bagging [21], AdaBoost [22], and random forest [23] have received increasing attention, but time complexity remains a consideration in real-world applications [19, 20].

Given the significance of credit scoring, this study proposes a novel multistage hybrid ensemble classification model that significantly advances the current state of credit scoring by combining multiple innovative methods. Despite the success of various ensemble classifiers, several challenges remain unaddressed in the existing credit scoring systems. Many ensemble models suffer from overfitting, particularly when dealing with small or unbalanced datasets. Others face a trade-off between predictive accuracy and interpretability, which limits their adoption in high-stakes financial environments, where transparency is essential. Moreover, although powerful, stacked and nested classifiers often introduce high computational complexity, making them impractical for real-time credit evaluation scenarios.

To address these challenges, the proposed model introduced a three-stage approach. The first stage introduces a novel weighted ensemble filter for feature selection, which not only enhances the dataset by selecting the optimal set of features, but also improves model interpretability and reduces complexity. The second stage introduces an improved borderline SMOTE for minority oversampling, addressing the issue of the imbalanced class distribution more effectively than traditional methods, thereby improving the predictive accuracy of the model for the minority class. Finally, a novel nested ensemble classifier is proposed to further enhance the predictive accuracy while minimizing the variance and limiting the time complexity often observed in deep ensemble models.

Unlike existing methods, the proposed model combines feature engineering, class balancing, and nested ensemble techniques into a single framework, making it innovative and highly applicable to practical financial situations. The proposed model enhances credit scoring accuracy and robustness by addressing dataset quality and class imbalance through a multistage process. Thus, this study contributes to the field of credit scoring for financial institutions by outperforming traditional models, as demonstrated through rigorous experimental validation.

The remainder of the paper is structured as follows: Section 2 reviews the existing literature related to the proposed research work; the detailed procedure and framework are discussed in Section 3; Section 4 explores the experimental setup, including datasets used, parameter setting, and the performance metrics used for evaluation; Section 5 analyzes the experimental results; Section 6 discusses the statistical analysis and interpretation of the results; and Section 7 draws conclusions and suggests further study.

2. Literature Review

The proposed study had three significant stages: feature selection, resampling, and ensemble classification. Given the importance of ML and credit scoring are so crucial, numerous researchers have concentrated on these three elements. This section reviews previous studies related to individual phases and the overall study.

2.1 Feature Selection Techniques

In general, feature selection reduces the number of irrelevant or redundant features to improve the classification performance [24]. Depending on the criteria used for assessment, feature selection can be classified as a filter technique, wrapper method, or embedding approach [25]. Assigning scores to features using distance, information, and correlation allows filter algorithms to calculate faster while still capturing an effective feature set [26]. Common methods used in filter-based approaches are Pearson's correlation [27], information gain, gain ratio, and relief algorithm [28]. The main drawback of filter-based feature selection is the identification of the stopping criteria. Moreover, with the development of ML algorithms, several ensemble methods have been proposed to improve the feature-selection process. It includes rank aggregation-based

feature selection [29], a hybrid data mining model of feature selection algorithms [30], local search-based methods [31], an optimized multiple rank score model [32], and ensemble feature ranking (median, correlation, and LR) [33]. Because each of the aforementioned feature ranking algorithms generates unique ratings for the same feature, ensemble feature selection is an area that needs to be investigated further.

2.2 Class Balancing Techniques

Most classification models struggle with class imbalance issues that can be solved by applying resampling techniques. Regular resampling methods include random oversampling (ROS) and random undersampling (RUS), which increase the number of samples from the minority class and decrease the number of samples from the majority class to achieve an even class distribution. Although these two methods appear to provide similar results, it has been proven that ROS outperforms RUS for most classifiers [34]. In addition, several researchers have created resampling strategies based on a wide range of intelligent methods to improve the performance of traditional methods. Such techniques include cluster-based oversampling (CBOS) [35], one-sided selection (OSS) [36], synthetic minority oversampling technique (SMOTE) [37], Wilson's edited nearest neighbor [38] combined with SMOTE (SMOTE + ENN) [39], adaptive synthetic sampling approach (ADASYN) [40], safe-level synthetic minority oversampling technique (SL-SMOTE) [41], majority weighted minority oversampling technique (MWMOTE) [42], KNN-SMOTE-LSTM [43], and borderline-SMOTE [44]. Several studies have been conducted to analyze and compare the performance of these methods, in which SMOTE offers better performance than several classifiers [45], and SMOTE + ENN provides good results for credit scoring [16]. Recently, generative adversarial networks (GANs) have gained prominence for handling imbalanced datasets by generating realistic synthetic samples [46]. However, additional resampling approaches must be developed to further enhance the efficiency of the classification system.

2.3 Ensemble Learning Techniques

Ensemble classifiers combine the strengths of several classifiers and rules to learn from training data. They are more stable and learn more quickly than the individual classifiers. Furthermore, it can be classified as either a homogeneous or heterogeneous ensemble, depending on the structure of the underlying classifiers. In general, homogeneous ensembles employ the same base classifiers, whereas heterogeneous ensembles employ different classifiers to predict class labels with improved predictive power [30, 47]. Furthermore, ensemble classification models can be categorized into bagging, boosting, and stacking [19]. Bagging or bootstrap aggregation combines many predicted model versions in which individual models are trained and averaged, thus reducing variance [48]. Boosting is a different approach than bagging, in which the base classifiers learn successively and adaptively to enhance the model predictions [49]. Bagging and boosting are homogeneous ensembles, whereas stacking is a heterogeneous ensemble. Stacking, also known as "stacked generalization," is a method for improving model performance by combining the results of numerous trained models that address this issue [50]. In this study, only bagging and stacking were included in the proposed ensemble classification model because of increased complexity and implementation challenges.

Several learning models for credit scoring have been proposed in previous studies. Recently, interpretable ML and responsible AI have been proposed. Explainable AI and deep learning have been suggested for credit card default prediction, overcoming challenges with complex explanations and user difficulties [51]. In addition, a system for synthetic financial documents and fraud detection was proposed [52]. However, this requires further investigation. A classifier consensus system has been proposed that merges various base classifiers, such as neural networks (NN), support vector machines (SVM), random forests (RF), decision trees (DT), and naïve Bayes (NB) [53]. A unique ensemble model for credit scoring was presented to achieve good performance and resilience across imbalanced ratio datasets [18]. Similarly, information-gain-based feature selection was proposed for credit scoring applications using KNN, NB, and SVM classifiers [54]. Other methods for credit scoring applications include high-accuracy priority rule extraction [55], ensemble classification models with optimization techniques [56], hybrid models with multi-population niche genetic algorithms [20], and ensemble models with enhanced outlier adaptation [2]. Accordingly, the performance of the proposed method was benchmarked against that of existing approaches to validate its effectiveness.

2.4 Research Gap

Although a wide range of studies has examined feature selection, resampling strategies, and ensemble classifiers separately or in combination, most methods focus on one or two of these elements, frequently ignoring how these phases interact within a comprehensive classification pipeline. Computationally efficient filter-based approaches might not be classification-aware, whereas wrapper and ensemble-based selection strategies provide superior performance at the cost of greater complexity [25, 29–33]. Although resampling techniques, such as SMOTE and its variants, have demonstrated improved handling of class imbalance, their effectiveness often depends on the base classifier used [37, 39, 41–44]. Ensemble classifiers such as bagging and stacking have shown

superior generalization [49, 51], whereas boosting offers strong predictive power but suffers from implementation challenges [49]. Only a few studies have jointly optimized feature selection, resampling, and ensemble learning to maximize predictive accuracy in credit scoring problems [2, 18, 20, 54]. The notable methods from the existing literature, along with their strengths and limitations, are summarized in Table 1.

Therefore, this paper proposes a unified framework that integrates intelligent resampling, improved ensemble feature selection, and a hybrid of bagging and stacking for classification. This design aims to overcome the individual limitations of earlier research and enhance the predictive performance and resilience in credit scoring applications.

3. Proposed Multi-Stage Hybrid Ensemble Classification Model

In this study, a new multistage ensemble model was suggested, and the overall framework is presented in Figure 1. The proposed model includes four significant stages: data preprocessing, feature selection, minority oversampling, and ensemble classification.

In the first stage, fundamental preprocessing steps are followed to enhance the accuracy and efficiency of the underlying model by transforming the raw data into formatted data. In this stage, missing values are handled, categorical values are encoded, and numerical values are normalized. Finally, the preprocessed data were divided into training and testing sets. In the feature selection stage, several filters are applied to the preprocessed training set, from which the feature importance scores are computed. The absolute feature importance was computed by combining the feature importance scores from the various filters, and the optimal feature subset was selected. In the next minority oversampling stage, the dataset with optimal features is given as input, and the k-nearest neighbors are selected for each minority sample, for which the new samples are generated using SMOTE by verifying the borderline conditions. Moreover, the consistency of the class labels was verified before they were included in the dataset. In the ensemble classification stage, bagging and stacking are applied to the classifiers, and the final prediction results are identified based on the average prediction probabilities of the classifiers.

3.1 Feature Selection With Weighted Ensemble Filters (WEF)

The majority of predictive models face difficulties such as slowing down model development and training and placing high demands on the system memory [2]. Removing irrelevant and redundant features and selecting the most important features are significant for improving the performance of the underlying model [31]. Filter methods are among the most common feature selection techniques that are widely used and work based on statistical or feature importance measures. The filters identify and remove irrelevant attributes based on scores and rankings, which improves the prediction efficiency [29]. In general, this method does not require any learning algorithms because the features are selected based on their association with the target feature. Filter techniques are used in this study because they do not overfit the data and require minimal computational complexity. Moreover, each filter was superior to the others in different aspects.

Table 1. Notable methods used in existing studies

Technique	Strengths	Limitations
Filter Feature Selection [26–28]	Fast, independent of classifier	May ignore feature dependencies; stopping criteria unclear
Ensemble Feature Selection [29–33]	Improved robustness, aggregation of multiple rankings	Computationally intensive; ranking inconsistency
SMOTE and Variants [37, 39, 41–44]	Widely adopted, improves minority class representation	Risk of overfitting; may generate noisy samples
GAN-based Resampling [46]	Generates realistic synthetic samples	Requires complex training; sensitive to mode collapse
Bagging [48]	Reduces variance, improves stability	Does not reduce bias; may underperform on small datasets
Boosting [49]	Sequential learning, often yields high accuracy	Prone to overfitting; complex implementation
Stacking [50]	Leverages diverse model strengths	Increased model complexity and interpretability challenges

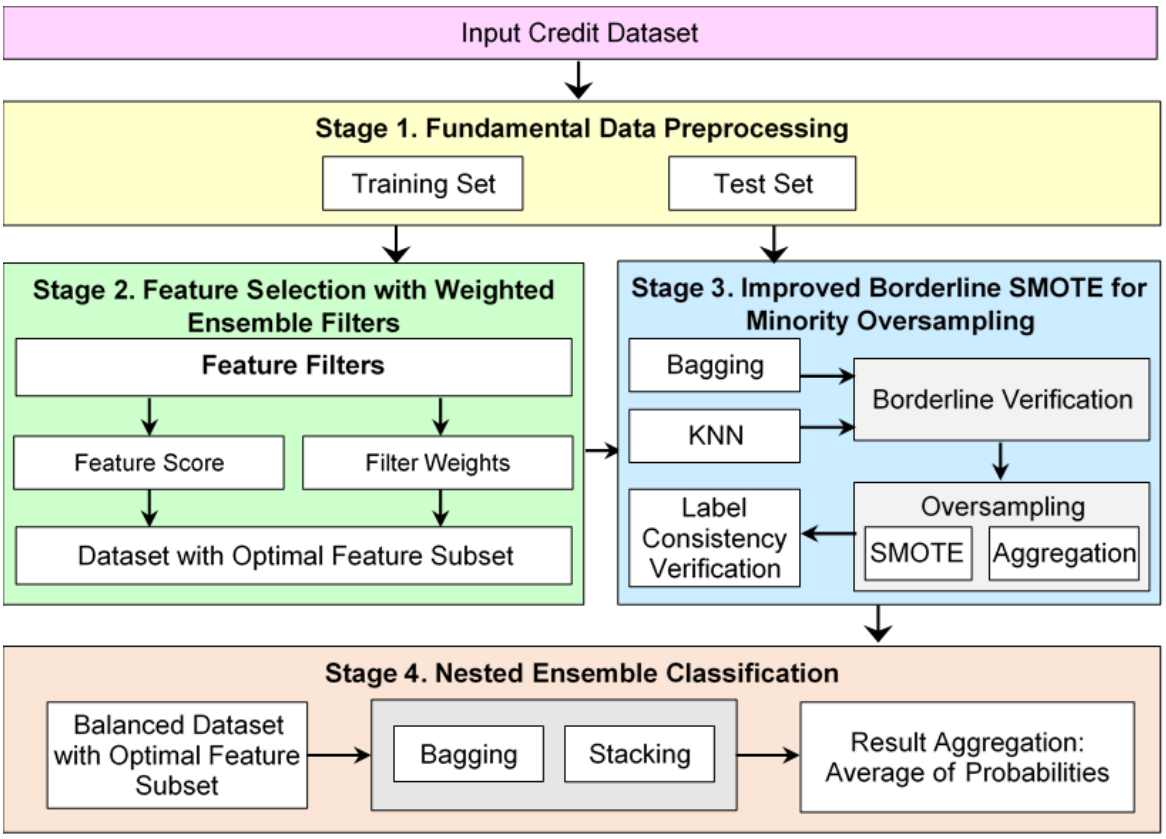

Figure 1. The overall framework of the proposed model

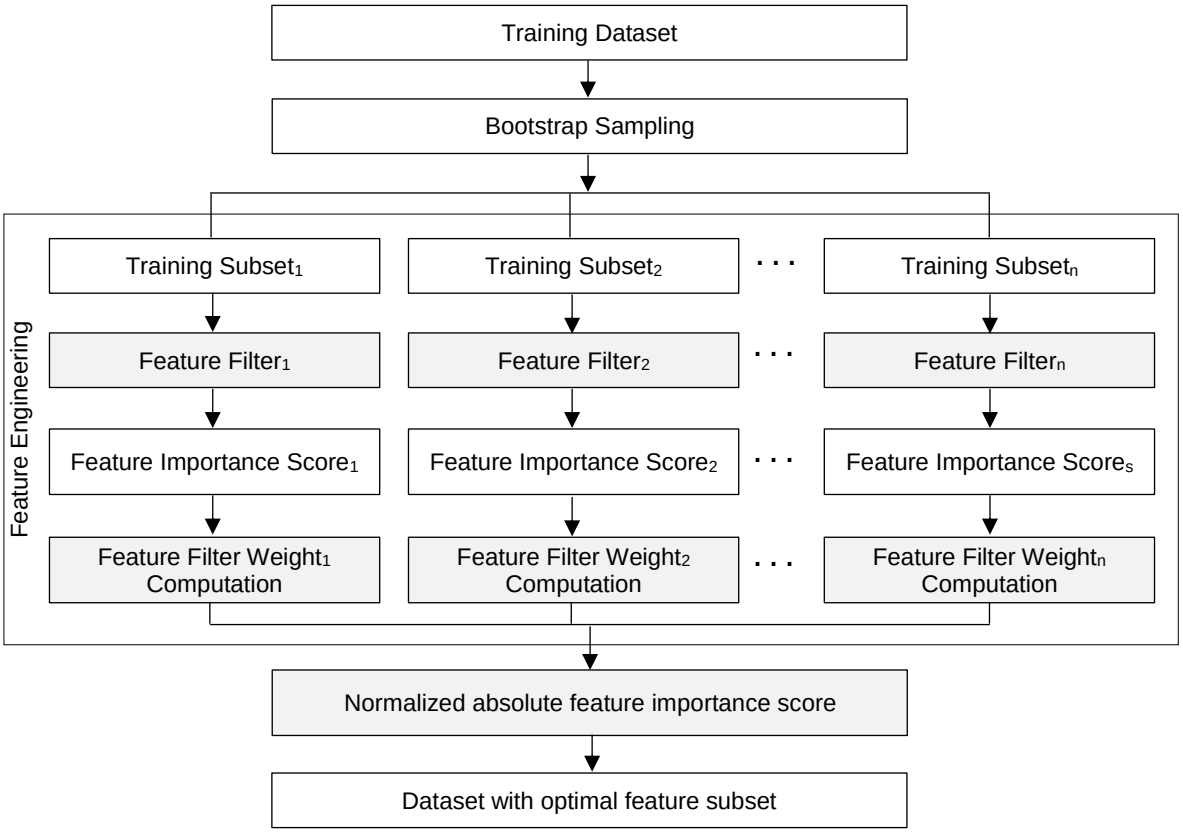

Figure 2. Concise diagram of feature selection using weighted ensemble filters

Thus, considering the significance of bagging and filters, a new weighted heterogeneous ensemble of filters for selecting significant features was proposed. Figure 2 shows the concise framework of the proposed feature selection method using weighted ensemble filters.

In this step, the preprocessed training set was used as the input for partitioning using the bagging technique of the bootstrap sampling method. Various filters were applied to each training subset and the feature importance scores (FIS) were examined. The proposed method employs four significant filter techniques to assess the importance of features such as the gain ratio, information gain, correlation, and symmetrical uncertainty concerning the class [32].

- **Correlation (COR)**: The process determines Pearson's correlation between the attribute and the class to compute the value of a feature.
- **Gain Ratio (GR):** This method computes the gain ratio relative to the class to assess the value of a feature.
- **Information Gain (IG):** This technique measures the information acquired relative to the class to assess a feature's relevance.
- **Symmetrical Uncertainty (SU)**: This technique uses symmetrical uncertainty relative to the class to establish a feature's significance.

With each filter, FISs greater than the threshold were selected and trained using the dominant classifier, for which the accuracy and area under the curve (AUC) values were estimated. Considering the efficiency of the random forest algorithm in credit scoring, the proposed model uses RF at each stage for various operations [24]. These estimated values were used to compute the weights of the filters, as specified in Eq. (1).

$$w(f_j) = \frac{ACC_j + AUC_j}{2} \quad (1)$$

where $w(f_j)$ indicates the weight of the j^{th} filter and ACC_j and AUC_j indicate the classification accuracy and area under the curve values obtained for the selected features using the j^{th} filter.

Upon estimating the weights for the filters based on the classification performance with the corresponding selected features, the final absolute feature importance score was evaluated using the equation proposed by [21], as in Eq. (2).

$$FIS_i = 1/n \sum_{j=1}^{n} w(f_j) \frac{FIS_{ij} - \min(FIS_j)}{\max(FIS_j) - \min(FIS_j)} \quad (2)$$

where FIS_i is the final absolute feature importance score for the i^{th} feature, n is the number of filters used, $w(f_j)$ is the weight of the j^{th} filter, FIS_{ij} is the feature importance score for the i^{th} feature using the j^{th} filter, $\min(FIS_j)$ and $\max(FIS_j)$ are the minimum and maximum feature importance scores computed for the j^{th} filter.

Based on the final obtained feature importance scores, features with scores above the specified threshold were selected for further processing, thereby achieving dimension reduction and resulting in a dataset with optimal features. Specifically, in the feature selection stage of the analysis, the bottom 25% of the features with the lowest final importance scores (FIS) were eliminated, while the remaining 75% were retained.

Algorithm 1. Feature Selection with Weighted Ensemble Filters (WEF)

Procedure featureselection_WEF (training_set, threshold)

Begin

Preprocess the dataset and partition using bootstrap sampling

Initialize the list of filters = [COR, GR, IG, SU]

For each subset in training_set

For each method in the filters

Compute the Feature Importance Scores (FIS)

End For

End For

For each method in the filters

Select the significant features based on the given threshold

Apply random forest classifier algorithm to the selected features

Compute the filter_performance using AUC and ACC

Compute the filter_weights as (AUC + ACC) / 2

End For

For each feature in the training set

Compute the final absolute feature importance score using normalization

End For

Select features with FIS greater than the threshold

Remove redundant features by checking the correlation

Output the final set of selected features

End Procedure

This threshold-based pruning technique was established through iterative experimentation across various datasets and classifier configurations rather than random selection. A sensitivity analysis further validated the robustness of this threshold, demonstrating that

classification metrics such as accuracy and AUC remained stable when the cutoff ranged from retaining 70% to 80% of the top-ranked features (corresponding to FIS thresholds between 0.55 and 0.65). This consistency strengthens the dependability and generalizability of the threshold for selecting the most informative feature subsets. Moreover, for features with the same score, their correlation was verified; if the correlation exceeded 0.95, one of the features was removed. The pseudocode for the WEF is presented in Algorithm 1.

3.2 Improved Borderline SMOTE for Minority Oversampling (IBL-SMOTE)

An imbalanced class distribution is one of the primary issues affecting the performance of predictive modelling for credit scoring applications. This indicates that the positive class (majority class) and negative class (minority class) samples of the training sets were not balanced or biased. Although predicting the minority class label is more significant than predicting the majority class in real-world applications, it is challenging to predict the minority class owing to its fewer number of samples. Most classification algorithms lean towards majority class instances; therefore, minority ones are poorly modelled into the final system, whose predictions are more important [16]. Thus, considering the efficiency of bagging and oversampling, an improved minority oversampling method is proposed that uses bootstrap sampling, borderline SMOTE, and consistency verification.

Figure 3 shows a concise diagram of the proposed improved borderline SMOTE minority oversampling method. The training set with optimal features is provided as the input for this stage, for which bootstrap sampling is applied. For each training subset m and minority class sample x, an improved borderline SMOTE was applied [44].

To determine the borderline region, the method calculates the number of majority class neighbours among the k-nearest neighbours (k=5 in this study) of a given minority instance. Let Nmaj(x) be the number of majority class samples among the k neighbours of x. Then,

- If Nmaj(x) = 0, the instance is considered "safe".
- If 0< Nmaj(x)<k, the instance is in the "danger" zone.
- If Nmaj(x) = k, the instance is considered "noisy" and not used for oversampling.

Thus, new samples were generated only from minority-class instances located in the "danger" zone.

Thus, for each minority class sample x, the k-nearest neighbor is identified and new samples are added based on the number of neighbors belonging to the majority class. Thus, if the number of neighbors belonging to the majority class is less than k/2, then the minority class samples are in a safe state, and thus the aggregation of minority class neighbors is performed. On the other hand, if the number of neighbors belonging to the majority class is greater than k/2, say l, then the minority class samples are in danger; thus, new samples are generated, as given in Eq. (3).

$$x^{\wedge}=x+rand(0,1)*|x-x\_l| \quad (3)$$

where x^' is the generated sample, x is the original sample for which the k-nearest neighbor is identified, rand(0,1) is a random number between 0 and 1, 〖and x〗 _l is the lth neighbor belonging to the minority class.

Algorithm 2. Improved Borderline SMOTE (IBL-SMOTE)

Procedure IBL_SMOTE(train_set, k)

Begin

Preprocess the dataset and apply bootstrap sampling to create subsets

For each minority sample x in the train_set

Find the k-nearest neighbors of x

If fewer than k/2 neighbours are from the majority class **then**

Mark x as "safe"

End If

If more than k/2 neighbours are from the majority class **then**

Mark x as "in danger"

Generate new sample x' using:

x' = x + random value * (x - majority_neighbor)

End If

End For

Check the class label of the new sample x' using KNN

If the label is "minority," **then**

keep x' in the dataset

Else discard x'

End If

Return the updated dataset

End Procedure

Finally, instead of directly including the generated samples back in the dataset, they were verified for class label consistency [43]. The KNN classifier is used to predict the class label for the newly generated minority samples.

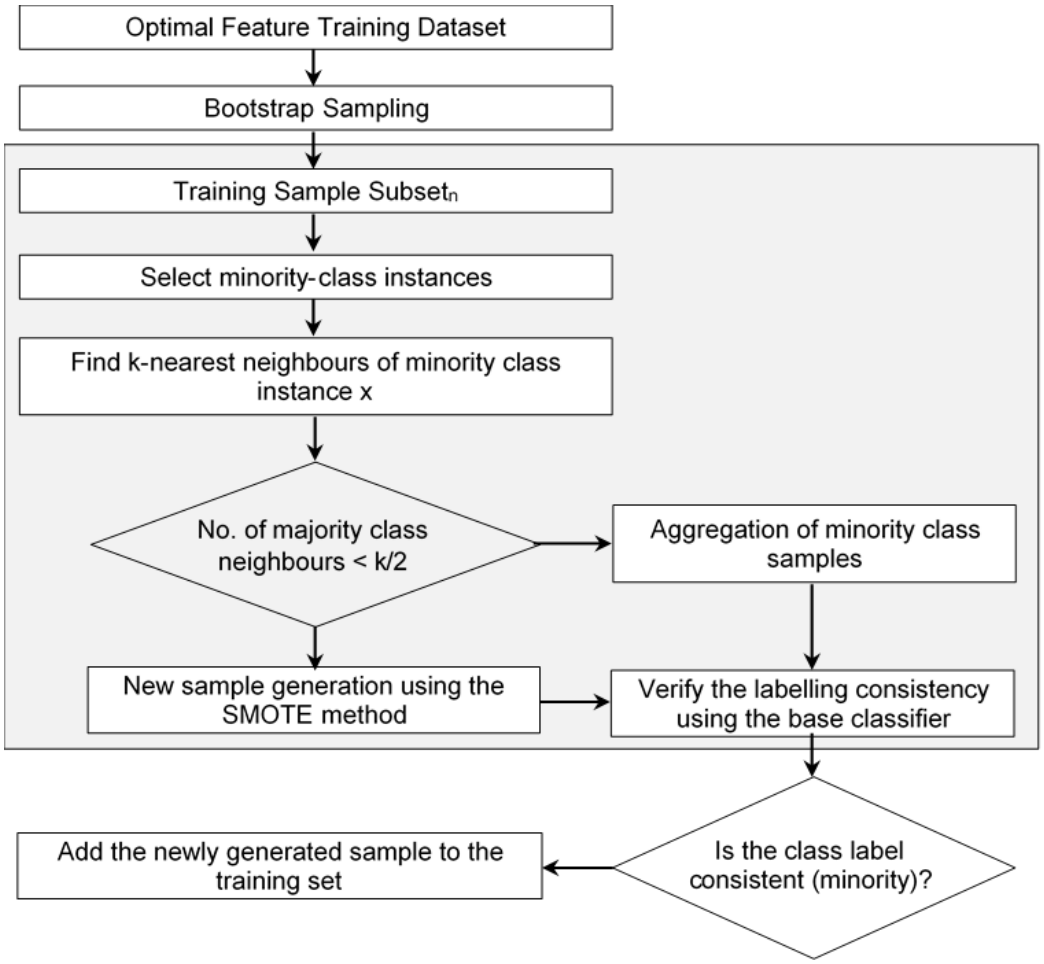

Figure 3. Concise diagram of improved minority oversampling method

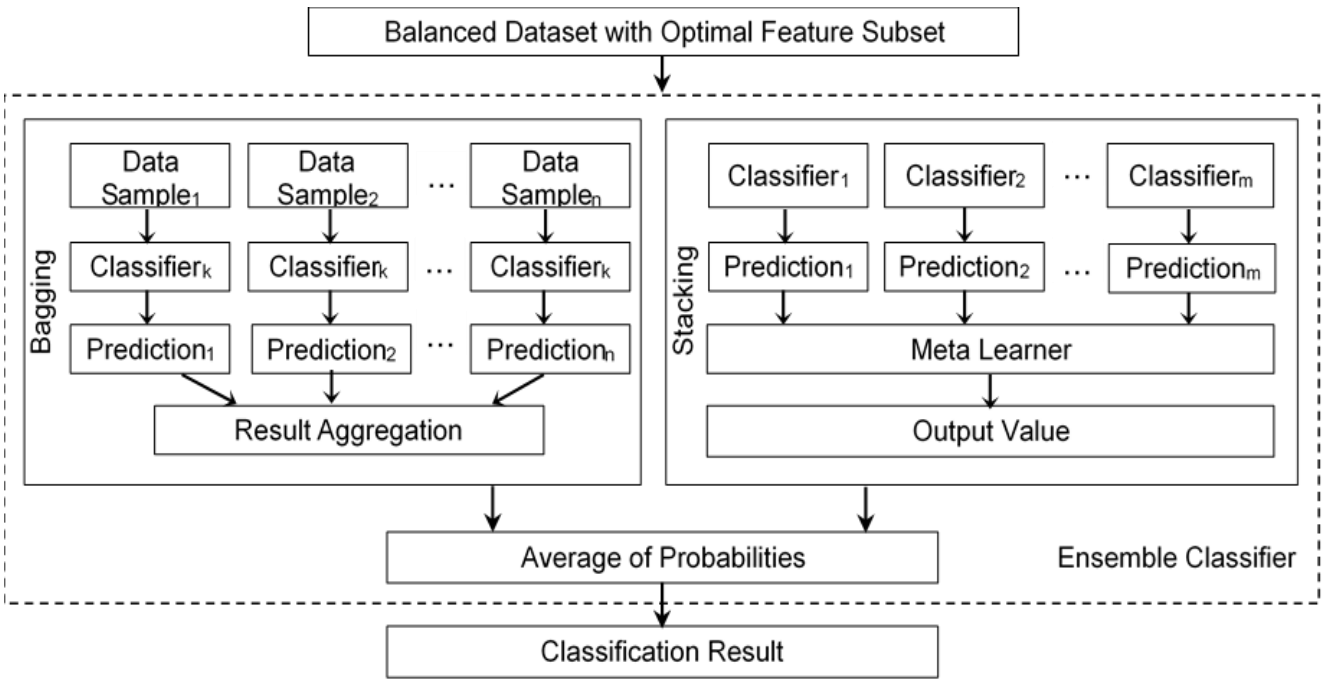

Figure 4. Concise diagram of proposed nested ensemble classification model

If the label is not consistent (classified as the majority class), then the samples are discarded; otherwise, the samples are included in the dataset. The pseudocode for IBL-SMOTE is presented in Algorithm 2.

3.3 Nested Ensemble Classification Model (NEC)

Ensemble models aim to utilize the predictive power of multiple learning algorithms within a single framework. Ensemble models can be implemented in three ways: bagging, boosting, and stacking [19].

- Bagging, which often considers homogeneous learners, teaches each one separately in parallel and then averages their predictions using an aggregation method [21].
- Boosting focuses on homogeneous weak learners, learning them sequentially in an adaptive manner, before merging them in a deterministic manner.
- Stacking involves training diverse weak learners in parallel and integrating them by training a metamodel to produce a forecast based on their predictions [20].

Bagging aims to decrease variance by increasing the size of the training sets through combinations with repetitions, while boosting and stacking improve prediction accuracy and reduce bias. Although boosting enhances prediction accuracy, it is difficult to implement owing to time and computational complexity.

Given the significance of ensemble models, a nested ensemble classification model is employed that combines bagging and stacking ensembles. A simple structure of the proposed ensemble classification model is shown in Fig. 4. For the nested ensemble classifiers, both bagging and stacking were used to improve the predictive power of the models. In the proposed model, two classifiers, RF and NB, are used with bagging homogeneous ensembles because these classifiers exhibit different credit scoring performances [15]. RF and NB complement each other, enhancing ensemble robustness and reducing overfitting compared with linear models or multiple decision trees. For stacking ensembles, Multinomial LR and Linear LR are employed as base classifiers, whereas RF is used as a meta-classifier owing to its superior performance. Because of their ease of use and efficacy in handling various data distributions, Multinomial LR and Linear LR were selected as base classifiers in stacking. RF was chosen as a meta-classifier owing to its robustness, faster training, and ability to handle high-dimensional data, requiring less computational effort than other models, such as XGBoost and meta-logistic regression. Finally, the classification outputs from both the bagging and stacking ensembles were aggregated by averaging the probabilities. This method reduces the variance and improves the classification stability efficiently without adding computational complexity.

Given the significance of ensemble models, a nested ensemble classification model is employed that combines bagging and stacking ensembles. A simple structure for the proposed ensemble classification model is presented.

Algorithm 3. Nested Ensemble Classification

Procedure NEC_Model(train_set)
Begin
Preprocess the dataset
Apply the bagging ensemble method
Use RF and NB as classifiers in bagging
Train the classifiers in parallel with the training data
Apply the stacking ensemble method
Use multinomial LR and linear LR as a base classifier
Train the classifiers in parallel with the training data
Use RF as the meta-classifier to combine predictions
Combine the outputs from the bagging and stacking ensembles
Aggregate the classification results using the average of probabilities
Output the final classification result for the nested ensemble
End Procedure

Moreover, to transform the qualitative data into a numerical form suitable for analysis, one hot encoding is used where each attribute value of a feature is converted into a new feature with only two values, “1” or “0” [5]. Finally, the larger values in the numerical values were normalized using min-max normalization [58].

4.2 Parameter Setting

After the data had been cleaned and organized, it was split into a training set and a testing set using the 80/20 rule that has been used in many studies [18, 20]. In general, cross-validation increased the accuracy of the proposed model and decreased the influence of data dependency. More specifically, in many cases, 10-fold cross-validation was applied to the proposed model because of its known efficacy [24]. To assess the proposed classification model, various training options such as 5-fold, 10-fold cross-validation, and percentage split (66-34) in which the dataset is split into 66% training set and 34% test set, were employed [16]. To reduce the impact of contingency, the average prediction result from 10 iterations was accepted as the outcome for each dataset.

Table 2. Dataset description

Dataset	#Features (Numerical/ Categorical)	#Samples	#Positive Samples	#Negative Samples
Australian	15 (6/9)	690	307	383
German	21 (7/14)	1000	300	700
Japanese	16 (5/11)	690	307	383
UK Thomas	14 (9/5)	1225	902	323

During the feature selection phase, 75% of the features with the highest scores were retained, whereas the lowest 25% were discarded. Pearson's method was used for the correlation analysis to remove redundant features. The proposed feature selection method retained 10 features for the Australian dataset, 15 for the German dataset, 11 for the Japanese dataset, and 10 for the UK-Thomas dataset.

Moreover, in the oversampling and classifier phases, the study follows the parameter setting of [16]. Thus, for SMOTE, SMOTE + ENN, ADASYN, SL-SMOTE, and the proposed method, the number of nearest neighbors was taken as five, and for MWMOTE, the parameter values were $k_1 = 5$, $k_2 = 3$, and $k_3 = N_{min}/2$. In addition, K = 5 was selected because it is widely used in the literature to balance noise reduction and oversampling effectiveness. Several values of k, such as 3, 5, 7, and 9, were empirically assessed in this study. Among these, k = 5 consistently yielded dependable and stable results across all datasets, demonstrating its suitability for SMOTE variations.

Similarly, Weka was used to implement classifiers such as Random Forest (RF), Naïve Bayes (NB), Logistic Regression (LR), Linear Logistic (LL), AdaBoost (AB), k-nearest neighbors (KNN), Multilayer Perceptron (MLP), and Decision Tree (J48). The default parameters were retained to ensure reproducibility, with the exception of KNN, where the number of neighbors was explicitly set to k = 3, based on a preliminary performance evaluation. The key parameters used were as follows:

- RF: numTrees = 100, numFeatures = 0, maxDepth = 0 (unlimited), seed = 1
- NB: useKernelEstimator = false, useSupervisedDiscretization = false
- LR: ridge = 1.0E-8, maxIts = -1 (no limit)
- LL: heuristicStop = 50, numBoostingIterations = -1 (no limit)
- AB: numIterations = 10, baseClassifier = DecisionStump, seed = 1
- KNN: K = 3, distanceFunction = Euclidean, crossValidate = false
- MLP: learningRate = 0.3, momentum = 0.2, trainingTime = 500, hiddenLayers = "a"
- J48: confidence factor = 0.25, minNumObj = 2, pruning = true, seed = 1.

4.3 Evaluation Metrics

To assess the performance of the proposed model at each stage, various evaluation measures, such as accuracy, balanced accuracy, AUC, f1-score, kappa statistics, mean absolute error, root mean squared error, and index of balanced accuracy, were used. Most of these measures are estimated through the elements of the confusion matrix: true positive (TP) indicates correctly predicted positive class samples, true negative (TN) indicates correctly predicted negative class samples, false positive (FP) specifies incorrectly predicted positive samples, and false negative (FN) specifies incorrectly predicted negative class samples [59]. The percentage of properly predicted samples among all the data samples is the accuracy, which summarizes the classification performance, and the formula is shown in Eq. (4).

$$Accuracy = \frac{(TP + TN)}{(TP + FN + FP + TN)} \quad (4)$$

Balanced accuracy is a good evaluation measure for imbalanced classification. It is the arithmetic mean of the sensitivity $\left(\frac{TP}{TP+FN}\right)$ and specificity (or TNR) $\left(\frac{TN}{TN+FP}\right)$ and is defined as in Eq. (5) [60]:

$$BA = \frac{Sensitivity + Specificity}{2} \quad (5)$$

The area under the curve (AUC) provides a concise summary of the compromise that must be made between a predictive model's true-positive and false-positive rates. The area under the ROC curve was always less than 1, and a higher AUC indicated a more accurate classification capability for the model.

The F1-score (FS) is the harmonic mean of the precision and recall values, and is often used for an uneven class distribution. Precision is the number of true positive predictions made out of all positive predictions $\left(\frac{TP}{TP+FP}\right)$ and recall (or TPR) is the number of true positive

predictions made out of all positive predictions$\left(\frac{TP}{TP+FN}\right)$. This formula is defined in Eq. (6).

$$F1-Score=\frac{2\times(presicion\times recall)}{(precision+recall)} \quad (6)$$

Cohen's kappa statistics (KS), mean absolute error (MAE), and root mean squared error (RMSE) are other common metrics used to assess classification models [23]. Cohen's kappa represents agreement between raters [61]. It is defined in Eq. (7), where P_o and P_c represent the fractions of actual agreement and agreement expected by chance, respectively.

$$KS=\frac{P_o-P_c}{1-P_c} \quad (7)$$

The MAE is the mean of the absolute values of the prediction errors for each occurrence in the test set. Prediction errors are calculated by subtracting the instance's actual value (y_i) from the predicted value $\lambda(x_i)$ of instancex_i. RSME is the square of the absolute values of the prediction errors for the test set instances. The square of the difference between the actual and expected values of the instance was used to assess the prediction errors. The formulas are presented in Eq. (8) and Eq. (9), where n is the total number of instances.

$$MAE=\frac{\sum_{i=1}^{n}abs(y_i-\lambda(x_i))}{n} \quad (8)$$

$$RMSE=\frac{\sum_{i=1}^{n}(y_i-\lambda(x_i))^2}{n} \quad (9)$$

The index of balanced accuracy (IBA) combines class precision with overall precision to enhance the geometric mean (GM) [62], and is defined as in Eq. (10).

$$\mathrm{IBA}_{\gamma}(GM)=(1+\gamma\times Dom)\times(GM)$$

Where $GM=\sqrt{TPR\times TNR}$, Dom = TPR -TNR denotes dominance, $\gamma\geq 0$ denotes the weight factor of Dom where it is taken as 0.05, and G-mean measures the overall precision.

5. Result Analysis

In the proposed study, eight existing classifiers (RF, NB, LR, LL, AB, KNN, ANN, and DT) are used to assess the proposed model at each stage. Moreover, seven evaluation assessments (ACC, KS, MAE, RMAE, FS, BA, and AUC) are commonly used to evaluate the methods, and $\mathrm{IBA}_{\gamma}(GM)$ is employed to assess the various oversampling methods in the phase of effectively balancing the class distribution. Python Version 3.7 was utilised in the study, which is installed on a personal computer with a 2.4 GHz Intel Pentium (R) processor. The configuration of the PC included 8 GB of RAM and was powered by Microsoft Windows 10 on a 64-bit processor.

To assess and compare the proposed methods at various stages, the baseline results were initially evaluated for various classifiers such as RF, NB, LR, LL, AB, KNN, ANN, and DT on the three credit scoring datasets (Australian, German, and Japanese), and were evaluated using various performance indicators such as ACC, KS, MAE, RMAE, FS, BA, and AUC. The results are presented in Table 3. The classifiers with the highest performance for the various indicators are shown in bold. It can be observed that the RF classifier offers better performance with most of the indicators.

5.1 Performance Evaluation of Feature Selection with Weighted Ensemble Filter (WEF)

The proposed feature selection technique with a weighted ensemble filter was assessed using various performance indicators on three credit-scoring datasets (Australian, German, and Japanese), adopting eight standard classifiers. The results are shown in Table 4, in which the bold values indicate the classifiers' improved performance compared to the baseline results after implementing feature selection. The results indicate that the weighted ensemble filter-based feature selection improves classifier performance, especially with the credit scoring application.

The proposed weighted ensemble filter-based feature selection, evaluated using various classifiers, was assessed by computing the critical difference (CD) between the models with respect to different performance metrics. CD diagrams for the classification models across the three datasets are shown in Figure 5. The figure indicates that the proposed feature selection method offers an improved performance, as reflected by the better ranks for RF (1.71), LR (2.99), and AB (2.35) on the Australian dataset. However, for the German dataset, the model shows an improved performance for LL (2.21), LR (2.71), and RF (2.79), which is similar to the results for the Japanese dataset, where LL (2.07), LR (2.93), and RF (3.29) perform well.

To compare the results of the proposed weighted ensemble filter-based feature selection method with the results of individual filters, such as gain ratio (GR), information gain (IG), correlation (CR), and symmetrical uncertainty (SU), the ACC and AUC values of these methods were measured, and the results are shown in Table 5 with their ranks at the end. The filters with the best performance with the eight classifiers on the Australian, German, and Japanese datasets are highlighted in boldface. From the results and the average ranks (ACC and AUC), it was found that the proposed weighted ensemble filter (average rank of 1.3) offers improved performance compared to other individual filters such as GR (2.8), IG (2.9), CR (3.4), and SU (2.8). Based on the computed accuracy and AUC values, the performance of the various feature selection methods was assessed using average rankings. Figure 6 shows the average rankings of the five feature selection techniques (GR, IG, CR, SU, and WEF).

Table 3. Baseline results

Dataset	Model	ACC	KS	MAE	RMAE	FS	BA	AUC
Australian	RF	**0.874**	**0.745**	0.202	**0.309**	**0.885**	**0.872**	**0.934**
	NB	0.772	0.524	0.226	0.439	0.818	0.796	0.893
	LR	0.870	0.737	0.192	0.315	0.880	0.868	0.929
	LL	0.859	0.718	0.206	0.319	0.869	0.858	0.924
	AB	0.862	0.722	0.201	0.320	0.875	0.860	0.930
	KNN	0.848	0.693	0.189	0.340	0.861	0.846	0.902
	ANN	0.849	0.694	**0.170**	0.366	0.866	0.849	0.895
	DT	0.852	0.700	0.182	0.352	0.869	0.851	0.876
German	RF	**0.764**	**0.370**	0.337	**0.405**	**0.844**	**0.729**	**0.787**
	NB	0.754	0.381	0.294	0.420	0.831	0.705	**0.787**
	LR	0.752	0.375	0.310	0.409	0.830	0.703	0.785
	LL	0.759	0.392	0.313	0.404	0.835	0.712	0.792
	AB	0.705	0.278	0.312	0.480	0.794	0.644	0.712
	KNN	0.742	0.317	0.320	0.427	0.829	0.691	0.727
	ANN	0.728	0.340	**0.279**	0.486	0.808	0.674	0.734
	DT	0.707	0.250	0.346	0.479	0.801	0.639	0.641
Japanese	RF	**0.861**	**0.730**	0.229	0.322	**0.881**	**0.866**	**0.926**
	NB	0.777	0.534	0.223	**0.436**	0.821	0.800	0.896
	LR	0.852	0.702	0.195	0.333	0.864	0.850	0.904
	LL	0.849	0.698	0.213	0.325	0.858	0.848	0.918
	AB	0.846	0.689	0.206	0.321	0.861	0.844	0.929
	KNN	0.812	0.618	0.189	0.433	0.832	0.810	0.808
	ANN	0.830	0.656	0.180	0.387	0.849	0.829	0.901
	DT	**0.861**	0.718	**0.192**	0.331	0.875	0.859	0.887

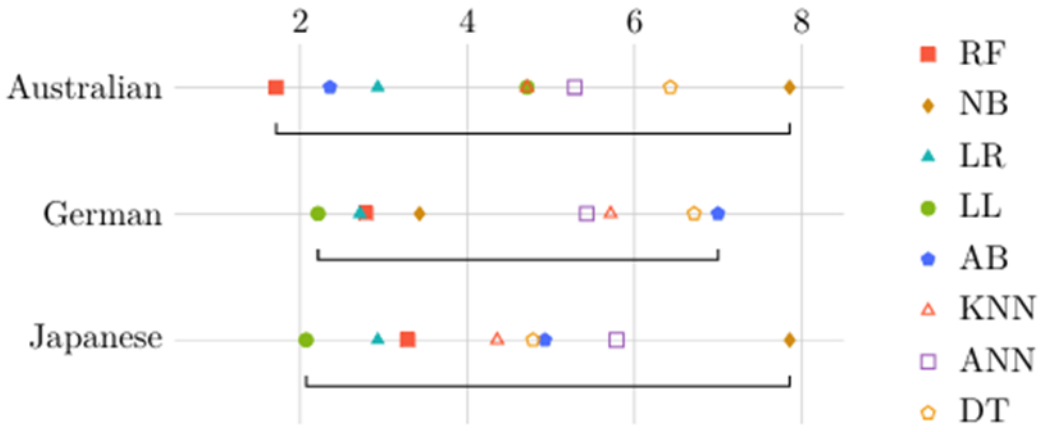

Figure 5. CD diagram for WEF feature selection using different classifiers

Table 4. Performance Evaluation Of The Proposed WEF Model

Dataset	Model	ACC	KS	MAE	RMAE	FS	BA	AUC
Australian	RF	**0.875**	**0.748**	**0.187**	**0.305**	**0.886**	**0.873**	**0.937**
	NB	**0.777**	**0.532**	**0.223**	**0.436**	**0.823**	**0.806**	**0.898**
	LR	**0.871**	**0.735**	**0.189**	**0.31**	**0.881**	**0.869**	**0.931**
	LL	**0.864**	**0.726**	**0.202**	**0.317**	**0.874**	**0.862**	**0.925**
	AB	**0.868**	**0.752**	**0.175**	**0.308**	**0.88**	**0.866**	**0.931**
	KNN	**0.864**	**0.725**	**0.185**	**0.335**	**0.876**	**0.862**	**0.909**
	ANN	**0.859**	**0.716**	**0.169**	**0.342**	**0.873**	**0.858**	**0.911**
	DT	**0.858**	**0.713**	**0.180**	**0.339**	**0.872**	**0.856**	**0.884**
German	RF	**0.778**	**0.378**	**0.334**	**0.401**	0.840	0.717	0.786
	NB	**0.756**	**0.385**	**0.292**	**0.417**	**0.833**	**0.708**	**0.791**
	LR	**0.761**	**0.398**	**0.308**	**0.405**	**0.836**	**0.715**	**0.792**
	LL	**0.762**	**0.394**	**0.310**	**0.403**	**0.838**	**0.717**	**0.795**
	AB	**0.728**	**0.284**	**0.305**	**0.466**	**0.796**	**0.647**	**0.726**
	KNN	0.733	0.295	**0.318**	0.433	0.822	0.677	**0.732**
	ANN	**0.748**	0.337	0.284	**0.482**	0.804	0.670	**0.736**
	DT	**0.732**	**0.323**	**0.339**	**0.458**	**0.816**	**0.675**	**0.671**
Japanese	RF	**0.881**	0.719	**0.221**	0.324	0.874	0.859	**0.934**
	NB	**0.781**	0.522	0.227	0.440	0.817	0.794	0.897
	LR	**0.868**	**0.734**	**0.192**	**0.322**	**0.879**	**0.866**	**0.921**
	LL	**0.870**	**0.738**	**0.198**	**0.317**	**0.879**	**0.868**	**0.927**
	AB	**0.846**	**0.689**	**0.206**	**0.321**	**0.861**	**0.844**	**0.929**
	KNN	**0.869**	**0.716**	**0.181**	**0.336**	**0.872**	**0.857**	**0.904**
	ANN	**0.846**	**0.670**	**0.176**	**0.375**	**0.850**	**0.834**	**0.905**
	DT	**0.865**	**0.721**	**0.19**	**0.327**	0.872	0.855	**0.891**

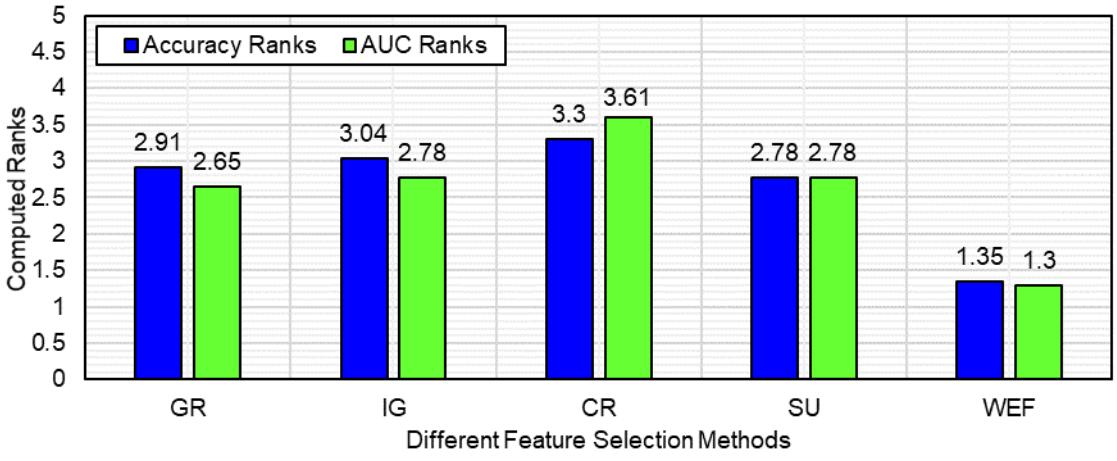

Figure 6. Average Ranks on different feature selection methods

Table 5. Comparison of the proposed WEF model with individual filters

Dataset	Model	Accuracy					AUC				
		GR	IG	CR	SU	P	GR	IG	CR	SU	P
Australian	RF	0.870	0.870	0.868	0.870	**0.875**	0.933	0.933	0.929	0.933	**0.937**
	NB	0.771	0.771	0.770	0.771	**0.777**	0.897	0.897	0.894	0.897	**0.898**
	LR	0.867	0.867	0.870	0.867	**0.871**	0.929	0.929	0.928	0.929	**0.931**
	LL	0.861	0.861	0.862	0.861	**0.864**	0.925	0.925	0.923	0.925	**0.925**
	AB	0.862	0.862	0.864	0.862	**0.868**	0.93	0.93	0.929	0.93	**0.931**
	KNN	0.807	0.807	0.855	0.807	**0.864**	0.803	0.803	0.904	0.803	**0.909**
	ANN	0.859	0.859	0.845	0.859	**0.859**	0.909	0.909	0.911	0.909	**0.911**
	DT	**0.862**	**0.862**	0.846	**0.862**	0.858	0.875	0.875	0.872	0.875	**0.884**
German	RF	0.769	0.756	0.771	0.769	**0.778**	0.781	0.783	0.781	**0.791**	0.786
	NB	0.750	0.748	0.754	0.754	**0.756**	0.79	0.788	0.789	0.786	**0.791**
	LR	0.761	0.759	0.754	0.758	**0.761**	0.788	0.785	0.789	0.784	**0.792**
	LL	0.754	0.755	0.756	0.753	**0.762**	0.789	0.789	0.791	0.785	**0.795**
	AB	0.723	0.711	**0.741**	0.705	0.728	**0.742**	0.713	0.722	0.716	0.726
	KNN	0.732	0.731	0.729	0.724	**0.733**	0.728	0.72	0.712	0.714	**0.732**
	ANN	0.715	0.716	0.713	0.737	**0.748**	0.716	0.739	0.719	**0.750**	0.736
	DT	0.725	0.722	0.728	0.729	**0.732**	0.665	**0.769**	0.665	0.688	0.671
Japanese	RF	0.858	0.824	0.881	0.862	**0.881**	0.923	0.883	0.826	0.923	**0.934**
	NB	0.778	0.778	0.762	0.778	**0.781**	0.896	0.896	0.887	0.896	**0.897**
	LR	0.859	0.859	0.860	0.859	**0.868**	0.903	0.905	0.92	0.905	**0.921**
	LL	0.849	0.865	0.855	0.865	**0.870**	0.919	0.924	0.918	0.924	**0.927**
	AB	0.846	0.846	0.845	0.846	**0.846**	0.929	0.929	**0.931**	0.929	0.929
	KNN	0.861	0.862	0.848	0.862	**0.869**	0.9	0.898	0.902	0.898	**0.904**
	ANN	**0.848**	0.824	0.843	0.824	0.846	0.899	0.883	0.893	0.883	**0.905**
	DT	0.851	0.864	0.857	0.864	**0.865**	0.888	0.875	0.839	0.875	**0.891**
Avg. Rank		*2.91*	*3.04*	*3.30*	*2.78*	*1.35*	*2.65*	*2.78*	*3.61*	*2.78*	*1.30*

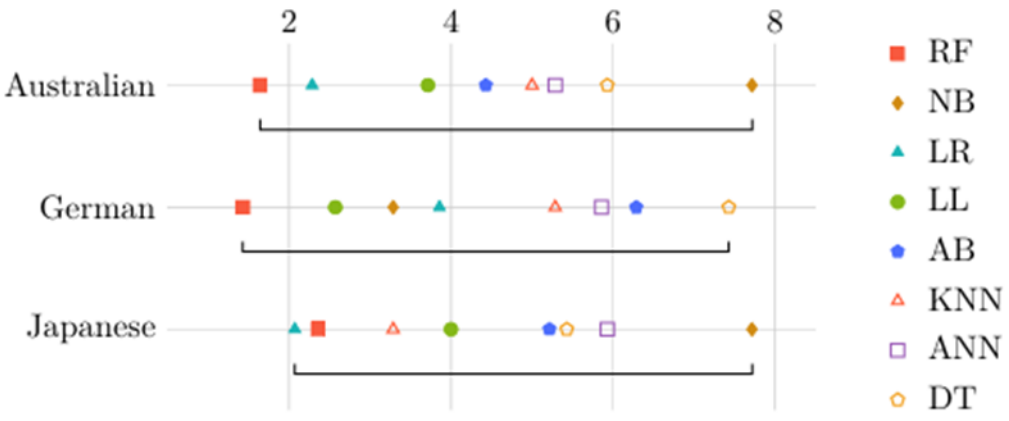

Figure 7. CD diagram for IBM-SMORE resampling using different classifiers

Table 6. Performance evaluation of the proposed minority oversampling technique

Dataset	Model	ACC	KS	MAE	RMAE	FS	BA	AUC
Australian	RF	**0.891**	**0.748**	**0.175**	**0.296**	**0.892**	**0.891**	**0.947**
	NB	**0.801**	**0.600**	**0.201**	**0.405**	0.823	**0.815**	**0.911**
	LR	**0.884**	**0.759**	**0.183**	**0.305**	**0.884**	**0.885**	**0.938**
	LL	**0.878**	**0.756**	**0.190**	**0.308**	**0.876**	**0.879**	**0.937**
	AB	**0.878**	**0.756**	0.187	0.310	0.879	**0.878**	0.878
	KNN	**0.866**	**0.732**	**0.181**	**0.330**	0.866	**0.866**	**0.915**
	ANN	**0.863**	**0.718**	0.172	**0.331**	0.866	**0.863**	**0.915**
	DT	0.857	**0.719**	**0.175**	**0.326**	0.861	**0.857**	**0.908**
German	RF	**0.814**	**0.628**	**0.283**	**0.361**	0.818	**0.814**	**0.896**
	NB	**0.788**	**0.576**	**0.279**	**0.395**	0.789	**0.788**	**0.851**
	LR	**0.783**	**0.566**	**0.288**	**0.388**	0.781	**0.784**	**0.861**
	LL	**0.793**	**0.586**	**0.298**	**0.387**	0.790	**0.794**	**0.863**
	AB	**0.770**	**0.542**	0.333	**0.408**	0.763	**0.773**	**0.832**
	KNN	**0.769**	**0.539**	**0.275**	**0.407**	0.752	**0.778**	**0.849**
	ANN	**0.765**	**0.531**	**0.239**	**0.456**	0.768	**0.765**	**0.842**
	DT	**0.763**	**0.526**	**0.312**	**0.433**	0.767	**0.763**	**0.77**
Japanese	RF	**0.883**	**0.75**	**0.207**	**0.31**	**0.885**	**0.883**	**0.939**
	NB	**0.788**	**0.546**	**0.202**	**0.412**	0.816	**0.809**	**0.901**
	LR	**0.880**	**0.761**	**0.181**	**0.310**	0.879	**0.881**	**0.934**
	LL	**0.874**	**0.747**	**0.193**	**0.311**	0.872	**0.875**	**0.932**
	AB	**0.862**	**0.723**	**0.195**	0.313	0.859	**0.863**	0.935
	KNN	**0.876**	**0.752**	**0.176**	**0.320**	**0.876**	**0.876**	**0.919**
	ANN	**0.856**	**0.712**	**0.157**	**0.347**	**0.859**	**0.856**	**0.909**
	DT	**0.868**	**0.736**	**0.188**	0.339	0.871	**0.868**	0.883

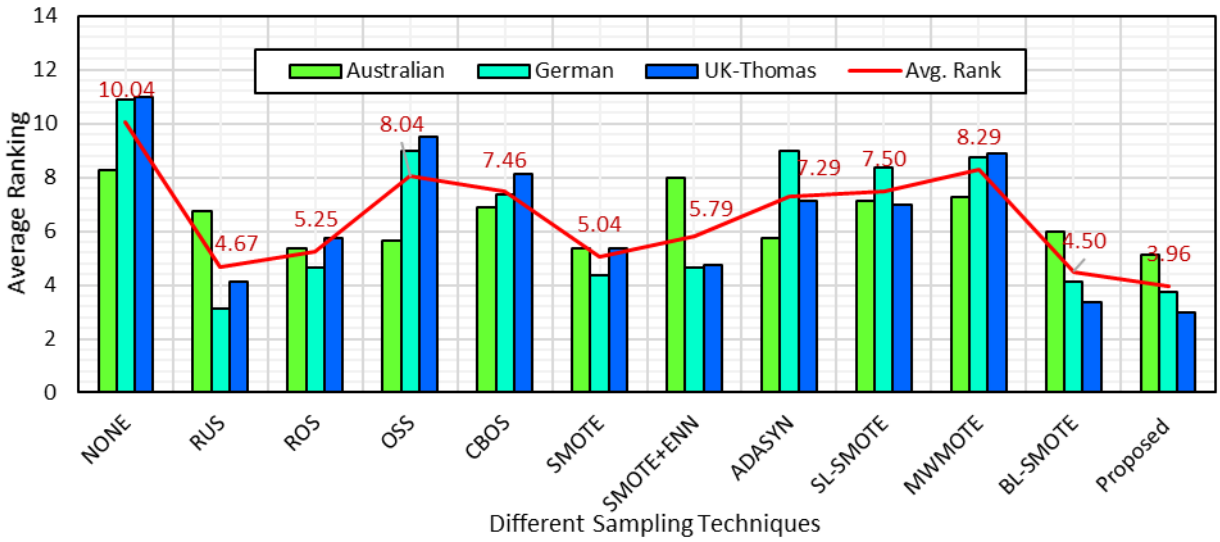

Figure 8. Average Ranks of Resampling Methods on IBA0.05(GM)

Table 7. Comparison of proposed minority oversampling technique

Dataset	Model	None	Rus	Ros	Oss	Cbos	Smote	Smote+ Enn	Adasyn	Sl-Smote	Mwmote	Bl-Smote	Proposed
Australian	RF	0.865	0.868	0.868	0.872	0.866	0.867	0.863	0.863	0.864	0.867	0.865	**0.873**
	NB	0.794	0.820	0.819	0.828	0.809	**0.853**	0.808	0.836	0.805	0.823	0.843	0.843
	LR	0.857	0.857	0.858	0.849	0.861	0.860	0.855	**0.865**	**0.865**	0.859	0.857	0.859
	LL	0.861	0.874	0.875	0.873	0.873	0.872	0.871	0.868	0.875	0.839	**0.879**	0.874
	AB	**0.870**	0.866	0.869	0.865	0.864	0.854	0.856	0.868	0.844	0.865	0.865	0.861
	KNN	0.843	0.852	0.861	0.861	0.860	0.864	0.863	**0.868**	0.863	0.858	0.863	0.863
	ANN	0.827	0.822	0.823	0.828	0.830	0.823	**0.850**	0.807	0.825	0.837	0.831	0.833
	DT	0.860	0.864	0.860	**0.868**	0.858	0.865	0.858	0.862	0.853	0.843	0.852	0.853
German	RF	0.607	0.687	0.652	0.645	0.640	0.724	0.694	0.643	0.644	0.634	0.715	**0.732**
	NB	0.636	0.703	**0.709**	0.677	0.691	0.675	0.685	0.647	0.686	0.680	0.691	0.689
	LR	0.623	0.712	0.713	0.663	0.705	0.697	0.690	0.706	0.709	0.705	0.722	**0.725**
	LL	0.614	**0.725**	0.669	0.618	0.655	0.649	0.658	0.609	0.588	0.615	0.653	0.655
	AB	0.622	0.717	0.708	0.669	0.700	**0.722**	0.705	0.687	0.693	0.644	0.695	0.698
	KNN	0.649	0.725	0.688	0.526	0.666	**0.726**	0.679	0.611	0.625	0.558	0.671	0.669
	ANN	0.612	0.644	0.616	0.628	0.603	0.668	0.666	0.635	0.643	0.648	**0.690**	0.681
	DT	0.605	0.648	0.611	0.625	0.602	0.663	**0.670**	0.612	0.582	0.622	0.645	0.647
UK-Thomas	RF	0.420	**0.590**	0.506	0.489	0.494	0.579	0.558	0.528	0.510	0.485	0.579	0.577
	NB	0.546	0.537	0.546	**0.569**	0.519	0.523	0.544	0.530	0.536	0.545	0.555	0.563
	LR	0.291	0.586	0.601	0.358	0.575	0.515	0.594	**0.617**	0.614	0.604	0.582	0.582
	LL	0.369	0.585	0.549	0.490	0.548	**0.587**	0.560	0.531	0.543	0.484	0.573	0.579
	AB	0.267	0.562	0.574	0.356	0.535	0.538	0.586	0.568	0.564	0.513	0.572	**0.589**
	KNN	0.215	0.581	0.527	0.236	0.515	**0.583**	0.566	0.487	0.504	0.419	**0.583**	0.582
	ANN	0.439	0.565	0.560	0.440	0.554	0.563	0.578	0.553	0.551	0.511	0.582	**0.587**
	DT	0.317	**0.563**	0.526	0.426	0.552	0.553	0.544	0.502	0.537	0.429	0.561	0.557
Avg. Rank		*10.042*	*4.667*	*5.250*	*8.042*	*7.458*	*5.042*	*5.792*	*7.292*	*7.500*	*8.292*	*4.500*	*3.958*

Table 8. Evaluation of the proposed NEC model

Dataset	Model	ACC	KS	MAE	RMAE	FS	BA	AUC
Australian	PS (66-34)	0.918	0.770	**0.125**	**0.257**	0.921	0.918	0.949
	5-fold	0.916	**0.764**	0.157	0.268	0.916	0.916	0.942
	10-fold	**0.924**	0.753	0.159	0.298	**0.925**	**0.924**	**0.967**
	Average	*0.918*	*0.762*	*0.147*	*0.274*	*0.919*	*0.918*	*0.948*
German	PS (66-34)	**0.828**	0.632	0.277	0.360	**0.840**	**0.828**	**0.902**
	5-fold	0.819	0.632	0.262	0.358	0.821	0.819	0.897
	10-fold	0.823	**0.636**	**0.253**	**0.357**	0.825	0.823	0.901
	Average	*0.823*	*0.633*	*0.264*	*0.358*	*0.829*	*0.823*	*0.900*
Japanese	PS (66-34)	**0.886**	0.772	0.196	0.302	**0.893**	**0.886**	0.935
	5-fold	0.885	0.771	0.197	0.304	0.886	**0.886**	0.933
	10-fold	0.879	**0.788**	**0.186**	**0.298**	0.879	0.879	**0.937**
	Average	*0.884*	*0.777*	*0.193*	*0.301*	*0.886*	*0.884*	*0.936*

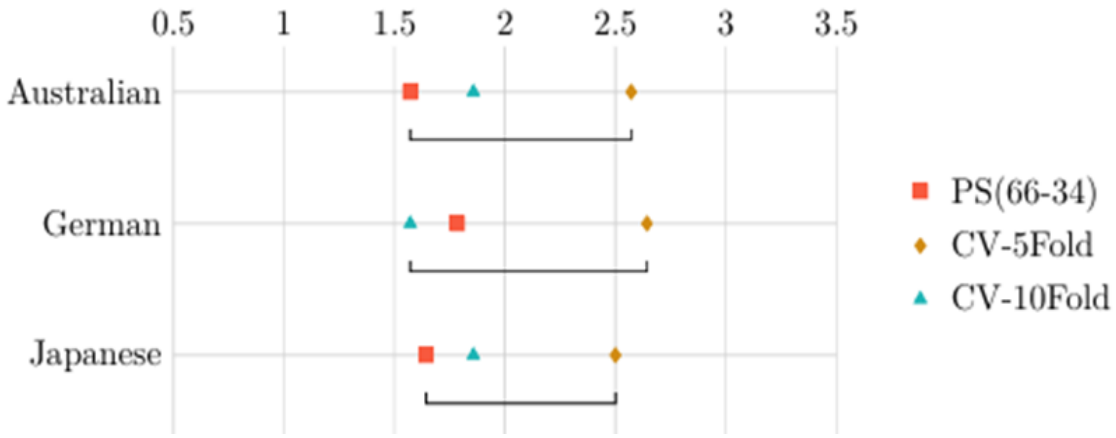

Figure 9. CD diagram for NEC model using different datasets

Table 9. Performance comparison of the proposed NEC model

Classifiers	ACC	AUC	BA	FM	KS	MAE	RSME
RF	2.67	1.67	2.67	2.33	3.67	5.67	2.00
NB	7.33	6.67	7.33	7.33	7.33	7.00	7.67
LR	3.00	3.67	3.00	3.33	2.67	5.00	3.00
LL	4.00	4.00	4.00	4.33	3.33	7.00	3.67
AB	5.67	6.67	6.33	6.33	5.33	7.67	5.67
KNN	5.33	5.67	5.00	6.33	5.33	3.33	6.33
ANN	7.67	6.33	7.67	7.00	8.00	1.33	8.33
DT	7.67	8.67	7.67	7.00	7.33	5.67	7.00
Proposed	1.33	1.33	1.33	1.00	2.00	2.33	1.33
Statistics of the Friedman Test	16.311	18.089	18.133	17.333	16.644	16.733	21.067
p-value	0.038	0.021	0.020	0.027	0.048	0.046	0.007

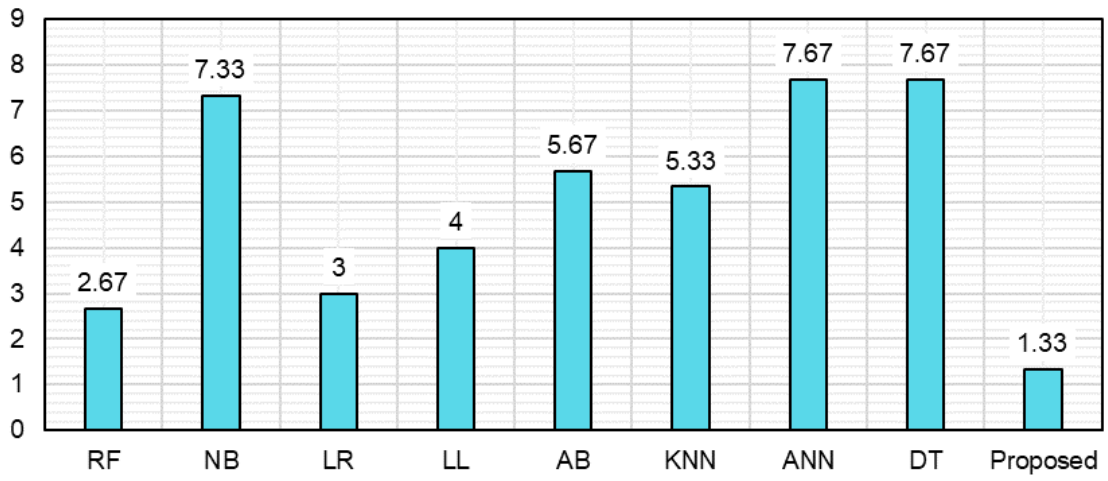

Figure 10. Average Ranks of Classifiers on Seven Evaluation Metrics

A better performance is indicated by lower values. While CR typically demonstrated the least effectiveness, with the highest average ranks (accuracy rank: 3.3, AUC rank: 3.61), WEF consistently produced the best results, ranking first across both measures (accuracy rank: 1.35, AUC rank: 1.3). This is followed by GR and SU, which have the second-best ranks.

5.2 Performance Evaluation of Improved Borderline SMOTE for Minority Oversampling (IBL-SMOTE)

On three credit scoring datasets—Australian, German, and Japanese—adopting eight conventional classifiers, the proposed improved borderline SMOTE for minority oversampling was evaluated using a variety of performance measures. The findings are shown in Table 6, where bold values indicate classifier performance that increased as a consequence of using the proposed resampling approaches. The findings suggest that classifier performance, particularly with the credit scoring application, is enhanced by the proposed borderline SMOTE for minority oversampling.

The proposed IBL-SMOTE-based resampling method, evaluated across various classifiers, was assessed by computing the critical difference between these models with respect to different performance metrics. The CD diagrams for the classification models across the three datasets are shown in Fig. 7. The figure indicates that the proposed resampling method offers an improved performance, as reflected by the better ranks for RF (1.64), LR (2.29), and LL (3.71) on the Australian dataset. Similarly, for the German dataset, the model exhibited improved performance for RF (1.43), LL (2.57), NB (3.29), and LR (3.86). However, on the Japanese dataset, the performance is reflected in the rankings of LR (2.07), RF (2.36), and KNN (3.29).

To compare the results of the proposed improved borderline SMOTE for minority oversampling techniques with various existing resampling techniques, such as RUS, ROS, OSS, CBOS, SMOTE, SMOTE+ENN, ADASYN, SL-SMOTE, MWMOTE, and BL-SMOTE, the evaluation metric $IBA_{0.05}(GM)$ was adopted. The results obtained for the Australian, German, and UK-Thomas credit datasets using the eight standard classifiers are shown in Table 7, in which the last row indicates the average ranks obtained after the analysis. Moreover, resampling techniques with a higher $IBA_{0.05}(GM)$ value are highlighted in boldface. The evaluation of the results indicated that the proposed modified borderline SMOTE for minority oversampling outperformed other prevailing resampling strategies in many cases.

Based on the IBA0.05(GM) values computed in Table 7, the performance of various sampling methods was assessed using average rankings. Figure 8 displays the average rankings of the 12 resampling techniques (NONE, RUS, ROS, OSS, CBOS, SMOTE, SMOTE + ENN, ADASYN, SL-SMOTE, MWMOTE, BL-SMOTE, and Proposed). For the Australian dataset, the proposed model (5.125), ROS (5.375), and SMOTE (5.375) exhibited better performance. In contrast, BL-SMOTE (German: 4.125; UK-Thomas: 3.375) and RUS (German: 3.125; UK-Thomas: 4.125) demonstrated improved performance, similar to the proposed model (German: 3.75; UK-Thomas: 3.0). Lower rank values indicate better performance. The Proposed approach consistently outperformed others in credit scoring tasks across different datasets, achieving the best average rank (3.96), followed by BL-SMOTE (4.50) and RUS (4.67).

5.3 Evaluation of Nested Ensemble Classification (NEC) Model

To assess the performance of the proposed nested ensemble classification model, the model was assessed with various options such as percentage split (66-34), 5-fold and 10-fold cross-validation, as well as the average of these results on the three credit scoring datasets, Australian, German, and Japanese.

The model was evaluated using various performance indicators, and the results are presented in Table 8. In the table, bold values indicate the best performance with various test options.

To compare the performance of the proposed nested ensemble classification model with the other eight existing classifiers, the results of the proposed model with 10-fold cross-validation from Table 8 were compared with the results of the classifiers shown in Table 6. The average ranks of the classifiers were computed for each performance indicator, and the results are listed in Table 9. From the obtained results, it is clear that the proposed nested ensemble classifier outperformed other existing standard classifiers, including RF and LR, with an average rank of 1.33.

Moreover, the average ranks of the classifiers across the seven performance metrics revealed that the proposed model consistently achieved the highest performance, with the lowest average rank of 1.33, indicating robust effectiveness across all evaluation measures. In addition, RF followed with an average rank of 2.67, maintaining a strong performance, particularly in the AUC and RMSE. LR also performed competitively, with a rank of 3.00, indicating balanced and reliable results across most metrics. Other models, such as LL and KNN, show moderate performance, while NB, ANN, and DT rank lower, reflecting relatively weaker consistency and performance.

Additionally, the statistical significance of the performance variations among classifiers across various parameters and datasets was evaluated using the Friedman test (See Table 9). This non-parametric test was selected because it is well suited for comparing

different methods over repeated measures and does not presume that the data distribution is normal. Metrics such as ACC (p = 0.038), AUC (p = 0.021), and RMSE (p = 0.007) yielded p-values ranging from 0.007 to 0.048, indicating statistically significant differences between the classifiers. The proposed model consistently achieved the lowest average rank (1.33), outperforming established methods, such as Logistic Regression (3.00) and Random Forest (2.67). These results not only demonstrate statistical significance, but also suggest practical relevance, highlighting the robustness and reliability of the proposed model across various evaluation metrics.

5.4 Computational Complexity Analysis

The computational complexity of these three phases is discussed in this section. In the proposed feature selection method, the number of filters used, size of the training dataset, and dimensionality of the features all significantly affect the computational cost of the WEF process. Multiple filters (COR, GR, IG, and SU) were applied to compute feature importance scores (FIS) for each bootstrap sample. Each filter typically has a complexity of $O(n \times d)$ or higher, where n is the number of instances, and d is the number of features.

In the second phase, the k-nearest neighbor (k-NN) search for every minority sample dominates the complexity of IBL-SMOTE. In a dataset of size n, finding neighbors using brute-force k-NN requires $O(n \times d)$ per query, resulting in $O(m \times n \times d)$ for m minority samples. The complexity is further increased by the generation of synthetic samples and the subsequent k-NN classification verification. Although this method is more computationally expensive than simple oversampling, it improves efficiency over naive SMOTE by focusing on minority samples that are in “danger.” Runtime can be reduced using approximate methods or efficient k-NN search algorithms (such as KD trees).

In the NEC model, several classifiers were trained concurrently in both bagging and stacking ensembles. The RF and NB classifiers are used in bagging; RF’s computational complexity of RF is approximately $O(T \times m \times \log n)$, where T is the number of trees, m is the number of features considered per split, and n is the number of samples. NB is computationally lightweight with a complexity of $O(n \times d)$, where d is the number of features. Owing to their linear nature, the multinomial LR and linear LR base classifiers used in stacking have a complexity of $O(n \times d)$, whereas RF as a meta-classifier has the same complexity as RF $(O(T \times m \times \log n))$. Combining predictions by averaging probabilities adds negligible overhead $O(n)$. The NEC model benefits from parallel execution and scalable base learners; however, its overall complexity remains high owing to the training of multiple classifiers. This trade-off in the computational cost results in improved accuracy.

5.5 Performance Comparison of Proposed Model with Benchmark Ensemble Models

The approximate performance of the proposed model is compared with various existing benchmark ensemble models with different performance indicators, such as ACC, AUC, BA, and FS. The values obtained for Australian, German, and Japanese are shown in Table 10. Better performance indications of the proposed model are highlighted in bold. The symbol “-” indicates that the values were missing in the respective work. From the results, it is clear that the proposed model offers a better performance across various metrics across all datasets. Moreover, this model outperformed other existing models with the German and Australian credit datasets.

The proposed model exhibits superior performance, specifically with highly imbalanced datasets, effectively addressing the challenges posed by unequal class distributions and enhancing the predictive accuracy for minority class predictions in credit scoring applications.

6. Statistical Analysis And Interpretation

A comparison of the proposed weighted ensemble filter-based feature selection, improved borderline SMOTE for minority oversampling, and nested ensemble classification model with various existing methods was statistically analyzed. A non-parametric statistical technique called the Friedman test was used to rank the various models according to their overall performance across the datasets used in the analysis. Thus, at the feature selection stage, the results of each filter on the two performance indicators (ACC and AUC) were assessed and represented by ranks, for which the average ranks were computed [63]. In addition, 12 resampling methods were assessed, their results were ranked, and the average ranks were assessed using the $IBA_{0.05}$(GM) metric. Moreover, nine classifiers were assessed using seven performance indicators (ACC, AUC, BA, FM, KS, MAE, and RSME), for which their performance was evaluated with individual and average ranks. The lower the ranking, the higher the performance.

To verify whether the performance of the various strategies differed significantly, the Friedman test was applied with the null hypothesis that the methods under comparison had equal performance by setting the significance level $\alpha = 0.05$. The χ^2 distribution was applied, and the results are presented in Table 11. In general, if the test value from the Friedman test is greater than the chi-squared distribution value and if the p-value is less than the alpha, then the null hypothesis can be rejected, which signifies that the performance of the methods is significantly different. Thus, from the results, it can be seen that all test values are greater than

Table 10. Performance comparison of the proposed model with benchmark ensemble models

Dataset	Benchmark Models	ACC	AUC	BA	FS
Australian	Multiple classifier systems (MCS) [53]	0.87980	0.94040	-	-
	Continuous recursive-rule extraction [55]	0.88400	0.88000	-	-
	Extended BalanceCascade approach [18]	-	0.93404	-	0.85020
	Information Gain Directed Feature Selection algorithm (IGDFS) [54]	0.90751	0.91278	-	-
	Novel ensemble model [56]	-	0.93880	-	-
	Multi-stage hybrid model [20]	0.87540	-	0.93700	-
	Multi-stage ensemble model [2]	0.92361	0.96656	0.92113	0.91803
	Proposed Ensemble	**0.92430**	**0.96782**	0.92438	**0.92206**
German	Multiple classifier systems (MCS) [53]	0.77720	0.80230	-	-
	Continuous recursive-rule extraction [55]	0.79000	0.75700	-	-
	Extended BalanceCascade approach [18]	-	0.80021	-	0.84439
	Information Gain Directed Feature Selection algorithm (IGDFS) [54]	0.82800	0.76680	-	-
	Novel ensemble model [56]	-	0.81020	-	-
	Multi-stage hybrid model [20]	0.76820	0.80290	-	-
	Multi-stage ensemble model [2]	0.79500	0.83122	0.73443	0.85091
	Proposed Ensemble	**0.83163**	**0.90101**	**0.83158**	0.83393
Japanese	Multiple classifier systems (MCS) [53]	0.87880	0.93280	-	-
	Extended BalanceCascade approach [18]	-	0.93058	-	0.87004
	Multi-stage hybrid model [20]	0.87200	0.93870	-	-
	Multi-stage ensemble model [2]	0.93163	0.96959	0.93226	0.93451
	Proposed Ensemble	0.87883	**0.93781**	0.87906	0.87915

Table 11. Statistical analysis of the results

Phase	Metrics	Deg_Freedom	Test value	Distribution value	p-value	Hypothesis
5 Feature Selection	ACC	4	60.392	9.487	2.39E-12	Reject
	AUC	4	69.879	9.487	2.40E-14	Reject
12 Resampling Methods	$IBA_{0.05}$(GM)	11	67.423	19.675	3.76E-10	Reject
9 Classifiers	ACC	8	16.311	15.507	0.038	Reject
	AUC	8	18.089	15.507	0.021	Reject
	BA	8	18.133	15.507	0.020	Reject
	FM	8	17.333	15.507	0.027	Reject
	KS	8	16.644	15.507	0.048	Reject
	MAE	8	16.733	15.507	0.046	Reject
	RSME	8	21.067	15.507	0.007	Reject

Thus, from the results, it can be seen that all test values are greater than the distribution value, and $p<0.05$, indicating the rejection of the null hypothesis.

6.1. Study Limitations

Although the study showed improved performance, it had several limitations. First, the high computational complexity of the proposed method is one of its drawbacks, particularly when considering the numerous feature selection filters and k-NN searches in the enhanced borderline SMOTE phase. Scalability for particularly large datasets or high-dimensional feature spaces may be affected by this extended processing time. Second, while the Friedman test successfully identified significant overall differences among the compared methods, it did not reveal which specific method pairs differed significantly. To address this, future work will incorporate post-hoc tests, such as the Nemenyi test, to enable detailed pairwise comparisons and strengthen the evidence for model superiority.

Third, the computational cost, which includes training time and model size, is an important consideration when assessing model performance. Although this study focused primarily on accuracy and other predictive metrics, future research will explore computational trade-offs to balance model complexity with resource efficiency. Additionally, the current implementation of the model is based on structured datasets typically found in credit scoring domains. Its generalizability to other data types, such as unstructured or streaming data or to other real-time high-throughput scoring systems, remains to be tested. Fourth, interpretability is a growing concern in financial applications. While this study focuses on improving predictive capacity, future work should include interpretability frameworks such as SHAP or LIME to support transparent, explainable credit decisions for regulatory compliance. Finally, to improve model performance, resilience, and adaptability across a variety of datasets and dynamic contexts, future research should also investigate different generative and AI-based models, such as deep generative models and transformer architectures.

7. Conclusion

The credit scoring system has long been a cornerstone of the growth and stability of financial companies by defending against credit risk. Over the past few decades, credit scoring has become indispensable for financial institutions to identify defaulters with the aid of ML and AI. In this study, a novel multistage hybrid ensemble classification model is proposed, which integrates a weighted ensemble filter for feature selection, improved borderline SMOTE for minority oversampling, and a nested ensemble classification model to enhance the effectiveness of credit scoring applications. This model addresses key limitations of previous models, such as feature redundancy and class imbalance, by combining advanced feature selection and oversampling techniques with a robust ensemble learning approach. Compared to prior studies, which often used isolated methods for feature selection or class balancing, the proposed hybrid model achieved superior predictive accuracy by integrating these components within a unified framework.

The analysis was conducted on four credit scoring datasets and evaluated using eight performance metrics. The experimental findings and statistical analysis validated the superior performance of the proposed model compared to benchmark models, demonstrating significant improvements in prediction accuracy and reliability. For example, the model achieved an AUC of 0.968 and accuracy of 0.924 for the Australian dataset. Similar improvements were observed on the German (AUC: 0.901; ACC: 0.831) and Japanese datasets (AUC: 0.879; ACC: 0.937). Furthermore, the statistically significant test values confirmed consistent superiority across the datasets and evaluation metrics. These findings demonstrate the robustness and consistency of the proposed model across a range of parameters, indicating both its statistical significance and practical relevance.

The proposed model provides financial institutions with a more reliable and accurate tool for credit scoring, which not only reduces the risk of bad credit decisions, but also streamlines operational processes and mitigates costs related to manual evaluations and fraud prevention. The model theoretically advances the field by demonstrating how the problems of redundancy and class imbalance in credit data can be resolved through a seamless combination of nested ensemble learning, synthetic data augmentation, and ensemble-based feature filtering. This highlights the importance of optimizing both the modelling and data preprocessing stages of credit risk analytics.

Practically, this model can be integrated into automated decision-support systems in financial institutions' backends for real-world credit scoring. As the suggested framework is modular, financial technology platforms can incorporate it into their scoring pipelines. This includes using the nested ensemble classifier for reliable predictions, applying real-time oversampling where class imbalance is identified, and utilizing the feature selection phase to minimize computational overhead. When implemented, it can improve the calibration of applicant risk profiling, facilitate regulatory compliance with comprehensible results, and eventually lead to more objective, data-driven lending decisions.

Future studies will focus on assessing the proposed model using various credit scoring datasets

and real-world datasets. To assess the creditability of the proposed model, it must also be tested using other learning classifiers. Furthermore, the framework can be adapted to other domains, such as healthcare or insurance, where predicting outcomes from imbalanced datasets is crucial. Future studies could include diverse feature types, such as temporal and behavioral data, to improve accuracy. This adaptability enhances the model's applicability across various fields while maintaining high predictive performance and low complexity.

References

[1] W. Zhang, J. Wang, Credit risk contagion in complex companies network–Empirical research based on listed agricultural companies, Economic Analysis and Policy, 82, (2024) 938-953. https://doi.org/10.1016/j.eap.2024.04.025

[2] W. Zhang, D. Yang, S. Zhang, J.H. Ablanedo-Rosas, X. Wu, Y. Lou, A novel multi-stage ensemble model with enhanced outlier adaptation for credit scoring, Expert Systems with Applications, 165, (2021) 113872. https://doi.org/10.1016/j.eswa.2020.113872

[3] W.A. Addy, A.O. Ajayi-Nifise, B.G. Bello, S.T. Tula, O. Odeyemi, T. Falaiye, AI in credit scoring: A comprehensive review of models and predictive analytics, Global Journal of Engineering and Technology Advances, 18(2), (2024) 118-129. https://doi.org/10.30574/gjeta.2024.18.2.0029

[4] M. Naved, R. Kumar, S.S Saheb, Analyzing financial stability by predicting bankruptcy situations with machine learning, Journal of Artificial Intelligence and System Modelling, 1(3), (2024) 18-35. https://doi.org/10.22034/jaism.2024.457068.1039

[5] S. Abimannan, E.S.M. El-Alfy, Y.S. Chang, S. Hussain, S. Shukla, D. Satheesh, Ensemble multifeatured deep learning models and applications: A survey IEEE Access, 11, (2023) 107194-107217. https://doi.org/10.1109/ACCESS.2023.3320042

[6] M. Mahbobi, S. Kimiagari, M. Vasudevan, Credit risk classification: an integrated predictive accuracy algorithm using artificial and deep neural networks, Annals of Operations Research, 330(1), (2023) 609-637. https://doi.org/10.1007/s10479-021-04114-z

[7] Ghavidel, P. Pazos, Machine learning (ML) techniques to predict breast cancer in imbalanced datasets: a systematic review, Journal of Cancer Survivorship, 19(1), (2025) 270-294. https://doi.org/10.1007/s11764-023-01465-3

[8] S.K. Rath, M. Sahu, S.P. Das, J.J. Jena, C. Jena, B. Khan, A. Ali, P. Bokoro, Software reliability prediction using ensemble learning on selected features in imbalanced and balanced datasets: A review, Computer Systems Science and Engineering, 48(6), (2024) 1513-1536. https://doi.org/10.32604/csse.2024.057067

[9] S. Farhadpour, T.A. Warner, A.E. Maxwell, Selecting and interpreting multiclass loss and accuracy assessment metrics for classifications with class imbalance: Guidance and best practices, Remote Sensing, 16(3), (2024) 533. https://doi.org/10.3390/rs16030533

[10] P. Ramila Rajaleximi, M.S. Irfan Ahmed, A. Alenezi, Classification of imbalanced class distribution using random forest with multiple weight based majority voting for credit scoring, International Journal of Recent Technology and Engineering, 7(6S5), (2019) 517-526. https://www.ijrte.org/wp-content/uploads/papers/v7i6s5/F10910476S519.pdf

[11] K. Hemapriya, K. Valarmathi, Innovative framework for thyroid disease detection by leveraging hybrid AGTEO feature selection and GRU classification model, International Research Journal of Multidisciplinary Technovation, 6(3), (2024) 112-127. https://doi.org/10.54392/irjmt2439

[12] P. Kumar, U.L. Maneesh, G.M. Sanjay, Optimizing Loan Approval Decisions: Harnessing Ensemble Learning for Credit Scoring, In Proc. 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI),IEEE, Chennai, India, 1-4. https://doi.org/10.1109/ACCAI61061.2024.10602097

[13] S.K. Trivedi, A study on credit scoring modelling with different feature selection and machine learning approaches, Technology in Society, 63, (2020) 1-9, https://doi.org/10.1016/j.techsoc.2020.101413

[14] M.Z. Abedin, C. Guotai, P. Hajek, T. Zhang, Combining weighted SMOTE with ensemble learning for the class-imbalanced prediction of small business credit risk. Complex & Intelligent Systems, 9(4), (2023) 3559-3579. https://doi.org/10.1007/s40747-021-00614-4

[15] M.S. Irfan Ahmed, P. Ramila Rajaleximi, A detailed analysis on classification algorithms for imbalanced class distribution on credit score

datasets, Adalya Journal, 9(7), (2020) 244-251, 2020. https://doi.org/10.37896/aj9.7/023

[16] J. Xiao, Y. Wang, J. Chen, L. Xie, J. Huang, Impact of resampling methods and classification models on the imbalanced credit scoring problems, Information Sciences, 569, (2021) 508-526. https://doi.org/10.1016/j.ins.2021.05.029

[17] Z. Zhao, T. Cui, S. Ding, J. Li, A.G. Bellotti, Resampling techniques study on class imbalance problem in credit risk prediction, Mathematics, 12(5), (2024) 1-27. https://doi.org/10.3390/math12050701

[18] H. He, W. Zhang, S. Zhang, A novel ensemble method for credit scoring: Adaption of different imbalance ratios, Expert Systems with Applications, 98, (2018) 105-117. https://doi.org/10.1016/j.eswa.2018.01.012

[19] J. Abellán, J.G. Castellano, A comparative study on base classifiers in ensemble methods for credit scoring, Expert systems with applications, 73, (2017) 1-10. https://doi.org/10.1016/j.eswa.2016.12.020

[20] W. Zhang, H. He, S. Zhang, A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring, Expert Systems with Applications, 121, (2019) 221-232. https://doi.org/10.1016/j.eswa.2018.12.020

[21] Luo, A comparison analysis for credit scoring using bagging ensembles, Expert Systems, 39(2), (2022) 1-7. https://doi.org/10.1111/exsy.12297

[22] L. Zhou, K.K. Lai, Adaboosting neural networks for credit scoring, In 2009 Proc. International Symposium on Neural Networks, Springer Berlin Heidelberg, 875-884. https://doi.org/10.1007/978-3-642-01216-7_93

[23] I. Ahmed, Credit risk management using hybrid scoring strategy and ensemble learning, Thesis Dissertation, Bharathiar University, India, 2020. http://hdl.handle.net/10603/397840

[24] Wang, Z. Zhang, R. Bai, Y. Mao, A hybrid system with filter approach and multiple population genetic algorithm for feature selection in credit scoring, Journal of Computational and Applied Mathematics, 329, (2018) 307-321. https://doi.org/10.1016/j.cam.2017.04.036

[25] Theng, K.K. Bhoyar, Feature selection techniques for machine learning: a survey of more than two decades of research. Knowledge and Information Systems, 66(3), (2024) 1575-1637. https://doi.org/10.1007/s10115-023-02010-5

[26] S. Sathya Bama, M.S. Irfan Ahmed, A. Saravanan, Relevance re-ranking through proximity based term frequency model, In ICT Innovations 2016: Cognitive Functions and Next Generation ICT Systems, Springer International Publishing, 219-229. https://doi.org/10.1007/978-3-319-68855-8_22

[27] S. Sathya Bama, M.I. Ahmed, A. Saravanan, A mathematical approach for mining web content outliers using term frequency ranking, Indian Journal of Science and Technology, 8(14), (2015) 1-5. https://dx.doi.org/10.17485/ijst/2015/v8i14/55679

[28] X. Cui, Y. Li, J. Fan, T. Wang, A novel filter feature selection algorithm based on relief, Applied Intelligence, 52(5), (2022) 5063-5081. https://doi.org/10.1007/s10489-021-02659-x

[29] W. Bouaguel, G. Bel Mufti, M. Limam, Rank aggregation for filter feature selection in credit scoring, In Proc. 2013 International Conference on Mining Intelligence and Knowledge Exploration, Springer International Publishing, 7-15. https://doi.org/10.1007/978-3-319-03844-5_2

[30] F.N. Koutanaei, H. Sajedi, M. Khanbabaei, A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring, Journal of Retailing and Consumer Services, 27, (2015) 11-23, https://doi.org/10.1016/j.jretconser.2015.07.003

[31] Boughaci, A.A.S. Alkhawaldeh, Three local search-based methods for feature selection in credit scoring, Vietnam Journal of Computer Science, 5, (2018), 107-121. https://doi.org/10.1007/s40595-018-0107-y

[32] P. Rajaleximi, M. Ahmed, A. Alenezi, Feature selection using optimized multiple rank score model for credit scoring, International Journal of Intelligent Engineering and Systems, 12(2), (2019) 74-84.

[33] D. Tripathi, D.R. Edla, R. Cheruku, V. Kuppili, A novel hybrid credit-scoring model based on ensemble feature selection and multilayer ensemble classification, Computational Intelligence, 35(2), (2019) 371-394. https://doi.org/10.1111/coin.12200

[34] S.F. Crone, S. Finlay, Instance sampling in credit scoring: An empirical study of sample size and balancing, International Journal of

Forecasting, 28(1), (2012) 224-238. https://doi.org/10.1016/j.ijforecast.2011.07.006

[35] T. Jo, N. Japkowicz, Class imbalances versus small disjuncts, ACM Sigkdd Explorations Newsletter, 6(1), (2004) 40-49. https://doi.org/10.1145/1007730.1007737

[36] S.J. Bennehalli, S. Vakkund, A. Hegde, B. Bhowmik, Navigating data imbalances in credit risk management: A one-sided selection approach. In IEEE 2024 Control Instrumentation System Conference, IEEE, Manipal, India, 1-6. https://doi.org/10.1109/CISCON62171.2024.10696124

[37] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of artificial intelligence research, 16, (2002) 321-357. https://doi.org/10.1613/jair.953

[38] D.L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems, Man, and Cybernetics, 3, (1972), 408-421. https://doi.org/10.1109/TSMC.1972.4309137

[39] G.E. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD explorations newsletter, 6(1), (2004) 20-29. https://doi.org/10.1145/1007730.1007735

[40] H. He, B. Yang, E.A. Garcia, S.A. Li, Adaptive synthetic sampling approach for imbalanced learning, In Proc. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China.1322-1328.

[41] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, In Proc. 2009 Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Bangkok, Thailand, Springer Berlin Heidelberg, 475-482. https://doi.org/10.1007/978-3-642-01307-2_43

[42] S. Barua, M.M. Islam, X. Yao, K. Murase, MWMOTE--majority weighted minority oversampling technique for imbalanced data set learning, IEEE Transactions on knowledge and data engineering, IEEE, 26(2), (2012) 405-425. https://doi.org/10.1109/TKDE.2012.232

[43] Bao, Y. Wu, Z. Li, Y. Li, L. Liu, G. Chen, Effect improved for high-dimensional and unbalanced data anomaly detection model based on KNN-SMOTE-LSTM, Complexity, 2020(1), (2020) 9084704. https://doi.org/10.1155/2020/9084704

[44] Han, W.Y. Wang, B.H. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, In Proc. 2005 International Conference on Advances in Intelligent Computing, China, Springer Berlin Heidelberg, 878-887. https://doi.org/10.1007/11538059_91

[45] V. García, A.I. Marqués, J.S. Sánchez, Improving risk predictions by preprocessing imbalanced credit data, In Proc. 2012 International Conference on Neural Information Processing, Doha, Qatar, Springer Berlin Heidelberg, 68-75. https://doi.org/10.1007/978-3-642-34481-7_9

[46] C. Jiang, W. Lu, Z. Wang, Y. Ding, Benchmarking state-of-the-art imbalanced data learning approaches for credit scoring, Expert systems with applications, 213(B), (2023) 118878. https://doi.org/10.1016/j.eswa.2022.118878

[47] Y. Xia, C. Liu, B. Da, F. Xie, A novel heterogeneous ensemble credit-scoring model based on bstacking approach, Expert Systems with Applications, 93, (2018) 182-199. https://doi.org/10.1016/j.eswa.2017.10.022

[48] M. Abdoli, M. Akbari, J. Shahrabi, Bagging supervised autoencoder classifier for credit scoring, Expert Systems with Applications, 213, (2023) 118991. https://doi.org/10.1016/j.eswa.2022.118991

[49] R.E. Schapire, The strength of weak learnability, Machine learning, 5, (1990) 197-227. https://doi.org/10.1007/BF00116037

[50] D.H. Wolpert, Stacked generalization, Neural networks, 5(2), (1992) 241-259. https://doi.org/10.1016/S0893-6080(05)80023-1

[51] F.M. Talaat, A. Aljadani, M. Badawy, M. Elhosseini, Toward interpretable credit scoring: integrating explainable artificial intelligence with deep learning for credit card default prediction, Neural Computing and Applications, 36(9), (2024) 4847-4865. https://doi.org/10.1007/s00521-023-09232-2

[52] A.V. Chaudhari, P.A. Charate, Synthetic Financial Document Generation and Fraud Detection Using Generative AI and Explainable ML, Journal of Recent Trends in Computer Science and Engineering, 13(2), (2025) 45-59. https://doi.org/10.70589/JRTCSE.2025.13.2.6

[53] M. Ala'raj, M.F. Abbod, Classifiers consensus system approach for credit scoring, Knowledge-Based Systems, 104, (2016) 89-105. https://doi.org/10.1016/j.knosys.2016.04.013

[54] S. Jadhav, H. He, K. Jenkins, Information gain directed genetic algorithm wrapper feature selection for credit rating, Applied Soft Computing, 69, (2018) 541-553. https://doi.org/10.1016/j.asoc.2018.04.033

[55] Y. Hayashi, T. Oishi, High accuracy-priority rule extraction for reconciling accuracy and interpretability in credit scoring, New Generation Computing, 36(4), (2018) 393-418. https://doi.org/10.1007/s00354-018-0043-5

[56] F. Shen, X. Zhao, Z. Li, K. Li, Z. Meng, A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation, Physica A: Statistical Mechanics and its Applications, 526, (2019) 121073. https://doi.org/10.1016/j.physa.2019.121073

[57] J. Xiao, Y. Wang, J. Chen, L. Xie, J. Huang, Impact of resampling methods and classification models on the imbalanced credit scoring problems. Information Sciences, 569, (2021) 508-526. https://doi.org/10.1016/j.ins.2021.05.029

[58] S.S. Bama, A. Saravanan, Efficient classification using average weighted pattern score with attribute rank based feature selection, International Journal of Intelligent Systems and Applications, 11(7), (2019) 29-42. https://doi.org/10.5815/ijisa.2019.07.04

[59] M. Owusu-Adjei, J. Ben Hayfron-Acquah, T. Frimpong, G. Abdul-Salaam, Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems, PLOS Digital Health, 2(11), (2023) e0000290. https://doi.org/10.1371/journal.pdig.0000290

[60] D. Chicco, N. Tötsch, G. Jurman, The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation, BioData mining, 14(1), (2021) 13. https://doi.org/10.1186/s13040-021-00244-z

[61] M. Li, Q. Gao, T. Yu, Kappa statistic considerations in evaluating inter-rater reliability between two raters: which, when and context matters, BMC cancer, 23(1), (2023) 799. https://doi.org/10.1186/s12885-023-11325-z

[62] V. García, R.A. Mollineda, J.S. Sánchez, Index of balanced accuracy: A performance measure for skewed class distributions, In Proc. 2009 Iberian Conference on Pattern Recognition and Image Analysis, Portugal, Springer Berlin Heidelberg, (2009) 441-448. https://doi.org/10.1007/978-3-642-02172-5_57

[63] S. Lessmann, B. Baesens, H.V. Seow, L.C. Thomas, Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research, European Journal of Operational Research, 247(1), (2015) 124-136. https://doi.org/10.1016/j.ejor.2015.05.030

Acknowledgement

We would like to thank our organizations for providing support and opportunities to carry out this research work.

Authors Contribution Statement

P. Ramila Rajaleximi: Conceptualization, methodology, data collection, investigation, analysis, validation, writing - original draft. A. Saravanan: Conceptualization, methodology, investigation, analysis, validation, supervision, writing - original draft. B. SivaSakthi: Investigation, analysis, validation, writing - original draft. Leena Jaganathan: Investigation, analysis, validation, writing -original draft. S. Sathya Bama: Investigation, analysis, validation, visualization, writing - original draft, writing-review, and editing. All the authors read and approved the final version of the manuscript.

Funding

This research was conducted without the aid of any financial grants.

.

Competing Interests

The authors declare that there are no conflicts of interest

Data Availability

The data supporting the findings of this study are openly available in the UCI Machine Learning Repository at https://archive.ics.uci.edu/.

Has this article screened for similarity?

Yes

About the License