Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 4, August 2026, pp. 3440~3456
ISSN: 2302-9285, DOI: 10.11591/eei.v15i4.10732

Optimizing bank customer churn prediction with machine learning models: a comparative study

Tran Kim Toai[1], Bui Tien Thinh[2]
[1]Department of E-commerce, Faculty of Economics, Ho Chi Minh City University of Technology and Engineering, Ho Chi Minh City, Vietnam
[2]Department of Logistics, Faculty of Economics, Ho Chi Minh City University of Technology and Engineering, Ho Chi Minh City, Vietnam

Article Info

Article history:

Received May 22, 2025
Revised Apr 23, 2026
Accepted Jul 9, 2026

Keywords:

Boosting algorithms
Customer churn
Data imbalance
Data mining
Machine learning
Model optimization

ABSTRACT

This study utilizes five machine learning algorithms to classify and predict churn behavior based on financial and demographic characteristics from a dataset of 10,000 customers of a confidential multinational bank. This dataset is publicly accessible on Kaggle. The models in this study are random forest (RF), decision tree (DT), light gradient boosting machine (LightGBM), categorical boosting (CatBoost), and multi-layer perceptron (MLP). After initial data preprocessing, the imbalanced data is addressed by the synthetic minority over-sampling technique (SMOTE). Based on the model performance evaluation, LightGBM and CatBoost are the best-performing models in terms of accuracy, F1-score, and average training time. Moreover, the correlation analysis, including chi-square, variance analysis, and pearson correlation, highlights eight critical factors influencing customer churn, including CreditScore, Geography, Gender, Age, Balance, NumOfProducts, IsActiveMember, and Exited. Based on the experimental results, LightGBM and CatBoost are recommended due to superior model performance and processing times, although these models demand significant computational resources based on CPU and RAM usage. The study suggests a new research direction involving the combination of LightGBM and MLP to better capture non-linear relationships and improve prediction performance.

Corresponding Author:

Tran Kim Toai
Department of E-commerce, Faculty of Economics
Ho Chi Minh City University of Technology and Engineering
No 1 Vo Van Ngan Street, Thu Duc Ward, Ho Chi Minh City, Vietnam
Email: toaitk@hcmute.edu.vn

1. INTRODUCTION

Customer churn prediction is pivotal for banks seeking to retain their customer base and increase business performance. Customer churn refers to the phenomenon in which a customer ceases to purchase a company's products or services. These client losses can be attributed to various factors, including intense competition, price increases, or declining service quality. The loss of existing customers presents a financial challenge due to the disproportionately high cost of acquiring new customers compared to retaining existing ones.

Currently, a robust approach to address this customer churn challenge involves developing predictive models to identify customers with high churn risk. This approach has recently gained significant academic attention. Various empirical studies have employed different machine learning algorithms to predict customer

churn in the banking sector, such as random forest (RF), decision tree (DT), artificial neural networks (ANN), k-nearest neighbors (KNN), extreme gradient boosting, light gradient boosting machine (LightGBM), and categorical boosting (CatBoost) [1]-[5]. Additionally, deep learning algorithms have recently been employed in customer churn prediction in other sectors [6]. Several studies suggest that gradient boosting algorithms may offer superior performance in terms of accuracy, speed, and efficiency compared to traditional machine learning algorithms, such as RF, DT, and ANN [4], [7].

Although these studies collectively highlight the potential of machine learning models in forecasting customer churn within the banking sector, several limitations still exist in prior studies. These limitations represent a literature gap in terms of optimizing machine learning models for customer churn prediction. Specifically, many studies have not focused on comprehensively addressing imbalanced data, leading to inaccurate results for high-risk customer segments. Additionally, prior studies using machine learning models have not applied correlation analysis to clearly identify the degree of correlation between variables in the data. This lack of correlation analysis leads to difficulties in identifying important patterns and reduces the accuracy of predicting high-risk churn customers.

This research aims to fill this literature gap and optimize machine learning models for a more accurate classification of bank customer churn by addressing these issues, namely data imbalance and variable correlation analysis. Through the development of predictive models, this study aims to uncover patterns and insights within customer behavior, thereby enabling businesses to reduce customer churn, enhance satisfaction and loyalty. Five machine learning models are employed in this study, namely RF, DT, LightGBM, CatBoost, and multi-layer perceptron (MLP) due to the following reasons. First, RF and DT are popular for their interpretability and robustness in handling large datasets of the banking sector or other industries [8]. Second, gradient boosting models, such as LightGBM and CatBoost, are known for their efficiency and performance, especially with large datasets; additionally, gradient boosting models have consistently shown improved prediction accuracy over traditional models [9]. Third, MLP can offer powerful predictive capabilities in capturing complex and non-linear relationships. In credit card churn prediction, neural networks are among the top-performing models, indicating their potential in banking applications [10].

This research employs the Python programming language to analyze and process all experimental data. The dataset is retrieved from a multinational digital commerce bank and published on Kaggle. This dataset contains information on 10,000 customers and 14 variables with Exited as the key variable. Data preprocessing methods include one-hot encoding for categorical variables, StandardScaler for standardization, and the synthetic minority over-sampling technique (SMOTE) for data imbalance handling. The data is divided into a 70:20:10 ratio for training, testing, and validation, respectively. The training dataset is further processed using SMOTE and then subjected to K-fold cross-validation with k=30 to assess stability. The model evaluation metrics include accuracy, precision, recall, F1-score, processing time, computational resources based on CPU and RAM usage, and execution speed after 30 training iterations, starting with an untrained model and ending with a trained model. The results are based on an average of 30 runs and include a confusion matrix.

The results of our study suggest that there is no optimal machine learning algorithm, as no model can obtain high accuracy in customer churn without demanding high computational resources. Based on the model assessment metrics, LightGBM and CatBoost are the optimal machine learning algorithms, although these models require significant computational resources. However, if the machine learning models are evaluated by the confusion matrix analysis, both MLP and RF demonstrated superior performance; additionally, MLP displayed advantages in capturing non-linear relationships and attained considerable accuracy in identifying the customer churn group. The comparison among these models illustrated the advantages of each method in optimizing classification strategies; additionally, these results also yielded practical applications for banks in terms of choosing the optimal model based on the bank's intended model usage purposes and the information technology infrastructure for model application.

This study contributes to the literature on the application of machine learning models for customer churn prediction in the following ways. First, this paper addresses the common shortcomings of prior research on customer churn in terms of addressing imbalanced data and the lack of a correlation analysis. By addressing these shortcomings, our results are more robust. Second, we acknowledge the limitations of our study and offer future research directions involving the combination of different machine learning algorithms. Based on our findings, no single machine learning model can accurately and efficiently predict customer churn. That is, a trade-off occurs when a machine learning model requires significant computational resources to provide highly accurate customer churn prediction and vice versa. Finally, the results of this study also offer practical implications for banks when it comes to choosing optimal machine learning models. This depends on the initial aim of customer churn prediction and the available information technology infrastructure and computational resources.

The remainder of this research is structured as follows: section 2 illustrates the literature review and the shortcomings of prior studies. In section 3, the data description, the imbalanced data handling, and the application and assessment of the employed machine learning models are provided. In section 4, the results of

the machine learning model application are demonstrated. Confusion matrix analysis and future research direction are provided in section 5. Finally, section 6 presents the conclusion of the study along with practical implications for the banking sector in terms of choosing the optimal machine learning model for customer churn prediction.

2. LITERATURE REVIEW

Given the important role of accurate prediction of customer churn in the banking sector, prediction using machine learning algorithms has received significant academic attention. Various machine learning algorithms have been employed to predict customer churn, with mixed results.

Regarding tree-based algorithms in machine learning, RF, and DT are two of the most commonly applied models in churn prediction. Various empirical studies suggested that RF consistently showed better performance compared to DT. For example, [11], [12] provided evidence of strong RF performance in predicting customer churn. The findings from these studies illustrate that RF achieved an accuracy of about 80%, which closely follows the performance of gradient boosting models and outperforms DT. RF models have also been reported to reach accuracy levels between 89.93% and 95.90% across different sectors [11]. A possible reason is that RF is based on multiple DT, which helps RF avoid the overfitting issue commonly experienced by DT.

In addition to RF and DT, boosting algorithms have been used to predict customer churn. Various studies suggested that LightGBM could outperform traditional machine learning algorithms in terms of accuracy, speed, or efficiency when other determinants of customer churn were considered, such as age and account balance. For example, [13] employed LightGBM, RF, DT, and KNN. The results showed that LightGBM delivered an accuracy of 98%, precision of 97%, recall of 100%, and AUC of 99%. These findings suggested that LightGBM significantly outperformed other machine learning models in the study.

In addition to DT, RF, and gradient boosting models, MLP has also been applied to customer churn prediction. Although the application of MLP in customer churn prediction in the banking sector currently remains limited, related studies using MLP in other sectors suggested that MLP performed well when compared to other models. The most direct empirical evidence came from [14], as the study showed a near-perfect prediction accuracy compared to other models.

Based on previous studies, the most commonly applied machine learning algorithms for customer churn prediction are DT, RF, CatBoost, and LightGBM. Additionally, although MLP has not been extensively employed in the banking sector, MLP still shows potential in customer churn prediction in other sectors. RF and other boosting techniques, like LightGBM and CatBoost, could outperform traditional DT in churn prediction. While RF provides strong predictive accuracy, gradient boosting methods achieve higher efficiency and precision, making these machine learning algorithms well-suited to address complex banking datasets.

Although these studies collectively highlight the potential of machine learning models in forecasting customer churn within the banking sector, several limitations still exist. Specifically, many studies have not focused on comprehensively addressing imbalanced data, leading to inaccurate results for high-risk customer segments. Additionally, previous studies on machine learning models have not applied correlation analysis to identify the degree of correlation between variables in the data. This also makes it difficult to identify important trends and patterns and reduces the accuracy of predicting customer churn among high-risk customers.

3. DATA DESCRIPTION AND METHOD

3.1. Data description

The dataset used in this study was obtained from a confidential multinational bank and is publicly available on the Kaggle platform. This dataset contains information on 10,000 customers and is designed to facilitate customer behavior analysis and churn prediction. The dataset consists of 14 variables, which reflect customers' demographic characteristics, financial status, and their relationship with the bank. The key variable in the dataset is "Exited," which indicates whether a customer has left the bank. Key customer attributes and their respective designations (in parameters) include credit score (CreditScore), age (Age), tenure (Tenure), balance (Balance), number of used products (NumOfProducts), credit card usage (HasCrCard), active members (IsActiveMember), and salary estimation (EstimatedSalary).

3.2. Methodology: machine learning application and data processing

The five machine learning algorithms, including DT, RF, MLP, CatBoost, and LightGBM, are employed to analyze data and predict customer churn. Analyses were conducted in Python using libraries such as scikit-learn, pandas, NumPy, and matplotlib. A 30-fold cross-validation procedure is applied to assess model

reliability, while SMOTE is used to address class imbalance issues. The data processing, data imbalance handling, and the machine learning application to predict customer churn are illustrated as follows.

In the first step, the task of predicting customer churn was clearly defined. This involved understanding the critical factors that affected customer churn and the development of a predictive model based on available data. Next, the initial data containing information about customer characteristics related to churn was collected, saved to a CSV file, and prepared for further processing.

After identifying the problem and collecting the data, data preprocessing techniques were applied. First, the collected data was preprocessed by handling missing values and outliers. Next, the correlation analysis was conducted to identify the variables with an important impact on customer churn. After that, chi-square, analysis of variance (ANOVA), and pearson correlation tests were implemented to test the critical factors that could affect customer churn.

The next step in data processing involved categorical variables. Specifically, because machine learning algorithms required numerical inputs, categorical variables were converted into numerical values. The encoding method chosen in this study was the One-Hot Encoding, which created binary columns corresponding to each value of the categorical variables. This method also reduced multicollinearity and overfitting.

After data encoding, data normalization was the next crucial step in preprocessing, especially when variables had different scales and units. For instance, the CreditScore variable ranges from 300 to 850, while EstimatedSalary can reach hundreds of thousands. This disparity could significantly impact the performance of machine learning algorithms. Normalization adjusts all variables to a common scale, ensuring that algorithms operate accurately and efficiently. The normalization method used in this study was StandardScaler, which standardized variables so that their distribution had a mean of 0 and a standard deviation of 1. The normalization function is expressed as (1):

$$z = \frac{X\text{-}Mean(X)}{Std(X)} \tag{1}$$

where X is the value of the variable to be standardized, mean(X) is the mean of the variable, and Std(X) is the standard deviation of the variable.

When analyzing the target variable distribution, a significant data imbalance issue was identified, which could negatively impact the performance of machine learning models. In imbalanced datasets, models might tend to predict the majority class (Exited=0) more frequently, leading to poor performance for the minority class (Exited=1). To address this issue, the SMOTE was applied. This technique synthetically generates new minority class samples by calculating the distances between data points within the minority class and creating new data points based on these distances. This process helps balance the number of samples between the two classes. As a result, this approach enhances the model's ability to learn from both classes while preserving the original dataset. The techniques for handling imbalance, including SMOTE, were applied only to the training set, enabling the models to learn the characteristics of both classes without being biased by the imbalance. The test and validation sets retained their original distributions to ensure unbiased model evaluation.

To optimize model performance, the dataset was divided into three subsets: a training set (70%), a test set (20%), and a validation set (10%). The number of observations for each set is illustrated in Table 1. The training set served as the foundation for model development, while the test set was used to fine-tune hyperparameters and select the best-performing model. The validation set was used to provide an unbiased evaluation of the final model's performance.

Table 1. The subsets of the original dataset

	Training	Testing	Validation
Number of observations	7000	2000	1000

In this study, an iterative process was required to identify the best-performing model between DT, RF, CatBoost, and LightGBM. After training, the models were evaluated based on accuracy, precision, recall, and F1-score metrics. This dataset was run based on 30 repetitions of K-fold cross-validation values. In the following step, GridSearchCV was utilized to optimize the hyperparameters. This process helped to enhance the accuracy of the models. Finally, a comparison result was provided to find a suitable model to predict customer churn.

By following these steps and conducting a thorough analysis, a model for customer churn prediction can be selected and applied. The accuracy and quality measurements for results are calculated after 30 runs, using accuracy, precision, recall, and F1-score metrics. An overview of data processing and machine learning algorithm application of this paper can be illustrated in Figure 1.

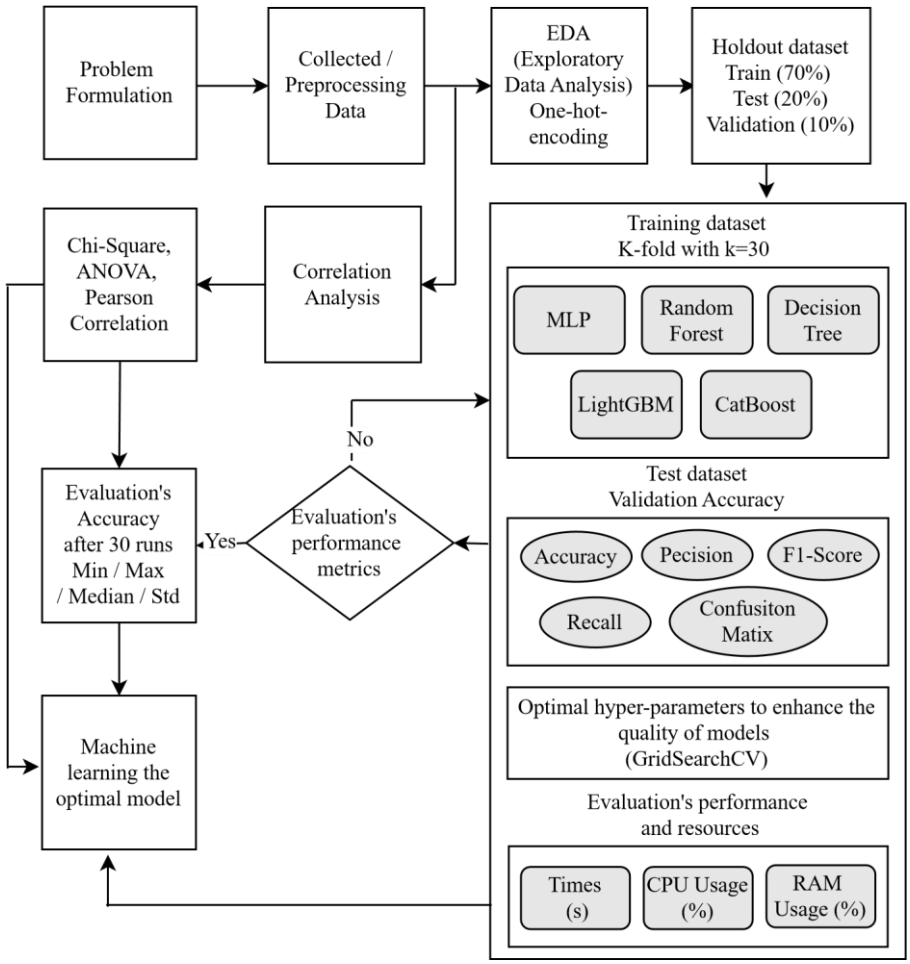

Figure 1. The flow diagram for selecting optimal models

3.3. Machine learning algorithms

This section provides a systematic and transparent account of the machine learning algorithms for predicting client churn to enable reproducibility and independent validation for readers.

3.3.1. Decision tree

DT is employed as a baseline due to its interpretability and ability to capture non-linear feature interactions. DT recursively partitions the dataset based on feature values to predict customer churn outcomes. Hyperparameters such as maximum depth and minimum samples per leaf act as pre-pruning mechanisms to prevent overfitting. By detailing the splitting criterion, stopping rules, and pruning strategy, this work ensures methodological transparency and replicability. The model is highly interpretable, providing insights into decision rules and feature contributions.

DT training procedure, also known as recursive top-down induction, is as follows. Splits are chosen to maximize information gain (entropy) or minimize Gini impurity; recursion terminates under predefined stopping conditions to control model complexity and overfitting. Iterative Dichotomiser 3 (ID3) is a greedy and classic algorithm for building DT [15]. This algorithm ensures that each attribute is used only once in each branch of the tree (greedy strategy). ID3 determines the best attribute for splitting every node in the DT using two key measures: entropy and information gain. Entropy measures the level of uncertainty or randomness in an information system. A high level of entropy means high uncertainty, and vice versa. The mathematical definitions (splitting criteria) are given as the equation below. Specifically, the formula for calculating entropy is as (2):

$$\text{Entropy}(S) = \sum - P(i) \times \log_2 P(i) \quad (2)$$

where P(i) is the probability of a state i.

Information gain is the criterion for evaluating the degree of entropy reduction when splitting an attribute. This value is determined by calculating the difference in entropy before and after the split. The goal of ID3 is to select the split attribute so that the maximum information is gained, i.e., this approach has reduced entropy and improved the accuracy of classifying observations [16]. The function for calculating information gain is described as (3):

$$\text{Information gain (S, A)} = \text{Entropy(S)} - \sum [\text{P(S|A)} \times \text{Entropy(S/A)}] \quad (3)$$

3.3.2. Random forest

RF is a machine learning method that combines many DT to improve prediction accuracy. Instead of relying on a single tree, RF builds multiple trees using different samples of the data and random subsets of features. Each tree makes its own prediction, and the final result comes from averaging the outcomes (for numbers) or taking a majority vote (for categories). This approach reduces errors, handles large and complex datasets well, and provides insights into which features are most important for predicting customer behavior. Furthermore, RF offers interpretable insights into variable importance, enabling both accurate prediction and meaningful understanding of the underlying data structure.

RF is an ensemble learning method and a supervised machine learning algorithm. This method utilizes a set of DTs to overcome the overfitting problem commonly encountered in individual trees [17]. RF is constructed by creating multiple bootstrap samples from the original data and then building DT for each sample. At each node, a random subset of features is selected to find the optimal split point, and the tree grows until a minimum number of observations is reached at the leaf nodes. The prediction results are aggregated from the trees by majority voting for classification problems and by averaging the predicted values for regression problems.

Although RF has good predictive performance in terms of predicting numerical outcomes or categories, RF requires an excessively high number of trees, which can affect computational efficiency [18].

3.3.3. Multi-layer perceptron

MLP is a deep learning architecture employed to model complex, nonlinear relationships in customer churn prediction. The model consists of an input layer, one or more hidden layers, and an output layer, where each layer transforms the input through weighted linear combinations and nonlinear activation functions. The training process begins with random initialization of weights and biases. Predictions are generated using forward propagation, in which each layer computes the linear transformation, followed by the activation function. Model performance is quantified through a loss function, commonly binary cross-entropy. Parameter optimization is conducted using backpropagation, which computes gradients of the loss with respect to weights and biases, followed by updates via gradient descent. This iterative process continues until convergence or until the maximum number of training epochs is reached. The final output is a trained MLP model capable of predicting churn probabilities with improved accuracy.

MLP is an ANN with three types of layers. These layers include an input layer, one or more hidden layers, and an output layer [19]. Neurons in MLP are fully connected, meaning the output of each neuron in one layer becomes the input of neurons in the subsequent layer. The operation of MLP is based on calculating the weighted sum of inputs, followed by applying an activation function to produce an output.

The following equations describe how the neural network processes information from the input layer, through the hidden layers, and to the output layer. First, the weighted sum of inputs for the ordinal number j^{th} hidden node is calculated as (4):

$$s_j = \sum_{i=1}^{n} (w_{ji} \times x_i) - \theta_j, \quad j = 1, 2, \ldots, m \quad (4)$$

where s_j is the sum of the weighted inputs to the ordinal number j^{th} hidden node, w_{ji} is the weight connecting the ordinal number i^{th} input node to the ordinal number j^{th} hidden node, x_i is the input value at the ordinal number i^{th} node, and θ_j is the bias of the ordinal number j^{th} hidden node. After calculating the weighted sum, the output from each hidden node is computed using an activation function. One of the most common activation functions is the sigmoid function, which is defined as (5):

$$f(j) = \frac{1}{(1+\exp(-s_j))}, \quad j = 1, 2, \ldots, m \quad (5)$$

where $f(j)$ is the output value of the ordinal number j^{th} hidden node after the activation function is applied, and s_j is the weighted sum at the ordinal number j^{th} hidden node. Finally, the network output is computed from the output values of the hidden layer through the weights connecting the hidden layer to the output layer. The equation for calculating the output of the ordinal number k^{th} output node is (6):

$$o_k = \sum_{j=1}^{m} (W_{jk} \times f(j)) - \theta_k^{'}, \quad k = 1, 2, \ldots, l \tag{6}$$

where W_{jk} is the weight linking the ordinal number j^{th} hidden node to the ordinal number k^{th} output node and θ_k is the bias of the ordinal number k^{th} output node.

MLP is trained using the backpropagation algorithm, which employs derivatives to adjust the weights between layers to minimize prediction error. This algorithm optimizes the connection weights by calculating and propagating the error backward from the output layer to the input layer. Therefore, this approach adjusts the weights to minimize mistakes during training iterations. MLP is one of the fundamental architectures of deep neural networks and is able to solve nonlinear classification and regression problems.

3.3.4. Categorical boosting

CatBoost is a gradient boosting algorithm designed to handle categorical data efficiently, making it well-suited for churn prediction tasks. It builds an ensemble of oblivious DT through sequential boosting, where each tree corrects the errors of the previous one. By applying ordered boosting and minimizing loss at each split, CatBoost reduces overfitting and improves predictive accuracy. The result is a robust model that enhances churn prediction performance on complex customer datasets.

This algorithm is an open-source gradient boosting library, and it effectively handles categorical features and outperforms other libraries in terms of quality on various datasets. Two key innovations of CatBoost are ordered boosting and a novel algorithm for handling categorical features and addressing the target leakage issue [20]. CatBoost has been successfully applied to problems with large and heterogeneous datasets, and this technique demonstrated superior performance in classification and regression tasks [21]. The system also supports computations on GPUs and CPUs. As a result, the approach has helped to accelerate the training and prediction process.

3.3.5. Light gradient boosting model

LightGBM is a gradient boosting framework designed for efficiency and scalability on large datasets. It predicts churn using histogram-based learning with leaf-wise tree growth, enabling faster training and higher accuracy than traditional boosting methods. By incorporating regularization (L1 and L2 penalties), the model mitigates overfitting while optimizing loss. This makes LightGBM highly effective for large-scale customer churn prediction tasks.

This model is a high-performance, open-source, distributed gradient boosting framework developed by Microsoft. The success of LightGBM comes from its leaf-wise tree growth strategy, combined with innovations such as histogram-based algorithms. These techniques help accelerate training, reduce memory consumption, and maintain high prediction accuracy. LightGBM is capable of handling large datasets and has high parallel efficiency. This technique also employs a depth-wise, leaf-wise growth strategy, which optimizes loss reduction at each leaf, addressing scalability and overfitting issues [22].

3.4. Model performance validation

In order to assess the performance of machine learning models in classification problems, this research utilized a number of machine learning algorithms and several critical evaluation metrics. Each metric provides a different perspective on the model's predictive capabilities. These metrics encompass accuracy, precision, recall, F1-score, and the confusion matrix [23].

Accuracy is the most commonly used metric for evaluating classification models. This metric measures the proportion of correct predictions to the total number of predictions. Nevertheless, in scenarios involving imbalanced datasets, accuracy might not accurately reflect the model's efficacy due to its susceptibility to being dominated by the majority class.

Given the accuracy issue in an imbalanced dataset scenario, the study incorporates additional metrics to provide a more accurate assessment of the model's performance. Precision evaluates a model's capability to accurately classify instances as belonging to the positive class. It is computed as the proportion of true positives to the total number of instances predicted as positive (true positives plus false positives). The mathematical formula for precision is given in (7):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

In this formula, TP refers to the count of true positive cases, while FP denotes the count of false positive cases. Precision assesses a model's capability to reduce the occurrence of false-positive classifications. Recall is a metric that evaluates a model's ability to identify all actual instances belonging to the positive class.

Recall is calculated as the ratio of true positives (TP) to the total number of actual positive instances (TP+FN). The formula for calculating recall is as (8):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

where FN represents the number of false negatives. This metric helps to evaluate the model's ability to accurately identify all positive instances.

F1-score is a metric that combines precision and recall, calculated as the harmonic mean of these two metrics; this metric offers a balance that enables accurate identification of positive instances and avoids false predictions. The formula for calculating the F1-score is (9):

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

4. RESULTS AND DISCUSSION

4.1. Correlation analysis

Correlation analysis is crucial for understanding the key factors influencing customer behavior. This data analysis provides insights into the relationship between variables, significant trends, and patterns. As depicted in Figure 2, a correlation graph between the inputs and output variables helps identify relationships between quantitative variables and the likelihood of customers leaving the bank (Exited). Correlation coefficients range from -1 to 1, with positive values indicating a positive correlation, and vice versa. If the values are close to 0, no strong linear relationship exists.

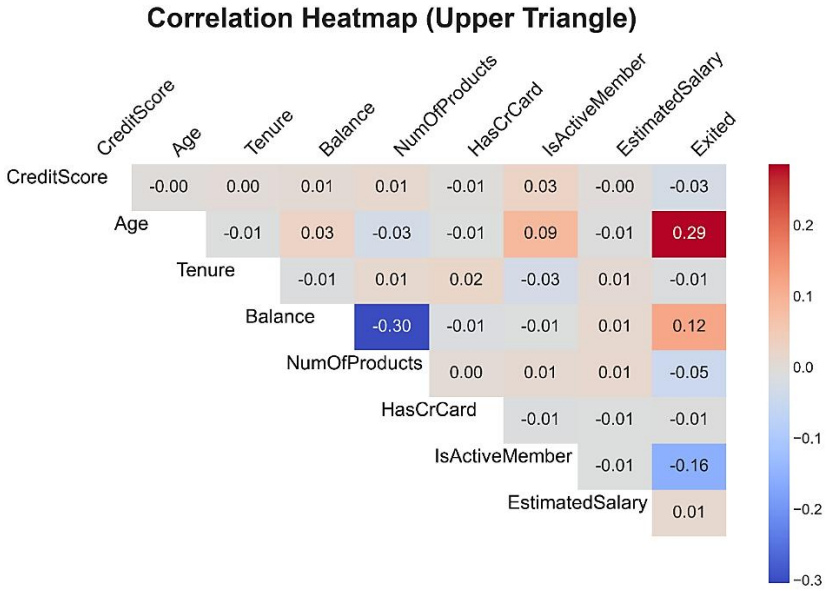

Figure 2. A correlation graph illustrates the relationships between the input and output variables

The analysis reveals that the strongest correlation is between “Age” and “Exited,” with a correlation coefficient of 0.29. This outcome suggests that older customers are more likely to churn. Additionally, “Balance” has a weak positive correlation with “Exited” (0.12), indicating that customers with higher balances may be more prone to churn. On the other hand, “IsActiveMember” has a negative correlation (-0.16) with “Exited”. This result indicates that active customers tend to maintain longer relationships with the bank. Furthermore, the correlation between “Balance” and “NumOfProducts” is negative (-0.30). This outcome suggests that customers who use multiple banking products typically do not maintain high balances in their accounts. Finally, the correlations between “CreditScore” and “EstimatedSalary” and “Exited” are very weak (0.01 and -0.03). Consequently, the findings suggest that these factors have little impact on the likelihood of customer churn.

In addition to pearson correlation analysis, a chi-squared test was also employed to validate assumptions and to ensure that the variables in the correlation analysis were continuous. The test was used to detect potential nonlinear relationships or interactions that correlation analysis might overlook. ANOVA was employed to analyze any identified relationships between the variables. ANOVA helps to determine which

groups have significantly different mean values and simultaneously examines the impact of categorical variables on continuous ones. Pearson correlation was used to confirm the correlation results and to compare with other methods. This approach contributed to identifying more complex interactions between variables and facilitated a deeper understanding of the nature of their relationships.

Table 2 illustrates the results of the chi-square test. The chi-square test (Table 2) indicates a strong association between "Geography," "Gender," and "IsActiveMember" and the likelihood of customers being "Exited" ($p<0.05$). Conversely, the variable "HasCrCard" does not exhibit a statistically significant relationship. These results further verify the correlation matrix shown in Figure 2.

Table 2. Chi-square test results between categorical variables and target variables

Feature	Chi-square (X2) statistic	P-value	Significance
Geography	301.2553368	3.83E-66	Yes
Gender	112.9185706	2.25E-26	Yes
HasCrCard	0.471337799	0.492372361	No
IsActiveMember	242.9853416	8.79E-55	Yes

Tables 3 and 4 demonstrate the results of the ANOVA test and pearson correlation, respectively. Both ANOVA (Table 3) and pearson correlation (Table 4) indicate that "Age," "Balance," "CreditScore," and "NumOfProducts" are strong predictors of customer churn, as indicated by the p-value being lower than 0.05. On the other hand, "Tenure" and "EstimatedSalary" do not significantly contribute to customer churn.

To conclude, through the correlation analysis, the results of the chi-square test and pearson correlation analysis, several variables are shown to be significantly correlated with the key variable, "Exited," namely "Geography," "Gender," and "IsActiveMember," "Age," "Balance," "CreditScore," and "NumOfProducts."

Table 3. Results of the ANOVA test comparing quantitative variables to the target variable

Feature	Chi-square (X2) statistic	P-value	Significance
CreditScore	7.344522164	0.006738214	Yes
Age	886.0632749	1.24E-186	Yes
Tenure	1.960163626	0.161526849	No
Balance	142.4738325	1.28E-32	Yes
NumOfProducts	22.9152225	1.72E-06	Yes
EstimatedSalary	1.463261924	0.226440428	No

Table 4. The results of the pearson correlation analysis were conducted to examine the linear relationship between quantitative variables and the target variable

Feature	Correlation coefficient	P-value	Significance
CreditScore	-0.02709	0.006738214	Yes
Age	0.285323	1.24E-186	Yes
Tenure	-0.014	0.161526849	No
Balance	0.118533	1.28E-32	Yes
NumOfProducts	-0.04782	1.72E-06	Yes
EstimatedSalary	0.012097	0.226440428	No

4.2. Comparison and evaluation of model performance

Five different models were applied and compared in this classification problem to optimize the models' effectiveness. These models include DT, RF, LightGBM, MLP, and CatBoost. Each model has distinct characteristics and different operating mechanisms, providing a comprehensive view of the classification capabilities in this specific problem.

Selecting and comparing multiple models could determine the best model based on performance evaluation criteria. Hyperparameter optimization by GridSearchCV is a crucial step in the model training process. The optimal parameter sets for each model are detailed in Tables 5 to 9.

Table 5. Key hyperparameters by GridSearchCV for DT

Parameters	Description	Best value
Criterion ['gini'; 'entropy']	Evaluating the quality of a split	'entropy'
max_depth [3; 5; 7; 9; none]	Controlling the maximum depth of the tree	9
min_samples_split [2; 9; 11; 13]	Minimum number of samples required to split a node	2
min_samples_leaf [1; 3; 5]	Minimum number of samples allowed in a leaf node	1
random_state [42]	Controlling the randomness of the estimator	42

Table 6. Key hyperparameters by GridSearchCV for RF

Parameters	Description	Best value
Criterion [‘Gini impurity’, ‘entropy’]	Evaluating the quality of splits	‘entropy’
n_estimators [100]	The number of trees to be built before taking the maximum voting or the average of predictions	100
max_features [‘sqrt’; ‘log2’]	The maximum number of features RF is allowed to try in an individual tree	‘sqrt’
max_depth [5; 6; 8; 10; 12; 15; none]	The maximum depth of each tree	10
random_state [1]	Random seed number	1

Table 7. Key hyperparameters by GridSearchCV for MLP

Parameters	Description	Best value
hidden_layer_sizes [(50,); (100,)]	The number of hidden layers and the number of neurons in each layer	(100)
Alpha (regularization parameter) [0.001; 0.005]	Controlling the amount of L2 regularization applied to the weights to prevent overfitting	0.001
max_iter [150]	A crucial parameter to tune for achieving convergence without overfitting. This parameter controls how many times the model updates its weights during training	150
random_state [42]	Determining random number generation for weights and bias initialization, this parameter is for the train-test split	42

Table 8. Key hyperparameters by GridSearchCV for CatBoost

Parameters	Description	Best value
Iterations [200]	This parameter specifies the number of boosting iterations (trees) to be used during training	200
Depth [3; 4; 5]	Specifying the maximum depth of each DT in the ensemble	5
Learning_rate [0.01; 0.1; 0.2]	Controlling the step size at each iteration while moving towards the minimum of the loss function.	0.2
Random_state [seed=42]	Fixing the random seed in random state ensures reproducibility	42
Loss_function [Logloss]	Specifying the loss function to be optimized during training	Logloss

Table 9. Key hyperparameters by GridSearchCV for LightGBM

Parameters	Description	Best value
n_estimators [100, 200]	Number of boosted trees to fit	200
Learning_rate [0.01; 0.1; 0.2]	This parameter controls how much to change the model in response to the estimated error each time the model weights are updated	0.2
Max_depth [3; 5; 7]	Maximum limitation of tree depth	7
Objective ["regression", "binary", "multiclass"]	Specifying the loss function to optimize based on the task	Binary
random_state [42]	Random seed number	42

The average values of model performance and computational resource requirements of various machine learning algorithms after 30 runs are illustrated in Tables 10 and 11, respectively. LightGBM emerges as the top-performing model, evidenced by its optimal metrics across all key indicators: accuracy (0.905802992), precision (0.907213318), recall (0.905802992), and F1-score (0.9057573). Additionally, LightGBM's average training time is 0.221188267 seconds, which is the second-fastest training time in the study. These results reflect the superior prediction accuracy and computational speed of LightGBM compared to CatBoost. A disadvantage of LightGBM is its computational resource consumption, as indicated by its second-highest CPU usage. The efficiency in data processing time of LightGBM can be attributed to the histogram-based algorithm that groups data into a histogram, from which a splitting point can be determined. This approach contributes to the efficiency of the split operation of LightGBM [24].

CatBoost is the second-best model in terms of the four key indicators: accuracy (0.904727723), precision (0.906749644), recall (0.904727723), and F1-score (0.904645032). These figures are marginally lower than those of LightGBM. This demonstrates CatBoost's ability not only to classify with high accuracy but also to maintain a balance between precision and recall. However, the main disadvantage of CatBoost is its average training time, at 1.343732556 seconds, which is the second highest out of the five models. Additionally, both CatBoost and LightGBM require significantly more CPU usage compared to RF, DT, and MLP. A fundamental reason for the excellent performance of CatBoost lies in its ability to handle categorical variables with its target statistics encoding technique and ordered boosting technique that prevents information

leakage from the training set [20]. As a result, CatBoost is able to prevent overfitting learning, which otherwise inflates the training performance.

The RF model ranks third with the following metrics: accuracy (0.883948082), precision (0.885347162), recall (0.883948082), and F1-score (0.883898536). Although the RF model does not reach the same performance levels as LightGBM and CatBoost do, RF still demonstrates robust and reliable performance. The average training time of this model, at 1.445653335 seconds, is significantly longer than both LightGBM and CatBoost. An advantage of RF over LightGBM and CatBoost lies in the CPU usage, as the CPU usage of RF is 18.78%, which is much lower compared to CatBoost and LightGBM.

After RF, DT is the next model in terms of performance, with the metric values as follows: accuracy (0.855924068), precision (0.858190065), recall (0.855924068), and F1-score (0.85576805). Although DT has the fastest average training time (0.067120592 seconds) and requires the least amount of CPU usage (16.75%), its performance falls considerably behind compared to LightGBM, CatBoost, and RF. As a result, these results highlight the limitations of the learning capacity of a single DT model in complex classification problems.

Finally, MLP is the lowest-performing model, with the following metrics: accuracy (0.826375076), precision (0.827540602), recall (0.826375076), and F1-score (0.826342522). Although it is a neural network model, MLP is not well-suited for this dataset and the classification tasks, and its most significant drawback is the longest average training time (5.098106201 seconds).

Table 10. Results of average accuracy performance metrics after repeating 30 runs of five models

Model	Accuracy	Precision	Recall	F1-score
DT	0.855924068	0.858190065	0.855924068	0.85576805
RF	0.883948082	0.885347162	0.883948082	0.883898536
MLP	0.826375076	0.827540602	0.826375076	0.826342522
CatBoost	0.904727723	0.906749644	0.904727723	0.904645032
LightGBM	0.905802992	0.907213318	0.905802992	0.9057573

Table 11. Analyzing the average performance metrics, resource consumption characteristics, and the speed of algorithms after repeating 30 runs of five models

Model	Time (s)	Latency (s)	CPU usage (%)	RAM usage (%)
DT	0.067120592	0.000409214	16.75	84.455
RF	1.445653335	0.021369163	18.78833333	84.67833333
MLP	5.098106201	0.001204809	33.41833333	84.17166667
CatBoost	1.343732556	0.010844453	64.99833333	84.12666667
LightGBM	0.221188267	0.009703167	43.46666667	84.18666667

The application of the SMOTE was crucial in this study. SMOTE was applied only to the training set after the initial train-test split. Specifically, oversampling was performed using SMOTE, with a random_state of 42, to generate synthetic minority class samples until the dataset was balanced. The resulting resampled training set was then used for 30-fold cross-validation to train and evaluate the model. This approach avoids data leakage, as no synthetic samples were generated from the test set. By restricting oversampling to the training portion, the evaluation remains representative of real-world class distributions. To highlight the effectiveness of SMOTE in handling class imbalance, the model performance metrics across 30 folds, with 95% confidence intervals before and after SMOTE are illustrated in Figure 3.

We illustrate model performance without SMOTE in Figure 3(a). Across accuracy, recall, precision, and F1-score, all models already had high values of these metrics. However, regarding metrics that are subject to class imbalance, such as MCC and Cohen's Kappa, the values are notably lower. These findings necessitate the inclusion of SMOTE in subsequent model performance.

As shown in Figure 3(b), with SMOTE, models consistently achieved higher median values in accuracy, recall, precision, and F1-score. The improvement was particularly strong in recall and F1-score, demonstrating SMOTE's ability to reduce false negatives and improve detection of the minority class. Metrics that are sensitive to class imbalance, such as MCC and Cohen's Kappa, also improved, confirming that SMOTE enhances the reliability of predictions across both classes.

At the same time, a slight trade-off was observed in Brier Scores, suggesting that the introduction of synthetic samples may affect probability calibration. This indicates that while SMOTE enhances discrimination performance, it may slightly distort well-calibrated probability estimates. Nonetheless, the overall gains across most performance measures—particularly those directly linked to imbalanced data handling—support the use of SMOTE as a beneficial preprocessing technique.

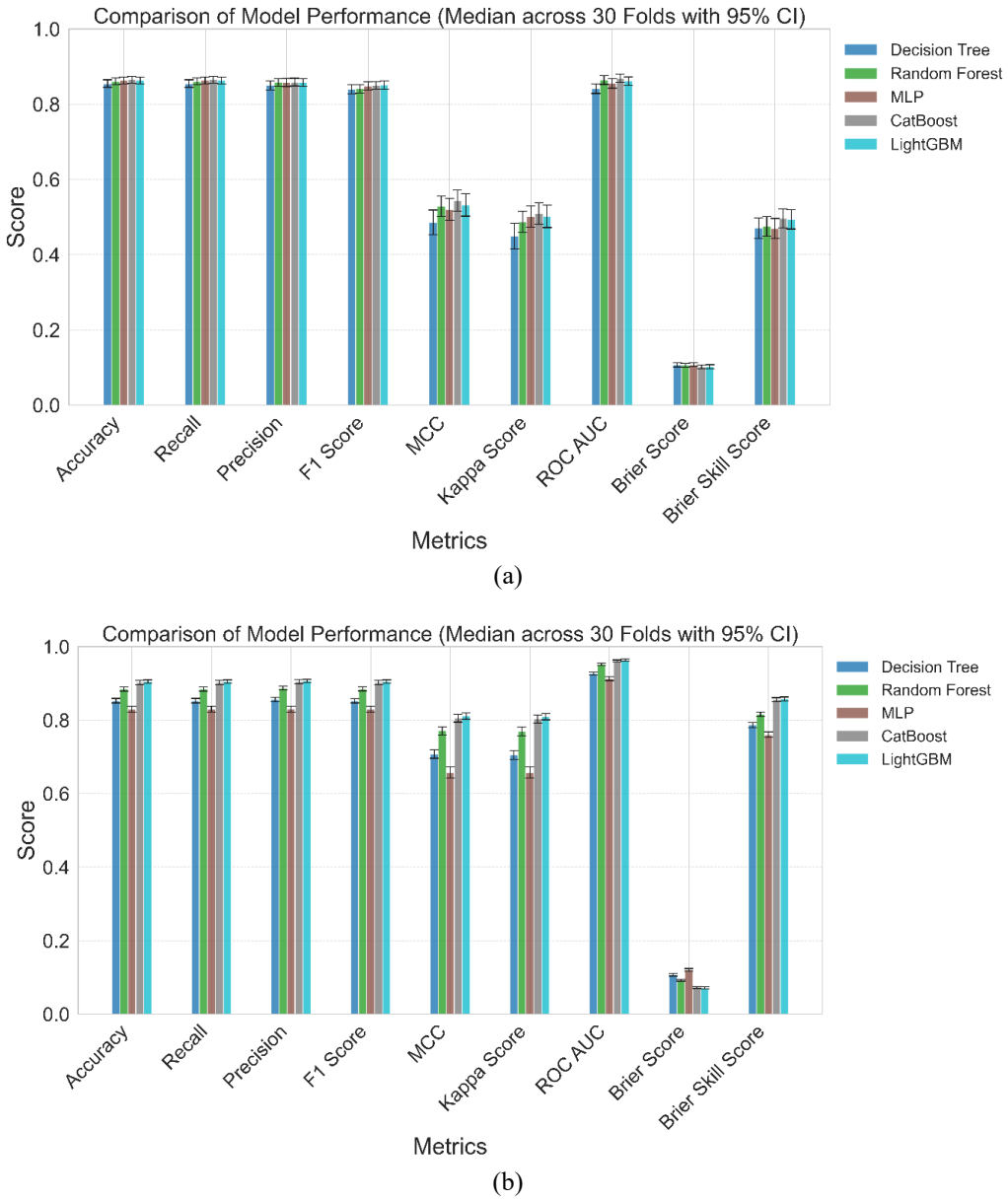

Figure 3. Comparison of model performance without SMOTE and with SMOTE; (a) model performance comparison without SMOTE (before SMOTE) and (b) model performance comparison with SMOTE (after SMOTE)

The ROC curves with 95% confidence intervals further confirm the advantages of SMOTE. As shown in Figure 4, models trained with SMOTE achieved higher AUC values and narrower confidence intervals, reflecting both stronger classification ability and more stable performance across folds. In contrast, without SMOTE, the ROC curves displayed broader confidence intervals and reduced AUCs, indicating less reliable separation between classes. This reinforces the observation that SMOTE not only improves minority class detection but also stabilizes model performance under cross-validation.

4.3. Cross-validation strategy

Cross-validation plays a critical role in evaluating machine learning models by balancing bias and variance trade-offs. Selecting an appropriate number of folds (k) directly influences the robustness and generalizability of the results. According to Prokhorenkova *et al.* [20], larger values of k typically reduce bias but may increase variance, and vice versa; therefore, the k value should be selected based on the dataset size, computational limitations, and the desired trade-off between accuracy and efficiency.

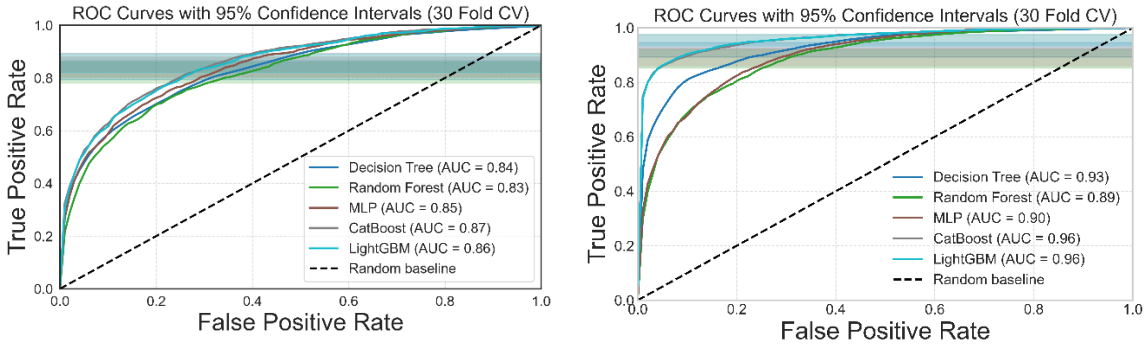

Figure 4. ROC curves without and with SMOTE

The performance of the five machine learning algorithms under different K-fold cross-validation settings, ranging from 5 to 30, is compared and illustrated in Tables 12 and 13. It is evident that K-fold CV provides superior metric values and ROC-AUC across all models compared to baseline single-split evaluation, with an average 4% increase in metric values. Additionally, while DT, RF, and MLP share the same optimal k value of 30, the optimal k values of CatBoost and LightGBM are 15 and 20, respectively.

In terms of algorithm performance at their respective optimal K-fold values, CatBoost and LightGBM are the best models, with all metric values being superior to the remaining algorithms. CatBoost obtains the best values at the optimal K-fold value of 15, with accuracy (0.905802), precision (0.907471), recall (0.905802), and F1-score (0.905733). LightGBM's metric values are very close, with the optimal K-fold value being 20. Most notably, while the values of all metrics of LightGBM at its optimal K-fold value are only about 0.2% less than the values of CatBoost, both models share almost identical ROC-AUC values at 0.9625.

Regarding DT, RF, and MLP, although all models share the same K-fold value, RF has the highest metric values out of the three models. The metric values of RF are as follows: accuracy (0.883948), precision (0.885347), recall (0.883948), and F1-score (0.883899). These figures are, on average, 3% and 6.5% higher than those of DT and MLP, respectively. These results suggest that CatBoost and LightGBM demonstrate superior performance at the optimal K-fold value compared to DT, RF, and MLP. These results support the findings as shown in Table 10.

Table 14 further supports the findings in Tables 12 and 13. CatBoost and LightGBM consistently deliver the highest accuracy across all statistics, with mean values of 0.9175 and 0.9161, respectively, and maximum accuracies exceeding 0.962. Their respective standard deviation values are also very low, indicating little fluctuation of the results. RF follows closely behind with a mean accuracy of 0.8979 and a maximum of 0.9522. DT and MLP show moderate performance. These values indicate that CatBoost and LightGBM can achieve the highest accuracies with greater consistency.

Table 12. Summary of model performance between the base split and the best k

Metric	DT		RF		MLP		CatBoost		LightGBM	
	Base split	(k=30)	Base split	(k=30)	Base split	(k=30)	Base split	(k=15)	Base split	(k=20)
Accuracy	0.829500	0.857625	0.845500	0.883948	0.788000	0.826375	0.852000	0.905802	0.858000	0.904190
Precision	0.839965	0.860086	0.851288	0.885347	0.836655	0.827541	0.845999	0.907471	0.852516	0.905513
Recall	0.829500	0.857625	0.845500	0.883948	0.788000	0.826375	0.852000	0.905802	0.858000	0.904190
F1-score	0.833800	0.857453	0.848013	0.883899	0.802516	0.826343	0.848235	0.905733	0.854539	0.904145
ROC-AUC	0.843459	0.924883	0.876756	0.952220	0.858674	0.911841	0.875563	0.962625	0.865695	0.962543

Table 13. Summary comparison

Metric	DT	RF	MLP	CatBoost	LightGBM
Accuracy	0.857625	0.883948	0.826375	0.905802	0.90419
Precision	0.860086	0.885347	0.827541	0.907471	0.905513
Recall	0.857625	0.883948	0.826375	0.905802	0.90419
F1-score	0.857453	0.883899	0.826343	0.905733	0.904145
ROC-AUC	0.924883	0.95222	0.911841	0.962625	0.962543

Table 14. Cross-validation summary

Statistical summary	DT	RF	MLP	CatBoost	LightGBM
min	0.857453288	0.883898536	0.826342522	0.905733	0.904145
max	0.924882706	0.952219879	0.911841276	0.962625	0.962543
median	0.857625136	0.883948082	0.826375076	0.905802	0.90419
mean	0.87153451	0.897872348	0.84369491	0.917487	0.916116
std	0.029842562	0.030387383	0.038098384	0.025244	0.02596

5. CONFUSION MATRIX ANALYSIS AND FURTHER DISCUSSION

To gain a deeper understanding of the strengths and weaknesses of each model, the authors further analyzed the confusion matrix. This technique facilitates a detailed analysis of the types of errors encountered by the models, which allows for the identification of essential improvements. Since DT, RF, and MLP achieved their highest stability and discriminative power at the K-fold value of 30, this value was selected as the optimal cross-validation setting for consistency and comparability across models.

The confusion matrix results following the validation process are presented in Figures 5(a)-(e). On the validation set, both LightGBM and CatBoost models maintain stable performance for non-churn customer prediction. The accuracy for the non-churn customer group reaches 92% for both models, while the accuracy for the churn customer group is 56% and 55%, respectively. However, the ability to accurately detect churn customers remains limited, with false negative rates of 44% and 45%, respectively.

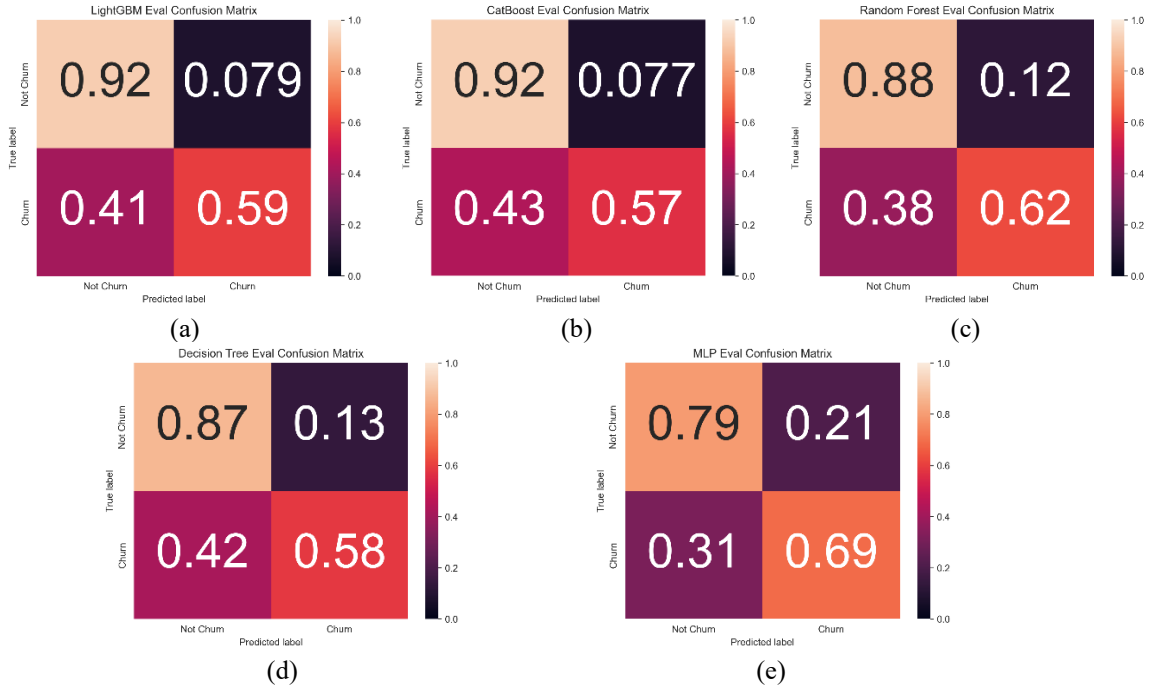

Figure 5. Post-validation confusion matrices for the; (a) LightGBM, (b) CatBoost, (c) RF, (d) DT, and (e) MLP models

Regarding the remaining models, RF maintains relatively good performance with an accuracy rate of 88% for non-churn customers and 62% for churn customers. This figure group is higher than that of LightGBM and CatBoost. The false negative rate of the RF model is 38%, significantly lower than the boosting models LightGBM and CatBoost, and this result indicates that this model is more balanced in classifying between the two groups. DT illustrates a lower performance compared to CatBoost and RF, with accuracy rates of 86% for the non-churn customer group and 59% for the churn customer group. This model continues to reveal limitations in identifying churn customers, with a false negative rate of 41%. This reflects the weakness of DT in capturing complex features, especially when working with small datasets such as the validation set.

Finally, the MLP model demonstrates the highest proficiency in accurately classifying the churn customer group with an accuracy rate of 69%, which is the highest among all models. However, MLP's performance for the non-churn customer group is the lowest, reaching only 79%. MLP also exhibits the highest false positive rate of 21%, which leads to the risk of unnecessary focus on customers who are not actually at

risk. Although MLP has an advantage in identifying the churn customer group, this model still requires further optimization to improve overall performance.

In conclusion, based on the findings, there is no optimal machine learning algorithm that could accurately and efficiently predict both churn and non-churn customers. Therefore, the optimal machine learning model depends on the initial aims of customer churn prediction and the current information technology infrastructure that supports the application of such model. If the bank prioritizes minimizing errors in the non-churn customer group and the information technology infrastructure is sufficient, the boosting models (LightGBM and CatBoost) remain the top choices. However, if the goal is to balance between the two groups or increase accuracy in the churn customer group, and if the infrastructure is not sufficient for the application of LightGBM or CatBoost, both RF and MLP could be considered. DT is easy to implement, but it is not the optimal choice for this classification problem in this research.

Given the limitations of individually employing machine learning models for customer churn prediction, a new approach may be tested to find the optimal model for such a task. By combining multiple models together, the prediction accuracy can be further enhanced. While various combined models have been proposed, the authors of this study propose a novel approach combining a Gaussian mixture model (GMM) clustering with an adaptive support vector machine (ASVM) to analyze customer behavior and predict churn in the banking industry through a clustering approach. The ASVM classifier achieved an accuracy of 98% in identifying at-risk customers, demonstrating superior performance compared to existing methods such as convolutional neural networks (CNN), decision tree-multinomial regression (DT-MR), and ANN [25]. This methodology provides bank administrators with insights into customer behavior, enabling the implementation of targeted strategies to enhance customer engagement and retention [26].

6. CONCLUSION

This study employed machine learning models to address the problem of classifying customer churn propensity for a bank. By leveraging personal characteristics, financial status, and interaction levels with the bank, the research provides valuable insights into customer behavior. This approach can help banks identify customer segments that require prioritized retention efforts effectively.

In this study, several machine learning algorithms are proposed for accurate customer churn prediction, namely RF, DT, LightGBM, CatBoost, and MLP. While LightGBM demonstrated superior performance and efficiency in handling large datasets to achieve 90.580% accuracy and 90.57% F1-score, with an average training time of only 0.271 seconds, MLP displayed strengths in capturing non-linear relationships and attained 69% accuracy in identifying the churn group. The comparison among these models highlighted the capabilities of each method in optimizing classification strategies and practical applications.

In addition to the performance evaluation of the five machine learning algorithms, the correlation analysis and chi-square, ANOVA, and pearson correlation tests also revealed crucial factors influencing customer churn propensity, including CreditScore, Geography, Gender, Age, Balance, NumOfProducts, IsActiveMember, and Exited. This highlights the necessity of accurate feature selection to enhance the effectiveness of classification models.

Based on both the comparison of performance metrics and the ROC curve analysis, the application of SMOTE demonstrates a clear positive impact on model performance in imbalanced classification. The improvements are most notable in recall, F1-score, MCC, and AUC, which are critical for detecting the minority class. While some trade-offs exist in probability calibration (Brier Score), the overall benefits outweigh the drawbacks. This confirms that applying SMOTE during training is an effective strategy to mitigate imbalance and enhance robustness in classification tasks.

In terms of practical application derived from the results of this study, several machine learning models can be recommended for bank applications. As there is no optimal machine learning algorithm that could perform churn and non-churn customer prediction accurately and efficiently, the choice of machine learning algorithm to be employed depends on the aims of customer churn prediction and the computational resources available to the banks. If the bank focuses on reducing errors in the non-churn customer group and has sufficient computational resources, LightGBM and CatBoost are excellent candidates. However, if balancing between the two groups or if accuracy in the churn customer group is the main focus, both RF and MLP could be considered. DT is poorly suited for customer churn prediction.

This study also acknowledges the significant challenge posed by data imbalance, where the proportion of churn customers was substantially lower than that of the non-churn group. Despite employing the SMOTE technique for data balancing, a gap remains in minimizing false negative errors, particularly for high-risk churn customer segments. Therefore, this study proposes a practical solution to enhance accuracy by combining LightGBM with an MLP. This approach presents a promising direction, where LightGBM can assume the role of data preprocessing, feature selection, and dimensionality reduction tasks. This result will help reduce complexity

for subsequent processing steps. MLP can capture non-linear relationships and use LightGBM's output to enhance classification accuracy. Integrating these two models through ensemble techniques, such as weighted voting or averaging, can create a robust solution that optimizes accuracy and stability in classification.

Overall, this study illustrates the potential of machine learning to support data-driven customer churn prediction in banking. The findings provide practical guidance for model selection and offer a foundation for developing more robust churn prediction systems in future research.

FUNDING INFORMATION

This research is partly funded by Ho Chi Minh City University of Technology and Engineering, Vietnam.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Tran Kim Toai	✓	✓	✓	✓	✓			✓	✓	✓				
Bui Tien Thinh		✓				✓			✓	✓	✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that there are no conflicts of interest regarding the publication of this paper.

DATA AVAILABILITY

The dataset used in this study is publicly available on Kaggle and can be accessed by interested researchers for replication and further analysis. The authors of this study are willing to share the data upon request.

REFERENCES

[1] Z. H. Kilimci, "The Effectiveness of Homogeneous Classifier Ensembles on Customer Churn Prediction in Banking, Insurance, and Telecommunication Sectors," *International Journal of Computational and Experimental Science and Engineering*, vol. 8, no. 3, pp. 77–84, Nov. 2022, doi: 10.22399/ijcesen.1163929.

[2] H. D. Tran, N. Le, and V.-H. Nguyen, "Customer Churn Prediction in the Banking Sector Using Machine Learning-Based Classification Models," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 18, pp. 087–105, 2023, doi: 10.28945/5086.

[3] H. Dalmia, C. V. S. S. Nikil, and S. Kumar, "Churning of Bank Customers Using Supervised Learning," in *Innovations in Electronics and Communication Engineering: Proceedings of the 8th ICIECE 2019,* 2020, pp. 681–691, doi: 10.1007/978-981-15-3172-9_64.

[4] N. T. M. Sagala and S. D. Permai, "Enhanced Churn Prediction Model with Boosted Trees Algorithms in The Banking Sector," in *2021 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Oct. 2021, pp. 240–245, doi: 10.1109/ICoDSA53588.2021.9617503.

[5] I. Kaur and J. Kaur, "Customer Churn Analysis and Prediction in Banking Industry using Machine Learning," in *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, IEEE, Nov. 2020, pp. 434–437, doi: 10.1109/PDGC50313.2020.9315761.

[6] M. A. Al Rahib, N. Saha, R. Mia, and A. Sattar, "Customer data prediction and analysis in e-commerce using machine learning," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 4, pp. 2624–2633, Aug. 2024, doi: 10.11591/eei.v13i4.6420.

[7] M. Imani and H. R. Arabnia, "Hyperparameter Optimization and Combined Data Sampling Techniques in Machine Learning for Customer Churn Prediction: A Comparative Analysis," *Technologies (Basel)*, vol. 11, no. 6, p. 167, Nov. 2023, doi: 10.3390/technologies11060167.

[8] D. Kotios, G. Makridis, G. Fatouros, and D. Kyriazis, "Deep learning enhancing banking services: a hybrid transaction classification and cash flow prediction approach," *Journal of Big Data*, vol. 9, no. 1, p. 100, Oct. 2022, doi: 10.1186/s40537-022-00651-x.

[9] J. Cai, "The causes of bank customer churn based on XGBoost and LightGBMmodels: the evidence from the Kaggle dataset," *Finance & Economics*, vol. 1, no. 4, Jan. 2024, doi: 10.61173/ggw4ga77.

[10] A. B. Zoric, "Predicting Customer Churn in Banking Industry using Neural Networks," *Interdisciplinary Description of Complex Systems*, vol. 14, no. 2, pp. 116–124, 2016, doi: 10.7906/indecs.14.2.1.

[11] Md P. Ahmed *et al.*, "A Comparative Study of Machine Learning Models for Predicting Customer Churn in Retail Banking: Insights from Logistic Regression, Random Forest, GBM, and SVM," *Journal of Computer Science and Technology Studies*, vol. 6, no. 4, pp. 92–101, Oct. 2024, doi: 10.32996/jbms.2024.6.4.12.
[12] F. Ehsani and M. Hosseini, "Customer churn analysis using feature optimization methods and tree-based classifiers," *Journal of Services Marketing*, vol. 39, no. 1, pp. 20–35, 2025, doi: 10.1108/JSM-04-2024-0156.
[13] X. Miao and H. Wang, "Customer Churn Prediction on Credit Card Services using Random Forest Method," in *Proceedings of the 2022 7th International Conference on Financial Innovation and Economic Development*, 2022, doi: 10.2991/aebmr.k.220307.104.
[14] M. Przybyła-Kasperek, K. F. Marfo, and P. Sulikowski, "Multi-Layer Perceptron and Radial Basis Function Networks in Predictive Modeling of Churn for Mobile Telecommunications Based on Usage Patterns," *Applied Sciences*, vol. 14, no. 20, p. 9226, Oct. 2024, doi: 10.3390/app14209226.
[15] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, Mar. 1986, doi: 10.1007/BF00116251.
[16] E. E. Ogheneovo and P. A. Nlerum, "Iterative dichotomizer 3 (ID3) decision tree: A machine learning algorithm for data classification and predictive analysis," *International Journal of Advanced Engineering Research and Science (IJAERS)*, vol. 7, no. 4, pp. 514–521, 2020.
[17] J. Zhang, "Impact of an improved random forest-based financial management model on the effectiveness of corporate sustainability decisions," *Systems and Soft Computing*, vol. 6, p. 200102, Dec. 2024, doi: 10.1016/j.sasc.2024.200102.
[18] M. Rose and H. R. Hassen, "A survey of random forest pruning techniques," in *CS & IT Conference Proceedings*, 2019, doi: 10.5121/csit.2019.91808.
[19] X. Pu, S. Chen, X. Yu, and L. Zhang, "Developing a Novel Hybrid Biogeography-Based Optimization Algorithm for Multilayer Perceptron Training under Big Data Challenge," *Scientific Programming*, vol. 2018, no. 1, p. 2943290, 2018, doi: 10.1155/2018/2943290.
[20] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
[21] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *Journal of Big Data*, vol. 7, no. 1, p. 94, 2020.
[22] D. M. Gala, B. Pawar, G. Band, P. Dua, B. R. Mohanty, and B. V. Patil, "Enhancing Predictive Accuracy for Customer Churn in Digital Banking: A Multi-Model Analysis," in *2024 3rd Edition of IEEE Delhi Section Flagship Conference (DELCON)*, 2024, pp. 1–5.
[23] J. C. Obi, "A comparative study of several classification metrics and their performances on data," *World Journal of Advanced Engineering Technology and Sciences*, vol. 8, no. 1, pp. 308–314, Feb. 2023, doi: 10.30574/wjaets.2023.8.1.0054.
[24] M. Saber *et al.*, "Examining LightGBM and CatBoost models for wadi flash flood susceptibility prediction," *Geocarto International*, vol. 37, no. 25, pp. 7462–7487, Dec. 2022, doi: 10.1080/10106049.2021.1974959.
[25] S. Kumar and H. Sharma, "A Survey on Decision Tree Algorithms of Classification in Data Mining," *International Journal of Science and Research (IJSR)*, vol. 5, no. 4, pp. 2094–2097, Apr. 2016, doi: 10.21275/v5i4.NOV162954.
[26] A. Saxena, A. Singh, and G. M., "Analyzing customer churn in banking: A data mining framework," *Multidisciplinary Science Journal*, vol. 5, 2023, doi: 10.31893/multiscience.2023ss0310.

BIOGRAPHIES OF AUTHORS

Tran Kim Toai received a Bachelor's degree in Information Technology Management and a Master's degree in Business Administration. He received a Ph.D. degree from the Department of Computer Science, VSB–Technical University of Ostrava, Czech Republic. He works as a lecturer at the Ho Chi Minh University of Technology and Education. His current academic research interests are machine learning, artificial intelligence, data mining, neural networks, data analysis, and finance. He can be contacted at email: trankimtoai@gmail.com and toaitk@hcmute.edu.vn.

Bui Tien Thinh holds a Bachelor's degree in Economics and a Master's degree in Applied Economics. He received his Ph.D. in Finance from the Department of Finance, National Central University, Taoyuan City, Taiwan. He is currently a lecturer at the Ho Chi Minh City University of Technology and Education. His research interests include corporate finance, corporate decision-making, capital structure, investment efficiency, as well as logistics, and supply chain management. He can be contacted at email: thinhbt@hcmute.edu.vn and buitienthinh.k11401@gmail.com.