Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

An Investigation of Credit Card Default Prediction in the Imbalanced Datasets

Talha Mahboob Alam[1, *], Kamran Shaukat[2, 3,*], Ibrahim A. Hameed[4], Suhuai Luo[2], Muhammad Umer Sarwar[5], Shakir Shabir[1], Jiaming Li[6], Matloob Khushi[7]

[1] Department of Management Sciences, Lahore College for Women University, Lahore 54000, Pakistan
[2] School of Electrical Engineering and Computing, The University of Newcastle, Callaghan, NSW 2308, Australia
[3] Punjab University College of Information Technology, University of the Punjab, Lahore 54590, Pakistan
[4] Department of ICT and Natural Sciences, Norwegian University of Science and Technology, 7491 Trondheim, Norway
[5] Department of Computer Science, Government College University, Faisalabad. Pakistan
[6] Data61, Commonwealth Scientific and Industrial Research Organization, Australia
[7] School of Computer Science, The University of Sydney, NSW, Australia

Corresponding authors: Kamran Shaukat (e-mail: Kamran.shaukat@uon.edu.au), Talha Mahboob Alam (Talhamahboob95@gmail.com), Ibrahim A. Hameed (ibib@ntnu.no), Matloob Khushi (matloob.khushi@sydney.edu.au)

ABSTRACT Financial threats are displaying a trend about the credit risk of commercial banks as the incredible improvement in the financial industry has arisen. In this way, one of the biggest threats faces by commercial banks is the risk prediction of credit clients. Recent studies mostly focus on enhancing the classifier performance for credit card default prediction rather than an interpretable model. In classification problems, an imbalanced dataset is also crucial to improve the performance of the model because most of the cases lied in one class, and only a few examples are in other categories. Traditional statistical approaches are not suitable to deal with imbalanced data. In this study, a model is developed for credit default prediction by employing various credit-related datasets. There is often a significant difference between the minimum and maximum values in different features, so Min-Max normalization is used to scale the features within one range. Data level resampling techniques are employed to overcome the problem of the data imbalance. Various undersampling and oversampling methods are used to resolve the issue of class imbalance. Different machine learning models are also employed to obtain efficient results. We developed the hypothesis of whether developed models using different machine learning techniques are significantly the same or different and whether resampling techniques significantly improves the performance of the proposed models. One-way Analysis of Variance is a hypothesis-testing technique, used to test the significance of the results. The split method is utilized to validate the results in which data has split into training and test sets. The results on imbalanced datasets show the accuracy of 66.9% on Taiwan clients credit dataset, 70.7% on South German clients credit dataset, and 65% on Belgium clients credit dataset. Conversely, the results using our proposed methods significantly improve the accuracy of 89% on Taiwan clients credit dataset, 84.6% on South German clients credit dataset, and 87.1% on Belgium clients credit dataset. The results show that the performance of classifiers is better on the balanced dataset as compared to the imbalanced dataset. It is also observed that the performance of data oversampling techniques are better than undersampling techniques. Overall, the Gradient Boosted Decision Tree method performs better than other traditional machine learning classifiers. The Gradient Boosted Decision Tree method gives the best results while utilizing the K-means SMOTE oversampling method. Using one-way ANOVA, the null hypothesis was rejected by a p-value <0.001, hence confirming that the proposed model improved performance is statistical significance. The interpretable model is also deployed on the web to ease the different stakeholders. This model will help commercial banks, financial organizations, loan institutes, and other decision-makers to predict the loan defaulter earlier.

[*] TMA and KS contributed equally as first authors.

INDEX TERMS Machine learning, Imbalanced data, Customer credit risk, Credit card default model, Interpretable model, gradient boosted decision tree

I. INTRODUCTION

According to the Federal Reserve economic data, the default rate on credit loans across all commercial banks is at an all-time high for the past 66 months, and it is likely to continue to climb throughout 2020. The delinquency rate indicates the percentage of past-due loans within the borrower's entire loan portfolio. The climbing delinquencies will result in a significant amount of money lose from the lending institutions, such as commercial banks. Therefore, banks must have a risk prediction model and be able to classify the most relative characteristics that are indicative of people who have a higher probability of default on credit. In 2013, consumer spending encompassed approximately 69% of USA gross domestic product. Of the $3.098 trillion of outstanding consumer credit in the United States in the last quarter of 2013, they were revolving credit card for over 25% of it ($857.6 billion). A small increase in the accuracy of identifying high-risk loans could prevent losses of over $8 billion [1]. Because of the risks inherent in such a large portion of the economy, building models for consumer spending behaviors to limit risk exposures in this sector is becoming more critical. For this to be a viable option, the predictions need to be reasonably accurate.

A robust model is not only a useful tool for the lending institutions to decide on credit applications, but it can also help the clients to be aware of the behaviors that may damage their credit scores [2]. The primary motivation behind risk prediction is to utilize financial data, for example, business transactional data, exchange records and client transactions, and so forth to foresee the client's business performance or individual credit card data and to decrease loos and vulnerability. Several risk prediction models are based on statistical methods, including nearest neighbor, discriminant analysis, and logistic regression [3]. With the advancement of machine learning and artificial intelligence techniques, classification, and regression models were additionally being utilized to predict credit risk [4]. Credit risk here means the likelihood of a postponement in the reimbursement of the credit granted [5]. The goal of credit default prediction is to help financial institutions decide whether or not to lend to a client. The resulting test is usually a threshold value that allows the decision-makers to make the lending decision. The standard model depends on the financial ratios, income account, and data on the balance sheet [6, 7]. These ratios reveal their accessibility and regularization capabilities of prediction. Usually, these ratios are utilized to classify the credit default client from non-defaulted [8] because these parameters may enhance the performance of models. Furthermore, the accounting-based models control default prediction, but these models utilized limited features [6, 9], which leads to model ineffectiveness.

Traditional credit risk prediction techniques utilized a balanced dataset, but it is more typical to handle imbalanced datasets. There was less work done by exploring different resampling approaches for data imbalance issues for credit risk prediction [7]. Considering the Binary classification, when number instances are far less than other class, then class imbalance issue has arisen. The performance of classifiers is compromised when good borrowers and defaulters have an imbalanced distribution of classes because classifiers focused on the majority class and overlook the minority class. Different traditional statistical models, including regression, nearest neighbor, and multiple discriminant analysis, were not given significant results as compared to machine learning models. Different classification models based on machine learning have been applied for default prediction in previous literature [10]. Decision tree-based classification models have been extensively used for machine learning tasks because these models are easy to understand by humans, and also their implementation is straightforward [11]. The indicators or features related to predicting the credit default are still questionable and also alternatively changed in the past years. Hence, the traditional statistical were not able to solve the problem, and there is a dire need to build a machine learning model to predict the credit default effectively of the client [12].

Various datasets were used in previous studies like lending club dataset [13, 14], Chinese P2P lending company dataset [15], German credit dataset, Australian credit dataset, and Dataset of We.com [16], Chinese consumer finance company dataset [17], Six major USA financial institutions [2], and Major commercial USA bank dataset [18]. All these datasets have few limitations concerning different aspects. Few studies utilized a limited number of features [13, 16] and also employed a limited amount of data [16] for modeling purposes. Various researchers used a large number of features [2, 15, 17, 18] and also trained the model with a massive amount of data [2, 13, 18]. Furthermore, these studies were not given efficient results due to a high imbalance of data because they were not balanced the dataset [2, 18]. Current credit bureau analytics, such as credit scores, are based on slowly varying consumer characteristics. They are not adaptable to changes in client's behaviors and market conditions over time. Besides, the behavior of the market has not been consistent over the years to the features to predict the default are always debatable [19]. Limited work was also done to solve the problem of data imbalance by using few resampling techniques [20, 21], but results were not efficient. To the best of our knowledge, there was no work done on the default credit card client's dataset by employing various resampling techniques. Several studies reported that effective results were not obtained [11, 22-26] when imbalanced data has utilized.

Contributions: This research possesses various contributions in the domain of credit risk prediction.

1. First, multiple latest datasets have been used to build a machine learning model for credit risk prediction.
2. Second, the data imbalance problem has been explored by comparing the different resampling techniques and evaluate the performance that which the resampling technique has given effective results with a machine learning classifier.
3. Limited work was done on resampling techniques for data balancing in this domain because only a few resampling techniques were employed and also obtained less efficient results [2].
4. Lastly, the interpretable model is also deployed on the web to ease the different stakeholders. This model will help commercial banks, financial organizations, loan institutes, and other decision-makers to predict the credit defaulter earlier.

The paper has organized as follows. Section 1 covers the background theory related to the credit card default prediction and resampling techniques. Section 2 includes the existing techniques related to the credit card default prediction. Section 3 primarily contains the proposed solution as well as explained the datasets used in the study. Section 4 mainly contains the evaluation metrics used in the study. Section 5 discussed and analyzed the results obtained from the implementation stage. In section 6, a framework has also been designed for the credit card default prediction. Section 7 summarizes the research as a whole, restating the problem definition, challenge, and limitation of the study, and suggestion for future improvement.

2. LITERATURE REVIEW

The previous studies have been incorporated that deal with credit card default prediction using imbalanced data. Multiple combinations such as 'machine learning and credit card default prediction' and 'credit card default prediction and imbalanced data' have been used to retrieve the journal papers and conference proceedings. Three databases, namely IEEE Xplore, ScienceDirect, and SpringerLink, have targeted. In total, 400 articles were retrieved, and 150 duplicated items were removed. The title and abstract were screened to identify potential articles. The full texts of 100 studies were assessed to find the relevancy with the inclusion criteria. The articles that were related to the loan prediction through images, corporate default prediction, and credit card threats prediction have excluded. In total, 30 studies were finally selected for data utilization purposes. Figure 1 shows the process of paper selection. The previous review articles were also used in addition to these included papers to provide a comprehensive performance evaluation.

Yufei Xia et al. [13] proposed a credit scoring model to classify the healthy and loan default customers. They utilized the P2P lending dataset to build a model and also preprocessed the data due to noisy values. Advanced gradient boosting models and keyword clustering-based techniques were employed to test the results. They extracted

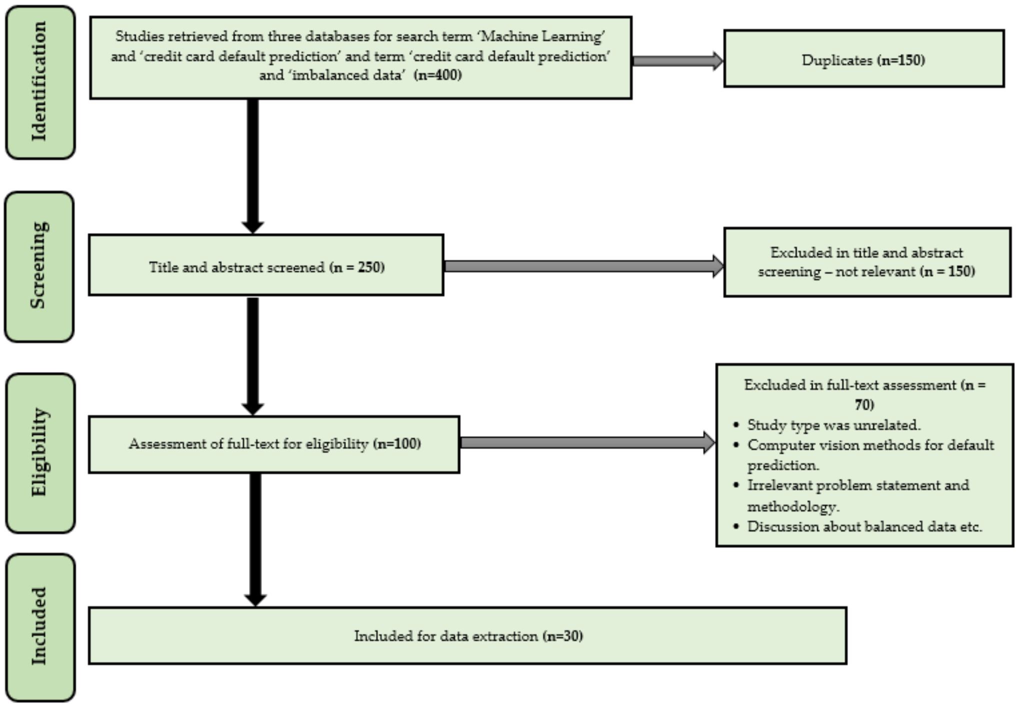

Identification
Studies retrieved from three databases for search term 'Machine Learning' and 'credit card default prediction' and term 'credit card default prediction' and 'imbalanced data' (n=400)
Duplicates (n=150)
Screening
Title and abstract screened (n = 250)
Excluded in title and abstract screening – not relevant (n = 150)
Eligibility
Assessment of full-text for eligibility (n=100)
Excluded in full-text assessment (n = 70)
Study type was unrelated.
Computer vision methods for default prediction.
Irrelevant problem statement and methodology.
Discussion about balanced data etc.
Included
Included for data extraction (n=30)

Figure 1. **Article Selection Process for related work**

dominant features to enhance the performance of classifiers. Their experiments indicated that the gradient boosting based Catboost model overtook other traditional models. Jing Zhou et al. [15] developed a decision tree-based model for customer default prediction in P2P lending. They employed different ensemble-based machine learning models for modeling purposes. The credit dataset contained 1138 features and 15000 instances of customers. Data preprocessing techniques were also utilized to deal with missing values and high scarcity. They also ranked the features, and less associated features were removed. The optimization of different Hyper-parameters was also done to improve the performance of classifiers. Their experiments showed promising results while using high-dimensional data to achieve desirable prediction. Leong and Jayabalan [22] investigated the different machine learning models to classify the default of credit card customers. The dataset used in their study was acquired from a bank in Taiwan to examine this task. Four machine learning algorithms were utilized in which neural networks were given the best results with an accuracy of 82%. Te-Wei Li et al. [17] developed the concept of transfer learning in which the learning was transferred from one dataset to another. The default risk prediction model was trained and also compared the results with traditional models. Shigeyuki Hamori [11] was done extensive work on analyzing payment data of defaulters and comparing the accuracy of four predictive machine learning methods; random forest, bagging, boosting alongside neural networks. The results gained by the boosting algorithm were best among the rest of the predictive machine learning methods. Yufei Xia et al. [16] developed an ensemble credit scoring model by combining the bagging and stacking model. Their model is different from traditional ensemble-based models based on trainable fuser, pool generation, and selection of base learners. German credit dataset, Australian credit dataset, the dataset of We.com, and Lending club dataset were utilized to measure the performance of the bstacking model. The performance of the model was estimated based on accuracy, the area under the curve (AUC), AUC-H measure, and Brier score. The results of their models outperformed traditional ensemble-based models. Maruf Pasha et al. [25] worked on a customer's default prediction and their predictive accuracy by utilizing specific data mining techniques. Six data mining techniques were used for modeling. The results described that the neural networks were the best method to generate predictions of the default credit cardholders. Che-hui Lien and Cheng Yun [26] proposed the model for default payments of the customers in Taiwan. Different data mining algorithms with the help of monetary related features were incorporated. Florentin Butaru et al. [2] utilized different machine learning techniques to predict the delinquency of credit card customers. The data was collected from six commercial banks, which contain the economic, credit bureau, and customer tradeline features. They observed that it not possible to build the generic model for all banks because the customer indicators are varied among banks. Their study concluded that delinquent accounts in all the banks are different so it was suggested that there is a dire need to build a generic model.

Only a few of them have talked about class imbalance but not realistically.

We proposed a machine learning model by analyzing the various credit default datasets. Since consumer credit models are relatively new in the space of machine learning, an overview of related articles is presented in Table 1.

TABLE 1: AN OVERVIEW OF RELATED ARTICLES

Reference	Year	Dataset Count	Machine Learning Techniques	Undersampling Techniques	Oversampling Techniques	Model Deployment
[26]	2009	1	✓	×	×	×
[18]	2010	1	✓	×	×	×
[2]	2016	1	✓	×	×	×
[25]	2017	1	✓	×	×	×
[16]	2018	4	✓	×	×	×
[11]	2018	1	✓	×	×	×
[17]	2018	1	✓	×	×	×
[24]	2018	1	✓	×	×	×
[23]	2019	1	✓	×	×	×
[15]	2019	1	✓	×	×	×
[22]	2019	1	✓	×	×	×
[13]	2020	1	✓	×	×	×
Our study	2020	3	✓	✓	✓	✓

3. METHODOLOGY

In this section, we explained the methodology of our study. Three imbalanced datasets have been employed to build a model for the effective prediction of credit default clients. After that, the data has been preprocessed to achieve effective results because real-world data leads to noisy values. Furthermore, to cater to the data imbalance problem, different resampling methods have been utilized to get the best results. After preprocessing, a Gradient Boosted Decision Tree (GBDT) model, which is an ensemble-based learning method, has been used for modeling and also compared the results with traditional machine learning models. At last, the credit default prediction model has been deployed for the end-users to predict the default risk earlier effectively. The proposed method is also explained in Figure 2. The following hypotheses have been developed to validate the significance of the proposed method.

First Null Hypothesis (H_0): There is no difference in the performance of various machine learning techniques.

Alternative Hypothesis (H_1): The improvement of performance by our developed model statistically significantly better.

Second Null Hypothesis (H_0): The use of imbalanced techniques do not improve the performance of the models.

Alternative Hypothesis (H_1): The use of imbalanced techniques significantly improves the performance of the models.

3.1. DATASETS

Three datasets related to credit default have been utilized in this study. Firstly, The data regarding payment employed by [26] in Taiwan have been engaged in this study. This payment data is obtained from the UCI machine learning repository in the form of a credit card client's dataset. The dataset includes 30000 observations in which 6636 are default payment observations, which also indicates an imbalance between the two classes. By using the approach of [26], default payment was designed as (yes= 1 and no= 0) as all the rest of the variables are described as shown in Table 2. Nevertheless, most of the records of the dataset regarding credit card clients are healthy. Secondly, the broadly utilized Statlog German credit data published on the UCI repository experiences extreme errors in the coding data, and any data foundation regarding economic features were not provided.

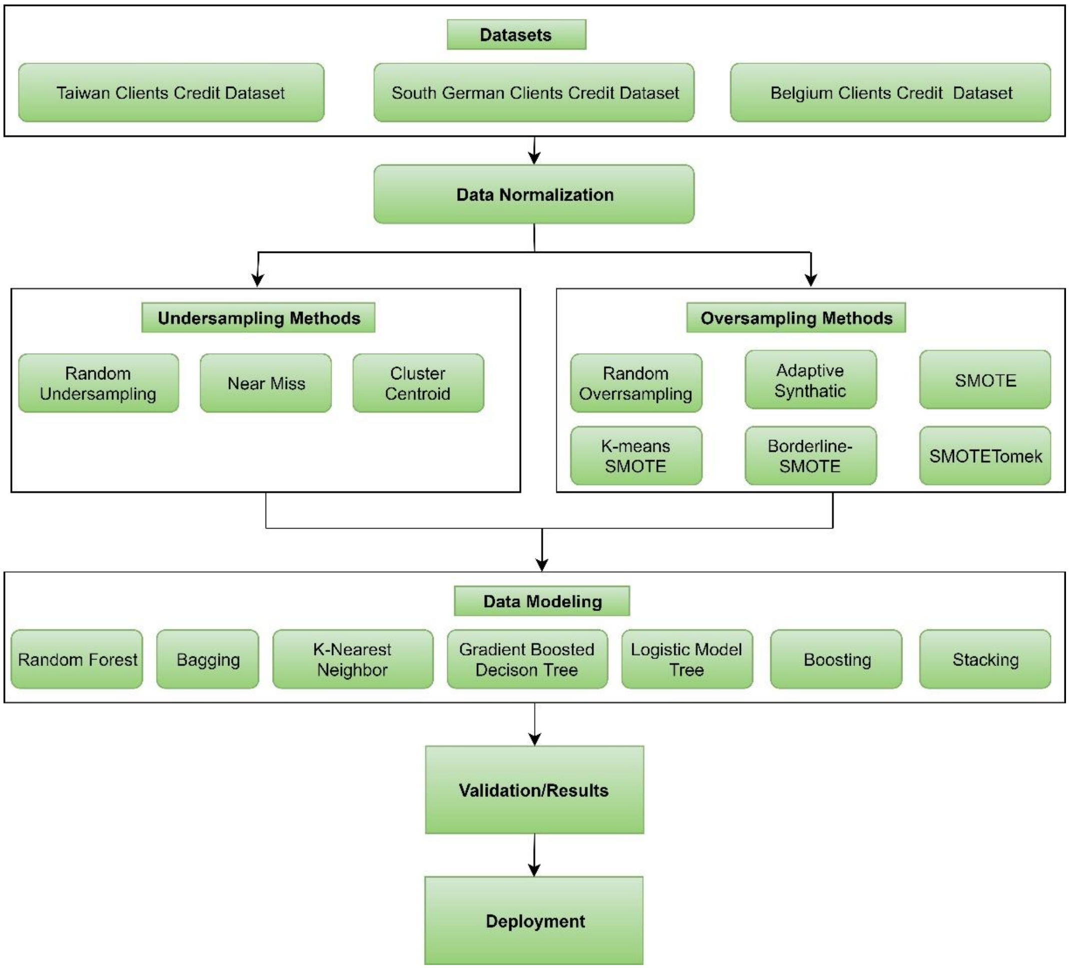


FIGURE 2. **Proposed methodology**

Data also contained an incorrect code table, and various features were wrongly represented, which implies that the data cannot be utilized for machine learning algorithms. The South German Credit data [27] published on the UCI repository, which amended the previous dataset and also added some background information relevant to features for a better understanding of data. The dataset contains 1000 instances and 21 features that indicated the financial status of clients. There are seven quantitative and thirteen categorical features. These features are related to financial records status, a measure of the advance, bank accounts or securities, a business term, Installment rate in the level of extra cash, property, age, and the number of existing credits. These data also have a target class that contains: Good or Bad. The data have a class imbalance problem because only 300 instances belong to bad credit clients, and 700 instances belong to good clients. The complete information of variables has presented in Table 3.

The credit card fraud dataset [28] was provided by a payment service provider in Belgium. The dataset was divided into daily chunks and contained fraudulent e-commerce transactions. It includes the transactions of credit cards of European cardholders in September 2013. The dataset contains the transactions in two days where 492 fraudulent and 284,807 non-fraudulent transactions were recorded. The dataset is highly imbalanced; the minority class (fraudulent) represent 0.172%. It contains just numerical factors that are the result of a PCA transformation.

Shockingly, because of confidentiality problems, the original features and data information has not provided. There are 28 features (V1, V2, … V28) in which only two features, Time and Amount, have not PCA transformed. The target class includes 0 for fraudulent and 1 for non-fraudulent transactions.

3.2. DATA NORMALIZATION

The major problem in the various datasets is that numerical features are all measured in different units. Therefore, data normalization is a useful data preparation scheme for tabular data, should be considered so that the comparison between measurements can be more accessible when building a model. Data normalization is a process of re-scaling the feature values to make the new inputs follow the standard normal distribution. Within the different features, there is often a significant difference between the minimum and maximum value. The most common normalization method is the Min-Max normalization. This technique scaled all the numerical values of a numerical feature to a specified range and computed through (1).

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (1)$$

All the features are scaled except categorical features.

TABLE 2: THE DESCRIPTION OF EACH ATTRIBUTE OF THE CREDIT CARD CLIENT'S DATASET

Attribute ID	Attribute Name	Attribute Description
X1	Limit_Bal	Amount Of Given Credit
X2	Sex	Gender (1 = Male; 2 = Female)
X3	Education	Education (1 = Graduate School; 2 = University; 3 = High School; 4: Others)
X4	Marriage	Marital Status (1 = Married; 2 = Single; 3 = Others).
X5	Age	Age (Year)
X6-X11	Pay_1 to Pay_6	History of past payment (From April to September 2005): X6 = The repayment status in September 2005… X11 = The repayment status in April, 2005 History of past payment tracked via past monthly payment records (-1 = Payment on time; 1 = Payment delay for one month; 2 = Payment delay for two months; . . . ; 8 = Payment delay for eight months; 9 = Payment delay for nine months and above).
X12–X17	Bill_Amt1 to Bill_Amt6	Amount of bill statement (Dollar) X12 = amount of bill statement in September 2005… X17 = amount of bill statement in April 2005

X18–X23	Pay_Amt1 to Pay_Amt6	Amount of previous payment (Dollar)
		X18 = Amount paid in September 2005…
		X23 = Amount paid in April 2005

X24	default payment next month	Default= 1 and healthy= 0

TABLE 3: THE DESCRIPTION OF EACH ATTRIBUTE OF THE SOUTH GERMAN CREDIT DATASET

Column Name	Variable Name	Attribute Description
Laufkont	Status(categorical)	The account status of the debtor with a bank.
Laufzeit	Duration(quantitative)	The duration of credit in months.
Moral	Credit history(categorical)	The contract's history of previous or current credit.
Verw	Purpose(categorical)	The reason behind credit.
hoehe	Amount(quantitative)	The total amount of credit.
sparkont	Savings(categorical)	Total savings of debtor.
beszeit	Employment duration(ordinal)	The duration of employment of a debtor with the current organization.
Rate	Installment rate(ordinal)	The credit installments of debtor's throwaway income.
famges	Personal status sex(categorical)	The information about both sex and marital status.
buerge	Other debtors(categorical)	Another debtor for the credit.
wohnzeit	Present residence(ordinal)	The duration of living in the present residence.
Verm	Property(ordinal)	The ranking of debtor's property in ascending order.
Alter	Age(quantitative)	The age of the debtor.
weitkred	Other installment plans(categorical)	Any credit/installment burden other than the credit-giving bank.
wohn	housing(categorical)	Status of current residence.
bishkred	Number credits(ordinal)	The complete history of the credit's taken.
beruf	Job(ordinal)	The level of the debtor's job.
Pers	People liable(quantitative)	The total number of peers depends on debtor financially.
Telef	Telephone (binary)	The status of a registered landline on the debtor's name.
gastarb	Foreign worker(binary)	Is the debtor a foreign worker?
kredit	Credit risk(binary)	Good or Bad

3.3. RESAMPLING METHODS

Any dataset can be considered as imbalanced if the number of instances between classes is not equal. Resampling methods for imbalanced learning applications typically means to add a bias to balance the dataset. Although classifiers absolutely can learn from imbalanced datasets, it is worthy of balancing the dataset to achieve more robust results. All the credit-related datasets employed in this study leads to the data imbalance problem. Besides, all of the resampling techniques allow resampling until reached the desired ratio of balance dataset, allowing us to directly compare different resampling methods for a given proportion of minority and majority class data points in the final training set. Resampling techniques have been implemented on the full datasets. Data level resampling approaches have most commonly used to deal with class imbalance, so various undersampling and oversampling based approaches have been used in this study.

3.3.1. RANDOM UNDERSAMPLING

Random undersampling is a simple undersampling based approach. Majority class instances in the training set are randomly eliminated until the ratio between the minority, and the majority class is at the desired level. Theoretically, one of the problems with random undersampling is that one cannot control what information about the majority class is thrown away. In particular, crucial details on the decision boundary between the minority and majority class may be eliminated. Despite its simplicity, random undersampling has empirically been shown to be one of the most effective resampling methods. In particular, few of the more sophisticated undersampling methods have outperformed random undersampling in empirical studies. In random undersampling, examples have been randomly removed from the majority class to balance the class instances, which results in the removal of vital information from the majority class. This approach also results in a downsizing of the training data considerably. Therefore it is the most naive approach in data undersampling.

3.3.2. NEAR MISS

Near Miss is an undersampling technique proposed by zhang and mani [29] that aims to mitigate the information loss during the undersampling of the majority class. Instead of resampling the minority class, using a distance, this will make the majority class equal to the minority class. Near Miss uses average distances between a given point and the nearest or farthest points of the opposite class. Near Miss, undersampling has three versions, all aimed at creating separation between the two class observations. In NearMiss-1, we need to select the majority class points up to the given percentage of the majority class size, which is close to some of the minority class points. It means to select major class points with the smallest average distance to the three nearest points from the minor class. In NearMiss-2, we need to choose the majority class points up to the given percentage of the majority class size, which is close to all points of the minority class. It means to select the majority class points with the smallest average distance to the three farthest points from the minority class. In NearMiss-3 for each minor class point, we need to select a given number of the closest majority class points. In this study, the NearMiss-1 method has been used in which majority class points are selected with the smallest average distance to the three nearest points from the minority class.

3.3.3. CLUSTER CENTROID

One major problem of using undersampling is that important information may be lost from the majority class, which can cause overly general rules, which means samples can be misclassified after classification. This cannot be afforded to develop the credit card default prediction model, especially for default samples. Hence, to overcome this problem, the Cluster Centroids method has been introduced in [30]. Cluster Centroids undersamples the majority class by replacing majority samples from clusters with the cluster of centroids using the K-means algorithm by considering the ratio of majority class samples to minority class samples. This technique performs undersampling by generating centroids based on k-means clustering methods. The data has grouped based on the similarity to preserve information. A K-means algorithm is fitted to the data, and the number of clusters (k) has been obtained by the level of undersampling. Then, the majority of samples from the clusters are entirely substituted by the sets of cluster centroids from K-Means. Cluster Centroids contain the most representative variations of the majority class in which features values would be visualized at the center. An attempt was made at remedying this issue by both underfitting and overfitting the data as well as combining the two. When underfitting was applied to the dataset, this was done by only considering the cluster centroids similar adapted from [30].

3.3.4. RANDOM OVERSAMPLING

Like random undersampling, random oversampling is a simple yet effective approach to resampling. Random Oversampling is a very naive approach to data oversampling. It merely replicates the minority class examples and adds them to the training data. By using this technique, new examples come from the existing minority class examples in the training set that results in the problem of over-fitting.

Over-fitting is a problem that occurs when all the training examples are very similar to each other, and the classifier correctly classifies these examples. In such a scenario, if a test example is slightly different from the training examples, then the classifier is not able to classify it correctly and results in poor classification for the new examples. In other words, the classifier is trained to classify only a very narrow set of examples correctly. The random oversampling method operates by replicating the randomly selected set of examples from the minority class so that the majority class does not have an overbearing presence during the training process. Since the resampling process is random, it becomes difficult for the decision function to find a clear borderline between the two classes. Therefore, although it is widely used, Random oversampling might be ineffective at improving recognition of the minority class by a large margin. Some potential drawbacks of random oversampling include an increase in training time for the classifier and overfitting on account of duplication of examples of the minority class. However, other oversampling methods have been built based on this method.

3.3.5. ADAPTIVE SYNTHETIC

Adaptive Synthetic (ADAYSN) oversampling technique is based on density distribution to generate synthetic data samples for each minority class inevitably. This method was proposed by Haibo He et al. [31] for two-class classification and describe the following.

Suppose, the training set is D_{tr} concerning m samples, I is *(1 to m)* in which X_i denotes the instance by considering n-dimensional space X. Therefore, $y\_i \in Y = \{1, -1\}$, which describes the class label association with X_i. m_0 represents the minority data points and m_1 denotes the majority data points which also implies $m_0 \leq m_1$ and $m_0 + m_1 = m$. First, calculate the rate of imbalance between two classes using (2):

$$d = \frac{m_0}{m_1} \qquad (2)$$

Then compute the synthetic data points that have to need to be generated from the minority class.

$$G = (m_1 - m_s) * \beta \quad (3)$$

In the above Equation (3), β specifies the generation of synthetic data after the desired balance level. If the value of (β =1), this implies that the dataset is balanced after generalization. For individual data point x_i belong to minority class, Euclidean distance has been calculated in n-dimensional space to determined k-nearest neighbors.

$$r_i = \frac{\triangle \mathrm{i}}{K}, \qquad i = 1,2,3, \ldots, m_s \qquad (4)$$

In which $\triangle$i denoted the number of instances in the K-nearest neighbor of x_i that is associated with the majority class. Hence, $r_i \in [0,1]$. The normalization has been done through (5).

$$\hat{r} = \frac{r_i}{\sum_{i=1}^{m_s} r_i} \qquad (5)$$

So, the synthetic data points that are necessary to be generated for individual minority data point's x_i are calculated:

$$g_i = \hat{r} * G \qquad (6)$$

G denotes the total number of synthetic data points that need to be created for the minority class is determined through (3). In the last step, one data point x_{zi} from minority class chosen randomly from the data x_i. Synthetic data points are generated through (7).

$$S_i = x_i + (x_{zi} - x_i) * \lambda \qquad (7)$$

In which $(x_{zi} - x_i)$ represented the n-dimensional space for difference vector, and λ denoted random number: $\lambda \in [0,1]$. This improves learning by reducing the bias introduced by the class imbalance and moving the classification decision boundary towards the samples that are more difficult to learn. The quoted mechanism shows that ADAYSN pays special attention to data samples that are particularly close to the majority of samples.

3.3.6. SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE

Chawla et al. [32] proposed a powerful oversampling approach called the Synthetic Minority Oversampling Technique (SMOTE) that improves the classification of minority classes in imbalanced data. It allows one to oversample the minority class and undersample the majority class. Unlike previous algorithms that oversample the minority class by replication, leading to over-fitting, SMOTE creates synthetic minority data. It over-samples the minority class by taking k (in our case, k = 5) nearest neighbors for a given minority data sample, finding the difference between the features of it and a randomly chosen neighbors, multiplying this difference by a random number between 0 and 1, and adding it to the feature vector.

$$x_{new} = x_i + (x_i' - x_i) * \alpha \qquad (8)$$

x_i' is one of the K- nearest neighbors of x_i, and $\alpha \in [0,1]$ is a real random number. SMOTE repeats this sampling and perturbation algorithm to create minority data samples according to the amount of over-sampling desired. For instance, over-sampling by 200% creates two new synthetic minority samples by separately perturbing a sample along the vectors of two different nearest neighbors. SMOTE also allows one to undersample the majority class by removing samples until the new majority class is a certain percentage of the original minority class' sample size. Depending upon the percentage of over and under-sampling, the resulting dataset may have more or fewer samples in the minority class than in the original data. With a slight variation, a similar technique can be used for categorical variables. In the case of mixed categorical and continuous variables, like our datasets, SMOTE calculates the nearest neighbors by first calculating the median of standard deviations of the continuous features in the minority class. If the categorical variables differ between the sample and its potential nearest neighbors, then the previously calculated median has been included in calculating the Euclidean distance between samples. After the k nearest neighbors are determined, the synthetic categorical features are assigned the majority occurring values amongst the nearest neighbors. At the same time, the continuous variables are calculated originally. By creating synthetic minority classes, SMOTE creates more general decision regions than the small, specific regions that result from the replication of minority classes. Because a factor between 0 and 1 only perturbs samples, this method does limit the synthesized data to be no more or less than the extreme values of the real data.

3.3.7. BORDERLINE-SMOTE

The Borderline-SMOTE algorithm was developed to help resolve some of the problems caused by borderline data. Han et al. [33] first introduced this algorithm as an extension of the SMOTE algorithm that focuses on generating synthetic data for minority class instances that are exclusively near the borderline. While [33] demonstrate that the borderline variant of SMOTE may increase the classification accuracy of the minority class, it is not clear how the overall performance of the models compare. One, in particular, that removes some of the randomnesses of the original SMOTE by considering both classes in the neighborhoods is Borderline-SMOTE. Equivalent to the innovative algorithm, it uses the k -nearest neighbors of a minority class point, but now while considering every remaining sample point in the training set. It uses this neighborhood to categorize the minority class point as follows:

a. If all of the k neighbors belong to the majority class, it is considered to be noise.
b. If less than half of its neighbors belong to the majority class, it is considered to be safe.
c. If at least half of its neighbors belong to the majority class (but not all of them), the point is labeled as danger.

This means that, by explicitly oversampling those points, it should aid the learner in incorrectly retrieving the decision bound. Synthetic instances are then created between the instance and a random sample of its m nearest neighbors. In this way, the border between the classes is strengthened, so it should be more comfortable for the classifier to recognize the difference between the two classes.

3.3.8. SMOTETOMEK

This algorithm is also an extension of the original SMOTE that was introduced by Batista et al.[34] to solve the class distribution problem more efficiently. The class distribution will always be a problem if the majority class invades into the minority class, and the same way after oversampling the minority class can do the same to the majority class. Similarly to the SMOTEENN technique, this technique is also a variant of SMOTE that utilizes an additional technique for data cleaning, in this case, that technique is Tomek's Link. It is a powerful method to combine SMOTE with Tomek's link removal for the sake of class balancing. SMOTE is applied first to the dataset, creating new synthetic observations. Subsequently, Tomek's link undersampling is applied to the new dataset (that contains the synthetic observations) to remove any pairs of examples that form a Tomek's link.

Tomek's link is a link between two data points that are defined by a combination of two things: Firstly, they must be nearest neighbors; secondly, they must have different class labels. Examples that are Tomek's link are more likely to be either noise or points that are close to the optimal decision boundary. Consider two examples x_i and x_j belong to different classes. Let $d(x_i, x_j)$be the distance between them. Examples (x_i, x_j) form a Tomek link if there is no other example $x_i (l \neq i \ and \ l \neq j)$ such that $d(x_i, x_l) < d(x_i, x_j)$ and $d(x_j, x_i) < d(x_i, x_j)$. If x_i and x_j create a Tomek link, then either one of them is noise, or both are borderline examples. Originally Tomek links are used to find out the noise and borderline examples, but this technique is also used as an undersampling method for majority class examples.

3.3.9. K-MEANS SMOTE

Douzas et al. [35] proposed a method that first separately partitions the minority and majority classes using the k-means algorithm, then performs over-sampling with duplication on the resulting clusters to re-balance the class distribution as well as inflate small clusters to counter the within-class imbalance through SMOTE. The specific way of over-sampling is as follows: In the majority class, clusters except the largest one are over-sampled up to the size of the largest cluster, then minority clusters are over-sampled until each cluster contains $\frac{Maj-size}{\#\text{min}-\text{clusters}}$ instances, where maj-size is the overall size of the majority class, and #min-clusters is the number of minority clusters. The oversampling has done through the SMOTE method, as explained in the above section. Furthermore, to divide the training set into clusters and then perform sampling locally for each cluster. Finally, all the clusters are combined to create only one training set for training a global classifier.

Different clusters in a dataset and each cluster seems to have distinct characteristics. If a cluster has more majority class instances and less minority class instances, it will behave more like the majority class. On the other hand, if a cluster has more minority class instances and less majority class instances, it will behave more like the minority class. The distribution of majority and minority classes employing different resampling techniques is shown in Table 4.

TABLE 4: CLASS DISTRIBUTION OF VARIOUS DATASET: BEFORE AND AFTER RESAMPLING TECHNIQUES

Credit Card Client's Dataset			
	Total instances	Defaulter clients	Healthy clients
Imbalanced Dataset	30000	6636	23364
Undersampled Dataset	13272	6636	6636
Oversampled Dataset	46728	23364	23364
South German Client's Credit Dataset			
	Total instances	Bad clients	Good clients
Imbalanced Dataset	1000	300	700
Undersampled Dataset	600	300	300
Oversampled Dataset	1400	700	700
Belgium Client's Credit Dataset			
	Total instances	Defaulter clients	Healthy clients
Imbalanced Dataset	285299	492	284807
Undersampled Dataset	984	492	492
Oversampled Dataset	569614	284807	284807

3.4. GRADIENT BOOSTING

The iterative machine learning method to solve the classification problem is known as gradient boosting. This technique is based on ensemble learning in which the model is trained in such a way that errors of the previous iteration are used. Gradient Boosting accounts for misclassified samples by fitting a new learner to the ensemble residual that is the difference between the target outputs and the current predictions of the ensemble. Gradient Boosting tries to maximize the predictive power of the ensemble, i.e., minimize the bias. The advantage of using a boosting approach is generally high predictive power, but it comes with the cost of being slow to train as each new learner is trained sequentially.

Consider the joint probability distribution $P(x,y)$ in which x is the input variable, and y is the output variable. The purpose is to determine the function $F(x)$ by utilizing the training set of N observations (x_i, y_i) to predict y in which the values of x are already known. When there are a finite number of known values y or classes, then it is described as a classification problem. When loss function L has minimized from the training set, then $F(x)$ determined.

$$\tau(F) = \sum_{i=1}^{N} L(y_i, F(x_i)) \quad (9)$$

$$F = argmin_F \tau(F) \quad (10)$$

The entire training data has used to calculate $\tau(F)$ through (9), which states the error furnished by learner F. The purpose is to determine the function F, which minimizes the error or loss through (10). The sum of $M+1$ base learners has constructed to approximate F_m to F through M iterations in gradient boosting.

$$F_m = \sum_{m=0}^{M} fm \quad (11)$$

The notion of gradient boosting has started with the initial guess F_0 then steepest descent follows iteratively to the negative gradient to minimize the error.

$$g_{m,i} = \nabla F_{m-1} L(y_i, F_{m-1}(x_i)) \quad (12)$$

$$F_m = F_{m-1} - \gamma_m g_m \quad (13)$$

$$\gamma_m = argmin_\gamma \, \tau(F_{m-1} - \gamma g_m) \quad (14)$$

The gradient loss of F_{m-1} on the training, data has been determined by (12). The negative gradient of the next classifier F_m has also been determined by (12). The length used to minimize the loss of optimal length r_m has been calculated through (14). The equation (11) given (m>0):

$$f_0 = F_0 \quad (15)$$

$$f_m = -\gamma_m g_m \quad (16)$$

It is not possible to use g_m directly because it only gives values on some points like training data, as shown in (12). It is mandatory to generalize the result because the model can also be suitable for unseen data, so the function from a restricted class for the best approximation has been used. To fit the gradient, a base learner h_m has used by utilizing the training set $(x_i, g_{m,i})_{i=1}^{N}$ and then updated equations are:

$$f_m = \gamma_m h_m, \quad m > 0 \quad (17)$$

$$F_m = F_{m-1} - \gamma_m h_m \quad (18)$$

$$\gamma_m = argmin_\gamma \, \tau(F_{m-1} - \gamma h_m) \quad (19)$$

Algorithm 1 is the gradient boosting algorithm for the above equations. The algorithm executes in such a way that it finds the local minimum L by iteratively increasing the step size. The step that reaches this minimum is chosen as the solution γ_m.

Algorithm 1: Gradient boosting algorithm

```
1. Initialize the f with the best constant in which,
   $F = argmin_F \tau(F)$
2. For-Do loop ($m = 1 \rightarrow M$)
3. Calculate the gradient at the training points:
   $g_{m,i} = \nabla F_{m-1} L(y_i, F_{m-1}(x_i))$
4. Fit a new base-learner to the target $g_m$
5. Find the best gradient step, which is as
   followed:
   $\gamma_m = argmin_\gamma \, \tau(F_{m-1} - \gamma h_m)$
    Update the function estimate
6. $F_m = F_{m-1} - \gamma_m h_m$
7. $m \leftarrow m + 1$
8. Loop end
9. Return $F_M$
```

3.4.1. GRADIENT BOOSTED DECISION TREE

Decision trees are intuitive models that resemble real-life thinking closely. This makes these kinds of models easy to work because it is easy to visualize them and can spot errors as well. Randomized decision trees and forests have a rich history in machine learning and have seen considerable success in various applications. However, they face fundamental limitations: given enough data, the number of nodes in decision trees will grow exponentially with depth, and the exponential growth of trees limits their depth. The GBDT model has been used to overcome the above problems. The GBDT utilized for decision trees of a fixed size as base learners. Friedman et al. [36] proposed a modification to gradient boosting, which enhances the performance of the base learner. Equation (19) has used to improve the optimization of m^{th} step size γ_m. It has also useful to execute the search for each tree to determine the optimal descent direction. It can defined as

$$\gamma_{mk} = argmin_\gamma \sum_{x_i \in R_k} L(y_i, F_{m-1}(x_i) + \gamma) \quad (20)$$

And an updated model becomes

$$F_m(x) = F_{m-1}(x) - \sum_{k=1}^{k} \gamma_{mk} \, . \, I(x \in R_k) \quad (21)$$

While a tree is growing, the best local gradient is approximated to find the γ_{mk} through (20). Hence, the updated Equation (22) for each node of boosted trees is:

$$min_{\gamma_{mk1,mk2}} \left[\sum_{x_i \mid x_{ip} \leq s} L((y_i, F_{m-1}(x) + \gamma_{mk1}) + \sum_{x_i \mid x_{ip} \leq s} L((y_i, F_{m-1}(x) + \gamma_{mk2}) \right] \quad (22)$$

3.4.1.1. LOSS FUNCTION

The conventional way to overcome the problem of loss or error function is to set a decision boundary on the real axis and assign the class on each side of the boundary. In other words, calculate the probabilities for each class. Binomial deviance enables to overcome the problem of the loss function. The decision boundary of this loss function is set to be zero. The probability with the learner output f has assigned by decision boundary and also calculated through the logistic function:

$$P(y = 1|x) = P(f) = \frac{1}{1+ e^{-f(x)}} \qquad (23)$$

$$\nabla f\, L_D = P(y = 1|x) - y = \frac{1}{1+ e^{-f(x)}} - y \qquad (24)$$

$$L_D = log\ (1 + e^{f}) - yf \qquad (25)$$

It is also noticed that binomial deviance also punished the correctly classified examples. Binomial deviance reduces the misclassification rate because it punishes the misclassifications examples more profoundly than the corrected ones. Furthermore, the penalty upturns linearly with f, which makes it more robust than other loss functions in which penalty rises at a high rate. The above reasons justify that binomial deviance is an ideal loss function for classification problems than others. This method can also be utilized for multi-class classification problems.

3.4.1.2. FEATURE SELECTION THROUGH GRADIENT BOOSTED DECISION TREE MODEL

It can be fascinating to comprehend where the model prediction originates from it. Furthermore, this also implies that how the model predicts decisions. In particular, when constructing a model with several features, some of them had a higher priority than others, while a few features may not be associated with predictions. Particular features and their various combinations contain different amounts of information suitable for class discrimination. Some features may be redundant and do not provide any new information for classification, or irrelevant, hence offer no relevant information at all. The occurrence of these features can influence the classifier design negatively and decrease its final performance.

The goal of feature selection is to find a reduced subset of the input features in which maximum redundant and irrelevant information is eliminated. The problem is how to define features that are better to keep or to remove. It is difficult to determine the informativeness of features correctly because of the limited sample size. Furthermore, the information content of features depends on a criterion function employed in the final performance evaluation. It is complicated to imagine all situations arising in the real-world data as there can be quite complex non-linear high-dimensional statistical dependencies between features [37]. It may be beneficial to determine which features clarify the dispersion of the data. It is easy to visualize the decision tree through a 2-D image to understand the selection of significant features. Various researches also extract significant features for better performance [38, 39]. However, it is more beneficial to calculate the importance of each feature to enhance the model's prediction.

The GBDT model that we have used there, i.e. the GBDT model automatically selects the significant features during the modeling phase that also given effective results [40-42]. The following method has utilized to estimate the significance of features:

$$I_p^2(T) = \sum_{splitsonX_p} I_k^2 \qquad (26)$$

It also gives the relative importance in the tree T of the feature p, also known as a relative influence. By splitting the variable p at node k, I_k^2 has obtained. This measure was extended later by averaging all the trees, which made the boosted model:

$$I_p^2 = \frac{1}{M}\sum_{m=1}^{M} I_p^2(T_m) \qquad (27)$$

3.4.1.3. The GRADIENT BOOSTED DECISION TREE MODEL AND OVERFITTING

The model has updated on each iteration of GBDT with the base learner to decrease the loss of gradient, which implies that with each iteration, the training loss decreased. The training error might be small if there are a large number of iterations M. The original GBDT algorithm might overfit when data is too much fitted, which leads to an increase in the error on the training set. It might be possible to optimize the number of iterations to decrease the overfitting risk to overcome this problem. A regularization method by shrinkage to overcome overfitting has been used in which a learning rate parameter V has added in Algorithm 2 by updating the (18):

$$F_m = F_{m-1} - V\gamma_m h_m \quad ,0 < V \leq 1 \qquad (28)$$

Equation (28) decreased the values V by 0.1, which control overfitting, but the training error becomes large if more shrinkage is performed.

Algorithm 2: Gradient boosted decision tree algorithm

```
1.  Initialize the f with the best constant in which, $F = argmin_F \tau(F)$
2.  For-Do loop ($m = 1 \rightarrow M$)
3.  Choose a suitably sized subsample $X'$ from the data points arbitrarily
4.  Calculate the negative gradient at the training points:
            $g_{m,i} = -\nabla F_{m-1} L(y_i, F_{m-1}(x_i))$
5.  By utilizing $X'$ fit the tree $h_m(x)$ to the target class
6.  Calculate the best terminal nodes which give predictions by using $X'$
        Update the function estimate
7.  $F_m = F_{m-1} + h_m$
8.  $m \leftarrow m + 1$
9.  Loop end
10. Return $F_M$
```

4. Evaluation Metrics

Model evaluation is of paramount importance in any predictive modeling task. It becomes even more critical in ensemble predictive modeling, where the relative performance and diversity of models must be thoroughly evaluated. All the evaluation metrics are built on four types of classifications: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

4.1 Accuracy

Typically, accuracy is used to assess the effectiveness of a model with the help of the confusion matrix. The accuracy of the model has been computed through (29).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (29)$$

4.2 Precision

Precision compares the number of true positives to the number of true positives and the number of false positives. That is, of all the instances the classifier said were positive, precision measure how many of them were positive. The Precision of the model has been computed through (30).

$$Precision = \frac{TP}{TP + FP} \quad (30)$$

4.3 Recall

Recall compares the number of true positives to the number of true positives and false negatives. The Recall of the model has been computed through (31).

$$Recall = \frac{TP}{TP + FN} \quad (31)$$

4.4 F-Measure

F-Measure combines precision and recall as the harmonic mean. The precision and recall trade-off with each other: higher precision generally associated with low recall. The value of F-Measure has been computed through (32).

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (32)$$

4.5 Receiver Operating Characteristic curve

A receiver operating characteristic (ROC) curve plot is also a widely used measure to evaluate the performance of classifiers. Specifically, the plot is created by plotting the true positive rate (recall) against the false positive rate at various threshold levels.

4.6 Geometric Mean

The Geometric Mean (G-Mean) is a metric that measures the balance between classification performances on both the majority and minority classes. This measure is essential in the avoidance of overfitting the negative class and underfitting the positive class [43]. The G-mean has been calculated through (33).

$$G - Mean = \sqrt{\left(\frac{TP}{TP + FN}\right) * \left(\frac{TN}{TN + FP}\right)} \quad (33)$$

Accuracy is a good measure to evaluate the performance of the balanced datasets but not on an imbalanced dataset. Measuring the performance of a classifier applied to imbalanced data using traditional metrics such as accuracy is difficult since it does not take into account the lower number of instances in the minority class. Previous studies [42, 44] also showed that imbalance could exert a significant impact on the value and meaning of accuracy and specific other well-known performance metrics. Another performance evaluation metric, F-Measure also neglects the correct classification of negative samples and only reflects the importance of retrieval of positive examples. Threshold metrics such as Precision and Recall have been used frequently for assessing the performance of a classifier in such cases. A combination of these measures, such as G-mean used different combinations of specificity and sensitivity of the classifiers to give a better indication of performance. Ranking order metrics such as ROC measure assess the performance of a classifier overall imbalance ratios and hence provide a summary of the entire range. Furthermore, several performance measures, i.e., accuracy, precision, recall, F-Measure, ROC, and G-Mean, have been employed.

5. RESULTS

Since we were trying to build a prediction model, there was a dire need for a dataset to build the model. There is a need for data to test whether our model made correct predictions on new data. We split our datasets into training and test data with a ratio of 70:30. Various classifiers have been utilized to evaluate the performance of imbalanced datasets. The performance of the GBDT method is also compared with traditional machine learning models in which the GBDT method outperformed traditional machine learning models on imbalanced datasets. The results are presented in Table 5 in which the GBDT model has been given the best results with the accuracy of 66.9% on Taiwan clients credit dataset, 70.7% on South German clients credit dataset, and 65% on Belgium clients credit dataset. After the GBDT method, different traditional models (random forest, bagging, K-Nearest Neighbor, Logistic Model Tree, Adaboost, and Stacking) have also been utilized. These models also showed significant results but outperformed by the GBDT model.

TABLE 5: THE PERFORMANCE OF MACHINE LEARNING TECHNIQUES WITHOUT OUR PROPOSED MODEL

Dataset	Models	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	ROC	G-Mean (%)
Taiwan clients credit dataset	Random Forest	65.5	72.7	60.7	66.2	0.68	66.4
	Bagging	63.6	78.9	55.5	65.2	0.65	66.2
	K-Nearest Neighbor	64.7	74.2	62.0	67.6	0.65	67.8
	Logistic Model Tree	61.1	66.6	60.0	63.1	0.63	63.2
	AdaBoost	63.8	69.2	66.6	67.9	0.64	67.9
	Stacking	46.8	50.0	52.9	51.4	0.50	51.4
	GBDT Model	66.9	67.1	72.4	69.6	0.70	69.7
South German clients credit dataset	Random Forest	69.3	68.7	70.9	69.8	0.72	69.8
	Bagging	66.1	65.6	67.7	66.6	0.68	66.6
	K-Nearest Neighbor	63.6	64.9	62.9	63.9	0.65	63.9

	Logistic Model Tree	62.7	63.1	63.1	63.1	0.63	63.1
	AdaBoost	68.8	66.6	73.3	69.8	0.70	69.9
	Stacking	46.2	52.7	50.0	51.3	0.49	51.3
	GBDT Model	70.7	68.5	75.0	71.6	0.73	71.7
Belgium clients credit dataset	Random Forest	63.5	62.5	67.5	64.9	0.65	64.9
	Bagging	61.9	61.2	66.2	63.6	0.63	63.6
	K-Nearest Neighbor	60.9	60.0	67.1	63.3	0.61	63.4
	Logistic Model Tree	60.2	62.9	59.7	61.2	0.61	61.3
	AdaBoost	43.2	49.5	38.4	43.2	0.47	43.6
	Stacking	44.8	50.0	44.4	47.0	0.48	47.1
	GBDT Model	65.0	63.7	66.6	65.1	0.68	65.2

A GBDT has been used with different combinations of tuning parameters. These tuning parameters were the learning rate and the number of decision trees to be constructed. First, Taiwan's client credit dataset has been employed to test the performance of classifiers. The GBDT model applied to random undersampling, Near Miss, and Cluster Centroid undersampled datasets. The results are presented in Table 6 in which Random undersampling, Near Miss, and Cluster Centroid method has given the accuracy of 70.3%, 82.8%, and 86%, respectively. The Cluster Centroid outperformed Near Miss and random undersampling method based on the accuracy. After the GBDT method, different traditional models (random forest, bagging, K-Nearest Neighbor, Logistic Model Tree, Adaboost, and Stacking) have also been utilized by utilizing undersampling techniques. These models also showed significant results on the Cluster Centroid method as compared to Near Miss and random undersampling. The performance of the GBDT method also compared with traditional machine learning models in which the GBDT method outperformed traditional machine learning models on undersampled datasets, as shown in Table 7. Furthermore, the performance of classifiers even better on the Cluster Centroid undersampled dataset as compared to the Near Miss and random undersampling dataset.

TABLE 6: THE PERFORMANCE OF GBDT MODEL USING UNDERSAMPLING TECHNIQUES ON TAIWAN CLIENTS CREDIT DATASET

Undersampling Methods	Learning Rate	Number of decision trees	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	ROC	G-Mean (%)
Random Undersampling	**0.1**	**50**	**70.3**	**67.6**	**75.9**	**71.5**	**0.77**	**71.6**
	0.1	100	70.1	67.6	74.9	71.1	0.77	71.2
	0.1	150	69.6	67.5	73.2	70.3	0.77	70.3
	0.1	200	69.2	67.3	72.1	69.6	0.78	69.7
	0.2	200	68.7	67.4	69.9	68.6	0.78	68.6
Near Miss	0.1	50	82.5	79.3	86.9	82.9	0.87	83.0
	0.1	100	82.7	79.8	86.7	83.1	0.87	83.2
	0.1	150	82.7	80.1	86.2	83.0	0.88	83.1
	0.1	200	82.7	80.4	85.6	82.9	0.88	83.0
	0.2	**200**	**82.8**	**80.5**	**87.0**	**83.3**	**0.88**	**83.7**
Cluster Centroids	0.1	50	84.6	84.0	84.6	84.3	0.91	84.3
	0.1	100	85.2	84.5	85.5	85.0	0.91	85.0
	0.1	150	85.8	85.0	86.3	85.6	0.91	85.6
	0.1	200	86.0	85.3	86.3	85.8	0.92	85.8
	0.2	**200**	**86.0**	**84.8**	**87.2**	**86.0**	**0.92**	**86.0**

TABLE 7: THE PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS USING UNDERSAMPLING TECHNIQUES ON TAIWAN CLIENTS CREDIT DATASET

Undersampling Methods	Performance Measures	Random Forest	Bagging	K-Nearest Neighbor	Logistic Model Tree	AdaBoost	Stacking	GBDT Model
Random Undersampling	Accuracy (%)	70.0	69.1	69.3	69.3	69.6	49.8	**70.3**
	Precision (%)	70.2	69.3	69.7	69.7	71.3	51.2	**67.6**
	Recall (%)	69.8	69.1	69.4	69.4	69.6	49.7	**75.9**
	F-Measure (%)	70.0	69.1	69.2	69.2	69.0	50.1	**71.5**
	ROC	0.76	0.75	0.61	0.75	0.75	0.50	**0.77**
	G-Mean (%)	70.0	69.2	69.5	69.5	70.4	50.4	**71.6**
Near Miss	Accuracy (%)	81.7	80.8	70.2	80.3	74.1	49.7	**82.8**
	Precision (%)	80.3	81.1	70.5	80.1	75.3	50.1	**80.5**
	Recall (%)	81.8	80.9	70.3	80.4	74.1	49.7	**87.0**
	F-Measure (%)	81.7	80.9	70.2	80.3	73.7	48.7	**83.3**
	ROC	0.87	0.87	0.70	0.85	0.77	0.50	**0.88**
	G-Mean (%)	81.0	81.0	70.4	80.2	74.7	49.9	**83.7**
Cluster Centroid	Accuracy (%)	82.4	81.5	63.1	78.6	73.0	49.7	**86.0**
	Precision (%)	82.5	81.6	63.2	78.7	73.0	52.1	**84.8**
	Recall (%)	82.5	81.6	63.2	78.7	73.0	49.7	**87.2**
	F-Measure (%)	82.5	81.6	63.2	78.7	73.0	48.1	**86.0**
	ROC	0.89	0.88	0.63	0.84	0.79	0.50	**0.92**

	G-Mean (%)	82.5	81.6	63.2	78.7	73.0	50.9	**86.0**

Secondly, Oversampling techniques (Random oversampling, ADASYN, SMOTE, Borderline-SMOTE, SMOTETomek, and K-means SMOTE) have also been utilized to enhance the performance of the classifier. The GBDT model is applied to oversampled datasets. The results are presented in Table 8, in which the K-means SMOTE method is given the best accuracy of 88.7%. The K-means SMOTE oversampling technique given better results than all other oversampling techniques based on accuracy. After the GBDT method, different traditional models (random forest, bagging, K-Nearest Neighbor, Logistic Model Tree, Adaboost, and Stacking) have also been used by utilizing oversampling techniques. These models also have shown effective results on the SMOTE based oversampling methods as compared to others. The performance of the GBDT method is also compared with traditional machine learning models in which the GBDT model outperformed traditional machine learning models on oversampled datasets, as shown in Table 9. Furthermore, the performance of classifiers also better on the SMOTE-based oversampled datasets as compared to ADASYN and random oversampled datasets.

TABLE 8: THE PERFORMANCE OF GBDT MODEL USING OVERSAMPLING TECHNIQUES ON TAIWAN CLIENTS CREDIT DATASET

Oversampling Methods	Learning Rate	Number of decision trees	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	ROC	G-Mean (%)
Random Oversampling	0.1	50	73.1	70.2	79.5	74.6	0.81	74.7
	0.1	100	74.4	72.0	79.1	75.4	0.81	75.5
	0.1	150	76.0	73.9	79.6	76.6	0.83	76.7
	0.1	200	76.9	75.2	79.8	77.4	0.85	77.5
	0.2	**200**	**80.4**	**79.9**	**80.7**	**80.3**	**0.87**	**80.3**
ADAYSN	0.1	50	82.6	80.2	87.1	83.6	0.89	83.6
	0.1	100	85.6	82.9	90.0	86.3	0.92	86.4
	0.1	150	86.2	83.7	90.3	86.9	0.92	86.9
	0.1	200	86.8	84.3	90.9	87.5	0.92	87.5
	0.2	**200**	**86.9**	**85.1**	**89.9**	**87.4**	**0.92**	**87.5**
SMOTE	0.1	50	83.6	81.0	87.5	84.1	0.90	84.2
	0.1	100	86.4	83.6	90.2	86.8	0.90	86.8
	0.1	150	87.3	84.6	90.8	87.6	0.91	87.6
	0.1	200	87.5	85.0	90.8	87.8	0.92	87.9
	0.2	**200**	**87.6**	**85.7**	**89.9**	**87.8**	**0.93**	**87.8**
Borderline-SMOTE	0.1	50	81.0	79.0	82.7	81.1	0.90	80.8
	0.1	100	82.2	80.8	84.0	82.4	0.91	82.4
	0.1	150	86.0	84.2	88.2	86.1	0.92	86.2
	0.1	200	87.0	85.1	89.4	87.2	0.93	87.2
	0.2	**200**	**87.3**	**85.4**	**89.6**	**87.6**	**0.93**	**87.5**
SMOTETomek	0.1	50	83.8	81.5	87.5	84.4	0.91	84.4
	0.1	100	86.4	84.0	88.0	86.9	0.91	86.0
	0.1	150	87.0	84.7	89.5	87.5	0.92	87.1
	0.1	200	87.1	85.0	90.0	87.9	0.93	87.5
	0.2	**200**	**87.4**	**85.3**	**90.2**	**87.9**	**0.93**	**87.7**
K-means SMOTE	0.1	50	87.0	85.1	89.4	87.2	0.91	87.2
	0.1	100	88.1	85.7	91.2	88.4	0.92	88.4
	0.1	150	87.3	84.6	90.8	87.6	0.92	87.6
	0.1	200	88.6	85.9	92.2	88.9	0.92	89.0
	0.2	**200**	**88.7**	**85.9**	**92.2**	**89.0**	**0.94**	**89.0**

TABLE 9: THE PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS USING OVERSAMPLING TECHNIQUES ON TAIWAN CLIENTS CREDIT DATASET

Oversampling Methods	Performance Measures	Random Forest	Bagging	K-Nearest Neighbor	Logistic Model Tree	AdaBoost	Stacking	GBDT Model
Random Oversampling	Accuracy (%)	**84.9**	84.1	83.1	80.1	69.3	49.6	80.4
	Precision (%)	**85.1**	85.1	84.1	81.1	71.0	53.1	79.9
	Recall (%)	**84.9**	84.1	83.1	80.1	69.3	49.6	80.7
	F-Measure (%)	**84.9**	84.1	83.1	80.1	68.6	52.2	80.3
	ROC	**0.85**	0.84	0.83	0.82	0.75	0.50	0.87
	G-Mean (%)	**85.0**	84.6	83.6	80.6	70.1	51.3	80.3
ADAYSN	Accuracy (%)	85.7	81.5	81.5	77.1	67.6	50.8	**86.9**
	Precision (%)	85.8	81.6	82.4	77.2	69.0	52.9	**85.1**
	Recall (%)	85.8	81.6	81.5	77.1	67.6	51.0	**89.9**
	F-Measure (%)	85.8	81.6	81.5	77.1	66.8	51.2	**87.4**
	ROC	0.86	0.82	0.84	0.80	0.72	0.50	**0.92**
	G-Mean (%)	85.8	81.6	81.9	77.1	68.3	51.9	**87.5**
SMOTE	Accuracy (%)	85.9	82.4	82.1	78.9	69.3	49.6	**87.6**
	Precision (%)	85.8	82.5	82.9	78.9	71.5	52.4	**85.7**

	Recall (%)	85.7	82.4	82.1	78.9	69.4	49.6	**89.9**
	F-Measure (%)	85.7	82.4	82.1	78.9	68.5	49.6	**87.8**
	ROC	0.87	0.83	0.83	0.81	0.76	0.50	**0.93**
	G-Mean (%)	85.7	82.4	82.5	78.9	70.4	51.0	**87.8**
Borderline-SMOTE	Accuracy (%)	86.2	82.5	82.5	78.5	66.6	49.6	**87.3**
	Precision (%)	86.3	82.5	83.4	78.7	67.1	51.3	**85.4**
	Recall (%)	86.3	82.5	82.5	78.6	66.6	49.6	**89.6**
	F-Measure (%)	86.2	82.5	82.4	78.5	66.3	50.2	**87.6**
	ROC	0.88	0.86	0.85	0.82	0.71	0.50	**0.93**
	G-Mean (%)	86.3	82.5	82.9	78.6	66.8	50.4	**87.5**
SMOTETomek	Accuracy (%)	86.5	82.8	84.4	79.5	70.8	49.9	**87.4**
	Precision (%)	86.6	83.0	85.1	79.1	71.9	53.6	**85.3**
	Recall (%)	86.6	82.9	84.4	79.6	70.9	49.9	**90.2**
	F-Measure (%)	86.6	82.8	84.4	79.8	70.5	51.2	**87.9**
	ROC	0.89	0.87	0.96	0.84	0.77	0.50	**0.93**
	G-Mean (%)	86.6	82.9	84.7	79.3	71.4	51.7	**87.7**
K-means SMOTE	Accuracy (%)	88.2	87.8	83.6	87.9	86.5	49.6	**88.7**
	Precision (%)	88.5	88.1	83.7	88.1	86.7	50.8	**85.9**
	Recall (%)	88.3	87.9	83.6	87.9	86.5	49.6	**92.2**
	F-Measure (%)	88.2	87.9	83.6	87.9	86.5	49.6	**89.0**
	ROC	0.91	0.89	0.87	0.89	0.91	0.50	**0.94**
	G-Mean (%)	88.4	88.0	83.6	88.0	86.6	50.2	**89.0**

After Taiwan's client's credit dataset, the South German client's credit dataset has been employed to test the performance of classifiers. The GBDT model applied to random undersampling, Near Miss, and Cluster Centroid undersampled datasets. The results are presented in Table 10 in which Random undersampling, Near Miss, and Cluster Centroid method has given the accuracy of 76.7, 74.4, and 73.3%, respectively. The Random undersampling technique outperformed the Near Miss and cluster centroids method based on accuracy. After the GBDT method, different traditional models (random forest, bagging, K-Nearest Neighbor, Logistic Model Tree, Adaboost, and Stacking) have also been utilized by utilizing undersampling techniques. These models also showed effective results on the Random undersampling method as compared to Near Miss and cluster centroids method. The performance of the GBDT method also compared with traditional machine learning models in which the GBDT method outperformed traditional machine learning models on undersampled datasets, as shown in Table 11. It has also been observed that the random forest outperformed the GBDT model while using the cluster centroids method. Furthermore, the performance of classifiers even better on the Random undersampled dataset as compared to the Near Miss and cluster centroids dataset.

TABLE 10: THE PERFORMANCE OF THE GBDT MODEL USING UNDERSAMPLING TECHNIQUES ON SOUTH GERMAN CLIENTS CREDIT DATASET

Undersampling Methods	Learning Rate	Number of decision trees	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	ROC	G-Mean (%)
Random Undersampling	0.1	50	73.9	73.3	74.2	73.7	0.83	73.7
	0.1	100	73.9	73.3	74.2	73.7	0.83	73.7
	0.1	150	74.4	74.2	74.2	74.2	0.82	74.2
	0.1	200	73.3	73.0	73.0	73.0	0.82	73.0
	0.2	**200**	**76.7**	**76.4**	**76.4**	**76.4**	**0.82**	**76.4**
Near Miss	0.1	50	72.8	73.3	70.8	72.0	0.82	72.0
	0.1	**100**	**74.4**	**73.6**	**75.3**	**74.4**	**0.83**	**74.4**
	0.1	150	74.4	73.1	76.4	74.7	0.83	74.7
	0.1	200	73.9	72.8	75.3	74.0	0.83	74.0
	0.2	200	73.9	74.4	71.9	73.1	0.82	73.1
Cluster Centroids	0.1	50	72.8	72.2	73.0	72.6	0.83	72.6
	0.1	100	72.8	73.3	70.8	72.0	0.83	72.0
	0.1	150	72.8	73.3	70.8	72.0	0.83	72.0
	0.1	200	72.8	73.8	69.7	71.7	0.83	71.7
	0.2	**200**	**73.3**	**74.1**	**70.8**	**72.4**	**0.82**	**72.4**

TABLE 11: THE PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS USING UNDERSAMPLING TECHNIQUES ON SOUTH GERMAN CLIENTS CREDIT DATASET

Undersampling Methods	Performance Measures	Random Forest	Bagging	K-Nearest Neighbor	Logistic Model Tree	AdaBoost	Stacking	GBDT Model
Random Undersampling	Accuracy (%)	75.5	73.3	73.3	72.7	71.6	48.3	**76.7**
	Precision (%)	75.8	73.7	73.5	73.1	71.7	48.9	**76.4**
	Recall (%)	75.6	73.3	73.3	72.8	71.7	48.3	**76.4**
	F-Measure (%)	75.5	73.3	73.3	72.8	71.6	48.4	**76.4**
	ROC	0.83	0.79	0.73	0.79	0.76	0.50	**0.82**

	G-Mean (%)	75.7	73.5	73.4	72.9	71.7	48.6	**76.4**
Near Miss	Accuracy (%)	73.8	72.2	58.3	75.0	68.3	48.3	**74.4**
	Precision (%)	74.0	73.3	58.4	75.1	69.3	49.2	**73.6**
	Recall (%)	73.9	72.2	58.3	75.0	68.3	48.3	**75.3**
	F-Measure (%)	73.9	72.0	58.3	75.0	67.7	49.1	**74.4**
	ROC	0.82	0.76	0.58	0.79	0.75	0.50	**0.83**
	G-Mean (%)	73.9	72.7	58.3	75.0	68.8	48.7	**74.4**
Cluster Centroid	Accuracy (%)	**75.3**	73.8	67.7	70.5	68.3	48.3	73.3
	Precision (%)	**75.3**	73.9	68.0	71.3	68.4	50.5	74.1
	Recall (%)	**75.3**	73.9	67.8	70.6	68.3	48.3	70.8
	F-Measure (%)	**75.3**	73.9	67.8	70.4	68.3	49.6	72.4
	ROC	**0.79**	0.79	0.67	0.78	0.77	0.50	0.82
	G-Mean (%)	**75.3**	73.9	67.9	70.9	68.3	49.4	72.4

Oversampling techniques (Random oversampling, ADASYN, SMOTE, Borderline-SMOTE, SMOTETomek, and K-means SMOTE) have also been utilized to enhance the performance of the classifiers on South German clients credit dataset. The GBDT model is applied to oversampled datasets. The results are presented in Table 12, in which the SMOTETomek method is given the best accuracy of 83.5%. The SMOTETomek oversampling technique given better results than all other oversampling techniques based on accuracy. After the GBDT method, different traditional models (random forest, bagging, K-Nearest Neighbor, Logistic Model Tree, Adaboost, and Stacking) have also been used by utilizing oversampling techniques. These models also have shown effective results on the SMOTE based oversampling methods as compared to others. The performance of the GBDT method is also compared with traditional machine learning models in which the GBDT model outperformed traditional machine learning models on oversampled datasets, as shown in Table 13. Furthermore, the performance of classifiers even better on the SMOTE-based oversampled datasets as compared to ADASYN and random oversampled datasets.

TABLE 12: THE PERFORMANCE OF THE GBDT MODEL USING OVERSAMPLING TECHNIQUES ON SOUTH GERMAN CLIENTS CREDIT DATASET

Oversampling Methods	Learning Rate	Number of decision trees	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	ROC	G-Mean (%)
Random Oversampling	0.1	50	73.9	73.3	74.2	73.7	0.83	73.7
	0.1	100	73.9	73.3	74.2	73.7	0.83	73.7
	0.1	150	74.4	74.2	74.2	74.2	0.82	74.2
	0.1	200	73.3	73.0	73.0	73.0	0.82	73.0
	0.2	**200**	**76.7**	**76.4**	**76.4**	**76.4**	**0.82**	**76.4**
ADAYSN	0.1	50	77.1	79.0	76.9	77.9	0.86	77.9
	0.1	100	77.8	78.0	80.6	79.3	0.87	79.3
	0.1	150	78.3	77.7	82.4	80.0	0.87	80.0
	0.1	**200**	**80.0**	**79.6**	**83.3**	**81.4**	**0.87**	**81.4**
	0.2	200	79.8	79.0	83.8	81.3	0.87	81.4
SMOTE	0.1	50	82.1	83.6	82.4	83.0	0.89	83.0
	0.1	**100**	**82.4**	**83.0**	**83.8**	**83.4**	**0.90**	**83.4**
	0.1	150	80.7	81.3	82.4	81.9	0.90	81.8
	0.1	200	80.0	81.1	82.9	82.0	0.89	82.0
	0.2	200	81.0	82.0	82.0	82.0	0.90	82.0
Borderline-SMOTE	0.1	50	81.2	82.1	82.4	82.2	0.89	82.2
	0.1	100	80.7	82.2	81.1	81.6	0.89	81.6
	0.1	150	79.3	80.5	80.2	80.4	0.89	80.3
	0.1	200	80.2	80.9	82.0	81.4	0.89	81.4
	0.2	**200**	**82.9**	**82.1**	**86.5**	**84.2**	**0.90**	**84.3**
SMOTETomek	0.1	50	82.8	82.8	84.0	83.4	0.90	83.4
	0.1	100	82.3	82.0	84.0	83.0	0.91	83.0
	0.1	150	82.8	82.2	84.9	83.5	0.91	83.5
	0.1	200	83.3	82.6	85.4	84.0	0.91	84.0
	0.2	**200**	**83.5**	**82.1**	**86.8**	**84.4**	**0.91**	**84.4**
K-means SMOTE	0.1	50	81.7	83.1	82.0	82.5	0.89	82.5
	0.1	100	82.6	82.8	84.7	83.7	0.89	83.7
	0.1	150	83.1	83.0	85.6	84.3	0.89	84.3
	0.1	200	83.1	82.7	86.0	84.3	0.90	84.3
	0.2	**200**	**83.3**	**82.8**	**86.5**	**84.6**	**0.90**	**84.6**

TABLE 13: THE PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS USING OVERSAMPLING TECHNIQUES ON SOUTH GERMAN CLIENTS CREDIT DATASET

Oversampling Methods	Performance Measures	Random Forest	Bagging	K-Nearest Neighbor	Logistic Model Tree	AdaBoost	Stacking	GBDT Model
	Accuracy (%)	73.8	76.0	78.3	78.9	71.1	48.5	**76.7**

Random Oversampling	Precision (%)	73.9	76.0	78.3	78.1	71.2	50.1	**76.4**
	Recall (%)	73.9	75.9	78.3	78.0	71.2	48.6	**76.4**
	F-Measure (%)	73.9	75.9	78.3	78.9	71.2	49.1	**76.4**
	ROC	0.79	0.80	0.81	0.81	0.77	0.50	**0.82**
	G-Mean (%)	73.9	75.9	78.3	78.0	71.2	49.3	**76.4**
ADAYSN	Accuracy (%)	**81.6**	76.0	82.3	71.8	73.5	50.8	80.0
	Precision (%)	**81.7**	76.1	84.3	71.9	73.8	49.9	79.6
	Recall (%)	**81.7**	76.0	82.4	71.9	73.6	49.9	83.3
	F-Measure (%)	**81.7**	76.0	82.1	71.9	73.5	49.9	81.4
	ROC	**0.90**	0.83	0.82	0.79	0.80	0.50	0.87
	G-Mean (%)	**81.7**	76.0	83.3	71.9	73.7	50.4	81.4
SMOTE	Accuracy (%)	82.3	80.2	82.1	77.3	74.0	48.5	**82.4**
	Precision (%)	82.6	80.2	82.2	77.4	74.3	49.2	**83.0**
	Recall (%)	82.3	80.2	82.1	77.4	74.0	48.6	**83.8**
	F-Measure (%)	82.3	80.2	82.1	77.3	74.0	48.7	**83.4**
	ROC	0.90	0.87	0.82	0.83	0.82	0.50	**0.90**
	G-Mean (%)	82.4	80.2	82.1	77.4	74.1	48.9	**83.4**
Borderline-SMOTE	Accuracy (%)	82.0	78.5	81.2	79.2	71.9	48.5	**82.9**
	Precision (%)	81.0	78.7	81.0	79.3	72.2	49.2	**82.1**
	Recall (%)	81.5	78.6	80.2	79.3	71.9	48.6	**86.5**
	F-Measure (%)	81.9	78.5	80.1	79.3	71.7	48.8	**84.2**
	ROC	0.88	0.86	0.85	0.82	0.79	0.50	**0.90**
	G-Mean (%)	81.2	78.6	80.6	79.3	72.0	48.9	**84.3**
SMOTETomek	Accuracy (%)	83.0	77.9	82.1	82.8	71.4	47.6	**83.5**
	Precision (%)	82.6	78.0	82.4	82.0	71.7	47.9	**82.1**
	Recall (%)	82.0	78.0	80.7	82.2	71.4	47.7	**86.8**
	F-Measure (%)	82.4	78.0	80.0	82.6	71.2	47.7	**84.4**
	ROC	0.88	0.85	81.0	0.82	0.79	0.50	**0.91**
	G-Mean (%)	82.3	78.0	81.5	82.1	71.5	47.8	**84.4**
K-means SMOTE	Accuracy (%)	82.4	81.9	80.7	82.6	80.0	48.5	**83.3**
	Precision (%)	82.8	81.4	80.8	82.7	80.4	48.7	**82.8**
	Recall (%)	82.1	81.2	80.7	82.6	80.0	48.6	**86.5**
	F-Measure (%)	81.1	81.2	80.7	82.6	80.0	48.6	**84.6**
	ROC	0.90	0.88	0.80	0.89	0.86	0.50	**0.90**
	G-Mean (%)	82.4	81.3	80.7	82.6	80.2	48.6	**84.6**

After the South German client's credit dataset, the Belgium client's credit dataset has been employed to test the performance of classifiers. The GBDT model applied to random undersampling, Near Miss, and Cluster Centroid undersampled datasets. The results are presented in Table 14 in which Random undersampling, Near Miss, and Cluster Centroid method has given the accuracy of 72.9, 73.9, and 76.0%, respectively. The cluster centroids technique outperformed Random undersampling and Near Miss methods based on accuracy. After the GBDT method, different traditional models (random forest, bagging, K-Nearest Neighbor, Logistic Model Tree, Adaboost, and Stacking) have also been utilized by utilizing undersampling techniques. These models also showed effective results on the cluster centroids method as compared to Random undersampling and Near Miss methods. The performance of the GBDT method also compared with traditional machine learning models in which the GBDT method outperformed traditional machine learning models on undersampled datasets, as shown in Table 15. It has also been observed that the random forest outperformed the GBDT model while using the cluster centroids method. Furthermore, the performance of classifiers even better on the Cluster Centroid dataset as compared to the Random undersampled and Near Miss dataset.

TABLE 14: THE PERFORMANCE OF THE GBDT MODEL USING UNDERSAMPLING TECHNIQUES ON BELGIUM CLIENTS CREDIT DATASET

Undersampling Methods	Learning Rate	Number of decision trees	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	ROC	G-Mean (%)
Random Undersampling	0.1	50	69.2	66.5	74.8	70.4	0.76	70.5
	0.1	**100**	**72.9**	**68.7**	**75.9**	**72.1**	**0.77**	**72.2**
	0.1	150	68.5	66.4	72.1	69.1	0.76	69.2
	0.1	200	69.8	67.5	72.4	69.9	0.77	69.9
	0.2	200	69.7	68.5	68.2	68.3	0.77	68.3
Near Miss	0.1	50	72.5	69.3	69.2	69.2	0.78	69.2
	0.1	100	70.5	70.5	75.5	72.9	0.78	73.0
	0.1	**150**	**73.9**	**70.1**	**76.1**	**73.0**	**0.79**	**73.0**
	0.1	200	71.2	70.5	71.5	71.0	0.79	71.0
	0.2	200	72.6	70.0	74.5	72.2	0.79	72.2
Cluster Centroids	0.1	50	74.6	74.0	74.6	74.3	0.81	74.3
	0.1	100	75.2	74.5	75.5	75.0	0.81	75.0

	0.1	150	75.8	75.0	76.3	75.6	0.81	75.6
	0.1	200	76.0	75.3	76.3	75.8	0.82	75.8
	0.2	**200**	**76.0**	**74.8**	**77.2**	**76.0**	**0.82**	**76.0**

TABLE 15: THE PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS USING UNDERSAMPLING TECHNIQUES ON BELGIUM CLIENTS CREDIT DATASET

Methods	Performance Measures	Random Forest	Bagging	K-Nearest Neighbor	Logistic Model Tree	Adaboost	Stacking	GBDT Model
Random Undersampling	Accuracy (%)	72.6	72.3	70.3	72.5	48.3	46.5	**72.9**
	Precision (%)	73.7	72.7	70.5	73.0	49.2	47.6	**68.7**
	Recall (%)	73.5	72.3	71.3	72.6	48.3	46.6	**75.9**
	F-Measure (%)	73.6	72.5	70.9	72.8	48.7	47.1	**72.1**
	ROC	0.78	0.77	0.77	0.78	0.49	0.50	**0.77**
	G-Mean (%)	73.6	72.5	70.9	72.8	48.7	47.1	**72.2**
Near Miss	Accuracy (%)	73.8	72.2	58.3	75.0	48.3	46.6	**73.9**
	Precision (%)	74.0	73.3	58.4	75.1	49.2	47.3	**70.1**
	Recall (%)	73.9	72.2	58.3	75.0	48.3	46.6	**76.1**
	F-Measure (%)	73.9	72.7	58.3	75.0	48.7	46.9	**73.0**
	ROC	0.82	0.76	0.58	0.79	0.49	0.50	**0.79**
	G-Mean (%)	73.9	72.7	58.3	75.0	48.7	46.9	**73.0**
Cluster Centroid	Accuracy (%)	**77.3**	71.9	70.5	76.6	53.5	46.6	76.0
	Precision (%)	**77.5**	71.8	70.4	76.4	54.5	47.4	74.8
	Recall (%)	**79.5**	71.8	70.9	77.2	53.4	46.6	77.2
	F-Measure (%)	**78.5**	71.8	70.6	76.8	53.9	47.0	76.0
	ROC	**0.83**	0.78	0.76	0.81	0.54	0.50	0.82
	G-Mean (%)	**78.5**	71.8	70.6	76.8	53.9	47.0	76.0

Oversampling techniques (Random oversampling, ADASYN, SMOTE, Borderline-SMOTE, SMOTETomek, and K-means SMOTE) have also been utilized to enhance the performance of the classifiers on Belgium clients credit dataset. The GBDT model is applied to oversampled datasets. The results are presented in Table 16, in which the K-Means SMOTE method is given the best accuracy of 86.3%. The K-Means SMOTE oversampling technique given better results than all other oversampling techniques based on accuracy. After the GBDT method, different traditional models (random forest, bagging, K-Nearest Neighbor, Logistic Model Tree, Adaboost, and Stacking) have also been used by utilizing oversampling techniques. These models also have shown effective results on the SMOTE based oversampling methods as compared to others. The performance of the GBDT method is also compared with traditional machine learning models in which the GBDT model outperformed traditional machine learning models on oversampled datasets, as shown in Table 17. It has also been observed that the random forest outperformed the GBDT model while using the ADAYSN oversampling method. Furthermore, the performance of classifiers even better on the SMOTE-based oversampled datasets as compared to ADASYN and random oversampled datasets.

TABLE 16: THE PERFORMANCE OF THE GBDT MODEL USING OVERSAMPLING TECHNIQUES ON BELGIUM CLIENTS CREDIT DATASET

Oversampling Methods	Learning Rate	Number of decision trees	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	ROC	G-Mean (%)
Random Oversampling	0.1	50	78.9	78.3	78.4	78.3	0.83	78.3
	0.1	100	78.9	78.3	78.1	78.2	0.83	78.2
	0.1	150	79.4	79.2	74.2	76.6	0.83	76.7
	0.1	**200**	**79.9**	**79.0**	**79.0**	**79.0**	**0.84**	**79.0**
	0.2	200	78.3	78.4	78.4	78.4	0.84	78.4
ADAYSN	0.1	50	79.9	79.0	79.9	79.4	0.84	79.4
	0.1	100	79.8	79.0	83.8	81.3	0.84	81.4
	0.1	150	78.3	77.7	82.4	80.0	0.85	80.0
	0.1	200	79.3	80.5	80.2	80.3	0.85	80.3
	0.2	**200**	**81.8**	**81.0**	**83.8**	**82.4**	**0.85**	**82.4**
SMOTE	0.1	50	79.2	82.6	81.6	82.1	0.89	82.1
	0.1	**100**	**82.5**	**82.0**	**81.8**	**81.9**	**0.89**	**81.9**
	0.1	150	79.6	80.5	81.6	81.0	0.88	81.0
	0.1	200	79.5	80.3	81.6	80.9	0.88	80.9
	0.2	200	80.2	82.0	83.6	82.8	0.88	82.8
Borderline-SMOTE	0.1	50	83.3	84.5	84.0	84.2	0.90	84.2
	0.1	100	82.6	84.6	83.6	84.1	0.90	84.1
	0.1	150	81.4	82.6	82.5	82.5	0.90	82.5
	0.1	200	82.1	82.6	84.3	83.4	0.91	83.4
	0.2	**200**	**82.9**	**82.1**	**88.6**	**85.2**	**0.91**	**85.3**
SMOTETomek	0.1	50	82.8	82.8	84.0	83.4	0.91	83.4
	0.1	100	83.0	83.0	84.5	83.7	0.91	83.7
	0.1	150	83.1	83.2	85.9	84.5	0.91	84.5

	0.1	**200**	**83.5**	**83.6**	**86.4**	**85.0**	**0.92**	**85.0**
	0.2	200	83.3	83.1	86.0	84.5	0.92	84.5
K-means SMOTE	0.1	50	85.7	86.1	87.0	86.5	0.92	86.5
	0.1	100	85.6	86.8	87.7	87.2	0.92	87.2
	0.1	150	86.1	86.1	87.6	86.8	0.93	86.8
	0.1	200	86.1	86.7	87.0	86.8	0.93	86.8
	0.2	**200**	**86.3**	**86.8**	**87.5**	**87.1**	**0.93**	**87.1**

TABLE 17: THE PERFORMANCE COMPARISON OF TRADITIONAL MACHINE LEARNING MODELS USING OVERSAMPLING TECHNIQUES ON BELGIUM CLIENTS CREDIT DATASET

Oversampling Methods	Performance Measures	Random Forest	Bagging	K-Nearest Neighbor	Logistic Model Tree	Adaboost	Stacking	GBDT Model
Random Oversampling	Accuracy (%)	78.0	77.6	77.5	76.9	58.4	49.6	**79.9**
	Precision (%)	78.5	77.9	77.5	76.1	58.4	50.1	**79.0**
	Recall (%)	78.5	77.5	77.5	76.0	58.4	49.6	**79.0**
	F-Measure (%)	78.5	77.7	77.5	76.0	58.4	49.8	**79.4**
	ROC	0.82	0.81	0.81	0.81	0.58	0.50	**0.84**
	G-Mean (%)	78.5	77.7	77.5	76.0	58.4	49.8	**79.0**
ADAYSN	Accuracy (%)	82.1	77.3	71.8	80.2	58.4	49.6	**81.8**
	Precision (%)	82.2	77.4	71.9	80.2	58.4	50.5	**81.0**
	Recall (%)	82.1	77.4	71.9	80.2	58.4	49.6	**83.8**
	F-Measure (%)	82.1	77.4	71.9	80.2	58.4	50.0	**82.4**
	ROC	0.82	0.83	0.79	0.87	0.58	0.50	**0.85**
	G-Mean (%)	82.1	77.4	71.9	80.2	58.4	50.0	**82.4**
SMOTE	Accuracy (%)	78.5	80.2	82.0	82.3	60.6	49.6	**82.5**
	Precision (%)	78.7	80.2	81.0	82.6	60.7	50.3	**82.0**
	Recall (%)	78.6	80.2	81.5	82.3	60.6	49.6	**81.8**
	F-Measure (%)	78.6	80.2	81.2	82.4	60.6	49.9	**81.9**
	ROC	0.85	0.88	0.88	0.88	0.60	0.50	**0.89**
	G-Mean (%)	78.6	80.2	81.2	82.4	60.6	49.9	**81.9**
Borderline-SMOTE	Accuracy (%)	81.2	82.6	82.4	79.2	58.4	49.6	**82.9**
	Precision (%)	81.0	82.7	82.8	79.3	58.4	50.2	**82.1**
	Recall (%)	80.2	82.6	82.1	79.3	58.4	49.6	**88.6**
	F-Measure (%)	80.6	82.6	82.4	79.3	58.4	49.9	**85.2**
	ROC	0.89	0.89	0.90	0.89	0.58	0.50	**0.91**
	G-Mean (%)	80.6	82.6	82.4	79.3	58.4	49.9	**85.3**
SMOTETomek	Accuracy (%)	82.0	79.9	84.1	84.8	54.8	49.6	**83.5**
	Precision (%)	84.6	80.0	84.4	84.0	58.8	50.5	**83.6**
	Recall (%)	84.0	80.0	82.7	84.2	54.8	49.6	**86.4**
	F-Measure (%)	84.3	80.0	83.5	84.1	56.7	50.0	**85.0**
	ROC	0.91	0.90	0.91	0.90	0.58	0.50	**0.92**
	G-Mean (%)	84.3	80.0	83.5	84.1	56.8	50.0	**85.0**
K-means SMOTE	Accuracy (%)	85.0	85.9	85.7	85.5	54.8	49.6	**86.3**
	Precision (%)	85.0	85.4	86.8	85.9	58.8	50.2	**86.8**
	Recall (%)	85.9	85.2	86.7	86.0	54.8	49.6	**87.5**
	F-Measure (%)	85.4	85.3	86.7	85.9	56.7	49.9	**87.1**
	ROC	0.93	0.92	0.92	0.92	0.58	0.50	**0.93**
	G-Mean (%)	85.4	85.3	86.7	85.9	56.8	49.9	**87.1**

The results obtained through various imbalanced datasets showed that the GBDT model outperformed the traditional machine learning models based on undersampling and oversampling techniques. While tuning the GBDT model, the learning rate and the number of constructed trees was tuned randomly. The best results have been obtained when the learning rate was set to 0.2, and the number of constructed trees was 200. The results of undersampling and oversampling techniques have also been compared while trained with a GBDT model. Results showed that SMOTE based oversampling methods outperformed other oversampling technique as well as undersampling techniques which are used in this experiment. The performance of the GBDT model on various datasets is shown in Figure 3. Various imbalanced datasets like lending club dataset [13, 14], Chinese P2P lending company dataset [15], German credit dataset, Australian credit dataset, and Dataset of We.com [16], Chinese consumer finance company dataset [17] were used in the past. Previous studies [2, 11, 13, 15-18, 22-26] were not deployed in the models for end-users.

Furthermore, these studies were not given efficient results due to a high imbalance of data because these studies were not balanced the dataset. But, in this study, we used various resampling techniques to cater to the class imbalance problem. Results from Figure 3 reveals that the critical behavior of the evaluated resampling techniques. After analyzing the results, the GBDT model produces better results compared to other classifiers. A deeper analysis shows that the GBDT model gives fair results on all balancing techniques, but other techniques give a lower performance. The results also show that the GBDT model outperformed other models on various datasets, i.e., Taiwan client's credit dataset, South German client's credit dataset, and Belgium client's credit dataset. The most effective results have been obtained on the Taiwan client's credit dataset. The performance of the proposed model has been significantly better than previous studies, as shown in Table 18.

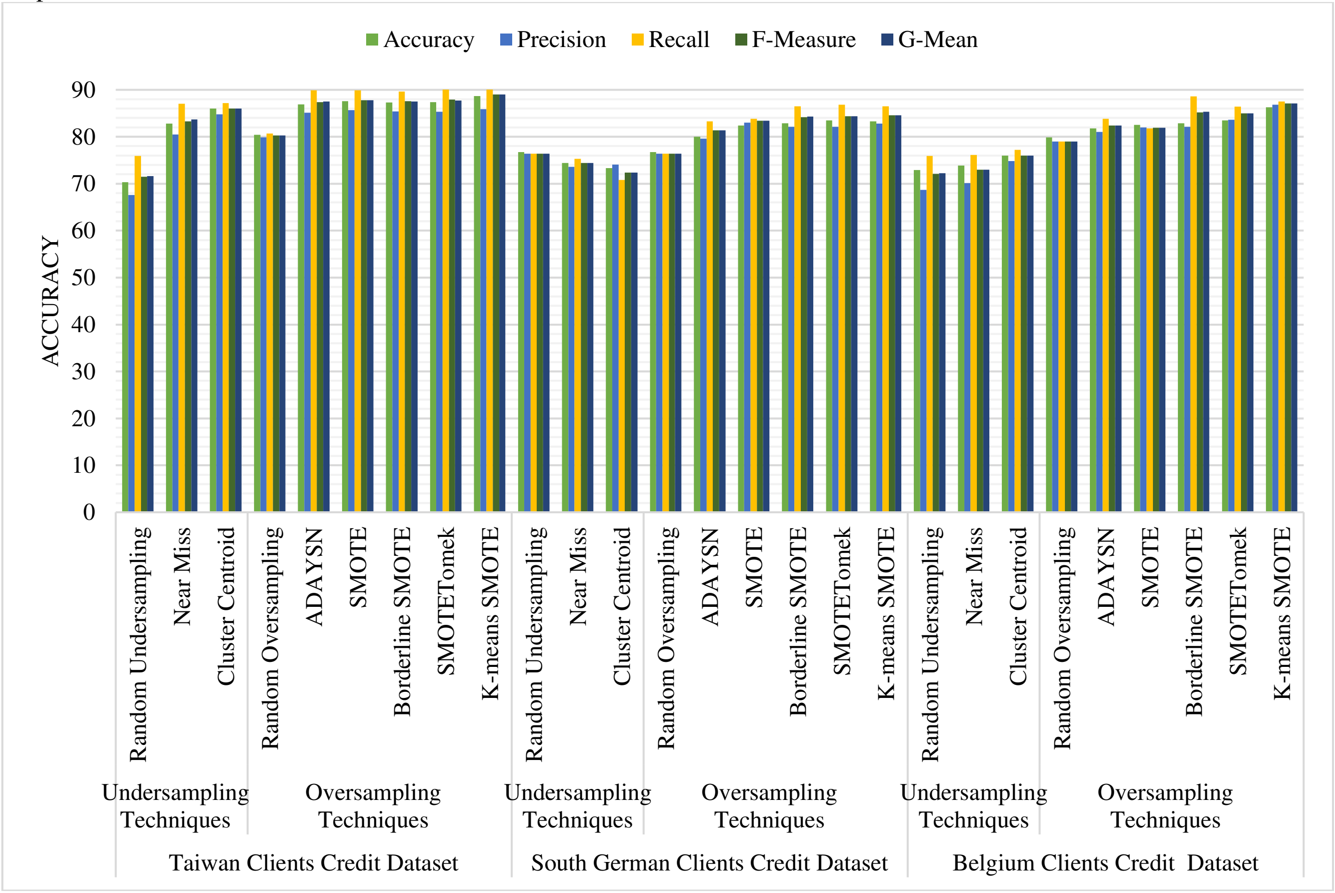


FIGURE 3. The performance comparison of the GBDT model with the combination of resampling techniques for each dataset

5.1 Statistical Analysis

We further performed hypothesis testing by calculating the p-value. Null hypothesis statistical testing (NHST) is beneficial to interpret results and ensures the claim of improved performance is backed up by statistical analysis. A p-value that is smaller than the significance level (often 0.05) is considered statistically significant. In this study, we used the Analysis of Variance (ANOVA) test. When the p-value is lower than the significance level, we reject the null hypothesis and conclude that the data support the alternative hypothesis. A one way ANOVA design has been used to detect differences in results based on accuracy, precision, recall, f-measure, ROC, and G-mean. The p-value of 0.05 (level of significance) has been used for each statistical analysis.

We addressed the issues, whether developed models using different machine learning techniques are significantly the same or different, and whether resampling techniques significantly improve the performance of the models. Figure 4 plots the 6 performance metric (accuracy, precision, recall, F-Measure, ROC, G-Mean) of base-line machine learning methods versus our proposed models with the use of the imbalance techniques for the three datasets including Taiwan, South German, and Belgium. As we can see in Figure 4 that the six performance evaluation metric are higher for our proposed models using over-sampling techniques for the three data dataset. P-value was calculated using the One-Way ANOVA test which turned out to be <0.002 for the Taiwan client credit dataset and <0.001 for South German, and Belgium client credit datasets.

Therefore our statistical testing rejected the Null hypothesis on three credit datasets, i.e., Taiwan, South German, and Belgium. All the results show that the proposed method using imbalanced techniques has significantly improved the performance from the baseline method as p-values in all cases were our statistical threshold of 0.05.

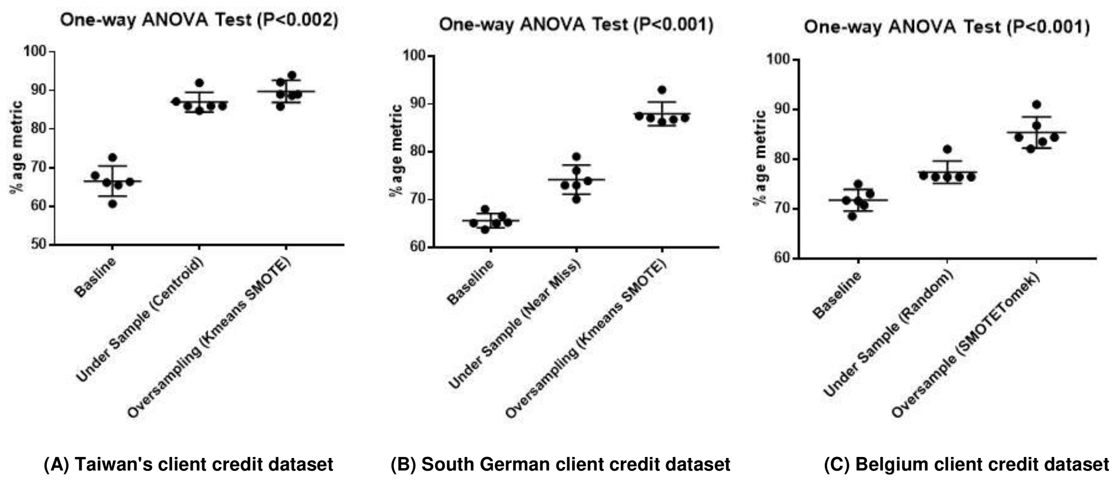


Figure 4: Hypothesis testing using One-way ANOVA test

5. DISCUSSION

For each classifier, we first described the general trends in our results and then analyzed these general trends. The clients we are trying to classify our account holders, so we have enough information about them. Without knowing anything about the client's spending patterns, it is hard to separate clients that have no intention of paying their debt from the clients that are merely taking advantage of the credit and will pay back later. Since we are making predictions based on salary statements of the previous quarter, bill payments of the previous quarter, and repayment status of the last quarter to predict delinquency in the next quarter, it is reassuring to see that credit amount, marital status, and education level are also significant features for our prediction problem. These features were generated by aggregating the data in the statements dataset.

Furthermore, socio-economic indicators, history of past payments, amount of bill statements, and amount of previous payments were features related to the loan itself, so it is reasonable that they are included to build a model. Finally, age and gender were also included as well: studies have shown that age is also correlated to income, and scoring is based on the creditworthiness of the client. It is also observed that the accuracy of the classifier is also associated with class balance; in each dataset, the accuracy improves where the number of minority samples increased. It also implies that as the number of instances increased then, the classifier has more chances to learn the patterns to separate the binary classes. For our results, this can suggest that the models recognize more generalized patterns of clients that are likely to default, and not necessarily patterns of clients that are soon going to default.

Generally, the performance of different classifiers is high on imbalanced datasets because of the overrepresentation of the majority class and under-representation of the minority class. This also implies the weakness of classifying minority class to calculate the accuracy of the model. Loyola-González et al. [45] pointed out that the accuracy of the model biased towards the majority class and minority class anticipated less to determine the accuracy even if the accuracy of the model is higher. A balanced dataset has been utilized to eliminate the biases of the majority class to overcome this problem. The GBDT model has the highest accuracy with the K-means SMOTE method (88.7%), followed by RF (88.2% with SMOTE). The best result has been obtained on Taiwan's client credit dataset. The results of our study outperformed other studies, as shown in Table 18. In credit assessment applications, a little enhancement in performance can prompt critical future investment funds and huge monetary effects. Therefore, improving the accuracy of resampling techniques can be significant for banks and financial institutions. The different classification techniques with nine datasets (Random Undersampling, Near Miss, Cluster Centroid, Random Oversampling, ADASYN, SMOTE, Borderline-SMOTE, SMOTETomek, K-means SMOTE) showed different performances to identify healthy and defaulter clients.

It has also been observed that in previous studies machine learning models, i.e., random forest [11], stacking [16], the GBDT model [22], logistic model tree [23], bagging [24], and k-nearest neighbor [26] were not given efficient results on imbalanced data on credit card default prediction data, as shown in Figure 5. In previous research, Random forest [11] was given the accuracy of 58.8%, but while combining the random forest with K-means SMOTE oversampling technique, the result has been significantly improved with the accuracy of 88.2%. On the contrary, stacking [16] was given accuracy of 78.8%, but while combining the stacking with SMOTETomek oversampling technique, the result has not been significantly improved. The GBDT model [22] was given accuracy of 82%, but while combining the GBDT model with K-means SMOTE oversampling technique, the results have been improved with the accuracy of 88.7%. The logistic model tree [23] was given the accuracy of 69% but while combining the logistic model tree with K-means SMOTE oversampling technique, the results have been significantly improved with the accuracy of 87.9%. The bagging model [24] and k-nearest neighbor was given the accuracy of 69 and 82%, but while combining the logistic model tree and k-nearest neighbor with K-means SMOTE oversampling technique, the results have been significantly improved with the accuracy of 87.9 and 83.1% respectively. The execution of oversampling methods could assist budgetary institutions with reducing the expense of misclassification contrasted with the original dataset.

Our model would have an extraordinary down to earth sway on banks and can ensure an upper hand over different banks that do not actualize this technique. These outcomes indicated that the resampling methodologies could more readily distinguish sound more significant part borrowers and wiped out minority borrowers than those in the original dataset. By creating similar instances to the existing minority instances, SMOTE based methods generate higher and less precise decision boundaries that enhance the generalization competences of the classifiers, therefore increasing their performance. However, SMOTE based methods have some associated issues, such as the problem of over-generalization (the new synthetic examples may be generated in overlapping areas) and also the possibility of augmenting noisy regions (since no distinction between different types of minority examples is performed). Despite this, it seems that its ability to generate more significant decision boundaries is still a considerable strength, even with its susceptibilities. Due to ADASYN adaptability nature that allows creating more data in neighborhoods with high amounts of majority class examples, the synthetic data generated might be very similar to the majority class data, potentially making many false positives. The other SMOTE variants and ADASYN differ from each other by selecting the samples x_i ahead of generating the new samples. For minority examples that are sparsely distributed, each neighborhood may only contain one minority example.

Credit institutions and consumer finance companies decide the issuance of the loan based on credit card records. However, it can be difficult for some borrowers to determine lending criteria and assess the trustworthiness of credit card sources over the Internet. A few borrowers may not give enough proof to reinforce their certainty. For instance, a few people did not have enough property, and most developing countries have constrained banking records of their clients. Subsequently, the proposed model can anticipate the default prediction of credit cards based on their previous data. The conveyed and progressively reliable credit card information is utilized to make the model increasingly all-inclusive, which can guarantee the performance of the proposed model. Consumer finance companies may settle on choices about various credits from the collection of applications. In any case, conventional financial organizations anticipate a credit choice through a manual audit, which leads to a high cost and also labor. This low-productivity, significant expense conjectures cannot meet consumer finance enterprise's credit choice requirements. Because of the abundance of money related and non-budgetary indicators, Decision-makers face the issue of choosing pertinent data. This model will help them to make the decision based on the history of credit card effectively. It also gives the results very fast, which will also save time as well as the cost of labor. Financial institutions may acquire additional information from the Internet to improve the performance of the model to increase the reliability of the model as well as decrease the risk of management capabilities. To validate the performance of our proposed technique, we have verified on three credit-related datasets of different countries (Taiwan, South German, and Belgium). The prototype has also developed, and it can be applied to any real-time dataset.

TABLE 18: COMPARISON OF PREVIOUS STUDIES FOR CREDIT CARD DEFAULT PREDICTION

Reference	Year	Dataset	Features Count	Total Instances	Method	Results
[26]	2009	Default credit card client's dataset	24	30000	K-nearest neighbor	Accuracy=82.0%
[18]	2010	Major commercial USA bank	138	2.2 million	Linear regression	Accuracy=85.0%
[2]	2016	Six major USA financial institutions	186	500 million	Random forest	Accuracy=39 to 82%
[25]	2017	Default credit card client's dataset	24	30000	Decision Tree	Accuracy=80.3%
[16]	2018	German Credit Dataset	24	1000	Ensemble fusion techniques based on bstacking method	Accuracy=78.8%
		Australian credit dataset	14	600		Accuracy=84.6%
		Dataset of We.com	17	1421		Accuracy=84.0%
		Lending Club dataset	11	2642		Accuracy=66.7%
[11]	2018	Default credit card client's dataset	24	30000	Random forest	Accuracy=58.83%
[17]	2018	Chinese consumer finance company	490	44000	XGBoost	AUC=0.71
[24]	2018	Default credit card client's dataset	24	30000	Bagging	Accuracy=73.4%
[23]	2019	Default credit card client's dataset	24	30000	Logistic model tree	Accuracy =69.0%
[15]	2019	Chinese P2P lending company dataset	1138	15000	XGBoost	AUC=0.71
[22]	2019	Default credit card client's dataset	24	30000	Gradient boosting tree	Accuracy=82.0%
[13]	2020	Lending Club dataset	15	64139	Logistic regression	Accuracy=79.2%
Our Method	**2020**	**Taiwan credit client's dataset**	**24**	**30000**	**Gradient Boosted Decision Tree Model**	**Accuracy=88.7%**
		South German credit client's dataset	**21**	**1000**		**Accuracy=83.5%**
		Belgium credit client's dataset	**28**	**285299**		**Accuracy=86.3%**

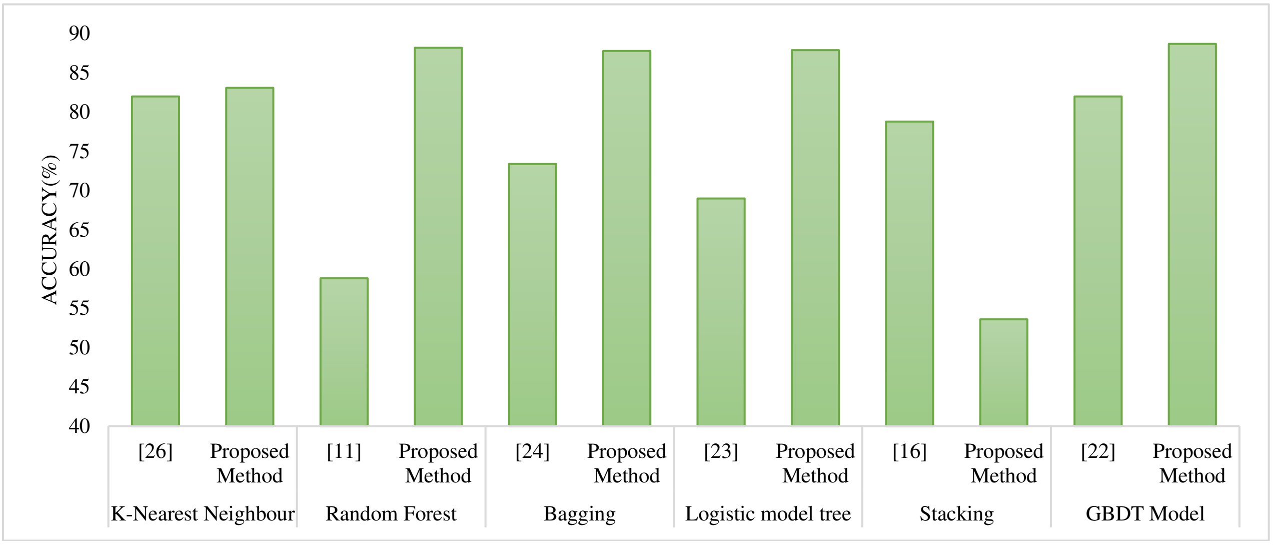


FIGURE 5. The comparison of previous studies with the proposed model

6. DEPLOYMENT

The experimental results review the effect of various degrees of the credit-related imbalanced datasets for training on credit card default prediction model. Various machine learning models were also deployed in the domain of cyber security [46, 47], healthcare [48, 49], education [50, 51] The most efficient results have been obtained through Taiwan's client credit dataset. So the learned weights of that dataset have been employed for the deployment of the model One of the principal objectives in building a model that precisely predicts results and is robust to changes in future information. Deployment is the phase of the proposed method that guarantees that the data mining process is repeatable for all organizations. The valuable information extracted from the data must be sorted out and introduced with the goal that any stakeholder can utilize it. Since the GBDT method is given the most effective result using K-Means SMOTE, so it has been employed for the deployment for training purposes.

Furthermore, we discovered that deployment choices could positively affect the acceptance of a data mining solution. Our method of implementation enabled the end-users that are not data miners to engage in scenario simulation activities of the complex system. In the past, the deployment was done manually by using traditional methods. It is also difficult to deploy the model using R or python because without using API, implementation is very tough. The businesses may be stuck with old models if the deployment is robust or expensive. Microsoft Azure Machine Learning Studio was utilized to actualize the model for credit card default prediction. This can not only deploy the model but also automatically sets up the model to work with Azure's load-balancing technology. Deployment is also a challenging task when data is multi-dimensional. The model has also been published to the Microsoft Azure Marketplace, and the URL of our model is [52]. A similar deployment was also done for corporate bankruptcy prediction with the help of Microsoft Azure Machine Learning Studio [53]. The model is visible to all, which is the smartest way to target the stakeholders. It is mandatory to provide all inputs in the right direction to get effective results. Just signing up for Microsoft Azure Machine Learning Studio, any stakeholder can use the model to check the status of clients, as shown in Figure 7. The complete workflow of deployment is shown in Figure 6.

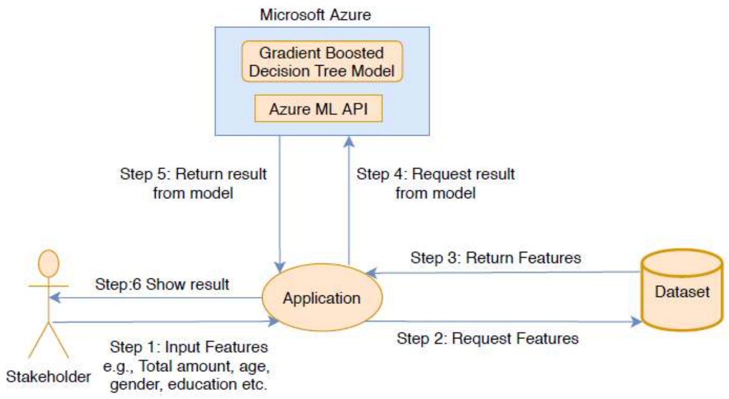
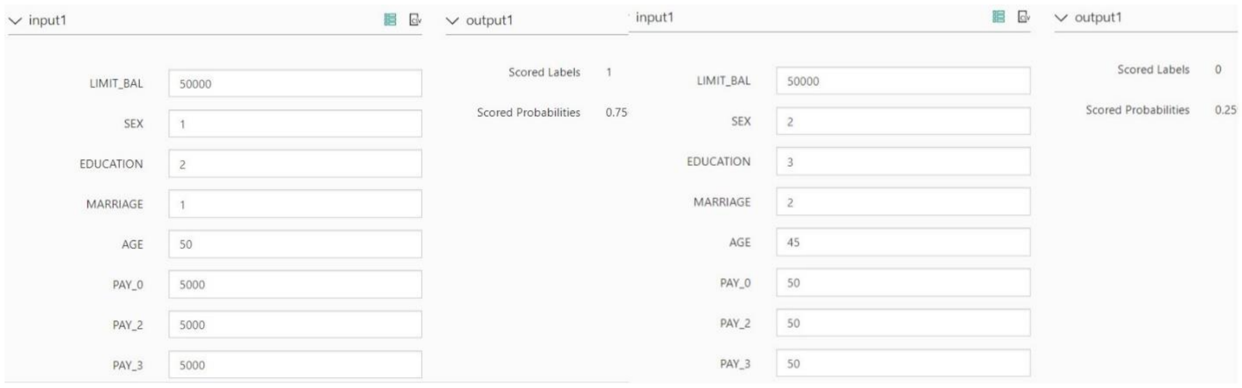

Microsoft Azure
Gradient Boosted Decision Tree Model
Azure ML API
Step 5: Return result from model
Step 4: Request result from model
Step 3: Return Features
Step:6 Show result
Application
Dataset
Step 2: Request Features
Step 1: Input Features e.g., Total amount, age, gender, education etc.
Stakeholder

FIGURE 6. The complete workflow of the deployment phase


input1

LIMIT_BAL	50000
SEX	1
EDUCATION	2
MARRIAGE	1
AGE	50
PAY_0	5000
PAY_2	5000
PAY_3	5000

output1

Scored Labels	1
Scored Probabilities	0.75

input1

LIMIT_BAL	50000
SEX	2
EDUCATION	3
MARRIAGE	2
AGE	45
PAY_0	50
PAY_2	50
PAY_3	50

output1

Scored Labels	0
Scored Probabilities	0.25

FIGURE 7. Deployment module which predicted defaulter and healthy clients

7. CONCLUSION

Machine learning methods, in conjunction with the use of imbalanced methods, have been utilized in various domains [54-55]. The objective of this paper is to train various supervised learning algorithms to predict the client's behavior in paying off the credit card balance. In classification problems, an imbalanced dataset is also crucial to enhance the performance of the model, so different resampling techniques were also used to balance the dataset. We first investigated the datasets by using exploratory data analysis techniques, including data normalization. We started with the GBDT model, then compared the results with traditional machine learning-based models. The prediction accuracy rate of the GBDT model is higher than the traditional machine learning-based models. The GBDT method given the best accuracy of 88.7% while utilizing the K-means SMOTE resampling method on Taiwan client's credit dataset. The results obtained through Taiwan client's credit dataset have significantly better than other datasets employed in this study.

In the end, the proposed method has also been deployed on the web to assist the different stakeholders. Therefore, when the financial institution considers issuing the client a credit card, the institution needs to check the payment history of that person because the decision on whether pay on duly or owe the bill on a specific month usually relates to the previous payment history. For instance, if a person owes numerous bills already, he or she is likely to delay the payment of the current month unless this person gets a windfall so that the total arrears can be paid off. Besides the payment history, it is also imperative to look at the applicants' credit limit of their current credit cards. This is a result of a virtuous circle: people who pay on duly tend to have better credit scores, so the banks prefer to increase these people's credit lines by taking less risk. As a result, if a potential client already has a credit card with a high credit limit line, this person is improbable to fail to pay the full amount owed in the future. Although the financial institution often collects clients' personal information such as age, educational level, and marital status when people apply for credit cards, this information also affects the default behavior. In other words, the financial institution should equally consider their potential clients who are men or women, obtain bachelor degrees or master degrees, single or married when decide whether approve their credit card/loan applications. We tried our best to make a thorough analysis, and there are still a few possible improvements that may require longer-term action. For the boosting models, only the GBDT method was trained, but various variants of boosting techniques may also be utilized in the future. The financial market changes rapidly every day, and people's economic status and performance are affected by the market all the time. So, if more economic indicators will be added to the dataset, this will leads to a more generic model.

8. REFERENCES

[1] H. Kim, H. Cho, and D. Ryu, "An empirical study on credit card loan delinquency," *Economic Systems,* vol. 42, pp. 437-449, 2018.

[2] F. Butaru, Q. Chen, B. Clark, S. Das, A. W. Lo, and A. Siddique, "Risk and risk management in the credit card industry," *Journal of Banking & Finance,* vol. 72, pp. 218-239, 2016.

[3] H. A. Bekhet and S. F. K. Eletter, "Credit risk assessment model for Jordanian commercial banks: Neural scoring approach," *Review of Development Finance,* vol. 4, pp. 20-28, 2014.

[4] M. Leo, S. Sharma, and K. Maddulety, "Machine learning in banking risk management: A literature review," *Risks,* vol. 7, p. 29, 2019.

[5] E. W. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decision support systems,* vol. 50, pp. 559-569, 2011.

[6] S. Tian and Y. Yu, "Financial ratios and bankruptcy predictions: An international evidence," *International Review of Economics & Finance,* vol. 51, pp. 510-526, 2017.

[7] G. Kou, X. Chao, Y. Peng, F. E. Alsaadi, and E. Herrera-Viedma, "Machine learning methods for systemic risk analysis in financial sectors," *Technological and Economic Development of Economy,* vol. 25, pp. 716-742, 2019.

[8] F. Barboza, H. Kimura, and E. Altman, "Machine learning models and bankruptcy prediction," *Expert Systems with Applications,* vol. 83, pp. 405-417, 2017.

[9] F. Ciampi, "Corporate governance characteristics and default prediction modeling for small enterprises. An empirical analysis of Italian firms," *Journal of Business Research,* vol. 68, pp. 1012-1025, 2015.

[10] M. Tkáč and R. Verner, "Artificial neural networks in business: Two decades of research," *Applied Soft Computing,* vol. 38, pp. 788-804, 2016.

[11] S. Hamori, M. Kawai, T. Kume, Y. Murakami, and C. Watanabe, "Ensemble learning or deep learning? Application to default risk analysis," *Journal of Risk and Financial Management,* vol. 11, p. 12, 2018.

[12] X. Feng, Z. Xiao, B. Zhong, J. Qiu, and Y. Dong, "Dynamic ensemble classification for credit scoring using soft probability," *Applied Soft Computing,* vol. 65, pp. 139-151, 2018.

[13] Y. Xia, L. He, Y. Li, N. Liu, and Y. Ding, "Predicting loan default in peer-to-peer lending using narrative data," *Journal of Forecasting,* vol. 39, pp. 260-280, 2020.

[14] H. Wang, G. Kou, and Y. Peng, "Multi-class misclassification cost matrix for credit ratings in peer-to-peer lending," *Journal of the Operational Research Society,* pp. 1-12, 2020.

[15] J. Zhou, W. Li, J. Wang, S. Ding, and C. Xia, "Default prediction in P2P lending from high-dimensional data based on machine learning," *Physica A: Statistical Mechanics and its Applications,* vol. 534, p. 122370, 2019.

[16] Y. Xia, C. Liu, B. Da, and F. Xie, "A novel heterogeneous ensemble credit scoring model based on bstacking approach," *Expert Systems with Applications,* vol. 93, pp. 182-199, 2018.

[17] W. Li, S. Ding, Y. Chen, H. Wang, and S. Yang, "Transfer learning-based default prediction model for consumer credit in China," *The Journal of Supercomputing,* vol. 75, pp. 862-884, 2019.

[18] A. E. Khandani, A. J. Kim, and A. W. Lo, "Consumer credit-risk models via machine-learning algorithms," *Journal of Banking & Finance,* vol. 34, pp. 2767-2787, 2010.

[19] F. Wen, L. Xu, G. Ouyang, and G. Kou, "Retail investor attention and stock price crash risk: Evidence from China," *International Review of Financial Analysis,* vol. 65, p. 101376, 2019.

[20] G. Kou, Y. Peng, and G. Wang, "Evaluation of clustering algorithms for financial risk analysis using MCDM methods," *Information Sciences,* vol. 275, pp. 1-12, 2014.

[21] D. Veganzones and E. Séverin, "An investigation of bankruptcy prediction in imbalanced datasets," *Decision Support Systems,* vol. 112, pp. 111-124, 2018.

[22] O. J. Leong and M. Jayabalan, "A Comparative Study on Credit Card Default Risk Predictive Model," *Journal of Computational and Theoretical Nanoscience,* vol. 16, pp. 3591-3595, 2019.

[23] T.-C. Hsu, S.-T. Liou, Y.-P. Wang, and Y.-S. Huang, "Enhanced Recurrent Neural Network for Combining Static and Dynamic Features for Credit Card Default Prediction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 1572-1576.

[24] P. Xu, Z. Ding, and M. Pan, "A hybrid interpretable credit card users default prediction model based on RIPPER," *Concurrency and Computation: Practice and Experience,* vol. 30, p. e4445, 2018.

[25] M. Pasha, M. Fatima, A. M. Dogar, and F. Shahzad, "Performance comparison of data mining algorithms for the predictive accuracy of credit card defaulters," *Int. J. Comput. Sci. Netw. Secur,* vol. 17, pp. 178-183, 2017.

[26] I.-C. Yeh and C.-h. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Systems with Applications,* vol. 36, pp. 2473-2480, 2009.

[27] U. Grömping, "South German Credit Data: Correcting a Widely Used Data Set," *Reports in Mathematics, Physics and Chemistry,* 2019.

[28] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert systems with applications,* vol. 41, pp. 4915-4928, 2014.

[29] I. Mani, "kNN approach to unbalanced data distributions: a case study involving information extraction."

[30] S.-J. Yen and Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Systems with Applications,* vol. 36, pp. 5718-5727, 2009.

[31] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, 2008, pp. 1322-1328.

[32] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research,* vol. 16, pp. 321-357, 2002.

[33] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering,* vol. 21, pp. 1263-1284, 2009.

[34] G. E. Batista, A. L. Bazzan, and M. C. Monard, "Balancing Training Data for Automated Annotation of Keywords: a Case Study," in *WOB*, 2003, pp. 10-18.

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

[36] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics,* pp. 1189-1232, 2001.

[37] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing,* vol. 300, pp. 70-79, 2018.

[38] M. Zaffar, M. A. Hashmani, and K. Savita, "Performance analysis of feature selection algorithm for educational data mining," in *2017 IEEE Conference on Big Data and Analytics (ICBDA)*, 2017, pp. 7-12.

[39] M. U. Ghani, T. M. Alam, and F. H. Jaskani, "Comparison of Classification Models for Early Prediction of Breast Cancer," in *2019 International Conference on Innovative Computing (ICIC)*, 2019, pp. 1-6.

[40] H. Rao, X. Shi, A. K. Rodrigue, J. Feng, Y. Xia, M. Elhoseny*, et al.*, "Feature selection based on artificial bee colony and gradient boosting decision tree," *Applied Soft Computing,* vol. 74, pp. 634-642, 2019.

[41] Z. Xu, G. Huang, K. Q. Weinberger, and A. X. Zheng, "Gradient boosted feature selection," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 522-531.

[42] T. M. Alam, M. M. A. Khan, M. A. Iqbal, W. Abdul, and M. Mushtaq, "Cervical cancer prediction through different screening methods using data mining," *IJACSA) International Journal of Advanced Computer Science and Applications,* vol. 10, 2019.

[43] X. Chao, G. Kou, Y. Peng, and F. E. Alsaadi, "Behavior monitoring methods for trade-based money laundering integrating macro and micro prudential regulation: a case from China," *Technological and Economic Development of Economy,* vol. 25, pp. 1081-1096, 2019.

[44] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognition,* vol. 91, pp. 216-231, 2019.

[45] O. Loyola-González, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, and M. García-Borroto, "Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases," *Neurocomputing,* vol. 175, pp. 935-947, 2016.

[46] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, S. Chen, D. Liu, *et al.*, "Performance Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity," *Energies,* vol. 13, p. 2509, 2020.

[47] K. Shaukat, A. Rubab, I. Shehzadi, and R. Iqbal, "A Socio-Technological analysis of Cyber Crime and Cyber Security in Pakistan," *Transylvanian Review,* vol. 1, 2017.

[48] T. M. Alam and M. J. Awan, "Domain Analysis of Information Extraction Techniques," *International Journal of Multidisciplinary Sciences and Engineering,* vol. 9, pp. 1-9.

[49] Y. Ali, A. Farooq, T. M. Alam, M. S. Farooq, M. J. Awan, and T. I. Baig, "Detection of Schistosomiasis Factors using Association Rule Mining," *IEEE Access,* vol. 7, pp. 186108-186114, 2019.

[50] K. Shaukat, I. Nawaz, S. Aslam, S. Zaheer, and U. Shaukat, "Student's performance in the context of data mining," in *2016 19th International Multi-Topic Conference (INMIC)*, 2016, pp. 1-8.

[51] K. Shaukat, I. Nawaz, and S. Zaheer, *Students Performance: A Data Mining Perspective*: LAP Lambert Academic Publishing, 2017.

[52] T. M. Alam. (2020). *Model Deployement on Microsoft Azure Machine Learning Studio*. Available: https://gallery.azure.ai/Experiment/Credit-Card-Default-Prediction-Model

[53] T. M. Alam, K. Shaukat, M. Mushtaq, Y. Ali, M. Khushi, S. Luo*, et al.*, "Corporate Bankruptcy Prediction: An Approach Towards Better Corporate World," *The Computer Journal,* 2020.

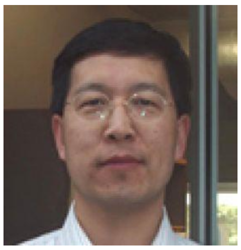
[54] Meng, T.L.; Khushi, M. Reinforcement Learning in Financial Markets. Data 2019, 4, 110.

[55] Barlow, H.; Mao, S.; Khushi, M. Predicting High-Risk Prostate Cancer Using Machine Learning Methods. Data 2019, 4, 129.

TALHA MAHBOOB ALAM was done a bachelor of software engineering from the University of Management and Technology (UMT), Lahore in 2017, and currently completed a master's in computer science from the University of Engineering and Technology (UET), Lahore. He has published more than 10 journal and conference articles in various journals of international repute. Recently, he published papers in IEEE Access and The Computer Journal. His research interests include Data Science, Data Mining, Knowledge Discovery in Databases, Bioinformatics, Big Data, Information Retrieval, and Artificial Intelligence.

KAMRAN SHAUKAT received the M.Sc. degree in computer science from Mohammad Ali Jinnah University, Pakistan. He is currently pursuing a Ph.D. degree with The University of Newcastle, Callaghan, NSW, Australia. He is the author of many articles in the area of machine learning, databases, and cyber security. He has served the University of the Punjab, Pakistan, for seven years, as a Lecturer. He has served as a Reviewer to many journals, including IEEE ACCESS. He has attended several international conferences, including the USA, U.K., Thailand, Turkey, and Pakistan. He received the Gold Medal for his M.Sc. degree.

IBRAHIM A. HAMEED is Professor and Deputy Head of research and innovation, Head of the international master program in simulation & visualization at Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, Norway. Since 2015, he has been an Associate Professor with the Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Trondheim, Norway.



SUHUAI LUO is an associate professor in information technology at the University of Newcastle. He received Bachelor and Master Degrees from Nanjing University of Posts and Telecommunications, and a Ph.D. degree from the University of Sydney, all in Electrical Engineering. His main research interests include image processing, computer vision, machine learning, cyber security, and media data mining. His diverse research focus has led him to conduct studies in areas ranging from medical imaging for computer-aided diagnoses, to computer vision for intelligent driving systems, and machine learning for enhancing cybersecurity.

MUHAMMAD UMER SARWAR is currently pursuing a Ph.D. degree. He is also an Assistant Professor with the Department of Computer Science, Government College University, Faisalabad. His research interests are database systems, text, and data mining.

Shakir Shabbir has done a bachelor's in Computer Science from the University of the Punjab Jhelum Campus. He is doing M.Sc in Computer science from the University of Engineering and Technology (UET), Lahore. His area of interest is big data, data mining, and machine learning. He also published a research paper entitled: Dengue Fever in Perspective of Clustering Algorithms in Journal of Data Mining and Genomics Proteomics. He has also more than 5 years of experience in web application development.

Jiaming LI received the B.E degree in 1986 and M.E. degree in 1989, both from the Electric Engineering Department of Nanjing University of Posts and Telecommunications, China, and a Ph.D. degree from the School of Computer Science and Engineering, the University of New South Wales, Australia in 1998. She is currently a senior research scientist with CSIRO Data61. Her main research interests include system modelling, machine learning, data fusion, image processing, pattern recognition, and multi-agent system design.

MATLOOB KHUSHI is the Director of Master of Data Science and Graduate Certificate in Data Science at the School of Computer Science, University of Sydney, Australia and has been lecturing at the school since 2017.