scientific reports

Check for updates

OPEN

An efficient churn prediction model using gradient boosting machine and metaheuristic optimization

Ibrahim AlShourbaji[1,2], Na Helian[1], Yi Sun[1], Abdelazim G. Hussien[3,4]✉, Laith Abualigah[5,6,7,8,9,10,11] & Bushra Elnaim[12]

Customer churn remains a critical challenge in telecommunications, necessitating effective churn prediction (CP) methodologies. This paper introduces the Enhanced Gradient Boosting Model (EGBM), which uses a Support Vector Machine with a Radial Basis Function kernel (SVM_{RBF}) as a base learner and exponential loss function to enhance the learning process of the GBM. The novel base learner significantly improves the initial classification performance of the traditional GBM and achieves enhanced performance in CP-EGBM after multiple boosting stages by utilizing state-of-the-art decision tree learners. Further, a modified version of Particle Swarm Optimization (PSO) using the consumption operator of the Artificial Ecosystem Optimization (AEO) method to prevent premature convergence of the PSO in the local optima is developed to tune the hyper-parameters of the CP-EGBM effectively. Seven open-source CP datasets are used to evaluate the performance of the developed CP-EGBM model using several quantitative evaluation metrics. The results showed that the CP-EGBM is significantly better than GBM and SVM models. Results are statistically validated using the Friedman ranking test. The proposed CP-EGBM is also compared with recently reported models in the literature. Comparative analysis with state-of-the-art models showcases CP-EGBM's promising improvements, making it a robust and effective solution for churn prediction in the telecommunications industry.

Abbreviations

CP	Churn prediction
EGBM	Enhanced gradient boosting machine
ACO	Ant colony optimization
ACO-RSA	Ant colony optimization-reptile search algorithm
AEO	Artificial ecosystem optimization
AUC-ROC	Area under the receiver operating characteristic curve
CNN	Convolutional neural network
CP-EGBM	Enhanced gradient boosting machine for churn prediction
CRM	Customer relationship management
CV	Cross validation
DT	Decision tree
FN	Flase negative
FP	False positive

[1]Department of Computer Science, University of Hertfordshire, Hatfield, UK. [2]Department of Computer and Network Engineering, Jazan University, 82822-6649 Jazan, Saudi Arabia. [3]Department of Computer and Information Science, Linköping University, Linköping, Sweden. [4]Faculty of Science, Fayoum University, Faiyum, Egypt. [5]Computer Science Department, Prince Hussein Bin Abdullah Faculty for Information Technology, Al Al-Bayt University, Mafraq 25113, Jordan. [6]Department of Electrical and Computer Engineering, Lebanese American University, 13-5053 Byblos, Lebanon. [7]Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan. [8]MEU Research Unit, Middle East University, Amman 11831, Jordan. [9]Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan. [10]School of Computer Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang, Malaysia. [11]School of Engineering and Technology, Sunway University Malaysia, 27500 Petaling Jaya, Malaysia. [12]Department of Computer Science, College of Science and Humanities in Al-Sulail, Prince Sattam Bin Abdulaziz University, 11671 Riyadh, Saudi Arabia. ✉email: abdelazim.hussien@liu.se; aga08@fayoum.edu.eg

GBM	Gradient boosting machine
GBT	Gradient boosted tree
GWO	Gray wolf optimizer
HEOMGA	Heterogeneous euclidean-overlap metric genetic algorithm
KNN	K-nearest neighbors
LR	Logistic regression
MH	Meta-heuristic
ML	Machine learning
mPSO	Modified particle swarm optimization
MVO	Multi-verse optimizer
NB	Naive Bayes
PSO	Particle swarm optimization
RBF	Radial basis functions
RF	Random forest
SMOTE	Synthetic minority oversampling technique
SVM	Support vector machine
SVM_{RBF}	Support vector machine with RBF kernel
TN	True negative
TP	True positive
WOA	Whale optimization algorithm
XGBoost	Extreme gradient boosting

Customers are the most valuable resource as they present the main reason for any industry's success. On the other hand, a churner is a customer who abandons his current company to join another competing company's service in the market. Customer churn is a common problem in the telecom business, and companies in this sector try to minimize churn rates. Studies show that reducing the customer churn rate saves money, as acquiring a new customer costs five times more than satisfying an existing one[1]. Therefore, reducing the churn rate has become particularly important for preserving revenues in this sector. Because of the significant financial implications of correctly predicting customer churn, CP models have become vital in CRM to identify customers most likely to terminate their relationships. As a result, there has been much focus on developing new methods to improve the accuracy of the CP using ML.

Nowadays, ML techniques are used to predict future patterns and behaviors of customers[2], so marketing strategies can be improved according to the produced results from these models. ML approaches can play a critical part in the success of different applications, such as oil price prediction[3], sentiment analysis[4], energy consumption[5], medical diagnosis[6], and CP[7]. These applications use one type of ML family of algorithms, called ensemble methods, which are inspirited by the human cognitive system. These methods have the powerful capability to deal with high-dimensional data and generate several diverse solutions for a given task[8].

Ensemble methods build many base models and then merge them into one to achieve better prediction results than using a single base model. Bagging and boosting are the most popular ensemble methods[9]. The bagging method, also known as "bootstrap aggregation," is based on averaging the base models, while the boosting methods are built upon a constructive iterative mechanism. In boosting algorithms, several weak learners are combined stage-wise to obtain a strong learner with improved prediction accuracy[10]. The family of boosting methods depends on different constructive strategies of ensemble formation. A gradient-descent-based formulation of boosting methods, called Gradient Boosting Machine (GBM), is derived by[11]. The GBM can be considered an optimization model aiming to train a series of weak-learner models, which sequentially minimizes a pre-defined loss function.

According to[12], several essential choices of differentiable weak-learner models and loss functions can be customized to a given task in the GBM model, making this model highly flexible to be applied in several ML applications based on the task requirements[13–15]. This paper aims to develop a new model by improving GBM's structure to effectively predict customer churn in the telecom sector. The main contributions of this paper can be summarized as follows:

- CP-EGBM is a new model with high predictive performance that may be used to develop effective strategies and contains customer churn risks in the telecom sector. It can enhance the learning ability of the GBM model structure by using SVM as a base learner and exponential loss as a loss function.
- Boosting the capability of the PSO in the exploration phase using the consumption operator of the AEO method could effectively find the most suitable values of the CP-EGBM's hyper-parameters.
- The performance of the proposed CP-EGBM is assessed using seven datasets in several evaluation metrics.
- The CP-EGBM model outperformed either GBM or SVM alone, and it is superior to several earlier reported models in the literature, making it more suitable for CP.

The rest of this paper is arranged as follows. "Literature review" provides a literature review on CP. The proposed CP-EGBM model is presented in "Proposed CP-EGBM", and "Experimental results" discusses the experimental results. Lastly, the conclusions of this paper and possible future works are provided in "Conclusion and future works".

Literature review

Many works applied ensemble ML models to predict customer churn[7,16,17]. Wang et al.[18] investigated the capability of the GBM model for CP. They used a large customer dataset obtained from the Bing-Ads platform company to identify whether the customers would leave or stay based on the analysis of their historical data records. The results showed that GBM was an effective and efficient model for predicting churner customers in the near future.

Several comparative analyses are conducted for CP using ML models. Ahmad et al.[19] compared four ML models, including Decision Trees (DTs), Random Forest (RF), GBM, and Extreme Gradient Boosting (XGBoost), for customer churn prediction. The results showed that the XGboost method outperformed other models when they evaluated the models using big data provided by a telecom company in Syria. Jain et al.[20] used four models for CP in the banking, telecom, and IT sectors, where they used Logistic Regression (LR), RF, SVM, and XGBoost. The results showed that XGBoost performed better than others in the telecom sector. In another work, Dhini et al.[21] compared RF and XGBoost to find the best model for CP. They used a private dataset collected from different companies in Indonesia to evaluate the models. The results showed that the predictive performance of the XGBoost was better than that of the RF model. In Sabbeh[22], the author compared a set of ML models using a publicly available dataset for CP. The results showed that RF attained the best results compared to other models used in their work.

Sandhya et al.[23] applied LR, K-Nearest Neighbors (KNN), SVM, and RF models to a publicly available dataset for CP. The authors first preprocessed the dataset and overcame the class imbalance problem using Synthetic Minority Oversampling Technique (SMOTE). The obtained results showed that RF performed better than the other models. Kimura[24] used six ML models: LR, RF, SVM, CatBoost, XGBoost and LightGBM. For data preprocessing, the authors used SMOTE Tomek Link and SMOTE-ENN sampling methods to balance class distribution in a publicly available dataset for CP. The results showed that CatBoost with SMOTE is the best model. Zhu & Liu[25] conducted a comparative study between ten ML models for churn prediction using a publicly available dataset; the results indicated that XGBoost obtained the best accuracy compared to the other models.

Kanwal et al.[26] employed a hybrid CP model using PSO to select the most informative features in a publicly available dataset for CP. Then, the selected features are used as inputs to DTs, KNN, Gradient Boosted Tree (GBT), and NB models. The findings indicate that the PSO with the GBT model obtained successful accuracy outcomes compared to the other models. Bilal et al.[27] introduced a CP model based on hybrid clustering and classification methods to predict customer churn from two publicly available datasets. The results showed that this model is more robust than the other existing models in the literature.

The stacking model technique (i.e., a mechanism that aims to leverage the benefits of a set of base models while ignoring their disadvantages) is also used for CP. Karuppaiah & Gopalan[28] presented a stacked Customer Lifetime Value-based heuristic incorporated ensemble model to predict customer churn. The authors used a publicly available dataset to evaluate the proposed model, and the obtained accuracy results showed good performance compared to the other existing models in the literature. Rabbah et al.[29] proposed a new CP model using deep learning and stacked models. They used a publicly available dataset to validate their model; the dataset is first preprocessed and balanced by the SMOTE method and then used a pre-trained Convolutional Neural Network (CNN) to select the essential features from the dataset. They employed the stacking model technique (i.e., a mechanism that aims to leverage the benefits of a set of base models while ignoring their disadvantages) to predict customer churn. The results demonstrated high efficacy of the developed model than the DTs, LR, RF, XGBoost, and Naive Bayes (NB) models.

Karamollaoglu et al.[30] used to separate datasets for CP in the telecommunication industry. Eight ML models are explored, including LR, KNN, DT, RF, SVM, AdaBoost, NB, and multi-layer perceptron. Although all models reported good performance, ensemble-based RF models showed the highest performance. Akinrotimi et al.[31] used oversampling techniques for class imbalance problems and applied the dimensionality reduction technique to pick out optimal features with strong predictive ability. They used LR and the NB models as classification strategies for CP. The results showed that NB provided more efficient results than LR. Akbar and Apriono[32] used XGBoost, Bernoulli NB, and DT models for CP and showed that XGBoost attained the best performance compared to other models.

Based on the provided research works on customer CP, the following research gaps can be identified:

- Limited exploration of ensemble models: while some studies have applied ensemble models for CP, such as stacking models, there is still a need for further exploration and evaluation of different ensemble techniques and their effectiveness in improving prediction accuracy.
- Limited investigation of hybrid models: hybrid models that combine different machine learning algorithms or feature selection techniques have shown promising results in CP. However, there is still a lack of comprehensive studies comparing various hybrid models and evaluating their performance on different datasets.
- Lack of focus on industry-specific CP: many studies have evaluated CP models on publicly available datasets, but there is a need for more research focusing on specific industries, such as banking, telecom, and IT. Different industries may have unique characteristics and churn patterns, requiring customized CP approaches.
- Preliminary analysis of feature selection techniques: feature selection plays a crucial role in CP, as it helps identify the most informative features for accurate prediction. However, the existing literature lacks comprehensive analyses and comparisons of different feature selection techniques and their impact on CP performance.
- Lack of comparison across multiple performance metrics: many studies focus on a single performance metric, such as accuracy or F1-measure, for evaluating CP models. However, a comprehensive comparison across multiple metrics, including precision, recall, and area under the receiver operating characteristic curve (AUC-ROC), is essential to understand different models overall performance and effectiveness.

Addressing these research gaps would contribute to advancing the field of customer churn prediction by providing insights into the effectiveness of different models, techniques, and approaches in various industry contexts and facilitating more accurate and proactive customer retention strategies. Although existing models based on ensemble methods achieved tremendous success in the application of CP, there is still a need for more efforts to provide this sector with an efficient and accurate model which can identify churner and non-churner customers accurately and can assess decision-makers in this sector to develop more effective strategies in order to reduce customer churn rate. The GBM model shows excellent potential in classification problems. It typically uses a DT as a base learner to initialize the model, which is sub-optimum[12]. SVM is a powerful mathematical model that proves its ability to solve CP problems[23,30]. Choosing an effective base learner as a starting point for the GBM learning process could produce an effective GBM model. Hence, in this work, the base- learner in the GBM is replaced with the SVM. In addition, the hyper-parameters for the modified GBM are optimized using a modified version of the PSO method. To the best knowledge of the authors, optimizing GBM has never been applied in CP so far. This paper presents a new model that improves the GBM structure and optimizes its hyper-parameters to predict customer churn effectively. The proposed model can assess improving CP's efficiency and designing optimal decisions and policies in this sector.

Proposed CP-EGBM

The overall process flow of our CP is depicted in Fig. 1, with the proposed CP-EGBM classification model in red. The following sub-sections provide the details of the model.

Data preprocessing and feature selection. Let the dataset consist of N examples of M-dimension feature vectors $\{x_{n,m}, 1 \le n \le N \text{and} 1 \le m \le M\}$ and target label $\{y, 1 \le y \le C\}$ where C is the number of classes. Each feature in the dataset is normalized in the range [0, 1] as per Eq. (1) to improve classification capability.

$$\widehat{x}_{n,m} = \frac{x_{n,j} - x_j^{min}}{x_j^{max} - x_j^{min}} \tag{1}$$

where, x_j^{min} and x_j^{max} are minimum and maximum values for the jth feature dimension and $\widehat{x}_{i,j}$ is the normalized value of jth feature for ith example.

The performance of most ML models degrades for a class-imbalanced dataset. A dataset balance can be checked by comparing the number of examples for each class label y. For balancing the dataset, the minority class examples are oversampled to match the number of examples using the Heterogeneous Euclidean-Overlap Metric Genetic Algorithm (HEOMGA) approach[33].

Another critical factor affecting the performance of ML models is the input feature dimensional space. The significant features for classification are selected from the normalized-balanced dataset using Ant Colony Optimization- Reptile Search Algorithm (ACO-RSA) approach[34]. The ACO-RSA is a recent Meta-Heuristic (MH) approach published as a feature selection method for CP. The optimal feature set comprises only the most significant features for classification. Finally, the datasets are split into two exclusive and exhaustive sets for training and testing the proposed CP-EGBM model.

Classification using CP-EGBM. An overview of the GBM, a description of the developed CP-EGBM, and Hyper-parameter optimization for the CP-EGBM are given in this section.

Gradient boosting machine (GBM). Gradient Boosting Machine (GBM)[11] combines a set of weak learners by focusing on the resulting error at each iteration until a strong learner is obtained as a sum of the successive weak ones.

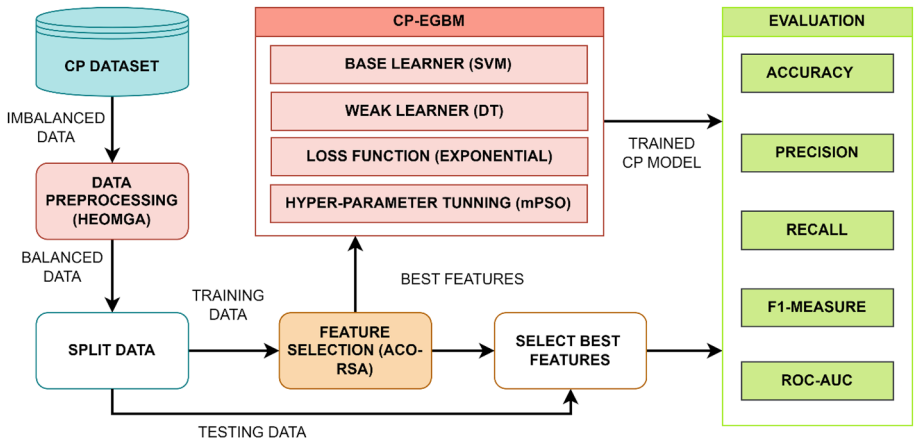

Figure 1. Flowchart of the proposed CP with the enhanced GBM (EGBM) model.

Let $D = \{x_n, y_n\}_{n=1}^{N}$ denote training examples where the goal of gradient boosting is to find an optimal estimate $F(x)$ of an approximation function $F^*(x)$, which maps the instances x_n to y_n to minimize the expected value of a given loss function $L(y, F(x))$ over the distribution of all training examples.

$$F^*(x) = \text{argmin}_{F(x)} L_{x,y}\big(y, F(x)\big) \tag{2}$$

GBM uses a logistic loss function for classification tasks to estimate approximation function $L\big(y, F(x)\big) = \big(y - F(x)\big)^2$[35]. GBM starts with a weak learner $F(x)$ that is usually a constant value, and then it fits each weak learner to correct the errors made by the previous weak learner to strengthen prediction performance by minimizing loss function over each boosting stage[36]. At each stage, the local minimum proportional takes steps to the loss function's negative gradient to find the local minimum. The gradient direction of the loss function at ith boosting stage can be calculated as

$$r_{i,n} = -\left[\frac{\partial L\big(y_n, F(x_n)\big)}{\partial F(x_n)}\right]_{F(x)=F_{i-1}(x)} \tag{3}$$

GBM generalizes the calculation range of the gradient when regression trees is used with parameter a as weak-learners, usually a parameterized function of the input variables x, characterized by the parameters a and ∂ indicates the partial derivative. The tree can be obtained by solving the following:

$$a_i = \text{argmin}_{a,\beta} \sum_{n=1}^{N} \lceil r_{i,n} - \beta h(x_n, a) \rceil^2 \tag{4}$$

where, a_i is a parameter that is obtained at iteration i, and β is the weight value (i.e., the expansion coefficient of the weak learner). Then the optimal length p_i is determined, and the model $F_i(x)$ is updated at each iteration i, with t = 1 to the number of iterations T, as in steps 5 and 6 below in the GBM algorithm. GBM is detailed in Algorithm 1[11].

Algorithm 1: Training process of stat-of-the-art GBM model.

Input: Training dataset $D = \{x_n, y_n\}_{n=1}^{N}$, the maximum number of boosting stages B

Output: GBM $F_i(x)$

1. $F_0(x) = \text{argmin}_p \sum_{n=1}^{N} L(y_n, p)$
2. For m =1 to B do
3. $\quad r_{i,n} = -\left[\frac{\partial L(y_n, F(x_n))}{\partial F(x_n)}\right]_{F(x)=F_{i-1}(x)}$
4. $\quad a_i = \text{argmin}_{a,\beta} \sum_{n=1}^{N} \lceil r_{i,n} - \beta h(x_n, a) \rceil^2$
5. $\quad p_i = \text{argmin}_p \sum_{n=1}^{N} L(y_n, F_{i-1}(x_n) + p\, h(x_n, a_i))$
6. $\quad F_i(x) = F_{i-1}(x) + p_i\, h(x, a_i)$
7. End for

The choices of base learners and loss functions derived from the GBM model facilitate the capacity to design and further development in this model by researchers based on the task requirements[11,12]. This work aims to develop a new classification model for the application of CP by enhancing the structure of the GBM and its hyper-parameters, as will be discussed in the following subsections.

Develop CP-EGBM. As mentioned earlier, the GBM model typically uses a DT as the base learner. At each boosting stage, a new DT (weak learner) is fitted to the current residual and concatenated to the previous model to update the residual. This process continues until the maximum number of boosting stages is reached[12]. However, using DT as a base learner might not optimally approximate a smooth function since DT extrapolates the relationship between the input/output data points with a constant value[5]. Thus, using a DT to start the GBM model training process could result in poor predictive performance and overfitting.

In the GBM model, various base-learners are derived, divided into linear, smooth, and DTs models[12], and optimized the GBM using different manners[37–39]. However, no previous works focused on changing the base learner of the GBM to improve its structure using the SVM in CP. The SVM model introduced in[40] proves its ability to solve various classification problems[41]. As for most classifiers, SVM depends on the training data to build its model by finding the best decision hyperplane that separates the class labels (i.e., response variables). The main goal of the SVM is to find the optimum hyperplane by maximizing the margin and minimizing

classification error between each class. In addition, using kernel functions strategy and its applicability to the linearly non-separable data can be extended to map input data into a higher dimensional space. The hyperplane can be described as follows[42]:

$$\mathrm{w}.x_i + \mathrm{b} = 0, \tag{5}$$

where, w is an average vector, and b is the position of the relative area to the coordinate center.

The optimization of the margin to its support vectors can be converted into a constrained programming problem as:

$$\min \frac{1}{2}[\![w]\!]^2 + C\sum_{i=1}^{N}\zeta_i \text{ s.t.} y_i\left(w^T x_i + b\right) \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0 \tag{6}$$

where, ζ_i represents the misclassified samples to the corresponding margin hyperplane, and C is the cost of the penalty.

The SVM model's most widely used kernel functions are Linear, Polynomial, Sigmoid, and Radial Basis functions (RBF). Among them, RBF is preferable due to its reliability in implementation, adaptability to handle very complex parameters and simplicity[42]. In this research, the SVM with RBF kernel ($\mathrm{SVM_{RBF}}$) is integrated as a base learner in the GBM's structure to boost its learning capability and provide a more accurate approximation of the target label. The RBF kernel function can be given as:

$$k(\mathrm{x_n}, \mathrm{x_i}) = \exp\left(-\gamma\|\mathrm{x_n} - \mathrm{x_i}\|^2 + C\right) \tag{7}$$

where $\mathrm{x_n}, \mathrm{x_i}$ are vectors of features computing from training or test data points, γ determines the influence of each training example, and C is the cost or penalty.

The GBM learning performance for a given task depends greatly on the loss function[12,36]. Therefore, it is essential to carefully select the loss function and the function to calculate the corresponding negative gradients in the GBM model's structure. Several loss functions are reported in the literature for classification, including logistic regression (i.e., deviance), Bernoulli, and exponential. A comparison of loss functions is presented in the next section. The pseudo-code of the developed CP-EGBM is given in Algorithm 2.

Algorithm 2: Pseudo-code of the developed CP-EGBM model for improving CP performance.

Data preprocessing and feature selection

1. Normalize the features in the dataset, Eq. (1).
2. Balance the dataset for all classes using HEOMGA [33].
3. Calculate the optimum feature set using the ACO-RSA approach [34].
4. Split the dataset into training and testing.

CP-EGBM training phase

5. Load training dataset.
6. Initialize the CP-EGBM model with SVM as the base learner, DT as weak learners, logistic/Bernoulli/exponential as a loss function.
7. Tune hyper-parameters of CP-EGBM using mPSO.
8. Train SVM as a base learner using optimum hyper-parameters, Eq. (5)–(7).
9. Train the GBM model using optimum hyper-parameters, as shown in Algorithm 1.

CP-EGBM testing phase

10. Load testing dataset.
11. Select only optimum features as calculated training phase.
12. Evaluate performance metrics using the trained CP-EGBM model.

Hyper-parameter optimization. Parameter setting is essential in enhancing the models' efficacy and performance. Traditionally, hyper-parameters can be selected using a trial-and-error. However, manually tuning the parameters is often time-consuming, yielding unsatisfactory results without deep expertise. MH method can tune the model's hyper-parameters for solving this problem. Two MH methods, PSO and AEO, are presented in the following subsections, and the modified PSO (mPSO) method is introduced.

Particle swarm optimization (PSO). PSO is an MH method inspired to simulate the social and group behaviors of animals, humans, and insects[43]. This method uses a set of particles (initial population) to traverse a given search space randomly. In each iteration, the position of each particle x and the velocity υ of this particle are updated using the best position in the current population.

Let there be P particles in the K-dimensional search space. The position $x(t)$ and velocity $\upsilon(t)$ at the time of t are expressed as:

$$x_i(t) = [x_{i1}(t), x_{i2}(t) \cdots x_{iK}(t)]^T \text{for} 1 \leq i \leq P$$

$$\upsilon_i(t) = [\upsilon_{i1}(t), \upsilon_{i2}(t) \cdots \upsilon_{iK}(t)]^T \tag{8}$$

The fitness, the local best position P_{best} and global best position G_{best} at time t are represented as:

$$P_{best}(t) = [\mathrm{P}_1(t), \mathrm{P}_2(t) \cdots \mathrm{P}_K(t)]^T$$

$$G_{best}(t) = [\mathrm{G}_1(t), \mathrm{G}_2(t) \cdots \mathrm{G}_K(t)]^T \tag{9}$$

At time $t+1$, the velocity $\upsilon(t+1)$ of the particle is updated as,

$$\upsilon_i(t+1) = w\upsilon_i(t) + c_1 r_1(\mathrm{P}_{besti}(t) - \mathrm{x}_1(t)) + c_2 r_2(G_{best}(t) - x_i(t)) \tag{10}$$

where w is an inertia weight factor that controls the velocity and allows the swarm to converge, c_1 is the cognitive factor and c_2 is the social factor that controls the randomness added to the velocity $\upsilon(t+1)$ for the next position $x_i(t+1)$, r_1 and r_2 are two random vectors in the range [0,1].

$$x_i(t+1) = x_i(t) + \upsilon_i(t+1) \tag{11}$$

where the next position $x_i(t+1)$ of ith particle is computed using the current position $x_i(t)$ and updated velocity $\upsilon_i(t+1)$ as generated in Eq. (10). Finally, x_i vectors present solutions while υ_i presents the momentum of particles.

Artificial ecosystem-based optimization (AEO). AEO is another MH method motivated by the energy flow in the natural ecosystem, introduced by[44]. AEO uses three operators to achieve optimal solutions, as described below.

1. Production
   In this operator, the producer represents the worst individual in the population. Thus, it must be updated concerning the best individual by considering the upper and lower boundaries of the given search space so that it can guide other individuals to search other regions. The operator generates a new individual between the best individual x_{best} (based on fitness) and the randomly produced position of individuals in the search space x_{rand} by replacing the previous one. This operator can be given as,

$$x_i(t+1) = (1-\alpha)x_{best}(t) + \alpha x_{rand}(t) \tag{12}$$

$$\alpha = (1 - t/\mathrm{T})r_1 \tag{13}$$

$$x_{rand} = r_2(UB - LB) + LB \tag{14}$$

   where $x_{rand}(t)$ guides the other individuals to explore search space in the subsequent iterations broadly, $x_i(t+1)$ leads the other individuals to exploitation in a region around $x_{best}(t)$ intensively, α is a linear weight coefficient to move the individual linearly from a random position to the position of the best individual $x_{best}(t)$ through the pre-defined maximum number of iterations T, r_1 and r_2 are random numbers in the interval [0, 1], and UB and LB represent the upper and lower boundaries of the search space.
2. 2. Consumption
   This operator starts after the production operator is completed. It may eat a randomly chosen low-energy consumer, a producer, or both to obtain food energy. A Levy flight-like random walk, called Consumption Factor (CF), is employed to enhance exploration capability, and it is defined as follows:

$$\mathrm{CF} = \frac{1}{2}\frac{v_1}{|v_2|}, \quad v_1, v_2 \in N(0,1) \tag{15}$$

   where, $N(0,1)$ is a normal distribution with zero mean and unity standard deviation.
   Different types of consumers adopt different consumption behaviors to update their positions. These strategies include:

   - Herbivore behavior: a herbivore consumer would eat only the producer and can be formulated as:

$$x_i(t+1) = x_i(t) + CF.(x_i(t) - x_1(t)), \quad i \in [2, \ldots P] \tag{16}$$

   - Carnivore behavior: A carnivore consumer would only eat another consumer with higher energy, and it can be modeled as:

$$x_i(t+1) = x_i(t) + CF.\left(x_i(t) - x_{rand\in(0,2i-1)}(t)\right), \quad i \in [3, \ldots P] \tag{17}$$

   - Omnivore behavior: An omnivore consumer can eat a random producer or a producer with higher energy, and this behavior can be formulated as:

$$x_i(t+1) = x_i(t) + CF(r_2(x_i(t) - x_1(t))) + (1 - r_2)(x_i(t) - x_{rand\in(0,2i-1)}(t)), \quad i \in [3, \ldots P] \tag{18}$$

3. Decomposition
   In this final phase, the ecosystem agent dissolves. The decomposer breaks down the remains of dead individuals to provide the required growth nutrients for producers. The decomposition operator can be expressed as:

$$x_i(t+1) = x_P(t) + De(e.x_P(t) - h.x_{\mathrm{rand}\in(0,2i-1)}(t)), i \in [1, \ldots P]$$

$$\text{where } De = 3u \quad u \in N(0,1), e = r_3.randi([1,2]) - 1, \text{and } h = 2r_3 - 1 \tag{19}$$

   and e, h, and De, are weight coefficients designed to model decomposition behavior.

Modified PSO (mPSO) method. The exploration phase is integral to MH algorithms, aiming to find better solutions by investigating search space. PSO suffers from premature convergence to a local minimum, which makes it spend most of the time on locally optimal solutions. Hence, it is weak in exploring new areas in the search space[45,46].

A modified PSO (mPSO) method aims to avoid premature convergence in the local optima and, thus, enhance its capability to tune optimum hyper-parameters for the CP-EGBM model. The mPSO method integrates the consumption operator of the AEO into the PSO method's structure. As discussed in the previous subsection, the consumption phase in the AEO method is responsible for exploration, and it has three leading operators: Herbivore, Carnivore, and Omnivore. Both herbivores and omnivores are based on the producer solution (i.e., equals to the best solution in the swarm); the last operator depends on two randomly selected solutions, which helps explore new regions in the search space. The mPSO method utilizes the strength of the AEO in exploration (Eq. 15) and the strength of the PSO in exploitation (Eq. 10) to select optimum hyper-parameters for the CP-EGBM model. The mPSO can be presented as (Eq. 20): The pseudo-code of the mPSO is described in Algorithm 3.

$$\upsilon_i(t+1) = w\upsilon_i(t) + c_1 r_1(CF - x_1(t)) + c_2 r_2(G_{best}(t) - x_i(t)) \tag{20}$$

Algorithm 3: Pseudo-code of the mPSO approach for GBM hyper-parameter optimization.

1. Initialize particles' positions and velocity, Eq. (8).
2. For t = 1 to T do
3. Calculate local and global best positions w.r.t. minimum fitness, Eq. (9).
5. Calculate CF, as in AEO Eq. (15).
6. Update the velocity of particles, Eq. (20).
7. Update the positions of particles, Eq. (11).
8. End for

Evaluation measures. In this study, the CP-EGBM model is assessed using a set of evaluation measures, including, Accuracy, Precision, Recall, F1-measure, and Area under the ROC Curve (AUC), and they are computed as follows:

$$AC = \frac{TP + TN}{TP + TN + FN + FP} \tag{21}$$

$$\text{Recall}(R) = \frac{TP}{TP + FN} \tag{22}$$

$$\text{F1-measure}(F) = \frac{(TP + FN)(TP + FP)}{TP(2TP + FN + FP)} \tag{23}$$

$$\text{AUC} = \frac{1}{2}\left(1 + \frac{TP}{TP + FN} - \frac{FP}{FP + TN}\right) \tag{24}$$

where True Positive and (TP) and True Negative (TN) denote the correctly detected samples as positive and negative, respectively; similarly, False Negative (FN) and False Positive (FP) represent the number of misclassified positive and negative examples.

Experimental results

The experiments performed to assess the CP-EGBM model, comparing its performance with the GBM and SVM_{RBF} models, are described.

Experimental setup. The performance of the CP-EGBM is validated by conducting experiments on publicly available datasets for CP. The characteristics of these Datasets (DSs) are presented in Table 1. The HEOMGA[33] is used for data balancing and ACO-RSA[34] is employed for FS on all the datasets. Possible bias in selecting the training and testing datasets is avoided using the tenfold cross-validation (CV) technique is employed. All the experiments are implemented using Python and executed on a 3.13 GHz PC with 16 GB RAM and Windows 10 operating system.

Base learner and its behavior in the GBM model. To examine the effect of changing the base learner from DT to SVM_{RBF} in the GBM model, Probability Density Distribution is used, and the test dataset classification score (which is a number between '0' and '1', indicating the degree how much a testing example belongs to Churner/Non-churner class) generated by both base learners are visualized using the Violin plot method[47], as shown in Fig. 2. A classification score is a raw continuous-valued probabilistic output of the ML model. For binary classification, one class (assume churner) has a classification score p then another class will have a score $1-p$.

The Violin plot is a method similar to the box plot with an additional characteristic called probability density, typically smoothed by a kernel density estimator. An interquartile range is calculated for each distribution to compare base learners' dispersion of non-churner and churner classes. The horizontal dotted lines in each class group indicate the first (25th percentile of the data), the second (50th percentile of the data or median), and the third (75th percentile of the data) quartiles to the corresponding distribution. The similarity/closeness of the two distributions is directly proportional to the closeness of these quartiles.

The visualization in Fig. 2 shows that the quartiles of classification score using SVM_{RBF} as a base learner in DS 1, DS 2, DS 5, DS 6, and DS 7 well-separate churners (in red) and non-churners (in green) than the quartiles using the DT. Using SVM_{RBF} as a base learner better classifies the Churner and Non-churner than DT. In DS 3 and DS 4, distributions for churners and non-churners are similar for both base learners, also indicated by closer quartiles for both classes, resulting in poor classification for both base learners. These results confirm and prove the suitability of the SVM_{RBF} to be used as a base learner in the developed CP-EGBM model.

Loss function selection. The loss function gives a general picture of how well the model is performed in predictions. If the predicted results are much closer to the actual values, the loss will be minimum, while if the results are far away from the original values, then the loss value will be the maximum.

In this section, an experiment is conducted using three loss functions to figure out the most suitable one for the application of CP, and they include:

- Logistic, deviance, or cross-entropy loss is the negative log-likelihood of the Bernoulli model. It is the default loss function in the GBM, and it is defined as[48]:

$$L_{Logi}(y,\widehat{y}) = -ylog(\widehat{y}) + (1-y)log(1-\widehat{y}) \tag{25}$$

- Bernoulli, it can be formulated as follows[46].

$$L_{Bern}(y,\widehat{y}) = \log\left(1+\exp\left(-2y.\widehat{y}\right)\right), \tag{26}$$

- Exponential is also used in the Adaboost algorithm, and it can be defined as[48]:

$$L_{Ada}(y,\widehat{y}) = \exp\left(-y.\widehat{y}\right), \tag{27}$$

where y is a binary class indicator, either 0 or 1, and $\widehat{y}$ is the probability of class 1, while $1-\widehat{y}$ is the probability of class 0.

Figure 3 plots the behavior of the loss functions over the defined number of iterations on all the DSs using the developed CP-EGBM. It can be seen in Fig. 3 that the exponential loss function obtains a smaller loss value on all the DSs. This can be explained by exponentially effectively contrasting misclassified data points much more, enabling the CP-EGBM to capture outlying data points much earlier than the logistic and Bernoulli functions.

Dataset description	DS 1	DS 2	DS 3	DS 4	DS 5	DS 6	DS 7
# of instances	3333	7043	71,047	100,000	3333	3150	50,375
# of features	21	21	58	100	11	16	10
# of class	2	2	2	2	2	2	2
Source	31,32	31,32	32	31,32	32	32	32

Table 1. Characteristics of the open-source CP datasets used for evaluating the developed CP-EGBM.

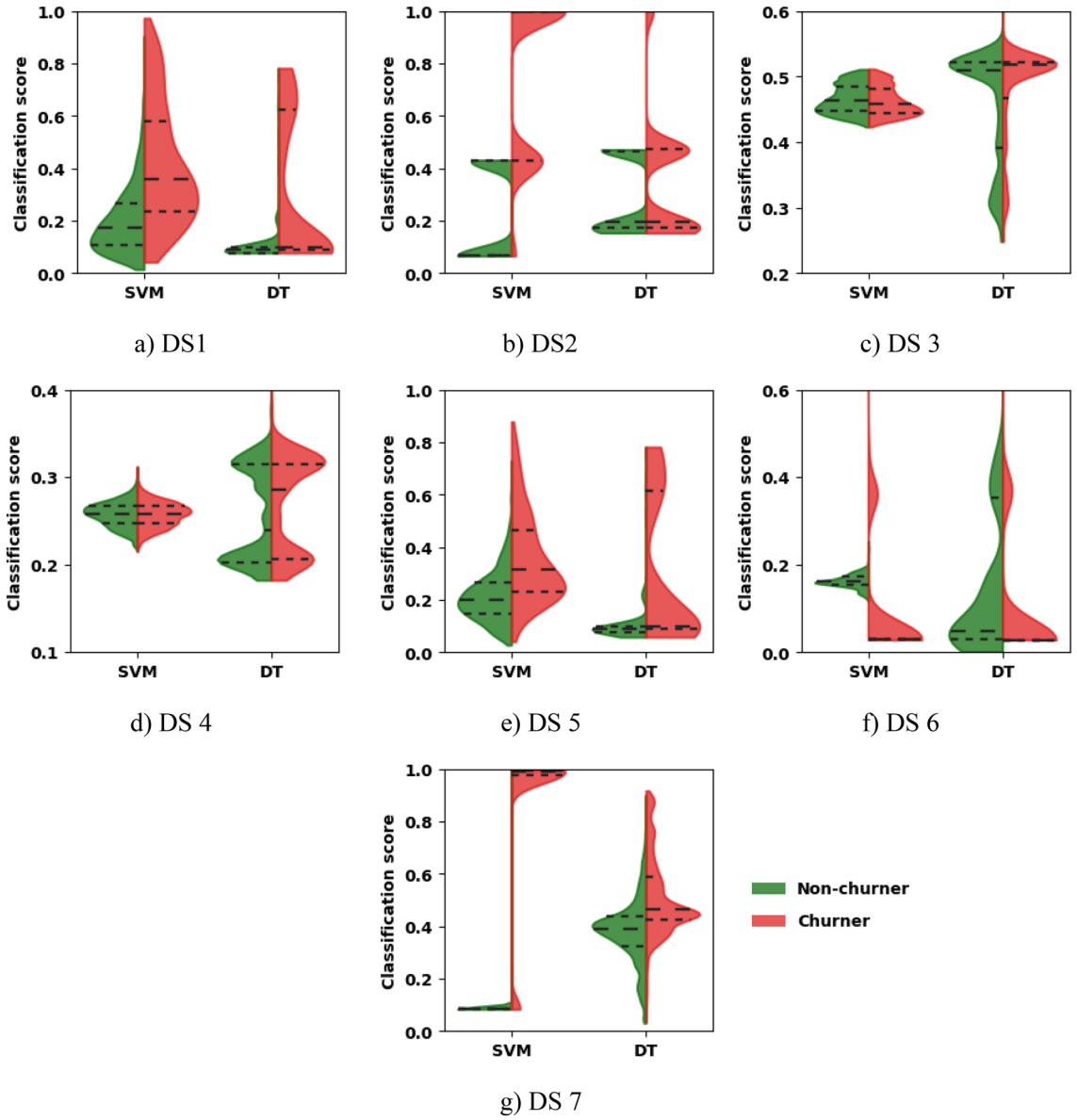

Figure 2. Comparative analysis of testing dataset classification scores generated by SVM_{RBF} and DT base learners based on probability density distribution of non-churners and churners in all the datasets.

The results from this experiment confirm that the exponential loss function is more suitable than the other two competitor loss functions for the application of CP.

Hyper-parameter setting. To better understand the behavior of the introduced mPSO, convergence curves are generated over 50 iterations on the x-axis and fitness values on the y-axis, as shown in Fig. 4. A wide range of MH methods introduced in the literature can be used for hyper-parameters tuning. However, the mPSO is compared with Multi-Verse Optimizer (MVO)[49], Whale Optimization Algorithm (WOA)[50], Gray Wolf Optimizer (GWO)[51], PSO[43], and AEO[44]. For all the methods, the population size is set to 20 and the maximum iterations equal 50. Each is run 20 times, and these settings are selected after empirically studying them. From Fig. 4, the convergence speed of the mPSO is faster than the other MH methods in five out of seven datasets, as it stabilizes to shallow fitness values in fewer iterations. Overall, the suggested improvement in the PSO leads to better convergence attributes and less computation time, making mPSO more suitable for tuning the CP-EGBM model's hyper-parameters.

Several hyper-parameters need to be initialized in the developed CP-EGBM. The mPSO method is used to optimize them. The hyper-parameter settings and the optimized information for each dataset are listed in Tables 2 and 3, respectively.

Experimental results and discussion. The results of the GBM, SVM_{RBF}, and the developed CP-EGBM models using evaluation metrics, Receiver Operating Characteristic (ROC), Statistical test, and model stability

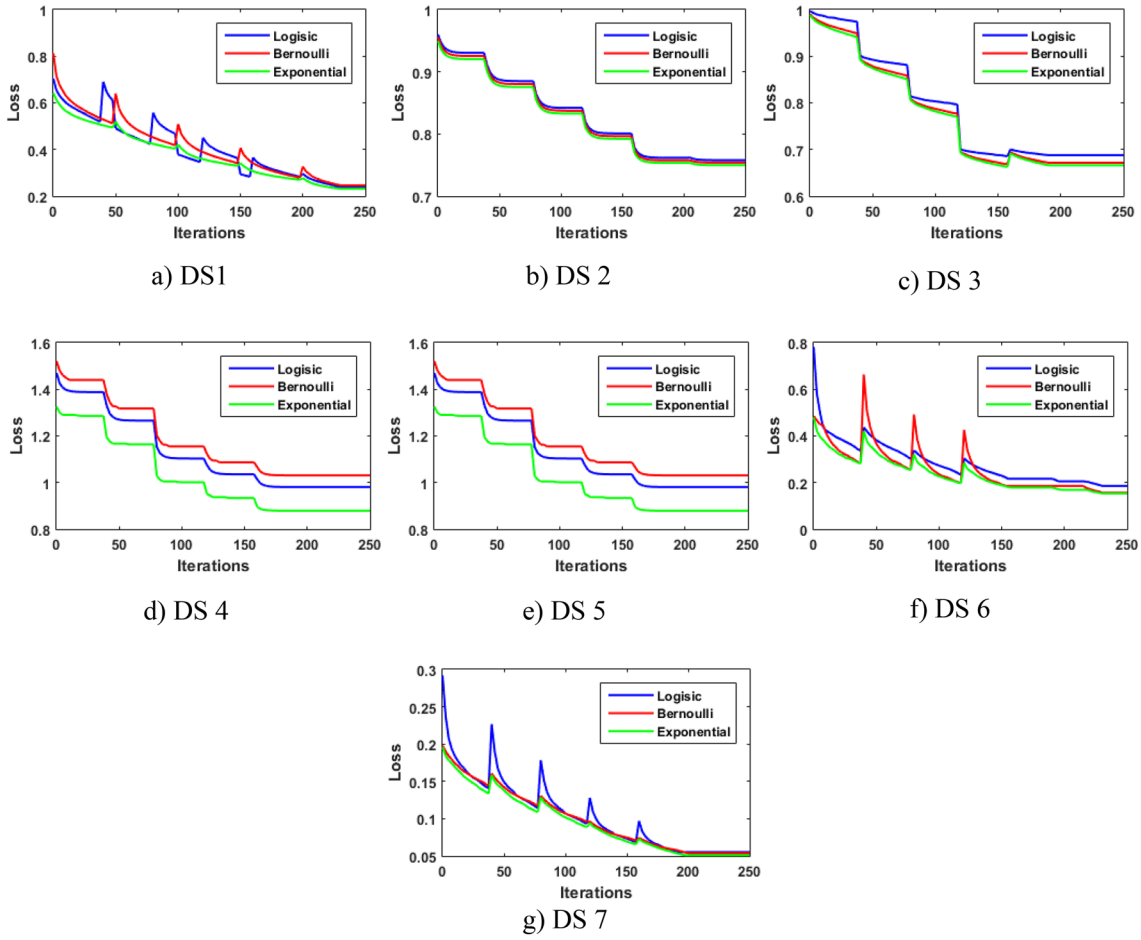

Figure 3. Comparative analysis of loss functions behavior on all the datasets in CP-EGBM framework.

are discussed in this section. Also, a comparison between the CP-EGBM and other used models in recent works is provided.

Performance results. The performance assessment of the GBM alone, SVM_{RBF} alone, and the developed CP-EGBM models on the datasets is carried out in this section. After applying tenfold-CV and fine-tuning the model's hyper-parameters using the mPSO, the average results are computed and recorded in Tables 4, 5, and 6, respectively.

The results in Tables 3, 4, and 5 show that the developed CP-EGBM performs better than the other models on all the datasets for individual evaluation metrics. Figures 5, 6, 7, and 8 show the models' performance on all the datasets. These figures reveal that the CP-EGBM has accomplished effective outcomes compared to GBM and SVM_{RBF}. For instance, in dataset 6, the CP-EGBM obtained an accuracy of 97.79%, a recall of 90.33%, an F1-measure of 91.52%, and an AUC of 92.73%. The results in Tables 3, 4, 5 and Figs. 5, 6, 7, 8 confirm the superiority of CP-EGBM compared to other models.

For DS 1, SVM has relatively good accuracy (0.8799) and F1-measure (0.8410), while GBM has high better accuracy (0.9401) and F1-measure (0.8439). The developed CP-EGBM outperforms both with the highest accuracy (0.9623) and F1-measure (0.8698). For DS 2, SVM alone has moderate performance with accuracy (0.8376) and F1-measure (0.7394), GBM provides accuracy (0.8677) and F1-measure (0.8200), while CP-EGBM provides relatively high accuracy (0.8649) and F1-measure (0.8211). In DS 3, GBM shows the worst performance in accuracy (0.6737) and F1-measure (0.6813). At the same time, CP-EGBM has the best accuracy (0.6949) and F1-measure (0.7044). Similarly, for DS 4 GBM has the lowest accuracy (0.5631) and F1-measure (0.5902) and CP-EGBM shows high performance with accuracy (0.6250) and F1-measure (0.6287). On the other hand, GBM performs better than SVM for DS5. CP-EGBM outperforms both with high accuracy (0.9482) and F1-measure (0.8727). Similar observations can be made for DS 6 with an outstanding performance of CP-EGBM by providing very high accuracy (0.9779) and F1-measure (0.9152). For DS 7, both GBM and CP-EGBM provide the same accuracy but later have higher F1-measure than earlier.

Overall, CP-EGBM consistently outperforms both GBM and SVM across most of the datasets in terms of accuracy, F1-measure, and AUC. However, GBM and SVM show competitive performance, achieving high accuracy and F1-measure on some datasets but lower performance on others.

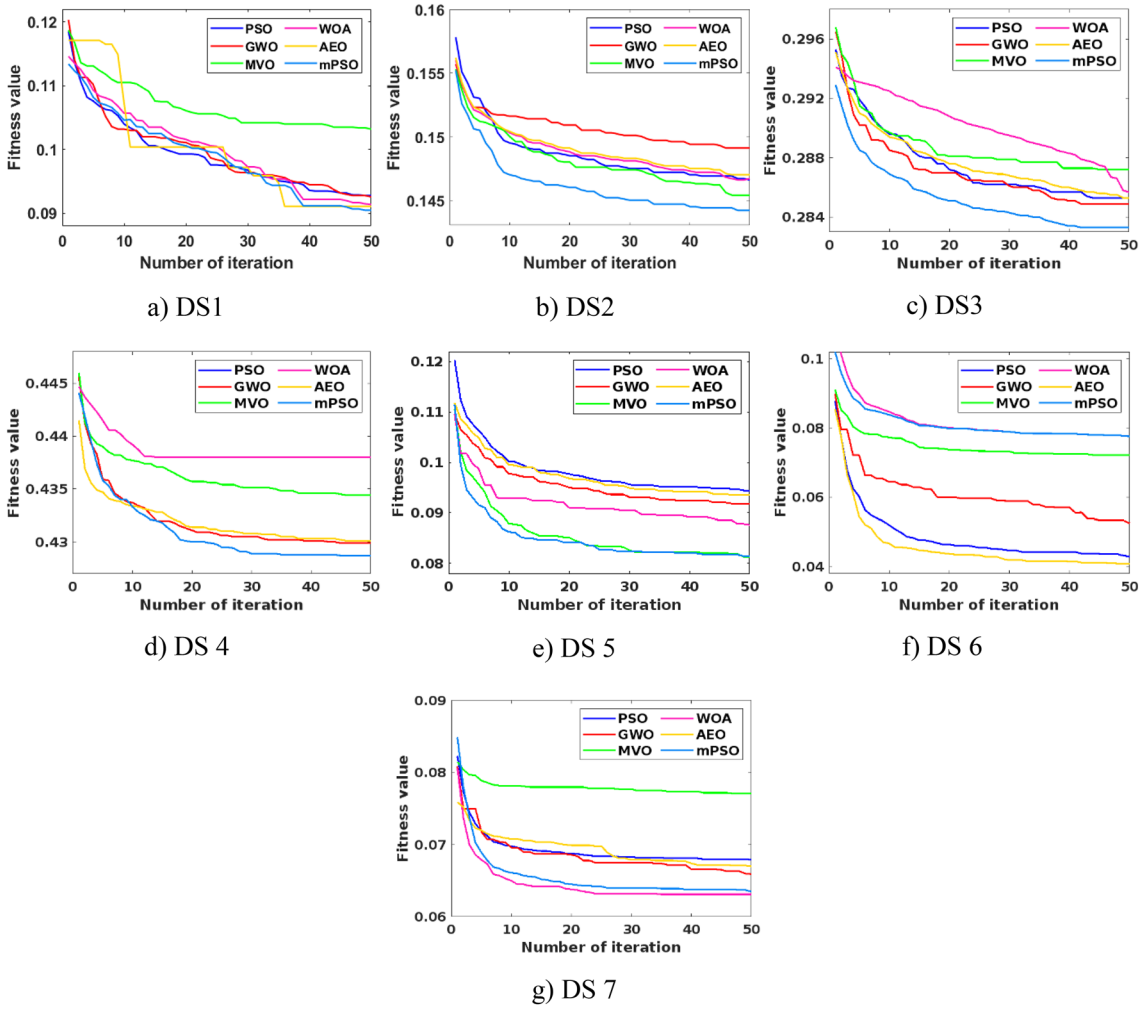

Figure 4. Comparison of convergence behavior of proposed mPSO and other MH algorithms on all the datasets for optimizing CP-EGBM model.

Model	Function	Default value	Search space
SVM_{RBF}	C	1	LB: 1E−1, UB: 1E
	Mapping of the feature space (γ)	1/(#features)	LB: 1E−4, UB:1E4
GBM	Number of estimators	100	LB: 100, UB:3000
	Learning rate	0.1	LB: 1E−3, UB:1
	Maximum depth of DTs	3	LB: 1, UB: 10
	Minimum samples for split	2	LB: 2, UB: 10
	Maximum features	Sqrt(#features)	LB: 1, UB: #features
	Sub-sample	1	LB: 0.5, UB:1

Table 2. Optimization hyper-parameters of different model for tuning the developed CP-EGBM. *LB* lower boundary, *UB* upper boundary.

ROC curve. The ROC curve computes model performance by changing the confidence level of the model score to get distinct values of the True-Positive Rate (TPR) and False Positive Rate (FPR), as illustrated in Fig. 9. As this figure shows, the CP-EGBM curves dominate the GBM and SVM_{RBF} models in all points on all the considered datasets, which indicates the suitability of the developed CP-EGBM.

Statistical test and model's stability. The developed CP-EGBM is selected as the control model in the Friedman ranks test, as shown in Fig. 10. In this figure, CP-EGBM gets the highest accuracy (Fig. 10a) and fitness values

Model	Function	DS 1	DS 2	DS 3	DS 4	DS 5	DS 6	DS 7
SVM_{RBF}	Regularization (C)	100	156	50	65	25	120	87
	Kernel coefficient (γ)	0.213	0.302	0.030	0.001	0.003	0.203	0.137
GBM	Number of estimators	315	503	223	418	438	250	305
	Learning rate	0.093	0.103	0.132	0.312	0.034	0.001	0.003
	Max. depth of DTs	5	5	4	6	6	7	6
	Min. samples for split	5	8	6	10	7	8	9
	Max. features	8	12	25	40	8	8	6
	Sub-sample	1	0.82	0.90	0.95	0.83	0.97	0.83

Table 3. Hyper-parameters of different models in CP-EGBM optimized by mPSO for all the datasets.

Dataset	*AC*	*R*	*F*	*AUC*
DS 1	0.9401	0.7931	0.8439	0.8246
DS 2	0.8677	0.8514	0.8200	0.8062
DS 3	0.6737	0.6528	0.6813	0.7062
DS 4	0.5631	0.6063	0.5902	0.6160
DS 5	0.9352	0.7825	0.8413	0.8187
DS 6	0.9520	0.8747	0.8672	0.8774
DS 7	0.9520	0.7747	0.8150	0.8274

Table 4. Performance evaluation of the GBM alone on all the datasets.

Dataset	*AC*	*R*	*F*	*AUC*
DS 1	0.8799	0.8050	0.8410	0.8462
DS 2	0.8376	0.6743	0.7394	0.7971
DS 3	0.6836	0.6889	0.7009	0.7070
DS 4	0.6157	0.6157	0.6146	0.6261
DS 5	0.8821	0.8050	0.8407	0.8462
DS 6	0.8711	0.8749	0.9004	0.8875
DS 7	0.8711	0.7549	0.8202	0.8275

Table 5. Performance evaluation of SVM_{RBF} alone on all the datasets.

Dataset	*AC*	*R*	*F*	*AUC*
DS 1	0.9623	0.9121	0.8698	0.8579
DS 2	0.8649	0.8456	0.8211	0.8991
DS 3	0.6949	0.7138	0.7044	0.7091
DS 4	0.6250	0.6298	0.6287	0.6329
DS 5	0.9482	0.9175	0.8727	0.8599
DS 6	0.9779	0.9033	0.9152	0.9273
DS 7	0.9520	0.9275	0.8609	0.8473

Table 6. Performance evaluation of the optimized CP-EGBM on all the datasets.

ranks (Fig. 10b), followed by GBM as the second and the SVMRBF ranked last. Therefore, this work concludes that the CP-EGBM is significantly better than the other models for CP.

The relative stability results associated with the standard deviation (Std) of the developed CP-EGBM and the other models are also calculated and provided in Table 7. According to the results in Table 6, the developed CP-EGBM model achieved the smallest Std values compared to the GBM and SVM_{RBF} models on all the datasets. This reflects the stability and robustness of the developed CP-EGBM for applying CP.

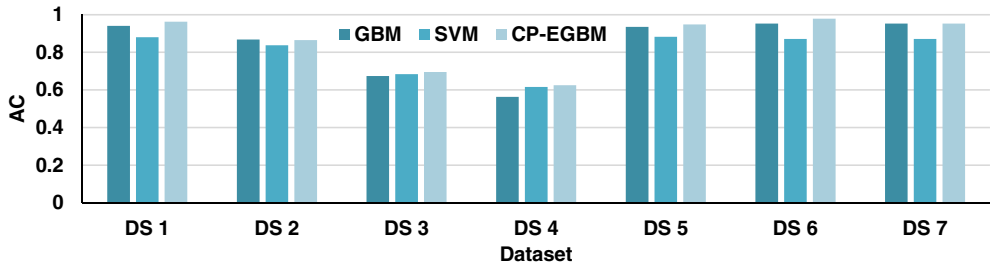

Figure 5. Comparative accuracy analysis of GBM, SVM_{RBF}, and CP-EGBM on all the datasets.

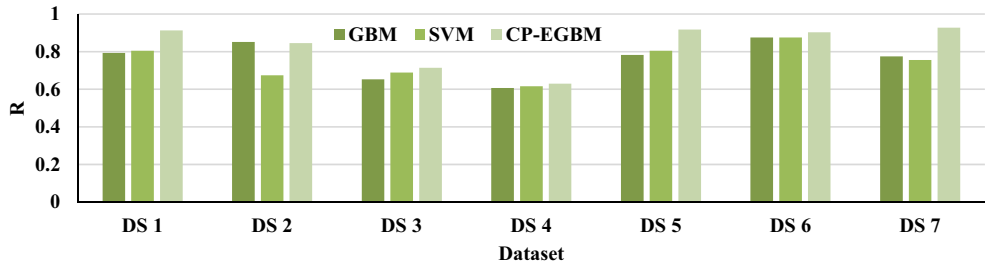

Figure 6. Comparative recall analysis of GBM, SBM_{RBF}, and CP-EGBM on all the datasets.

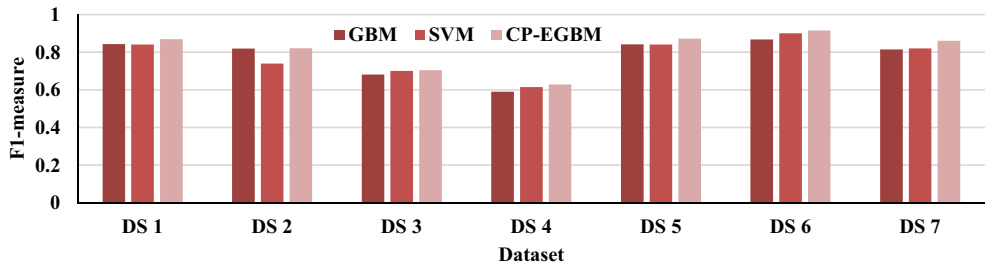

Figure 7. Comparative F1-measure analysis of the GBM, SVM_{RBF}, and CP-EGBM on all the datasets.

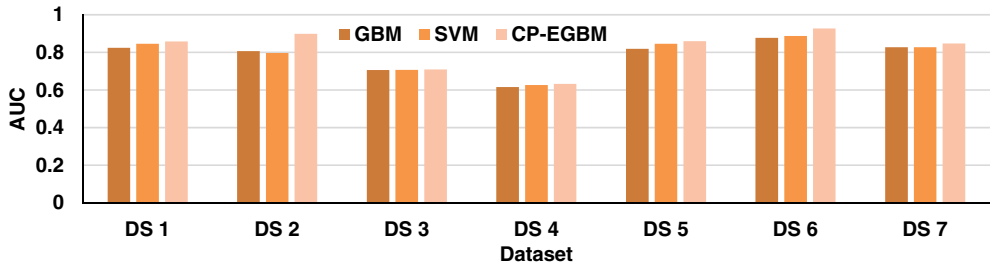

Figure 8. Comparative AUC analysis of the GBM, SVM_{RBF}, and CP-EGBM on all the datasets.

Performance comparison with existing models. Several studies have recently used ML models to predict customer churn in the telecom sector. A comparison between the developed CP-EGBM and other studies for CP is given in Table 8. We can see in Table 7 that the studies utilized DS 1, DS 2, and DS 5 to evaluate ML models used in their works. Therefore, we can use the same DSs to compare the performance of the CP-EGBM with them. As per the results in Table 8, the developed CP-EGBM model has great potential to predict customer churn in terms of accuracy and F1-measure with better prediction results than the existing models.

The proposed framework uses MH algorithms for feature selection and hyper-parameter tuning. Although MH algorithms have shown effectiveness in many domains, they also have certain limitations. MH algorithms

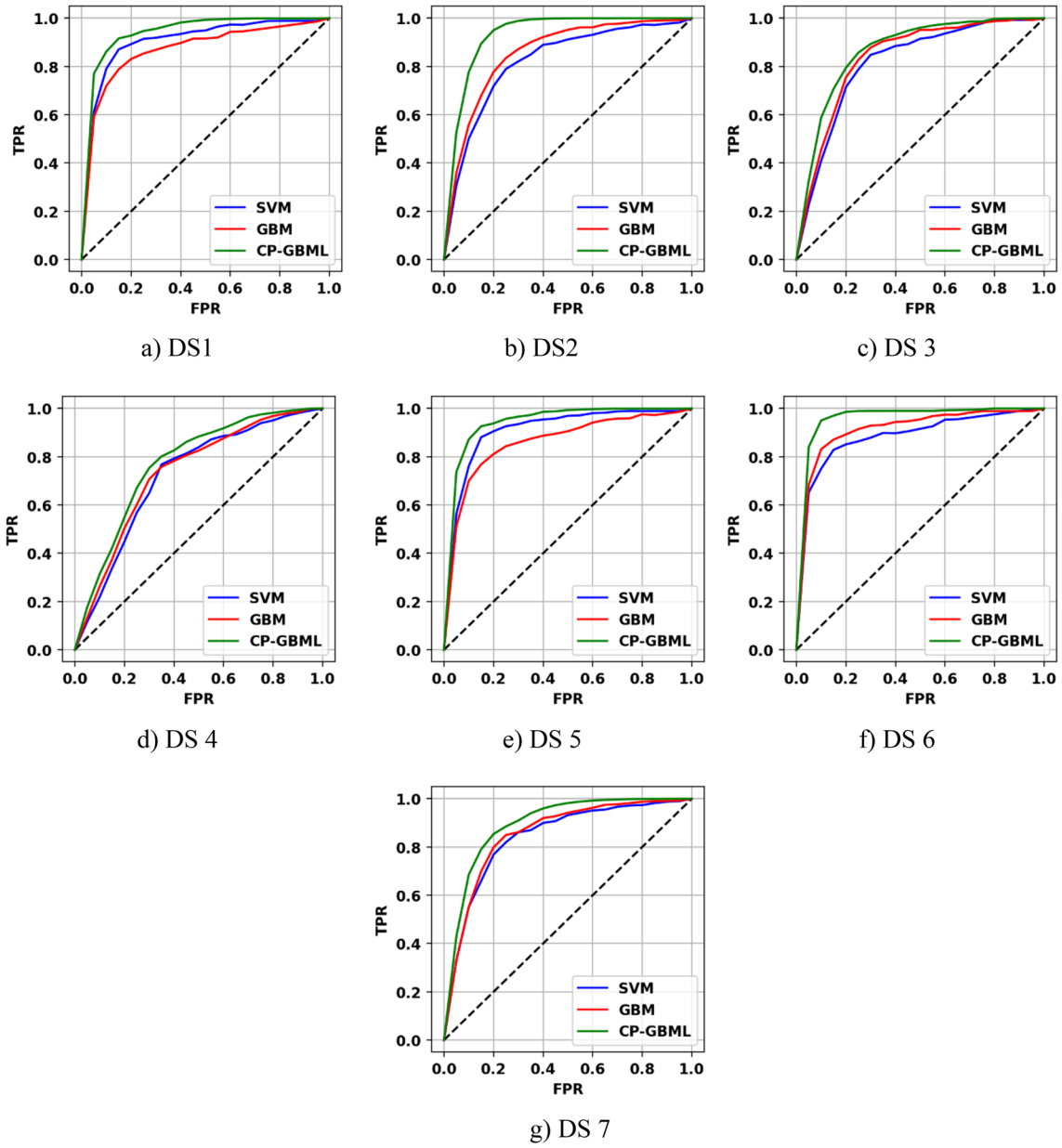

Figure 9. ROC-based performance comparison of GBM, SVM_{RBF}, and CP-EGBM for all the datasets.

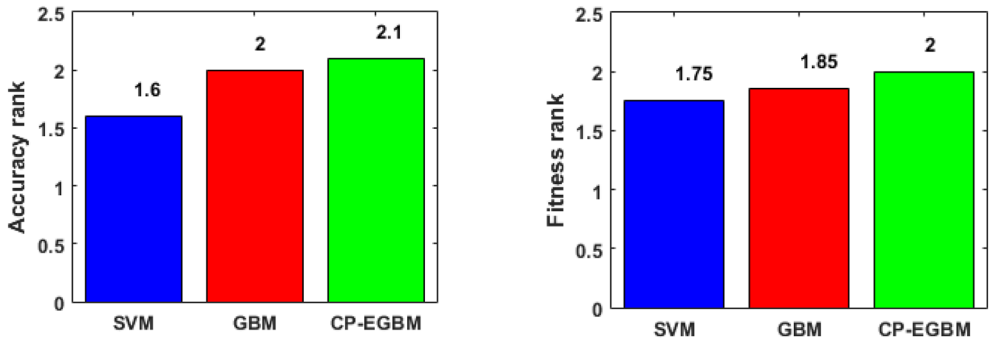

Figure 10. Friedman ranks test for statistical comparison of (**a**) accuracy, (**b**) fitness values of different model.

Dataset	Measure	Model		
		GBM	SVM_{RBF}	CP-EGBM
DS 1	Std	0.0137	0.0111	**0.0091**
DS 2	Std	0.0678	0.0401	**0.0204**
DS 3	Std	0.1025	0.0913	**0.0467**
DS 4	Std	0.1008	0.0925	**0.0381**
DS 5	Std	0.0123	0.0102	**0.0086**
DS 6	Std	0.0106	0.0097	**0.0055**
DS 7	Std	0.0107	0.0094	**0.0078**

Table 7. Comparison of stability and robustness of all models using Std values for all the datasets. Significant values are in bold.

Author (s)	DS 1		DS 2		DS 5	
	AC	*F*	*AC*	*F*	*AC*	*F*
Sabbeh (2018)[22]	0.9600	–	–	–	–	–
Sandhya et al. (2021)[23]	0.9550	0.8210	–	–	–	–
Kimura (2022)[24]	–	–	0.7710	0.613	–	–
Zhu and Liu (2021)[25]	–	–	0.7998	–	–	–
Kanwal et al. (2021)[26]	0.9300	0.8110	–	–	–	–
Bilal et al. (2022)[27]	0.9243	0.7181	–	–	0.9470	0.8063
Karuppaiah and Gopalan (2021)[28]	0.8900	–	–	–	–	
Rabbah et al. (2022)[29],	–	–	0.8350	0.8190	–	–
Karamollaoglu et al.[30]	0.9540	0.9440	–	–	0.7900	0.8630
Akbar and Apriono (2023)[32],	–	–	0.8156	0.7476	–	–
Developed CP-EGBM	**0.9623**	**0.8698**	**0.8649**	**0.8211**	**0.9482**	**0.8727**

Table 8. Comparison between the existing models and the proposed CP-EGBM model in terms of accuracy and F1-measure on DS 1, DS 2, and DS 5. Significant values are in bold.

may converge prematurely to get stuck in a local optimum or fail to explore the search space adequately. MH algorithms require many iterations and evaluations of objective functions, which can be computationally expensive for complex problems. Several control parameters need to be set appropriately to achieve good performance. The search process becomes more challenging in high-dimensional spaces, and MH algorithms may struggle to explore and exploit the search space effectively. Despite these limitations, MH algorithms remain valuable tools for solving complex optimization problems.

Conclusion and future works

The telecom sector has accumulated a massive amount of customer information during its development, and on the other hand, the widespread data warehouses technology and application make it possible to gain insight into historical customer data. Therefore, it has become clear to managers in this sector that customer information can be used to create prediction models to contain customer churn and risk. A CP-EGBM model is developed to provide an efficient prediction model for the application of CP. The CP-EGBM model uses SVM as a base learner and DTs as weak learners in the GBM's structure. Moreover, a modified version of PSO, mPSO, is introduced to optimize the CP-EGBM model's hyper-parameters by injecting the AEO consumption operator into the PSO's structure. The CP-EGBM is assessed using seven CP datasets. The experimental results and statistical test analysis show higher efficacy of the CP-EGBM model than GBM, SVM, and reported models in the literature. The results confirm the ability of the developed model for CP in the telecommunications sector. In the future, we will use the developed CP-EGBM to address other CP-related problems in e-commerce, businesses, and online shopping applications. Also, will deploy CP-EGBM on more datasets to make its results more robust. We will also apply the suggested mPSO method to solve feature selection problems in different fields, such as sentiment analysis and the Internet of Things. In sentimental analysis, each sentence is represented using a high dimensional sparse vector due to tokenization in natural language processing. Similarly, in IoT applications, inputs are represented using high-dimensional vectors because of the large number of sensing nodes. In these applications, mPSO can be used to reduce feature dimensionality, reducing the computational complexity of such systems.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Received: 29 October 2022; Accepted: 22 August 2023
Published online: 02 September 2023

References

1. Huang, Y. & Kechadi, T. An effective hybrid learning system for telecommunication churn prediction. *Expert Syst. Appl.* **40**, 5635–5647 (2013).
2. De Bock, K. W. *et al.* Ensemble classification based on generalized additive models. *Comput. Stat. Data Anal.* **54**, 1535–1546 (2010).
3. Zhou, Y. *et al.* A CEEMDAN and XGBOOST-based approach to forecast crude oil prices. *Complexity* **8**, 1–15 (2019).
4. Athanasiou, V. & Maragoudakis, M. A novel gradient boosting framework for sentiment analysis in languages where NLP resources are not plentiful: A case study for modern Greek. *Algorithms* **10**, 34 (2017).
5. Touzani, S. *et al.* Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy Build.* **158**, 1533–1543 (2018).
6. Bibault, J. E. *et al.* Development and validation of a model to predict survival in colorectal cancer using a gradient-boosted machine. *Gut* **70**, 884–889 (2021).
7. Sharma, T. *et al.* Customer churn prediction in telecommunications using gradient boosted trees. *Int. Conf. Innov. Comput. Commun.* **10**, 235–246 (2020).
8. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).
9. Sagi, O. & Rokach, L. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **8**, e1249 (2018).
10. Freund, Y. & Schapire, R. E. A decision-theoretic generalization of online learning and an application to boosting. *J. Comput. Syst. Sci.* **55**, 119–139 (1997).
11. Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Ann. Stat.* **29**, 1189–1232 (2001).
12. Natekin, A. & Knoll, A. Gradient boosting machines, a tutorial. *Front. Neurorobot.* **7**, 21 (2013).
13. Fan, J. *et al.* Light gradient boosting machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data. *Agric. Water Manag.* **225**, 105758 (2019).
14. Martınez-Munoz, G. & Superior, E. P. Sequential training of neural networks with gradient boosting. arXiv preprint: 1909.12098 (2019).
15. Feng, J. *et al.* Multi-layered gradient boosting decision trees. *Adv. Neural Inform. Process. Syst.* **31**, 134 (2018).
16. Gregory, B. Predicting customer churn: Extreme gradient boosting with temporal data. arXiv preprint: 1802.03396 (2018).
17. Jaisakthi, S. M. *et al.* Customer churn prediction using stochastic gradient boosting technique. *J. Comput. Theor. Nanosci.* **15**, 2410–2414 (2018).
18. Wang, Q. F. *et al.* Large-scale ensemble model for customer churn prediction in search ads. *Cogn. Comput.* **11**, 262–270 (2019).
19. Ahmad, A. K. *et al.* Customer churn prediction in telecom using machine learning in big data platform. *J. Big Data* **6**, 1–24 (2019).
20. Jain, H. *et al.* Churn prediction and retention in banking, telecom and IT sectors using machine learning techniques. *Adv. Mach. Learn. Comput. Intell.* **4**, 137–156 (2021).
21. Dhini, A. & Fauzan, M. Predicting customer churn using ensemble learning: Case study of a fixed broadband company. *Int. J. Technol.* **12**, 1030–1037 (2021).
22. Sabbeh, S. F. Machine-learning techniques for customer retention: A comparative study. *Int. J. Adv. Comput. Sci. Appl.* **9**, 273–281 (2018).
23. Sandhya, G. *et al.* A hybrid learning system for telecom churn prediction using ensemble learning. *Comput. Netw. Inventive Commun. Technol.* **58**, 927–934 (2021).
24. Kimura, T. Customer churn prediction with hybrid resampling and ensemble learning. *J. Manag. Inform. Decis. Sci.* **25**, 1–23 (2022).
25. Zhu, M. & Liu, J. Telecom customer churn prediction based on classification algorithm. *Int. Conf. Aviat. Saf. Inform. Technol.* **16**, 268–273 (2021).
26. Kanwal, S. *et al.* An attribute weight estimation using particle swarm optimization and machine learning approaches for customer churn prediction. *Int. Conf. Innov. Comput.* **16**, 1–6 (2021).
27. Bilal, S. F. *et al.* An ensemble based approach using a combination of clustering and classification algorithms to enhance customer churn prediction in telecom industry. *Peer J. Comput. Sci.* **8**, e854 (2022).
28. Karuppaiah, S. & Gopalan, N. P. Enhanced churn prediction using stacked heuristic incorporated ensemble model. *J. Inform. Technol. Res.* **14**, 174–186 (2021).
29. Rabbah, J. *et al.* A new churn prediction model based on deep insight features transformation for convolution neural network architecture and stacknet. *Int. J. Web-Based Learn. Teach. Technol.* **17**, 1–18 (2022).
30. Karamollaoglu, H. Customer churn prediction using machine learning methods: A comparative analysis. *6th Int. Conf. Comput. Sci. Eng.* **18**, 139–144 (2021).
31. Akinrotimi, A. O. *et al.* A smote-based churn prediction system using machine learning techniques. *Int. Conf. Sci. Eng. Bus. Sustain. Dev. Goals* **1**, 1–6 (2023).
32. Akbar, T. A. R. & Apriono, C. Machine learning predictive models analysis on telecommunications service churn rate. *Green Intell. Syst. Appl.* **3**, 22–34 (2023).
33. AlShourbaji, I. *et al.* A novel HEOMGA approach for class imbalance problem in the application of customer churn prediction. *SN Comput. Sci.* **2**, 1–12 (2021).
34. Al-Shourbaji, I. Boosting ant colony optimization with reptile search algorithm for churn prediction. *Mathematics* **10**, 1031 (2022).
35. Freund, Y. *et al.* A short introduction to boosting. *J. Jpn. Soc. Artif. Intell.* **14**, 771–780 (1999).
36. Badirli, S. Gradient boosting neural networks: Grownet. arXiv preprint: 2002.07971 (2020).
37. Martınez-Mufioz, G. & Superior, E. P. Sequential training of neural networks with gradient boosting. arXiv preprint: 1909.12098 (2019).
38. Feng, J. *et al.* Soft gradient boosting machine. arXiv preprint: 2006.04059 (2020).
39. Zhou, Z. H. *et al.* Ensembling neural networks: Many could be better than all. *Artif. Intell.* **137**, 239–263 (2002).
40. Vapnik, V. N. An overview of statistical learning theory. *IEEE Trans. Neural Netw.* **10**, 988–999 (1999).
41. Patle, A. & Chouhan, D. S. SVM kernel functions for classification. *Int. Conf. Adv. Technol. Eng.* 1–9 (2013).
42. Xia, J. *et al.* Performance optimization of support vector machine with oppositional grasshopper optimization for acute appendicitis diagnosis. *Comput. Biol. Med.* **143**, 105206 (2022).
43. Kennedy, J. & Eberhart, R. Particle swarm optimization. *Int. Conf. Neural Netw.* 1942–1948 (1995).
44. Zhao, W. *et al.* Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm. *Neural Comput. Appl.* **32**, 9383–9425 (2020).
45. Haklı, H. & Uğuz, H. A novel particle swarm optimization algorithm with Levy flight. *Appl. Soft Comput.* **23**, 333–345 (2014).
46. Kołodziejczyk, J. & Tarasenko, Y. Particle swarm optimization and Levy flight integration. *Proc. Comput. Sci.* **192**, 4658–4671 (2021).
47. Hintze, J. L. & Nelson, R. D. Violin plots: A box plot-density trace synergism. *Am. Stat.* **52**, 181–184 (1998).
48. Painsky, A. & Wornell, G. On the universality of the logistic loss function. *IEEE Int. Symp. Inform. Theory* 936–940 (2018).
49. Mirjalili, S. *et al.* Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **27**, 495–513 (2016).

50. Mirjalili, S. & Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016).
51. Mirjalili, S. *et al.* Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014).

Author contributions

Conceptualization, I.A. N.H. and Y.S, methodology, I.A, software, I.A, validation, I.A, N.H. and Y.S.; formal analysis, I.A, N.H, Y.S. A.G.H. L.A, and B.E; writing—original draft preparation, I.A.; review and editing, I.A. N.H., Y.S., and L.A, supervision N.H, Y.S. A.G.H. L.A. All authors have read and agreed to the published version of the manuscript.

Funding

Open access funding provided by Linköping University.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.G.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.