Hindawi
Discrete Dynamics in Nature and Society
Volume 2022, Article ID 5134356, 20 pages
https://doi.org/10.1155/2022/5134356

Research Article

Model Optimization Analysis of Customer Churn Prediction Using Machine Learning Algorithms with Focus on Feature Reductions

Seyed Mohammad Sina Mirabdolbaghi[1] **and Babak Amiri**[2]

[1]*Department of Industrial Engineering, Kharazmi University, Tehran, Iran*
[2]*Industrial Engineering School, Iran University of Science and Technology, Tehran, Iran*

Correspondence should be addressed to Babak Amiri; babakamiri@iust.ac.ir

Received 6 April 2022; Revised 15 May 2022; Accepted 25 May 2022; Published 29 June 2022

Academic Editor: Wen-Tsao Pan

Currently, Customers are struggling to retain their business in today's competitive markets. Thus, the issue of customer churn becomes a significant challenge for the industries. In order to achieve this, it is vital to have an efficient churn prediction system. In this paper, we discuss methods for reducing features using PCA, Autoencoders, LDA, T-SNE, and Xgboost. In this paper, a model for predicting light GBM churn is proposed. The model consists of five steps. The first step is to preprocess the data so that missing and corrupt values can be handled and the data can be scaled. Secondly, implementing a comprehensive feature reduction system based on popular algorithms reduces the features and selects the most suitable one. In the third step, light GBM's hyperparameter is tuned using Bayesian hyperparameter optimization and genetic optimization algorithms. Lastly, interpreting the model and evaluating the impact of the features on model outputs by using the SHAP method, and finally ranking the churners by customer lifetime value. Aside from evaluating and choosing the best feature reduction methods, the proposed method is also evaluated using four famous datasets. It outperforms other ensemble and ML algorithms like AdaBoost, SVM, and decision tree on over seven evaluation metrics: accuracy, area under the curve (AUC), Kappa, Mathews correlation coefficient (MCC), Brier score, F1 score, and EMPC. In light of the evaluation metrics, our model shows a significant improvement in handling imbalanced datasets in churn prediction. Finally, in this paper, interpretability and how the features affect the model's output are presented by the SHAP method. Then CLV ranking is suggested for better decision-making.

1. Introduction

Customer churn problem is a big challenge in industries, especially in the telecommunication industry. A churner quits the service provided by operators and yields no profit to any further extent. One of the most effective ways to deal with customer churn is to evaluate customer behavior to detect churn probability. Due to the development of centralized information systems, more information can be obtained about customers' behavior through customer data analysis. Customer retention is one of the primary goals of CRM (customer relationship management). Its significance has resulted in the creation of a number of tools that aid in several critical tasks in predictive modeling and categorization [1]. Employing many different analysis techniques can help evaluate customer behavior. Churn probability calculation can be the basis for calculating the risk of resource depletion, which is one of the most critical parameters affecting an industry's status. To effectively handle customers' churn in a company, it is essential to produce a valuable and accurate customer churn prediction [2]. Churn prediction methods are highly based on artificial intelligence (AI) classification algorithms. These kinds of classification methods face high dimensionality and imbalanced datasets impeding accurate churn prediction. The amount of data accessible for investigation, as well as the balanced or unbalanced nature of the dataset, has a considerable impact on the efficacy of certain data mining approaches [3]. Churn datasets demonstrate the class imbalance problem, in which the churn class (minority class or class of interest) has fewer samples than the nonchurn class (majority class). This makes it harder for some machine learning approaches to identify the minority class [4]. So the ensemble methods demonstrate excellent results in handling imbalanced datasets in churn prediction.

Furthermore, ensemble methods are widely used in churn prediction models. They are accurate and straightforward to use for exploiting sophisticated models with low computational power. They are categorized into two groups based on their structures: parallel and sequential. The parallel ensemble is a set of different machine learning approaches that create an independent model in parallel. Sequential uses first algorithms to learn and generate a model. Then the second algorithm learns to modify the previous model, and thus, it will continue by its specific depths. Recent studies show that the boosting method outperforms the other ensemble methods in accuracy and imbalanced performances [5]. However, there is little focus on light GBM capabilities on churn prediction in literature. In some of the researches, it is utilized as a benchmark model. However, it shows a significant ability to improve models and win challenges in Kaggle.

Additionally, due to the light GBM algorithm's numerous hyperparameters, it is necessary to have an efficient and robust optimization algorithm to enhance the model's performance. The latest improvements in Bayesian optimization (BO) help us do faster hyperparameter tuning and avoid time-consuming computational calculations [6]. Thus, the BO enhances the model hyperparameter tuning efficiency and model performance metrics. Also, genetic optimization (GO) is a powerful optimization algorithm too. This paper uses the genetic algorithm for hyperparameter tuning. It compares it to the Bayesian optimization by their performance in churn prediction. In addition, the churn prediction systems calculate the probability of each customer's churn potentiality. In this procedure, the companies should determine the threshold of churn probability. The determined threshold is crucial for the analysis of customer churn prediction. Therefore, this paper uses threshold interval as a hyperparameter in the tuning algorithm for the first time.

Moreover, when performing complex data analysis, one of the major problems is the number of variables involved. Analysis with many variables generally requires a large amount of memory and computational power or classification algorithm that uses the instructional sample and generalizes to the new instances. There is a need to efficiently tackle the difficulties of high dimensional data analysis as the speed of current computers increases and parallel processing facilities become more affordable. Data mining and statistical approaches are utilized to deal with many feature sets for this purpose, and strategies such as feature reduction and subset selection methods may be employed to increase classification accuracy while requiring a low computing cost of the data mining process [7]. Feature reduction is a general term for constructing a combination of variables to solve high-precision problems [8]. Explicitly, feature reduction aims to select more representative features with more discriminative power over a particular dataset [9].

The following are the main objectives of dimensionality/feature reduction:

(i) The main goal is to select the best features for distinguishing classes.

(ii) Eliminate unnecessary data to reduce complex computations.

(iii) Improve data quality for data compression tasks such as pattern recognition and data mining.

(iv) Increase computational speed.

Therefore, in this paper, the comprehensive feature reduction methods such as Autoencoders, PCA, T-SNE, Xgboost feature importance selection, and LDA are discussed and compared. As a result, the best method has been chosen for the data feature engineering process and then in machine learning models analyses. Moreover, interpretability is necessary for customer churn prediction (CCP) models and helps decision-makers understand the model and features better. Feature or dimensionality reduction helps to understand more about crucial features. The rest of the paper is organized as follows: Section 2 reviews the churn literature and the process of churn prediction systems in the literature. Section 3 discusses the methodologies that are used in this paper. In Section 4, the proposed method has been discussed. Section 5 contributes to the results of experiments of the proposed method and benchmark models. Finally, Section 6 concludes the paper and suggests future research direction.

2. Literature Review of Churn Prediction Methods

Xiao et al. [10] presented a feature selection model of churn prediction using an ensemble algorithm with a dynamic environment. Their suggested model includes two main layers of the GMDH-type neural network and the classification layer. The resulted model has high performance regarding other methods of transfer learning. Idris and Khan [11] overcome the churn prediction imbalanced dataset and presented a classification method. The presented method is based on a filter wrapper and ensemble algorithm. In the first step, they used PSO-based methods for the extraction of the governing features. In the second step, GA was used for dimension reduction. The final phase consists of designing a new feature space and classification using an ensemble algorithm. Results show optimum dimension reduction for two datasets of orange and Cell2Cell. Saghir et al. [12] assessed the churn prediction using a method of neural network begging. The main goal is to increase the evaluation accuracy. Their results show that the presented method reaches 81% accuracy for several datasets. Amin et al. [7] used the feature of distance for churn prediction based on classification certainty. Their categorized dataset consists of two high and low certainty classifiers for churn prediction. The evaluation metrics include accuracy, precision, sensitivity, and f-score. Results show that there is a direct relationship between distance and certainty of classification factors. If distance factors were high, therefore, the classifier obtains high accuracy. This finding was correct for low distance with low certainty. Mou et al. [13] presented the energy-efficient distributed permutation flow-shop inverse scheduling issue to reduce adjustment and energy consumption at the same time. In their work, Zenggang et al. [14] used the service quality of nodes and their own resources to design message pricing functions that restrict the capacity of nodes to make fraudulent offers by disclosing the resource status of both nodes. Höppner et al. [15] presented models of churn detection for achieving a high-profit model. The model was based on the expected optimum profit for the churn of customers. They presented a model of ProfTree for classification and designing decision trees. The model outperformed state-of-the-art based on the article's results. Maldonado et al. [16] provide a new churn prediction method that drives profit. The method is a developed version of the minimax probability machine algorithm for the classification of churns. The presented method consists of LASSO and Tikhonov algorithms for regularization. Results show that the presented method plays an essential role in increasing company profits. Calzada-Infante et al. [17] presented a churn prediction method based on customer behavior analysis. They used the social network to assess customer behavior using classification methods. They divide customers into churners and nonchurner. The results were illustrated on-time order and static temporal diagram with proximal metrics. Results show outperform of the presented method in comparison with other approaches. Taghizadeh and Ahmadi [18] purposed an innovative strategy to classify and analyze a long-run link and convergence among a variety of knowledge variations in the context of the knowledge-based economy (KBE). Zhu et al. [19] suggested a unique two-step reconstruction approach for enhancing the resolution of reconstructed temperature distribution pictures while maintaining a high degree of precision. According to Yang et al. [20], the aggregated vehicle fuel consumption data are protected against time-series-based differential attacks using a negative survey method. Khodadadi et al. [21] used deep learning methods for the prediction of customer churn. They used return time for the analysis, using recurrent neural networks. Their dependent variable is customer loyalty. Their ChOracle methods presented with high accuracy and performance for churn classification. Amin et al. [22] provide a method of churn prediction with the use of cross-company data. Mahmoudi [23] has developed an innovative strategy for mitigating the detrimental impact of coronavirus on the U.S. economy. Li et al. [24] introduced an innovative collaborative prediction unit (CoPU) that combines the predictions from many collaborative predictors based on a collaborative graph. Özmen et al. [25] developed ant colony optimization methods for optimization and genetic algorithm for feature extraction of data. They applied the model to 100 companies of Turkey to detect customer churn. The AUC of the model reached 99% that is competitive in comparison with other methods. Li et al. [26] developed a unique tutorial that uses the viewpoint of pre-coding design in a multiantenna wireless communication system to address interference exploitation strategies. Kostić et al. [27] used graph theory for churn analysis of telecom companies from the social network. Customer friendship and connection in social media are also effective in churn prediction in their model. They used graph theory to find effective nodes of customer connection in social media. The paper resulted in a useful model for churn prediction classified customers based on a telecom company's churn. Li et al. [28] have examined how symbol-level precoding may enhance the error-rate performance of large MIMO systems deploying 1-bit digital converters. Using logarithmic fuzzy preference programming, Ahmadi [29] has ranked and categorized the growth-affected knowledge-based economic indicators. Cai et al. [30] presented multiple relaxed control techniques for complex-valued memristive neural networks based on data quantization (CNNs). Ahmed et al. [31] presented an ensemble learner for achieving their aims. They used a hybrid method of boosted-stacked learners and bagged-stacked learners. They applied the method to the SATO company for performance analysis. Mahmoudi [23] investigated genetic algorithm evidence in determining the optimal point of purchase intention. The study has been used extensively by automotive activists. Liang et al. [32] suggested an agent-based model (ABM) to simulate the impact of policy initiatives on inhabitants' decisions on the usage of green space. The result shows that the method accuracy reaches 94.4% for the SATO database. Ahmadi and Qaisari Hasan Abadi [33] studied the object-oriented properties of C++ to design expert systems for strategic planning. Jain et al. [34] used logistic regression and logit boost method for churn analysis in the telecom industry. They first used the PCA method for feature selection of customer churn. Then the floating-searching approach is applied for churn analysis. The method outperforms other related approaches. Castanedo et al. [35] used different algorithms, including autoencoders, deep belief networks, and multilayer feedforwards networks for CCP by exploiting business intelligence dataset of millions of call records. They achieve higher AUC than traditional methods by using deep models. Nekah et al. [36] examined the relationship between brand personality, brand reputation, and self-image of Iran Khodro Company with the customers' purchasing intention. Spanoudes et al. [37] test abstract datasets for a company's software in churn prediction on many datasets attained by subscription and nonsubscription companies. They designed a deep learning pipeline that has a vast application on any subscription type business. However, their model does not show improvements in nonsubscription in comparison to subscription-type companies. They examine different layers and architecture and data reduction techniques over churn prediction problems. Their results show that data dimensionality reduction techniques like t-SNE are helpful in the preprocessing phase of deep models. Halibas et al. [38] consider different approaches and deep models in the telecommunication sector. The models are constructed on RapidMiner software. Their results compare the algorithms by various metrics by exploiting upsampling techniques. However, their results show that GBT is better in comparison to the deep model. Saghir assessed the churn prediction using a method of neural network begging. The main goal is to increase the evaluation accuracy. Their results show that the presented method reaches 81% accuracy for several datasets. Table 1 shows the summary of research papers.

As the literature shows, the majority of papers emphasize one or two methods of feature reduction. Also, they did not consider both machine learning and deep learning methods on feature reductions. It is essential to have a comprehensive feature reduction algorithm evaluation because, in industrial applications, the bulk of features may cause high computational time and cost. Moreover, there is no way to understand, which feature reduction methods make a more reliable and good result according to the business goals. So, it is critical to evaluate all-powerful methods to choose the best feature reduction method. This paper discusses the most common feature reduction methods, including machine and deep learning algorithms.

Furthermore, the evaluation metrics are not sufficient to churn problems. In churn classification, methods should always compare models by their performance towards the imbalanced nature of datasets since the churn problem is highly imbalanced due to the lack of churners' data records. It is also essential to understand the model prediction outputs strength and their meaningfulness. Therefore, it is necessary to evaluate churn classification models with all related metrics to conclude the best method. Additionally, it is a priority to enhance model performance due to its vast hyperparameters. The ensemble methods lack concentration on the hyperparameter tuning algorithm, so the hyperparameter tuning-based light GBM method has been presented in this paper. Finally, a few papers focused on churn prediction interpretability as the main problem. Accordingly, in the industry, managers are also interested in knowing the model behavior and understanding how and why the model works correctly.

Furthermore, customer lifetime value is an essential key performance metric (KPI) for marketing efforts. This topic has received little attention in the literature. The customer lifetime value is presented as an essential component of the CCP model in this article. Thus, in this paper, one of the main focuses is model interpretability.

3. Churn Prediction

This section presents the stages of implementation of the churn prediction system in the literature. As illustrated in Figure 1, the process of churn prediction consists of five stages, including understanding business and customer features for analyzing churn problems, collecting customers' data, preprocessing the data, building the machine learning models, and finally, evaluation of prediction. Then researchers decide on conclusions and strategies towards churners.

3.1. Data Sources. Many datasets have been used in the existing solutions with different features of customers. Each of them is related to a specific industry. Some papers have been conducted on the finance sector, including banking, insurance, and funds [2, 48, 49], retail industry [50, 51], game [6], music [52], e-commerce [53], and telecommunication [15, 22, 25, 54–56]. It is worth noting that there are many datasets in the vertical of telecommunication due to the vast number of customers and available data.

3.2. Data Features. In business understanding and the customer features stage, the crucial attributes for measuring churn will be determined. Customer features, such as demographics, financial information, usage behavior, call detail records, tenure, and network structure of subscribers, have been used for more accurate customer churn prediction (CCP) models. Moreover, individual features such as recency, frequency, and monetary (RFM) [50], texts [57] and tweets [58], and log data [6] have been used in the churn prediction models. Table 2 shows some features of each financial, telecommunication, and e-commerce source.

3.3. Pre-Processing Methods. First, in the preprocessing section, most of the researchers exploit data cleaning techniques. Data cleaning is about detecting, correcting, or removing corrupt data. Furthermore, the next part is handling categorical features. The categorical features of customers like demographic data are usually vital in CCP models analyses. The critical part of data preprocessing is the imbalanced control of datasets. Churn datasets are highly imbalanced. Imbalanced data make a considerable impact on model evaluation metrics. There are several techniques for handling an imbalanced dataset. Under-sampling, upsampling, and algorithmic approaches are prevalent among imbalanced learning techniques. In this stage, some researchers also conduct descriptive data analysis (EDA) to understand the data better.

4. Methodology

In this section, the methods which are used in this paper have been discussed. It is essential to understand how these models work for more understanding of the results.

4.1. Feature Reduction Methods. Features are properties of objects whose values must be the same for objects in a particular class and objects in other classes (or backgrounds). Features may be continuous (such as numerical values) or deductions (such as tagged values). Length, area, and texture are examples of continuous variables, in contrast to the deductive or arranged features, such as where the ordering of labels has meaning (for example, class buildings, military grades, and degrees of satisfaction) or nominal, such as when there is no meaning (e.g., name, postal code, and department). Choosing the right features depends on the type of image and application that is desired. Feature reduction is used to simplify the features required to accurately describe a large set of data [8].

Table 1: Literature review summary.

References	Prediction method	Industry	Evaluation metrics
[35]	Feedforward network, deep belief network, and Autoencoder	Telecom	AUC
[39]	SMOTE, decision trees, random forest, AdaBoost, multilayer perceptron, K-means and Bayesian logistic regression	Telecom	P-value, accuracy, AUC, precision, recall, F1-Score
[40]	CCPBI-TAMO, CPIO-FS	Telecom	Precision, recall, accuracy, F-Score, ROC
[41]	Xgboost, AdaBoost, catboost, decision trees, SVM, KNN	Telecom	Accuracy, AUC, precision, recall, F-Measures
[37]	Deep feed-forward networks	Subscription companies	Accuracy
[38]	Deep ANN, machine learning algorithms	Telecom	Accuracy, precision, recall, F1-score, and AUC
[12]	Neural network with bagging	Telecom	Accuracy, precision, recall, F-score, kappa, absolute error, relative error, and classification error
[10]	Transfer learning of ensemble	Telecom	Area under curve of ROC (AUC) and complexity
[11]	Ensemble algorithm	Telecom	Area under curve of ROC (AUC)
[12]	Begging and neural network	Telecom	Accuracy and precision of classification
[42]	Artificial neural network (ANN) and self-organized map (SOM)	Telecom	Accuracy, recall, F-score, and precision
[15]	Profit tree	Telecom	Accuracy, cost, and profit
[16]	Minimax probability machines	Telecom	AUC and EMPC
[17]	similarity forests	Telecom	AUC, and tenlift AUPR
[21]	Temporal point processes (TPP) and recurrent neural networks (RNN)	Telecom	MAE and MRE
[22]	Cross-company just-in-time approach	Telecom	Accuracy, Kappa, and Recall
[25]	Multiobjective and colony optimization	Telecom	AUC
[27]	graph theory	Telecom	Top decile lift
[31]	Boosted-stacked learners and bagged-stacked learners	Telecom	G means and AUC
[34]	Logistic regression and Logit Boost	Telecom	Accuracy, Kappa, MAE, coverage case, and RMSE
[43]	Decision tree, artificial neural, networks, K-Nearest neighbors, and support vector machine	Telecom	Confusion matrix, precision, and recall
[44]	Logistic regression and random forest	Telecom, bank, and retail	AUC and cross-validation
[45]	Genetic algorithm, classification, and covering & LEM2 (rules generation algorithms)	Telecom	Accuracy, recall, mutual information, coverage, and F measure
[46]	Data transformations, DT, KNN, DNN, GBT, and SRI	Telecom	Probability of detection, probability of false alarm, area under the curve, and g-mean
[47]	Exhaustive, genetic, covering and LEM2	Telecom	TP, FP, FN, TN, COV, PRE REC, ER, ACC, and SPEFM

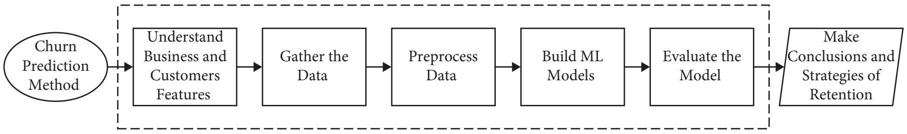
Churn Prediction Method
Understand Business and Customers Features
Gather the Data
Preprocess Data
Build ML Models
Evaluate the Model
Make Conclusions and Strategies of Retention

Figure 1: The process of churn prediction.

Table 2: Industries and their features.

Industry	Features
Financial	Credit score, balance, services, customer transactions
Telecommunication	Payment methods, monthly charges, internet services, number of lines, call details records
E-commerce	Frequency, recency, monetary, products, log data

4.2. Principal Component Analysis (PCA). The study of principal components is one of the outcomes of linear algebra mathematics because the nonparametric and straightforward method extracts relevant information from confusing sets. The transformation of the T can be obtained by minimizing the least-squares error, assuming that the vector of properties is $X \in R^d$ (d-dimensional space), then the space of the reduced features [59]. $Y \in R^h$. If the vector of properties is $X \in R^d$ then, the least-squares error can be in the form of the following equation:

$$\mathrm{MSE} = E\left[\|x - \widehat{x}\|^2\right]. \tag{1}$$

In PCA, we look for a conversion that obtains the least-squares error, which means the following equation is desired.

$$T_D = \arg\min_{r}\left\{E\left[\|x - \widehat{x}\|^2\right]\right\}. \tag{2}$$

As stated, the PCA goes to find a linear transformation T that produces the least-squares error and maximizes this linear transformation $T^T \operatorname{cov}_{x-\widehat{x}} T$ is maximized, in which $\operatorname{cov}_{x-\widehat{x}}$ is the covariance matrix of data with an average of zero of X. Therefore, the PCA solves the special problem, and the vector of special features of the linear transformation columns forms the T-matrix. The data are shown in the lower dimension $Y = (x - \widehat{x})$, namely Y. According to the above, MSE has at least $d - h$ value; that is, a special large vector corresponding to a large specific value should be taken to reduce the dimension or to the category objective for the minimum MSE. Here it is necessary to address some of the properties of the covariance matrix because in fact, the main problem is to find the special values of the covariance matrix. For example, in three dimensions, the covariance matrix is as follows [59]:

$$C = \begin{pmatrix} \operatorname{cov}(x, x) & \operatorname{cov}(x, y) & \operatorname{cov}(x, z) \\ \operatorname{cov}(y, x) & \operatorname{cov}(y, y) & \operatorname{cov}(y, z) \\ \operatorname{cov}(z, x) & \operatorname{cov}(z, y) & \operatorname{cov}(z, z) \end{pmatrix}. \tag{3}$$

The main goal of PCA is to obtain components that do not affect the external data layer, and a powerful covariance estimate matrix has replaced the covariance matrix. If the primary data of the problem is matrix $n \times p$ or $X = X_{n,p}$ N is the number of observations, and p specifies the number of variables.

4.3. *t-distributed Stochastic Neighbor Embedding (t-SNE).*

One of the feature extraction algorithms is t-SNE, which defines a way to convert an extensive dataset into two or three dimensions. Randomly t-distributed neighborhood embedding can maintain a large part of the local structure of high-dimensional data, while also showing the overall structure, such as clusters at different scales. Randomly distributed neighborhood embedding is a nonlinear non-observational method used to explore and visualize data. Simply put, this method provides the user with an understanding of how data is organized in a high-dimensional space.

The built-in algorithm randomly estimates a similarity criterion between pairs of samples in high-dimensional data and low-dimensional space. It then optimizes these two similarity criteria using the cost function. The similarity between points in space dimensions is calculated in the first step. To better understand the subject, we can consider data points scattered in a two-dimensional space. The user around that point centers on the Gaussian distribution. Then the density of all points of the flat Gaussian distribution is calculated. Normalization is then performed for all data points. This provides a set of probabilities (P_{ij}) for all data points. These probabilities are comparable to the similarities and mean that their ratios and observations are identical if the data points x_1 and x_2 have the same values under the Gaussian circle. Then local similarities are given in the structure of the high-dimensional space and distribution or circle. The embedding similarity q_{ij} between the two points y_i and y_j is calculated as a normalized Student-t kernel with a single degree of freedom [60].

$$q_{ij} = \frac{\left(1 + \left\|y_i + y_j\right\|^2\right)^{-1}}{\sum_{k \neq 1}\left(1 + \left\|y_k - y_l\right\|^2\right)^{-1}}, \quad q_{ij} = 0. \tag{4}$$

The place of the y_i is computed by minimizing the Kullback Leibler divergence between the joint distributions P and Q as follows [60]:

$$C(\epsilon) = L(P\|\mathrm{Q}) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{5}$$

The second step is identical to the first step. However, the Student T distribution is used with a degree of freedom rather than the Gaussian distribution. This gives the second collection of Q_{ij} probabilities in a low-dimensional space. The final step is for this set of probabilities from the lower Q_{ij} space to better match those in the higher P_{ij} Space. The nonconvex objective function is reduced by reducing along the gradient as follows [60]:

$$\frac{\partial C}{\partial y_i} = 4 \sum_{j \neq i}\left(p_{ij} - q_{ij}\right) q_{ij} Z\left(y_i - y_j\right). \tag{6}$$

4.4. *Autoencoders.*

Autoencoders, of which several versions have been presented, are helpful tools in various fields such as data compression, extraction of features without supervision, reducing the size of data, and building productive models. Figure 2 shows the architecture of Autoencoder models.

The autoencoders are neural networks that try to copy their input to the network output. In general, they have a hidden layer h that creates a coded expression of the input data. These networks can be divided into two parts: an encoder function in the form of $h = E(x)$, which is the back of the neural network that maps the input to the hidden h. It also consists of a decoder function in the form of $r = D(h)$, like the encoder of a neural network, which is responsible for reconstructing the input data based on its encoded expression in the hidden layer. The following figure shows this architecture. It would not be helpful if the autoencoders only learned to map their input to output as $r = D(E(x))$. What matters about these networks is how they are designed [61]. These networks are designed in a way that they can not completely copy the input to their output. Restrictions are usually imposed on these networks that prevent them from

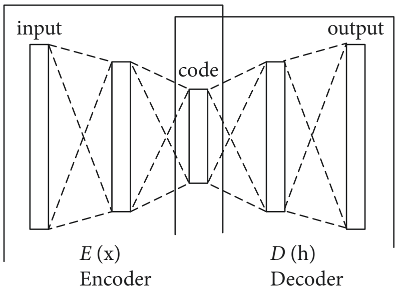

Figure 2: The Autoencoder architecture.

completely copying the input to the network's output and can only do so on an estimated basis. Due to limitations, the network is forced to learn, which aspects of the data are more prioritized for better data recovery at the output to maintain them. This allows the network to extract useful features from the data. One of the limitations is reducing the number of middle layer neurons relative to their encoder input dimensions. This example is shown in Figure 1, where the number of middle layer dimensions p and the number of input dimensions d are less. The MSE cost function or cross-entropy is usually used to train these networks. Also, like most networks, weights are updated using backward propagation algorithms. Equation (8) shows the definition of a network and equation (9) shows the cost function of a typical autoencoder network [61].

$$\begin{aligned} h_{W,b}(x) &= D_{W,b}\left(E_{W,b}(x)\right), \\ L(x; W, b) &= \left\|x - h_{W,b}(x)\right\|^2. \end{aligned} \tag{7}$$

4.5. Linea Discrimination Analysis (LDA). As the name implies, the goal is to find linear discrimination that separates data into known classes. This separator can be a line or, in higher dimensions, a surface. It is noteworthy that this method, in addition to data classification, also reduces the size of the data space. This classification model writes the data into a new space, which is one unit less than the classes defined for the data. These new dimensions are calculated under the components of the lines or surface, separating the data [62]. Suppose the input data are defined in two dimensions. In that case, the primary purpose of this supervised model is to find the direction by which the minimum in-class and out-of-class distance is maximized by looking at the data on it, i.e., in-class data have a minimum distance in the class and data in two different classes have a maximum distance from each other. In the LDA method, first, the model can be trained by training data, and the separating matrices inside the class make the classification model of appropriate directions S_w (total variance of members of each class) and outside class (S_B) and then the performance of the model by test data are benchmarked. The primary criterion of separation in LDA method calculations is the distance criterion [62]. In this method, a two-dimensional matrix with dimensions $a \times b$ is first converted into elemental $a \times b$ vectors. The two scattering matrices inside the class S_w and outside class S_b are defined as follows [62]:

$$S_w = \frac{1}{n}\sum_{j=1}^{c}\sum_{x\in c_j}\left(x-\overline{x}_j\right)\left(x-\overline{x}_j\right)^T, \tag{8}$$

$$S_b = \frac{1}{n}\sum_{j=1}^{c} n_j\left(\overline{x}_j-\overline{x}\right)\left(\overline{x_j}-\overline{x}\right)^T, \tag{9}$$

where c is the number of classes and $\overline{x_j}$ is the average class of j. One way to obtain the W_{LDA} conversion matrix is to use the Fisher criterion, which is obtained by maximizing the following equation:

$$J\left(W_{LDA}\right) = \frac{W_{LDA}^T S_b W_{LDA}}{W_{LDA}^T S_w W_{LDA}}, \tag{10}$$

where W_{LDA} is obtained by a set of particular vectors corresponding to the largest eigenvalues $S_W^{-1}S_b$.

4.6. Baseline Model. In this section, the explanation of light GBM is explained. It is included the main steps of performing these approaches.

4.6.1. Light GBM. Boosting is a sequential decision-tree-based ensemble machine learning algorithm. The algorithm uses a gradient boosting approach. The trees are built successively in order to minimize each previous tree loss function. Each tree learns from its predecessors in this procedure and then updates the residual errors until reaching specific depths. Later, the next tree that grows in the sequence will learn from an informed type of errors [63]. Light GBM is another boosting decision tree that uses a robust classifier with multiple weak classifiers [64]. It differs from Xgboost by its structure of one-side sampling to filter out the data splits. The model helps to enhance the training speed and efficiency.

4.7. Benchmark Models. The benchmark models that have been used in this paper for evaluating the results against the proposed method are explained in this section.

4.7.1. Support Vector Machines (SVM). Support vector machine is a supervised learning model that is structured based on risk minimization. The kernel function is used to improve performance. Research is taking place to find the best kernel or combination of them. The loss of prediction with SVM from the decision tree and sometimes ANN worked best. For example, the reference [43] has used the SVM method in predicting loss and has shown that SVM performs better than other methods.

4.7.2. Decision Tree (DT). The decision tree is a tree-type structure in which decisions are made that can create rules to classify specific datasets. In this tree structure, the leaves represent the class label. The branches represent the intersection of the attributes that lead to those labels. The decision tree does not perform well in complicated problems and nonlinear relations. While in the case of customer loss, the accuracy of the decision tree can be high, which depends on the data. This method has been used to predict loss in many articles, including [43, 65, 66]. The high popularity of this method in the field of loss is due to recognizing the causes and characteristics of falls based on the division of branches. The random forest algorithm, which is a type of random tree learning, also has similar features.

4.7.3. Random Forest. The stochastic forest is an ensemble learning algorithm that consists of a set of primary regression trees (basic algorithm). A regression tree generates a partition of a data set using consecutive binary divisions based on covariance variables. In each obtained subset (called leaves), the observations belong to a homogeneous group. The key to forming a group is to choose a division criterion (e.g., least-squares) that aims to reduce heterogeneity at each stage [67].

4.7.4. Adaboost. Adaboost or adaptive boosting is a popular boosting technique that helps weak classifiers become a single robust classifier. Weak learners are classifiers that produce a prediction that is better than random guessing. The structure of AdaBoost works by sequentially applying a classification algorithm to reweighted forms of the training dataset. Lastly, the algorithm will take a weighted majority vote of all classifiers [68].

4.8. Optimization Algorithms. In this part, the optimization algorithm's structure will be discussed. Genetic and Bayesian algorithms will be used in hyperparameter tuning of boosting algorithms.

4.8.1. Genetic Optimization (GO). Inspired by nature, the evolutionary principle of "survival of fitness" is the genetic algorithm (GA). Although GA was proposed after the evolutionary strategy algorithm, it is the most popular evolutionary algorithm method. In a genetic algorithm, a population of individuals survives in the environment according to their best adoption in an environment based on a fitness function. A greater chance of marriage and childbirth would be for those with higher skills. Therefore, with improved skills and adoptability, children will be born to new generations. Each person in the population is inserted as a chromosome in the genetic algorithm. Over a few years, chromosomes have increasingly become complete.

Chromosomes are tested in each generation and allowed to survive and replicate by their fitness function value. Generation is discussed in the genetic algorithms with intersection and mutation operators [69]. A collection of search space points is randomly treated at each stage of implementing the genetic algorithm. Each point is assigned a character set in this process, and these sets are applied to genetic operators. For obtaining new points in the search space, the resulting sets are then decoded. Finally, the likelihood of their involvement in the next step is determined by evaluating each point's objective function. Genetic algorithms can be considered as a directional random optimization method that progressively moves towards the optimal point. Regarding the characteristics of the genetic algorithm compared with other optimization methods, it can be said that it is an algorithm that can be applied to any problem without any restrictions on the type of its variables and has a demonstrated efficiency of obtaining the global optimum. This method's ability to solve complex optimization problems is why this algorithm is so popular. Also, the classical methods are either not applicable or unreliable to find the general optimization is not reliable [69]. Figure 3 shows the diagram of genetic algorithms.

4.8.2. Bayesian Optimization (BO). In Bayesian optimization, the goal is to find a new solution y_{N+1} that maximizes the chance of refining the accuracy score of f_{N+1}, where the chance is often defined as a surrogate function $a(y_{N+1}|D_N)$. In Bayesian optimization $\{y_1, \ldots . yn\}$ is a set of solutions, $f = f(x, y)$ is assumed to be drawn from a probabilistic model $p(f|D_N|)\alpha p(D_N f)p(f)$, where $D_N = \{(y_j, f_j)\}_{j=1}^{N}$ and $f_j = f(x, y_j)$. Then, the algorithm proceeds by recursively generating a new solution y_{N+1} from $D_{N+(t-1)}$ and update the set $D_{N+t} = D_{N+(t-1)}U\{y_{N+t}\}$ to make a new sample set of solution $y_{N+(t+1)}$ with an informed, updated observation [70]. The whole process of Bayesian optimization is shown in Figure 4.

4.9. Model Explainability. The accuracy and interpretability of machine learning models are essential for data scientists, researchers, and practitioners to explain their models and comprehend the worth and accuracy of their findings. Meanwhile, explainability refers to how well the internal mechanics of a machine or deep learning system can be articulated in human words.

4.9.1. SHAP. SHAP (SHAPly Additive explanations) is mainly used in clarifying model outputs. SHAP is predominantly founded on game theory and neighboring clarifications. It suggests a way of assessing the dedication of each element. Assuming a boosting model where an output (N) is foreseen using a group N (with n features). Depending on their marginal commitment in SHAP, each variable (φ_i is the engagement or commitment value of feature i) on the model performance (N) is distributed [71].

$$\varnothing_t = \sum_{s \subseteq N_{\{i\}}} \frac{s!(n-|s|-1)!}{n!}[\nu(s \cup \{i\}) - v(s)]. \qquad (11)$$

Depending on the subsequent preservative element, the attribution method is considered to be a linear function of binary features t

$$t(z') = \phi_0 + \sum_{i=1}^{I} \phi_i z'_i. \qquad (12)$$

4.10. Evaluation Measures. The evaluation measures are the most critical part of identifying the best model in our churn prediction methods. This paper use seven different measures, such as simple accuracy, imbalanced measures, AUC, kappa, Mathew's correlation coefficient, f1-score, and Brier score. Moreover, the EMPC metric is used to understand the model's performance towards profitability for the business.

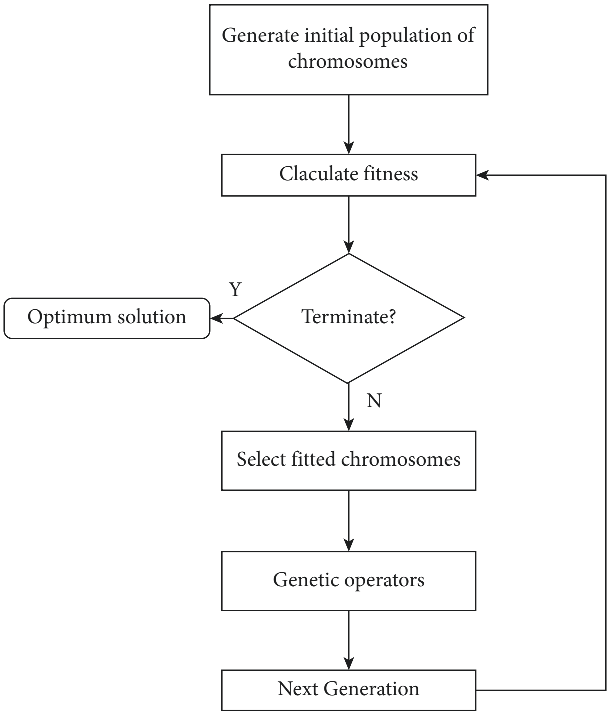

Figure 3: Genetic hyperparameter tuning flowchart.

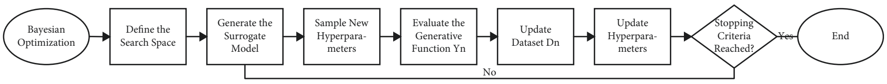
Bayesian Optimization
Define the Search Space
Generate the Surrogate Model
Sample New Hyperpara-meters
Evaluate the Generative Function Yn
Update Dataset Dn
Update Hyperpara-meters
Stopping Criteria Reached?
~~Yes~~
End
No

Figure 4: Bayesian optimization flowchart.

4.10.1. Accuracy. The accuracy informs the proximity of the model to the known values. There are four separate outputs in the binary classification, namely true-positive (TP), false-positive (FP), true-negative (TN), and false-negative (FN). The accuracy is simply calculated by the formula TP + FN/ TP + TN + FP + FN. It is commonly used as a priority criterion for assessing the models in churn prediction systems.

4.10.2. Area Under the Curve (AUC). The receiver operating characteristic curve (AUC) indicates the model's power to separate groups. The receiver operating characteristic (ROC) curve is a central assessment of the classification model in the imbalanced data type distribution. In binary predictions, the ROC graph is a true-positive (TP/(TP + FN)×100%) or false-positive (FP/(FP + TN)×100%) graph. It is not easy to compare the ROC of different models, so it is better to use the AUC [72]. Table 3 shows the interpretability of AUC results.

4.10.3. F1-Score. F1-score is one of the critical measures to evaluate imbalanced classification models. It works based on false positive and false negative rates.

It is calculated by 2 ∗ (Precision ∗ Recall)/(Precision + Recall) formula. The higher score indicates better the model, especially toward imbalanced datasets.

Table 3: AUC value interpretation.

Interpretation	AUC
Weak	(0.5, 0.6)
Fair	(0.6, 0.7)
Good	(0.7, 0.8)
Very good	(0.8, 0.9)
Excellent	(0.9, 1.0)

4.10.4. Mathews Correlation Coefficient (MCC). MCC is a performance metric that is less influenced by imbalanced datasets by considering both accuracy and error rates in both classes [73]. MCC is obtained from (13). MCC results between −1 (worst) to 1 (best). So, a more positive rate will consider that classifier as a better classifier.

$$\mathrm{MCC} = \frac{(\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN})}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}. \tag{13}$$

4.10.5. Cohen Kappa Score. Kappa score will tell about how much agreement between observed agreement and expected agreement. In simple words, it tells how much the model is better than a random classifier [73]. Table 4 indicates the interpretation of the kappa score output.

4.10.6. Brier Score. Brier score is a score function that shows the accuracy of the probabilistic prediction of the classifier. Brier score ranges from 0 (perfect prediction) to 1 (poor prediction) [5]. Brier score is obtained from the following equation:

$$\frac{1}{N}\sum_{t=1}^{N}\left(F_t - O_t\right)\wedge 2. \tag{14}$$

N = the number of items, F_t is the forecast probability, and O_t is the outcome.

4.10.7. The Expected Maximum Profit of the Customer Churn (EMPC). The EMPC measures the profitability of the model in churn prediction systems. It was first suggested by [74]. The function measures the model's profitability by considering accuracy, cost of retention, cost of contacting, and customer lifetime value (CLV). The higher rates indicate the best good classifier in churn prediction. So, it is widely used to measure the performance of churn prediction systems due to its business functionality. In this paper, the constant CLV (200) and cost of retention (10), and the cost of contact (1) are being used.

5. Proposed Method

The proposed method flowchart is shown in Figure 5. The proposed method is conducted in three steps:

Step 1. The aim is to find the best feature reduction method for further model implementation. First, data are uploaded, and preprocessing methods like cleaning and scaling are executed. Then, the feature reduction methods by considering light GBM as a baseline model are evaluated on the datasets. Finally, the best feature reduction method is chosen for further analysis.

Step 2. The aim here is to use the reduced feature datasets to train and test on the benchmarks models. The code is prepared, and the reduced feature dataset is utilized for comparing the results with the optimization model.

Step 3. It is the foremost step in this method. It incorporates optimizing the light GBM hyperparameters by genetic and Bayesian, then training on reduced features dataset and evaluating the test dataset each iteration. Then, the best results will be published after the completion of tuning algorithms. Furthermore, evaluation results will be compared to benchmark models. After that, the model results and influences of each feature on outputs are going to be presented. Finally, churners by their CLV for other marketing campaigns will be ranked.

Table 4: Kappa score interpretation.

Agreement interpretability	Kappa
Minor agreement	(0.01,0.20)
Reasonable agreement	(0.21,0.40)
Moderate agreement	(0.41,0.60)
Considerable agreement	(0.61,0.80)
Perfect agreement	(0.81,0.99)

5.1. Feature Reduction. The preprocessed dataset is composed of numerous features. Handling such enormous features causes high computational costs and storage space [75]. It is necessary to avoid this volume of features by feature reduction methods. Feature reduction methods can be categorized into two groups: dimensionality reduction and feature selection methods. In this study, we examined different popular approaches on datasets. Light GBM as a base learner will choose the best model. PCA uses principal components to reduce the feature numbers and map them on two or three eigenvectors. However, LDA uses the best discrimination line to separate. Moreover, T-SNE uses T distribution to find the best feature reduction line. Autoencoder uses a neural network approach to encode the features into some blocks and decode them to make those features again. It will update the weight results after each epoch. Autoencoder's structure is built by one neuron, and the encoded blocks will be used in the baseline models. However, the feature importance by Xgboost is different from other approaches. It identifies the ten most important features used to find churners in the rules of a decision tree structure.

5.2. Hyperparameter Tuning of Light GBM. The algorithm has implemented the Bayesian optimization. It conducts by determined iteration, and it updates the hyperparameters by considering the surrogate function and the chances of getting better accuracy. The genetic algorithms, as discussed, will choose the best hyperparameters based on the best parents mating and their offsprings. Regarding having optimum and efficient hyperparameters, this paper uses Bayesian and genetic optimizations. On the other hand, there are some other methods.

In comparison, they are not as fast as algorithms like Bayesian optimization and genetic algorithm. The model optimized six hyperparameters of boosting algorithms that these hyperparameters by few minutes for full model training in IBM. Besides the long duration of training by grid search, it wastes many resources of storage and computational space. So, it is not applicable for efficient software to use this method as a hyperparameter tuning. Although it is possible to use random search for time-efficiency, this method does not update the hyperparameters as informed as Bayesian and genetics do. Thus, the model may get a local optimum than more general optimums of hyperparameters. Random search works with different subsets of hyperparameters that are chosen in randomness. Therefore, it does not always make good predictions. Then it could not be reliable to use the random search for tuning the algorithm.

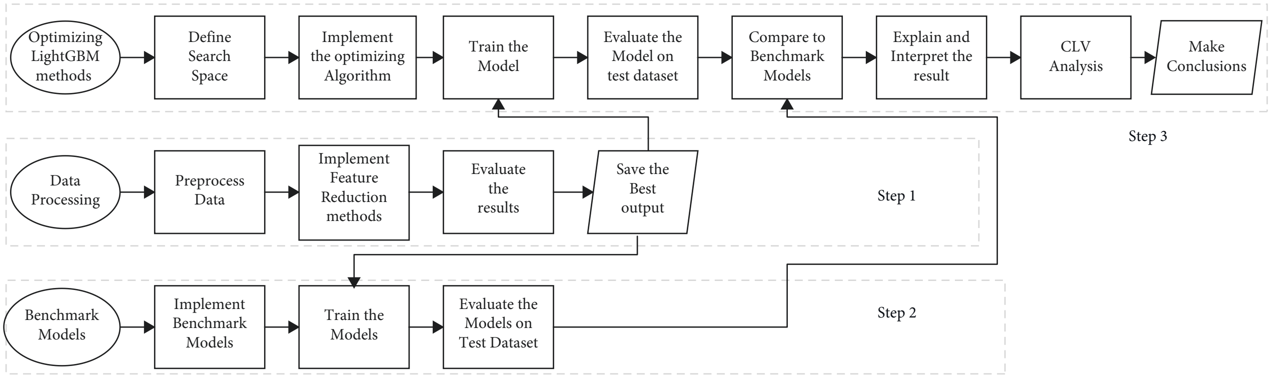
Optimizing LightGBM methods
Define Search Space
Implement the optimizing Algorithm
Train the Model
Evaluate the Model on test dataset
Compare to Benchmark Models
Explain and Interpret the result
CLV Analysis
Make Conclusions
Step 3
Data Processing
Preprocess Data
Implement Feature Reduction methods
Evaluate the results
Save the Best output
Step 1
Benchmark Models
Implement Benchmark Models
Train the Models
Evaluate the Models on Test Dataset
Step 2

Figure 5: Flowchart of proposed method.

Also, it should be considered that grid search and random search are not optimization algorithms. They only use a subset of hyperparameters, but in genetic, it will update hyperparameters based on the merits of parents and their offspring in the genetic natural selection process. Bayesian optimization will update hyperparameters by their impact on the likelihood of better prediction by model training. As optimization and time-efficiency matter in churn prediction systems, the two optimization algorithms' Bayesian and genetic algorithms are conducted and compared together in this paper.

5.3. Benchmark Models. In order to evaluate the proposed model performance, it is compared to other ML models such as AdaBoost, support vector machine (SVM), decision tree (DT), light GBM, Xgboost, and random forest. These models are trained on the preprocessed and reduced features of the datasets and then evaluated on the testing dataset. Finally, the results of these methods are going to compare against the baseline model. The benchmark models are evaluated by 10-fold cross validation.

5.4. Model Explainability. According to the importance of churn prediction, it is essential to understand the model. There are so many features in datasets that are irrelevant or redundant. Therefore, many papers conduct feature importance analysis [74]. Also, SHAP is newly introduced for feature explanations. Shapely values are obtained by integrating concepts from game theory and local explanations. For each feature, the SHAP value explains the contribution to explain the difference between the average model prediction and the instance's actual prediction.

5.5. Customer Lifetime Value. In customer relationship management, customer lifetime value (CLV) is a critical measure. It may be used to enhance market segmentation and resource allocation, analyze competitors, personalize marketing communication, optimize product offering timing, and estimate a firm's market worth. The present value of all future cash flows from a client is the CLV [76].

5.6. Experiment. In this paper, the tuned light GBM churn prediction is considered as a baseline model. The model implementation can be divided into five main steps: collecting the dataset, data preprocessing, feature reduction conducted by evaluating five main approaches, and finally, hyperparameter tuning evaluation on baseline model and evaluating the model with other machine learning algorithms in churn prediction. Also, describing the model interpretation.

5.7. Datasets. The datasets used in this study are from 3 real datasets. They are utilized to verify the performance of the different approaches. IBM telecom (https://www.kaggle.com/blastchar/telco-customer-churn), bank customers (https://www.kaggle.com/santoshd3/bank-customers), and orange (https://www.kaggle.com/mnassrib/telecom-churn-datasets) datasets are downloaded from the Kaggle website. They have also been used in other papers. They have a different ratio of churners that show the highly imbalanced nature of the CCP problems. The summary of the datasets is presented in Table 5.

6. Results

6.1. Data Preprocessing. When dealing with real-world data, there is always the chance that it is imperfect, i.e., it may be noisy, have some missing values, or be inconsistent, and all these factors can lead to misleading results by the machine learning system. Therefore, improving the overall quality of the data will consequently improve the results. In order to do this, the categorical columns transfer into numerical using python libraries (LabelEncoder and get_dummies method of Pandas library). These libraries make the categorical columns into separate columns that each column is represented the individual category. Mathematically, this process could be represented as $1_A(x) = \begin{cases} 1; & \text{if } x \in A \\ 0; & \text{if } x \notin A \end{cases}$. For example, $A = \begin{bmatrix} A \\ B \\ C \end{bmatrix}$ will become $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Thus, it only contains zero and one in each column that represents its category, for example, A, B, and C. So, the number of columns will be

Table 5: Data resources summary.

Dataset	Rows	Columns	Ratio of churners to whole dataset (%)
Bank Customer	10000	14	20.3
IBM Telecom	7043	21	26.5
Orange	3333	20	14.5

increased by two. Therefore, it makes the model training more time-consuming. So, more resources needed to compute these number of features. The amount of increase in features after preprocessing is shown in Table 6. Also, we conduct a standard scale for all columns and transfer them into a standard normal distribution. Therefore, the different scales of columns could not affect our datasets.

According to Table 7, there is an increase in the number of columns (dimensions) after using the processing method. It will be exacerbated by adding even one or two new categorical columns to this dataset. The data can be multiplied by the number of dimensions, so in actual use, it is necessary to create a reliable dimensional controlling channel that controls the number of features since only a small number of features are effective in the final result, and many features are extra [77]. Therefore, this number should be reduced automatically using known methods. In the next section, this issue will be discussed more. Also, dimensionality reduction makes it easier to visualize more significant features, which is advantageous for a better understanding of the data.

6.2. Feature Reduction Methods Evaluation. In order to find the best feature reduction method, the different algorithms have been adopted on the Light GBM classifier that shows the performance of each of them in Table 8. The results show feature selection based on Xgboost feature importance has higher achievement on all metrics. However, in the Bank dataset, it is a bit worse than LDA and Autoencoder. However, overall with considering two other datasets, the Xgboost has superiority. It enhances all the crucial metrics for accurately and meaningfully predicting the churners. As a result, the Xgboost feature selection will be selected in the following model implementation. At this stage, it is not necessary to compare methods based on their EMPC because the main reason for feature reduction is to enhance the model accuracy and computational speed.

One of the benefits of dimensional reduction, as mentioned, is the increase in computational speed. Table 8 illustrates this well before and after applying the dimension reduction by the method of feature importance of the Xgboost. Now suppose that a company's actual dataset will grow much more extensive over time, and as shown, as the data set grows larger in time or features are going to be added or changed, the need to increase speed and efficient use of resources and maintain the best accuracy increases. For example, the orange dataset has the highest speed improvement. Therefore, with time and also increasing the number of customers and features, there is an urgent need for methods to reduce and optimize features.

Table 6: Features amount before and after preprocessing.

Datasets	Features before preprocessing	Features after preprocessing	Ratio of increase (%)
Bank	14	20	142
IBM	21	33	157
Orange	20	71	355

6.3. Hyperparameter Tuning Implementation. Boosting algorithms are robust classifiers with many hyperparameters that require tuning. The model has been built for Bayesian and genetic tuning using the Google collab platform. The Hyperopt library has utilized the Bayesian optimization in python. It conducts by two hundred iterations. Because of the structure of TPE finding the local minimums, the minus value of accuracy calculates at each iteration. The genetic optimization uses eight parents. The number of parents mating is four, and also the generations will continue four times. The important boosting hyperparameters, such as learning rate and depth, and others have been considered in the model. Table 9 shows the different ranges of each hyperparameter of light GBM. In this paper, a range of 0.35 to 0.65 is considered as the threshold range. Usually, in the researches, it is placed on 0.5, but this paper took a more extensive scale.

6.4. Evaluation of the Result of Baselines and Benchmark Models. The evaluation metrics of customer chum prediction by different algorithms, including accuracy, AUC, MCC, Kappa score, Brier score, F1 score, and EMPC indicators, on the four datasets are shown in Tables 10–12. The results of all metrics are discussed below.

(1) Accuracy: In this metric in the bank dataset, the best accuracy belongs to Bayesian-based light. In the IBM dataset, the highest accuracy belongs to the Bayesian-based light GBM, with very little difference from the genetic-based light GBM. In orange, same as other experiments, Bayesian-based light GBM outperforms others. In all datasets, all the optimizer-based algorithms plus light GBM worked well together.

(2) AUC: The area under the curve is a good indication of an algorithm's power to identify all classes of a classification problem. In the orange and bank datasets, the genetics and Bayesian-based light GBM algorithms performed best, respectively. In the IBM dataset, genetic-based light GBM outperformed the Bayesian-based light GBM by a narrow margin of 0.0015. Also, Xgboost performed slightly better in this metric. According to Tables 10 to 12, all evaluations are in the sound and excellent range.

Table 7: Evaluation results of feature reduction methods.

Method	IBM Telecom	Bank Customer	Orange
PCA ($n = 3$)	Accuracy: 0.7923 AUC: 0.7026 MCC: 0.4481 Kappa_score: 0.4397 Brier_score: 0.2076 F1_score: 0.5730	Accuracy: 0.788 AUC: 0.5276 MCC: 0.1154 Kappa_score: 0.07888 Brier_score: 0.212 F1_score: 0.1396	Accuracy: 0.8110 AUC: 0.5036 MCC: 0.0101 Kappa_score: 0.0092 Brier_score: 0.1889 F1_score: 0.0999
PCA ($n = 4$)	Accuracy: 0.7889 AUC: 0.6971 MCC: 0.4377 Kappa_score: 0.4290 Brier_score: 0.2110 F1_score: 0.5640	Accuracy: 0.786 AUC: 0.5263 MCC: 0.1075 Kappa_score: 0.0749 Brier_score: 0.214 F1_score: 0.1384	Accuracy: 0.8080 AUC: 0.5018 MCC: 0.0051 Kappa_score: 0.0047 Brier_score: 0.1919 F1_score: 0.0985
PCA ($n = 5$)	Accuracy: 0.7889 AUC: 0.6990 MCC: 0.4392 Kappa_score: 0.4313 Brier_score: 0.2110 F1_score: 0.5670	Accuracy: 0.75 AUC: 0.5607 MCC: 0.1413 Kappa_score: 0.1375 Brier_score: 0.25 F1_score: 0.2824	Accuracy: 0.8035 AUC: 0.5168 MCC: 0.0417 Kappa_score: 0.0401 Brier_score: 0.1964 F1_score: 0.1437
LDA ($n = 2$)	Accuracy: 0.7906 AUC: 0.7046 MCC: 0.4466 Kappa_score: 0.4401 Brier_score: 0.2093 F1_score: 0.5760	Accuracy: 0.8604 AUC: 0.7237 MCC: 0.5297 Kappa_score: 0.5129 Brier_score: 0.1396 F1_score: 0.5927	Accuracy: 0.9505 AUC: 0.8438 MCC: 0.7842 Kappa_score: 0.7725 Brier_score: 0.0494 F1_score: 0.7999
t-SNE ($n = 3$)	Accuracy: 0.6387 AUC: 0.4684 MCC: −0.0842 Kappa_score: −0.0754 Brier_score: 0.3612 F1_score: 0.1143	Accuracy: 0.6944 AUC: 0.4864 MCC: −0.0305 Kappa_score: −0.0300 Brier_score: 0.3056 F1_score: 0.1511	Accuracy: 0.7361 AUC: 0.4862 MCC: −0.0261 Kappa_score: −0.0261 Brier_score: 0.2638 F1_score: 0.1287
Autoencoder	Accuracy: 0.7906 AUC: 0.7046 MCC: 0.4466 Kappa_score: 0.4401 Brier_score: 0.2093 F1_score: 0.5760	Accuracy: 0.8604 AUC: 0.7237 MCC: 0.5297 Kappa_score: 0.5129 Brier_score: 0.1396 F1_score: 0.5927	Accuracy: 0.9505 AUC: 0.8438 MCC: 0.7842 Kappa_score: 0.7725 Brier_score: 0.0494 F1_score: 0.7999
Feature Importances (features = 10)	Accuracy: 0.7969 AUC: 0.7152 MCC: 0.4658 Kappa_score: 0.4601 Brier_score: 0.2030 F1_score: 0.5929	Accuracy: 0.8548 AUC: 0.7223 MCC: 0.5139 Kappa_score: 0.5015 Brier_score: 0.1452 F1_score: 0.5860	Accuracy: 0.9505 AUC: 0.8614 MCC: 0.7861 Kappa_score: 0.7811 Brier_score: 0.0494 F1_score: 0.8092

Table 8: Computational speed before and after feature reduction.

Dataset	Computation speed before Feature reduction	Computation speed after Feature reduction	Improvement (%)
Bank	5 m 12 s	4 m 36 s	126
IBM	3 m 51 s	3 m 2 s	127
Orange	5 m 6 s	2 m 21 s	217

Table 9: Range and numbers of hyperparameters For Light GBM.

Hyperparameter	Range
Number of estimators	(150,300)
Max depth	(1,10)
Gamma	(0.01,9)
Cosample by tree	(0.01,1)
Subsample	(0.01,1)
Learning rate	(log(0.01),log(1))
Threshold	(0.35,0.65)

(3) MCC: Like AUC, it is a good indicator of the correct diagnosis of classification classes. In general, all algorithms based on genetics and Bayesian light GBM have achieved good results. However, in the bank dataset, light GBM did a bit better.

(4) Brier Score: This metric indicates that the prediction outputs strength, which Bayesian-based Light GBM performed the best in all datasets.

Table 10: Evaluation results in bank dataset.

Method	Accuracy	AUC	EMPC	MCC	Brier	Kappa	F1
BO-LightGBM	0.8688	0.7261	5.4771	0.5562	0.1312	0.5311	0.6038
GO-LightGBM	0.8624	0.6943	4.9208	0.5269	0.1376	0.4813	0.5509
Decision Tree	0.7936	0.6666	4.5939	0.3474	0.2064	0.3466	0.4745
Random forest	0.8412	0.7173	5.3804	0.479	0.1588	0.4734	0.5689
SVM	0.8548	0.6845	4.7804	0.496	0.1452	0.455	0.5291
Logistic regression	0.8332	0.6559	4.4109	0.4106	0.1668	0.3805	0.4674
Xgboost	0.8604	0.70733	5.1438	0.5224	0.1396	0.4938	0.5696
LightGBM	0.8604	0.7237	5.4526	0.5297	0.1396	0.5129	0.5927
Adaboost	0.8524	0.7044	5.1096	0.4966	0.1476	0.4764	0.5591

Table 11: Evaluation result of IBM dataset.

Method	Accuracy	AUC	EMPC	MCC	Brier	Kappa	F1
BO-LightGBM	0.8048	0.7132	8.5254	0.4793	0.1951	0.4677	0.5911
GO-LightGBM	0.8026	0.7147	8.5525	0.4758	0.1973	0.4666	0.5932
Decision tree	0.7645	0.6877	8.3582	0.3922	0.2354	0.3906	0.549
Random forest	0.765	0.6975	8.4663	0.4038	0.2349	0.4033	0.4033
SVM	0.7838	0.6729	8.1984	0.4101	0.2161	0.3915	0.3915
Logistic regression	0.7963	0.711	8.5276	0.4616	0.2036	0.4545	0.4545
Xgboost	0.8003	0.7182	8.6066	0.474	0.1996	0.4677	0.4677
LightGBM	0.7906	0.7046	8.4687	0.4466	0.2093	0.4401	0.4401
Adaboost	0.8026	0.7147	8.5525	0.4758	0.1973	0.4666	0.4666

Table 12: Evaluation result of orange dataset.

Method	Accuracy	AUC	EMPC	MCC	Kappa	Brier	F1
BO-LightGBM	0.958	0.8657	5.8275	0.8189	0.8097	0.0419	0.8333
GO-LightGBM	0.949	0.8649	5.7961	0.7809	0.7776	0.0509	0.8068
Decision tree	0.9265	0.8518	5.54872	0.7006	0.7005	0.0734	0.7434
Random forest	0.955	0.8596	5.7271	0.8052	0.7962	0.0449	0.8214
SVM	0.928	0.7649	4.2158	0.6732	0.6426	0.0719	0.68
Logistic regression	0.8575	0.5746	1.3099	0.2371	0.2037	0.1424	0.2635
Xgboost	0.949	0.8386	5.3913	0.7771	0.7644	0.0509	0.7926
LightGBM	0.9505	0.8438	5.4752	0.7842	0.7725	0.0494	0.7999
Adaboost	0.886	0.6746	2.7491	0.4512	0.4281	0.1139	0.4864

(5) Kappa Score: Kappa score means that the model is more meaningful than a random model. All algorithms based on optimizers have performed better. Bayesian-based Light GBM has better performance.

(6) F1-score: At the F1 score, similar to the results of the previous metric, the optimizers perform better overall.

(7) EMPC: The most important criterion for businesses is the estimator of the profitability of the created model. All optimizer-based algorithms have maximized profitability in boosting algorithms. This improvement reflects the excellent performance of optimizers with profitability. Bayesian-based light GBM achieved more in this metric except IBM that Xgboost achieved a slightly better result.

For better understanding of the best model, BO-light GBM Figures 6 to 8 show the ROC curve of each dataset. The AUC of the ROC curve is mentioned in the results tables.

The estimation of time complexities of the machine learning model is very difficult to consider. Especially, the proposed model is more complex by combining ML and optimization models. However, the execution time and parameters are mentioned to understand the model complexities better. All models incorporated 10 features due to

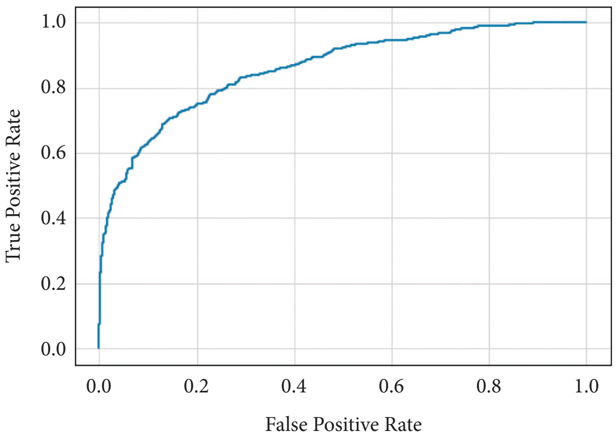

Figure 6: ROC curve IBM.

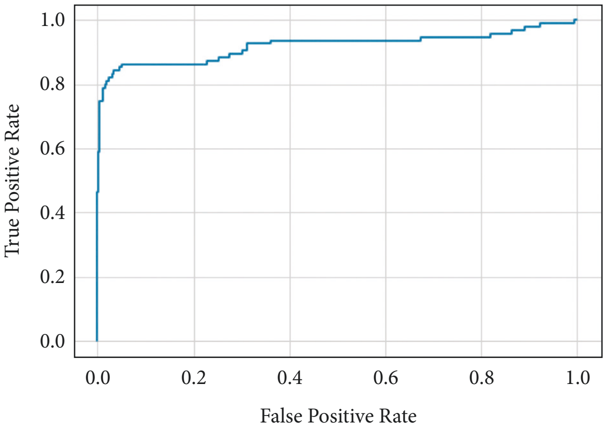

Figure 7: ROC curve orange.

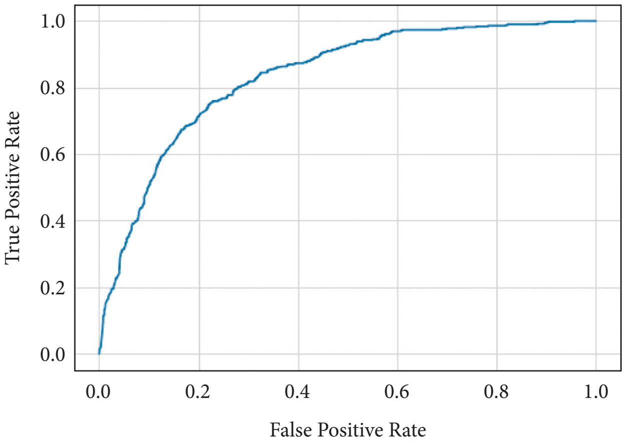

Figure 8: ROC curve bank.

the feature reduction methods that were implemented before this step (see Table 13).

Table 13: Samples and execution time.

Samples	Execution Time
10000	3 minutes and 21 seconds
7032	2 minutes and 51 seconds
2666	2 minutes and 23 seconds

6.5. Model Explainability. Light GBM constitutes regression trees. In Figure 9, the roles and branches of leaves of model implementation on bank datasets have been illustrated. It is split first by age and then uses the number of products count to split the churners. In the past, models were interpreted based on how features affected the machine learning models. Figure 9 shows how some features influence and play the most crucial role in separating tree branches, which can be a good indicator of the importance of features. For example, in the bank dataset, the number of products and user activity are essential features that can help decision-making. However, the SHAP model helps us take a more comprehensive look at the features and how they affect the model output. Model-independent SHAP is measured using embedded algorithms to analyze features' effect on the target variable (be churner or not). The horizontal line in Figure 10 shows the effect on the output, such as the probability of being a churner (target). Red and blue indicate the value of the desired feature, increase and decrease, respectively.

For example, for the IBM dataset (Figure 10), it can be interpreted that if one person has a month-to-month contract value (value 1) is more likely to churn than other nonmonth to month contracts (value zero). The next item is the number of tenure years. The longer the contract, the less likely it is to churn, and vice versa, which logically dictates the same thing in mind. Therefore, this method enhances the interpretation of the boosting models well and makes more effective and accurate management decisions. Figure 11 also shows the SHAP summary of the orange dataset. Hence, there is much possibility for understanding the model, especially with boosting tree models light GBM. It helps managerial decisions be more productive and focused on customers by understanding their customer churn's underlying reasons.

6.6. CLV Analysis. After determining the fundamental causes of customer churn, marketing efforts should target those consumers with various offers such as discounts. Customer lifetime value, or CLV, is a valuable tool for estimating consumers' true values. CLV calculates the actual value of a company's customers [78]. According to Figure 12, the majority of customers have a low CLV. Targeting those clients raises marketing expenditures while providing little benefit to the firm. Targeting identified churners based on their real value to the organization improves performance and lowers the cost of marketing efforts.

In Figure 9, the CLV of churners is calculated by each customer's monthly charges multiplied by the company's gross margin. The company's gross margin is assumed to be 0.2. However, more data and company-specific metrics are required to compute true CLV.

7. Discussion

According to the result analysis, it is clear to distinguish that the Bayesian-based light GBM algorithm's prediction performance is best since it outperforms well in almost all metrics in all datasets, especially on imbalanced related metrics evaluation. The experimental results are concluded in four parts:

(1) All ensemble methods perform better in all metrics in all datasets. Bayesian-based light GBM outperforms mostly in all datasets.

(2) According to MCC and Brier scores, the BO and GO hyperparameter tuning makes the predictions more meaningful and better for imbalanced datasets.

(3) Other methods such as support vector machine, decision tree, random forest, and AdaBoost did not perform well. They were lower in all metrics than boosting algorithms, significantly light GBM optimized with optimizers.

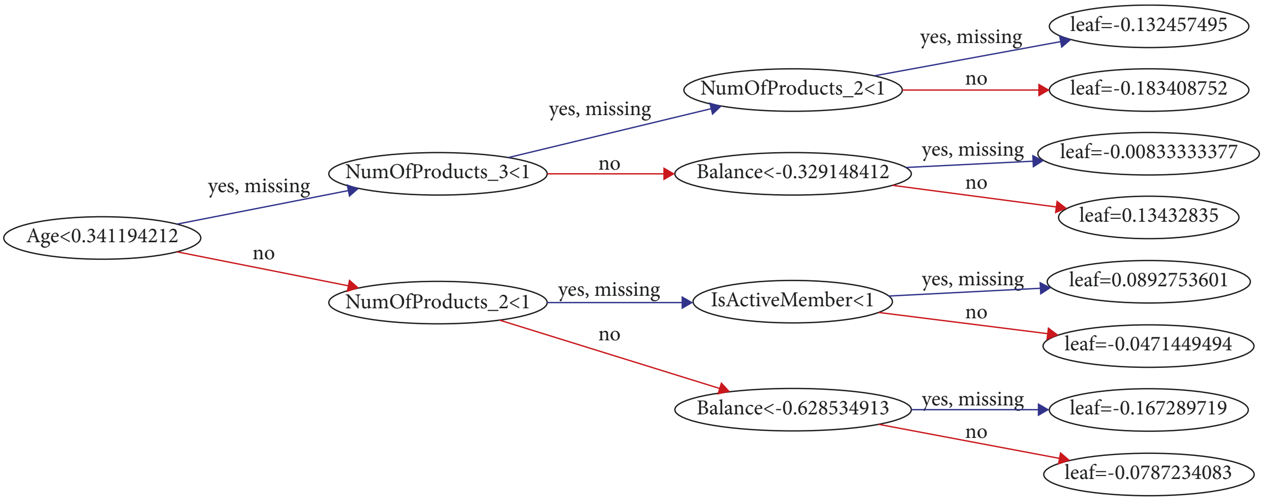

Figure 9: Tree chart of bank dataset.

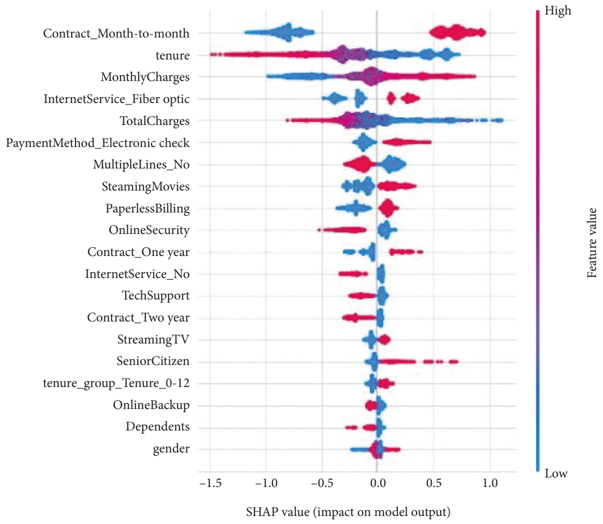

Figure 10: SHAP summary of IBM dataset.

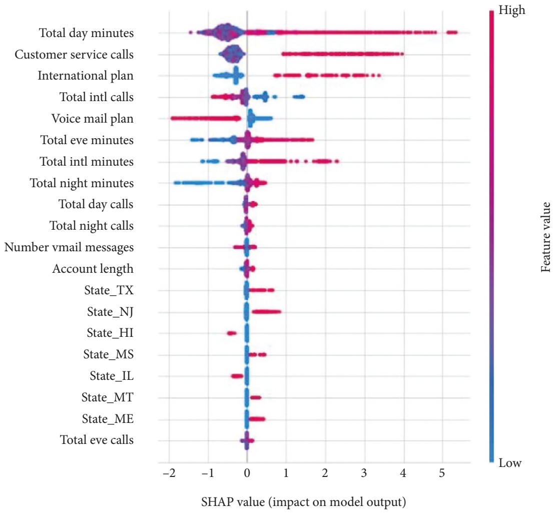

Figure 11: SHAP summary of orange dataset.

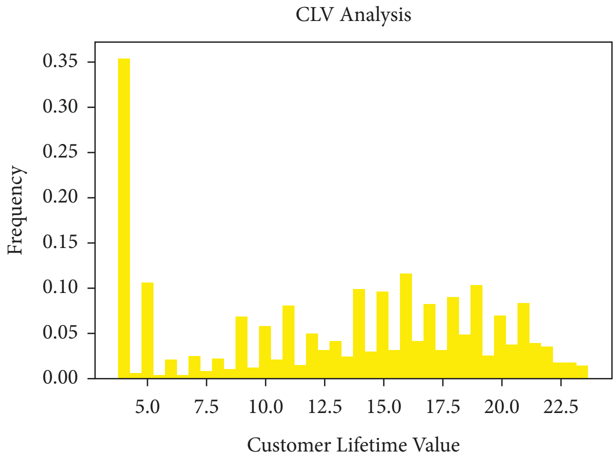

Figure 12: Ranking the IBM churners by CLV.

(4) Light GBM shows better results in imbalanced metrics and accuracy. Xgboost has slightly better AUC and EPMC in IBM datasets. However, the light GBM has better accuracy, brier score, and MCC in all datasets.

8. Conclusion

Given the importance of customer retention management in various businesses, including telecommunications, banking, insurance, and even online gaming, numerous studies have been conducted on predicting customer churn and the factors affecting it. This paper has implemented churn prediction systems, such as feature reduction and model selection, to compare various algorithms. However, there is a lack of concentration on ensemble methods implementation, especially the boosting methods like light GBM on CCP models. This paper focuses on enhancing the churn prediction models by the light GBM method and also hyperparameter tuning. The evaluation shows that the light GBM method obtains the best performances, especially when tuned by Bayesian optimization.

The results show that feature reduction effectively increases the speed and the important feature selection of the Xgboost algorithm has the best performance among feature reduction algorithms. Also, evaluating the algorithms on the datasets shows that the Bayesian-based light GBM is more successful and improves imbalanced evaluation metrics. Also, considering the computational speed, better prediction, and more profit in the business, the Bayesian-based light GBM method should be used over other ML and ensemble models. Furthermore, interpreting the results with SHAP helps the marketing team understand the underlying reasons for churn and helps them identify the customer's pain points better; also, targeting customers based on CLV and real value lowers marketing campaign costs and improves campaign performance. As a result, it leads to better-informed decision-making in client retention initiatives and tactics.

8.1. Limitations and Future Suggestions. For future studies, it is suggested studying the combination of more feature reduction methods like LDA and Autoencoder methods with ensemble learning. Other hyperparameter optimization algorithms can be implemented on light GBM in CCP models. One of the limitations of this study is the range of hyperparameters and the structure of hyperparameter tuning. It could be examined by other types, ranges, and structures to find the best-fitted model for the CCP problem. Also, the other limitation is the data structure. All datasets are comprised of structured inputs. It is highly recommended to use other unstructured data like text and images for future research, and therefore, the feature reduction methods will be more practical in this case. Also, it is worth measuring other marketing metrics and different methods in order to segment churners. It will be a broad research topic on how to handle churners after identifying them [78].

Data Availability

Data are available and can be provided over the emails querying directly to the author at the corresponding author (babakamiri@iust.ac.ir).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Amin, S. Shehzad, C. Khan, I. Ali, and S. Anwar, "Churn prediction in telecommunication industry using rough set approach," *New Trends in Computational Collective Intelligence*, vol. 572, pp. 83–95, 2015b.

[2] C. Wang, D. Han, W. Fan, and Q. Liu, "Customer churn prediction with feature embedded convolutional neural network: an empirical study in the internet funds industry," *International Journal of Computational Intelligence and Applications*, vol. 18, no. 1, Article ID 1950003, 2019.

[3] A. Amin, F. Rahim, I. Ali, C. Khan, and S. Anwar, "A comparison of two oversampling techniques (smote vs mtdf) for handling class imbalance problem: a case study of customer churn prediction," in *New Contributions in Information Systems and Technologies*, pp. 215–225, Springer, Berlin, Germany, 2015.

[4] A. Amin, S. Anwar, A. Adnan et al., “Customer churn prediction in the telecommunication sector using a rough set approach,” *Neurocomputing*, vol. 237, pp. 242–254, 2017.

[5] Y. Xia, C. Liu, Y. Y. Li, and N. Liu, “A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring,” *Expert Systems with Applications*, vol. 78, pp. 225–241, 2017.

[6] S. Kim, D. Choi, E. Lee, and W. Rhee, “Churn prediction of mobile and online casual games using play log data,” *PLoS ONE*, vol. 12, pp. 1–19, 2018.

[7] A. Amin, B. Shah, A. Abbas, S. Anwar, O. Alfandi, and F. Moreira, “Features Weight Estimation Using a Genetic Algorithm for Customer Churn Prediction in the Telecom Sector,” in *Proceedings of the World Conference on Information Systems and Technologies*, Galicia, Spain, April 2019.

[8] D. Han, N. Zhao, and P. Shi, “Gear fault feature extraction and diagnosis method under different load excitation based on EMD, PSO-SVM and fractal box dimension,” *Journal of Mechanical Science and Technology*, vol. 33, no. 2, pp. 487–494, 2019.

[9] W. C. Lin, C. F. Tsai, and S. W. Ke, “Dimensionality and data reduction in telecom churn prediction,” *Kybernetes*, vol. 43, no. 5, pp. 737–749, 2014.

[10] J. Xiao, Y. Wang, and S. Wang, “A dynamic transfer ensemble model for customer churn prediction,” in *Proceedings of the 2013 6th International Conference on Business Intelligence and Financial Engineering BIFE*, vol. 43, no. 1, pp. 115–119, Hangzhou, China, November 2013.

[11] A. Idris and A. Khan, “Churn prediction system for telecom using filter-wrapper and ensemble classification,” *The Computer Journal*, vol. 60, no. 3, Article ID bxv123, 2016.

[12] M. Saghir, Z. Bibi, S. Bashir, and F. H. Khan, “Churn prediction using neural network based individual and ensemble models,” in *Proceedings of the 2019 16th International Bhurban Conference on Applied Sciences and Technology*, pp. 634–639, IBCAST, Islamabad, Pakistan, January 2019.

[13] J. Mou, P. Duan, L. Gao, X. Liu, and J. Li, “An effective hybrid collaborative algorithm for energy-efficient distributed permutation flow-shop inverse scheduling,” *Future Generation Computer Systems*, vol. 128, no. 2022, pp. 521–537, 2022.

[14] X. Zenggang, L. Xiang, Z. Xueming et al., “A service pricing-based two-stage incentive algorithm for socially aware networks,” *Journal of Signal Processing Systems*, pp. 1–16, 2022.

[15] S. Höppner, E. Stripling, B. Baesens, S. vanden Broucke, and T. Verdonck, “Profit driven decision trees for churn prediction,” *European Journal of Operational Research*, vol. 284, no. 3, pp. 920–933, 2020.

[16] S. Maldonado, J. López, and C. Vairetti, “Profit-based churn prediction based on Minimax probability machines,” *European Journal of Operational Research*, vol. 284, no. 1, pp. 273–284, 2020.

[17] L. Calzada-Infante, M. Óskarsdóttir, and B. Baesens, “Evaluation of customer behavior with temporal centrality metrics for churn prediction of prepaid contracts,” *Expert Systems with Applications*, vol. 160, Article ID 113553, 2020.

[18] R. Taghizadech and M. Ahmadi, “Statistical and econometrical analysis of knowledge-based economy indicators affecting economic growth in Iran: the new evidence of principal component analysis—tukey and ARDL bound test,” *Preprint: January*, vol. 10, 2019.

[19] B. Zhu, Q. Zhong, Y. Chen et al., “A novel reconstruction method for temperature distribution measurement based on ultrasonic tomography,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, p. 1, 2022.

[20] W. Yang, X. Chen, Z. Xiong, Z. Xu, G. Liu, and X. Zhang, “A privacy-preserving aggregation scheme based on negative survey for vehicle fuel consumption data,” *Information Sciences*, vol. 570, no. 2021, pp. 526–544, 2021.

[21] A. Khodadadi, S. Hosseini, E. Pajouheshgar, F. Mansouri, and H. R. Rabiee, “ChOracle: a unified statistical framework for churn prediction,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 1, p. 1, 2020.

[22] A. Amin, F. Al-Obeidat, B. Shah et al., “Just-in-time customer churn prediction in the telecommunication sector,” *The Journal of Supercomputing*, vol. 76, no. 6, pp. 3924–3948, 2020.

[23] M. Mahmoudi, “COVID Lessons: was there any way to reduce the negative effect of COVID-19 on the United States economy?,” 2022, https://arxiv.org/abs/2201.00274.

[24] M. Li, S. Chen, Y. Shen, G. Liu, I. W. Tsang, and Y. Zhang, “Online multi-agent forecasting with interpretable collaborative graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[25] M. Özmen, E. K. Aydoğan, Y. Delice, and M. D. Toksarı, “Churn prediction in Turkey’s telecommunications sector: a proposed multiobjective–cost-sensitive ant colony optimization,” *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 1, Article ID e1338, 2020.

[26] A. Li, D. Spano, J. Krivochiza et al., “A tutorial on interference exploitation via symbol-level precoding: overview, state-of-the-art and future directions,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 796–839, 2020.

[27] S. M. Kostić, M. I. Simić, and M. V. Kostić, “Social network analysis and churn prediction in telecommunications using graph theory,” *Entropy*, vol. 22, no. 7, p. 753, 2020.

[28] A. Li, C. Masouros, A. L. Swindlehurst, and W. Yu, “1-bit massive MIMO transmission: embracing interference with symbol-level precoding,” *IEEE Communications Magazine*, vol. 59, no. 5, pp. 121–127, 2021.

[29] M. Ahmadi, “A computational approach to uncovering economic growth factors,” *Computational Economics*, vol. 58, no. 4, pp. 1051–1076, 2021.

[30] L. Cai, L. Xiong, J. Cao, H. Zhang, and F. E. Alsaadi, “State Quantized Sampled-Data Control Design for Complex-Valued Memristive Neural Networks,” *Journal of the Franklin Institute*, vol. 359, 2022.

[31] M. Ahmed, H. Afzal, I. Siddiqi, M. F. Amjad, and K. Khurshid, “Exploring nested ensemble learners using overproduction and choose approach for churn prediction in telecom industry,” *Neural Computing & Applications*, vol. 32, no. 8, pp. 3237–3251, 2020.

[32] X. Liang, T. Lu, and G. Yishake, “How to promote residents’ use of green space: an empirically grounded agent-based modeling approach,” *Urban Forestry and Urban Greening*, vol. 67, Article ID 127435, 2022.

[33] M. Ahmadi and M. Qaisari Hasan Abadi, “A review of using object-orientation properties of C++ for designing expert system in strategic planning,” *Computer Science Review*, vol. 37, Article ID 100282, 2020.

[34] H. Jain, A. Khunteta, and S. Srivastava, “Churn prediction in telecommunication using logistic regression and Logit Boost,” *Procedia Computer Science*, vol. 167, pp. 101–112, 2020.

[35] F. Castanedo, G. Valverde, J. Zaratiegui, and A. Vazquez, “Using Deep Learning to Predict Customer Churn in a Mobile Telecommunication Network Federico,” *Computer Science*, pp. 1–8, 2014.

[36] P. H. Nekah, T. Nabizadeh, and Z. Ali, “Determining the optimal point of purchase intention: using genetic algorithm

evidence from Iran khodro auto industry," *The IIOAB Journal*, vol. 7, pp. 455–462, 2016.

[37] P. Spanoudes and T. Nguyen, "Deep Learning in Customer Churn Prediction: Unsupervised Feature Learning on Abstract Company Independent Feature Vectors," pp. 1–22, 2017, http://arxiv.org/abs/1703.03869.

[38] A. S. Halibas, A. C. Matthew, I. G. Pillai, J. H. Reazol, E. G. Delvo, and L. B. Reazol, "Determining the intervening effects of exploratory data analysis and feature engineering in telecoms customer churn modelling," in *Proceedings of the 2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)*, Muscat, Oman, January 2019.

[39] S. Wu, W.-C. Yau, T.-S. Ong, and S.-C. Chong, "Integrated Churn Prediction and Customer Segmentation Framework for Telco Business," *IEEE Access*, vol. 9, 2021.

[40] I. V. Pustokhina, D. A. Pustokhin, A. Rh et al., "Dynamic customer churn prediction strategy for business intelligence using text analytics with evolutionary optimization algorithms," *Information Processing & Management*, vol. 58, no. 6, Article ID 102706, 2021.

[41] P. Lalwani, M. K. Mishra, J. S. Chadha, and P. Sethi, "Customer churn prediction system: a machine learning approach," *Computing*, vol. 104, no. 2, pp. 271–294, 2021.

[42] A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, and S. Anwar, "Customer churn prediction in telecommunication industry using data certainty," *Journal of Business Research*, vol. 94, pp. 290–301, 2019a.

[43] A. Keramati, R. Jafari-Marandi, M. Aliannejadi, I. Ahmadian, M. Mozaffari, and U. Abbasi, "Improved churn prediction in telecommunication industry using data mining techniques," *Applied Soft Computing*, vol. 24, pp. 994–1012, 2014.

[44] J. Burez and D. Van den Poel, "Handling class imbalance in customer churn prediction," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4626–4636, 2009.

[45] A. Amin, S. Anwar, A. Adnan et al., "Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study," *IEEE Access*, vol. 4, pp. 7940–7957, 2016.

[46] A. Amin, B. Shah, A. M. Khattak et al., "Cross-company customer churn prediction in telecommunication: a comparison of data transformation methods," *International Journal of Information Management*, vol. 46, pp. 304–319, 2019c.

[47] A. Amin, C. Khan, I. Ali, and S. Anwar, "Customer churn prediction in telecommunication industry: with and without counter-example, Nature-Inspired Computation and Machine Learning," in *Proceedings of the Mexican International Conference on Artificial Intelligence*, pp. 206–218, Tuxtla Gutierrez, Mexico, November 2014.

[48] Y. Chen, Y. R. Gel, V. Lyubchich, and T. Winship, "Deep ensemble classifiers and peer effects analysis for churn forecasting in retail banking," *Advances in Knowledge Discovery and Data Mining*, Springer, Berlin, Germany, pp. 373–385, 2018.

[49] R. Zhang, W. Li, T. Mo, and W. Tan, *Deep and Shallow Model for Insurance Churn Prediction Service*, in *Proceedings of the 2017 IEEE International Conference on Services Computing (SCC)*, Honolulu, HI, USA, June 2017.

[50] A. Dingli, V. Marmara, and N. S. Fournier, "Comparison of deep learning algorithms to predict customer churn within a local retail industry," *International Journal of Machine Learning and Computing*, vol. 7, no. 5, pp. 128–132, 2017.

[51] A. Simion-constantinescu and A. Ionu, "Deep Neural Pipeline for Churn Prediction," in *Proceedings of the 2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Cluj-Napoca, Romania, September 2018.

[52] J. Zhou, J. f. Yan, L. Yang, M. Wang, and P. Xia, "Customer churn prediction model based on LSTM and CNN in music streaming," *DEStech Transactions on Engineering and Technology Research*, pp. 254–261, 2019.

[53] S. Dhote, C. Vichoray, R. Pais, S. Baskar, and P. Mohamed Shakeel, "Hybrid geometric sampling and AdaBoost based deep learning approach for data imbalance in E-commerce," *Electronic Commerce Research*, vol. 20, no. 2, pp. 259–274, 2019.

[54] J. Hu, Y. Zhuang, J. Yang et al., "pRNN: a recurrent neural network based approach for customer churn prediction in telecommunication sector," in *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, December 2018.

[55] A. Idris and A. Iftikhar, "Intelligent Churn Prediction for Telecom Using GP-AdaBoost Learning and PSO Undersampling," *Cluster Computing*, vol. 22, 2017.

[56] A. Wangperawong, C. Brun, O. Laudy, and R. Pavasuthipaisit, "Churn Analysis Using Deep Convolutional Neural Networks and Autoencoders," pp. 1–6, 2016, http://arxiv.org/abs/1604.05377.

[57] J. Zhong and W. Li, "Predicting Customer Churn in the Telecommunication Industry by Analyzing Phone Call Transcripts with Convolutional Neural Networks," in *Proceedings of the CIAI 2019: 2019 The 3rd International Conference on Innovation in Artificial Intelligence*, Suzhou China, March 2019.

[58] M. Gridach, "Churn Identification in Microblogs using Convolutional Neural Networks with Structured Logical Knowledge," in *Proceedings of the Empirical methods in natural language processing*, pp. 21–30, November 2017.

[59] T. Zhang and B. Yang, "Big data dimension reduction using PCA," in *Proceedings of the 2016 IEEE International Conference on Smart Cloud*, pp. 152–157, New York, NY, USA, November 2016.

[60] M. Wattenberg, F. Viégas, and I. Johnson, "How to use t-SNE effectively," *Distill*, vol. 1, no. 10, p. e2, 2016.

[61] A. Myronenko, "3D MRI brain tumor segmentation using autoencoder regularization," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11384 LNCS*, pp. 311–320, Springer, Berlin, Germany, 2019.

[62] J. Wen, X. Fang, J. Cui et al., "Robust sparse linear discriminant analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 390–403, 2019.

[63] X. Ren, H. Guo, S. Li, S. Wang, and J. Li, "A novel image classification method with CNN-XGBoost model," in *Proceedings of the International Workshop on Digital Watermarking*, pp. 378–390, Magdeburg, Germany, August 2017.

[64] M. R. Machado, S. Karray, and I. T. de Sousa, "LightGBM: An Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry," in *Proceedings of the 2019 14th International Conference on Computer Science & Education (ICCSE)*, pp. 1111–1116, Toronto, ON, Canada, August 2019.

[65] P. K. Dalvi, S. K. Khandge, A. Deomore, A. Bankar, and V. A. Kanade, "Analysis of customer churn prediction in telecom industry using decision trees and logistic regression," in *Proceedings of the 2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, pp. 1–4, Indore, India, March 2016.

[66] M. A. H. Farquad, V. Ravi, and S. B. Raju, “Churn prediction using comprehensible support vector machine: an analytical CRM application,” *Applied Soft Computing*, vol. 19, pp. 31–40, 2014.

[67] G. Gerber, Y. L. Faou, O. Lopez, and M. Trupin, “The impact of churn on client value in health insurance, evaluation using a random forest under various censoring mechanisms,” *Journal of the American Statistical Association*, vol. 536, pp. 2053–2064, 2020.

[68] X. Wu and S. Meng, “E-commerce Customer Churn Prediction Based on Improved SMOTE and AdaBoost,” in *Proceedings of the 2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, pp. 1–5, Kunming, China, June 2016.

[69] S. Mirjalili, “Genetic algorithm,” in *Studies in Computational Intelligence*, vol. 780, pp. 43–55, Springer, 2019.

[70] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee, “Improving object detection with deep convolutional networks via bayesian optimization and structured prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 249–258, Boston, MA, USA, June 2015.

[71] A. B. Parsa, A. Movahedi, H. Taghipour, S. Derrible, and A. K. Mohammadian, “Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis,” *Accident Analysis & Prevention*, vol. 136, Article ID 105405, 2020.

[72] J. Xiao, Y. Xiao, A. Huang, D. Liu, and S. Wang, “Feature-selection-based dynamic transfer ensemble model for customer churn prediction,” *Knowledge and Information Systems*, vol. 43, no. 1, pp. 29–51, 2015.

[73] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, “Evaluation measures for models assessment over imbalanced data sets,” *Journal of Information Engineering and Applications*, vol. 3, no. 10, pp. 27–38, 2013.

[74] T. Verbraken, W. Verbeke, and B. Baesens, “A novel profit maximizing metric for measuring classification performance of customer churn prediction models,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 5, pp. 961–973, 2013.

[75] Z. Liu, Z. Lai, W. Ou, K. Zhang, and R. Zheng, “Structured optimal graph based sparse feature extraction for semi-supervised learning,” *Signal Processing*, vol. 170, Article ID 107456, 2020.

[76] W. D. Dahana, Y. Miwa, and M. Morisada, “Linking lifestyle to customer lifetime value: an exploratory study in an online fashion retail market,” *Journal of Business Research*, vol. 99, pp. 319–331, 2019.

[77] M. Xiang, C. Chunguang, D. Yan, and H. Hongbo, “A Feature Reduction Method by Grey Theory and Rough Set,” in *Proceedings of the 2010 Second WRI Global Congress on Intelligent Systems*, Wuhan, China, December 2010.

[78] M. S. Kahreh, M. Tive, A. Babania, and M. Hesan, “Analyzing the applications of customer lifetime value (CLV) based on benefit segmentation for the banking sector,” *Procedia-Social and Behavioral Sciences*, vol. 109, no. 8, pp. 590–594, 2014.