Scientific African 23 (2024) e02054

Contents lists available at ScienceDirect

Scientific African

journal homepage: www.elsevier.com/locate/sciaf

Empirical analysis of tree-based classification models for customer churn prediction

Fatima E. Usman-Hamza[a], Abdullateef O. Balogun[b,*], Salahdeen K. Nasiru[a], Luiz Fernando Capretz[c], Hammed A. Mojeed[a,d], Shakirat A. Salihu[a], Abimbola G. Akintola[a], Modinat A. Mabayoje[a], Joseph B. Awotunde[a]

[a] Department of Computer Science, University of Ilorin, Ilorin 1515 Ilorin, Nigeria
[b] Department of Computer and Information Science, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 32610 Perak, Malaysia
[c] Department of Electrical and Computer Engineering, Western University, London, Ontario, N6A 5B9, Canada
[d] Department of Technical Informatics and Telecommunications, Gdańsk University of Technology, Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland

ARTICLE INFO

Editor: DR B Gyampoh

Keywords:
Customer churn prediction (CCP)
Class imbalance problem (CIP)
Tree-based classifiers

ABSTRACT

Customer churn is a vital and reoccurring problem facing most business industries, particularly the telecommunications industry. Considering the fierce competition among telecommunications firms and the high expenses of attracting and gaining new subscribers, keeping existing loyal subscribers becomes crucial. Early prediction of disgruntled subscribers can assist telecommunications firms in identifying the reasons for churn and in deploying applicable innovative policies to boost productivity, maintain market competitiveness, and reduce monetary damages. Controlling customer churn through the development of efficient and dependable customer churn prediction (CCP) solutions is imperative to attaining this goal. According to the outcomes of current CCP research, several strategies, including rule-based and machine-learning (ML) processes, have been proposed to handle the CCP phenomenon. However, the lack of flexibility and robustness of rule based CCP solutions is a fundamental shortcoming, and the lopsided distribution of churn datasets is deleterious to the efficacy of most traditional ML techniques in CCP. Regardless, ML-based CCP solutions have been reported to be more effective than other forms of CCP solutions. Unlike linear-based, instance-based, and function-based ML classifiers, tree-based ML classifiers are known to generate predictive models with high accuracy, high stability, and ease of interpretation. However, the deployment of tree-based classifiers for CCP is limited in most cases to the decision tree (DT) and random forest (RF). Hence, this research investigated the effectiveness of tree-based classifiers with diverse computational properties in CCP. Specifically, the CCP performances of diverse tree-based classifiers such as the single, ensemble, enhanced, and hybrid tree-based classifiers are investigated. Also, the effects of data quality problems such as the class imbalance problem (CIP) on the predictive performances of tree-based classifiers and their homogeneous ensemble variants on CCP were assessed. From the experimental results, it was observed that the investigated tree-based classifiers outperformed other forms of classifiers such as linear-based (Support Vector Machine (SVM)), instance-based (K-Nearest Neighbour (KNN)), Bayesian-based (Naïve Bayes (NB)) and function-based (MultiLayer Perceptron (MLP)) classifiers in most cases with or without the CIP. Also, it was observed that the CIP has a significant effect on

* Corresponding author.
E-mail addresses: abdullateef.ob@utp.edu.my (A.O. Balogun), hammed.mojeed@pg.edu.pl (H.A. Mojeed).

https://doi.org/10.1016/j.sciaf.2023.e02054
Received 6 August 2023; Received in revised form 7 November 2023; Accepted 22 December 2023
Available online 23 December 2023

the CCP performances of investigated tree-based classifiers, but the combination of a data sampling technique and a homogeneous ensemble method can be an effective solution to CIP and also generate efficient CCP models.

Introduction

To address the lingering constraints to information transformation or economy digitization, organizations have been shifting towards advanced technological concepts over the last decade [1]. Due to these technological advancements, organizations were able to adopt and implement emerging technology trends such as digital networking, mobile application solutions, cloud storage, and big data analytics [2]. In today’s information era, organizations that use such methods may gain a substantial competitive edge, especially those that provide telecommunications services [3]. By 2030, it is estimated that the proportion of the global economy that will be conducted digitally would increase by more than twice the current value (from 22 % to 50 %), with the telecommunications industry leading the way [4]. This may be due to the fierce market competition, rapid pace of technological advancement, and customer expectations in the telecommunication industry. Regardless, the influence of the stake of customers in the telecommunication sector in actualizing digital transformation is as critical as its technological prowess [5]. That is, customers are considered essential stakeholders in today’s rapidly changing corporate world. Decisions and policymakers in the telecommunications industry can see clearly that customer data may be utilized to assess customer behaviour for retention reasons. When customers in a free market have several different service providers to choose from, they are more likely to jump ship fast which can be regarded as customer churn [6,7].

Customer churn is the quick shift in patronage that may occur in response to a variety of factors, including but not limited to discontent, a hike in price, poor quality, a lack of functions, and so on [8,9]. It has been noticed that customer churn has an immediate negative effect on multiple firms in the telecommunication sector and across other sectors including financial services, airline services, and gaming services [3,10–12]. Specifically, Customer Churn Prediction (CCP) facilitates the assessment of the likelihood of future customer churn based on information on the customer’s past behaviours or the customer information such as call logs, messages (voice and text), financial records etc. This information is crucial for determining whether customers are on the verge of leaving, or "churning, ". For this reason, it is crucial for firms to accurately anticipate their customers’ actions [6,7,11]. Customers may be retained based on either a proactive or reactive business model. A reactive business model allows a customer’s departure before providing incentives to keep them staying. On the other hand, proactive strategies consider consumers’ inclination to depart and provide them with appropriate advantages. Under the proactive approach, identifying churners and non-churners may be formulated as a binary classification problem. Also, the proactive approach is more realistic and aligns with business principles than the reactive approach. Regardless, CCP’s primary objective is to facilitate the creation of customer retention strategies that will eventually increase corporate income and garner an industry reputation [13–15].

Several CCP solutions with varying computational characteristics such as rule-based and machine learning (ML)-based techniques have been formulated and deployed with different outcomes [1815–21]. Rule-based CCP methods chiefly based on rough-set theories have been deployed with comparable prediction performances [16,17,22]. However, the scalability and adaptability issues are the primary drawbacks of rule based CCP solutions. Regarding ML-based CCP models, classifiers such as Naïve Bayes (NB), Bayesian Network (BN), Support Vector Machine (SVM), k Nearest Neighbour (kNN), Multi-layer Perceptron (MLP), Decision Tree (DT), Random Forest (RF) have been extensively used to develop suitable CCP solutions [6,9,10,15,19,20]. However, some of these ML classifiers such as SVM, MLP, and kNN do not perform well on large datasets, are heavily dependent on the state of their parameters, and often generate unstable models that are not easily interpretable [12,15,18,19,23]. Also, NB and BN as instances of Bayesian learning models necessitate a considerable amount of computational resources to ascertain an optimal Bayesian hypothesis [24,25]. DT and RF are somewhat unique in their capabilities and prediction performances as they can generate models with high accuracy and stability and are interpretable. However, it was observed from existing studies that only DT and RF as instances of tree-based classifiers have been prominently used in CCP tasks [26–29]. This may be due to their respective simplicity and high prediction performances in ML processes. Regardless, there are other tree-based classifiers with varying computational characteristics that can be comparable in prediction performances to DT and RF classifiers. Hence, this research investigates and conducts a comprehensive empirical analysis of the effectiveness of tree-based classifiers in CCP. Firstly, the CCP performances of thirteen tree-based classifiers (RF, DT, logistic model tree (LMT), functional tree (FT), credal decision tree (CDT), cost-sensitive forest (CS-Forest), decision stump (DS), hoeffding tree (HT), random tree (RT), reduced error pruning tree (REPTree), alternating decision tree (ADT), systematically developed forest of multiple decision trees (SysFor), and logit-boost alternating decision tree (LADTree)) from three variations (single, ensemble, and hybrid) of tree-based classifiers are investigated. Secondly, the effectiveness of the homogeneous ensemble (Bagging, Boosting, Cascade generalization, Rotation Forest, Dagging, and Real Boosting) variants of the tree-based classifiers is evaluated. Lastly, the effects of the data quality problems such as the class imbalance problem (CIP) on the prediction performances of the investigated tree-based classifiers and their respective homogeneous ensemble variants for CCP are analyzed. To alleviate the CIP, the synthetic minority over-sampling technique (SMOTE) is used as a practical solution to the inherent CIP in customer churn datasets, and it is combined with the tree-based classifiers and their enhanced variants to build improved CCP solutions.

In summary, the major contributions of this research are stated as follows:

1. Extensive empirical performance analysis of thirteen tree-based classifiers based on different computational variants for CCP with and without the CIP.

2. Extensive empirical performance analysis of six homogeneous ensemble variants of tree-based classifiers based on different computational variants (78 models) for CCP with and without the CIP.
3. The combination of data sampling and ensemble methods as an effective solution to CIP and the generation of effective tree-based CCP models.
4. The CCP performances of tree-based classifiers are evaluated and validated with diverse baseline (individual and homogeneous ensemble variants) classifiers and state-of-the-art existing rule, ML, and DL-based CCP models.

Additionally, for a comprehensive analysis, the following research questions (RQs) are identified to assess the efficacy of the tree-based CCP models.

1. How effective are the tree-based CCP models in comparison with baseline methods in CCP with and without the CIP?
2. What are the intra and inter-tree-based CCP performances with and without the CIP?
3. How effective are the homogeneous ensemble variants of the tree-based classifiers with and without CIP?
4. How effective are the tree-based CCP models in comparison with the state-of-the-art existing rule, ML, and DL-based CCP solutions?

The rest of the paper is structured as follows. Section 2 investigates current CCP research. Section 3 examines the research methods and experimental design used in this research. Section 4 presents the experimental findings and their explanation. Section 5 presents the challenge to the validity of this research, while Section 6 ends this research and emphasizes future work.

Literature review

In this part, we delve into the different ML-based strategies used by current CCP systems and provide an in-depth analysis of their efficacy.

Customer defection to a rival brand is referred to as customer churn and what constitutes customer churn varies from industry to industry and domain to domain. Long-term inactivity is a common indicator of churn which can be caused by several factors. Several researchers and experts in the telecommunications sector have used different tactics from a variety of studies to deal with the issue of customer churn.

The modern telecommunications sector, like many other service-based industries, is focused on growing its customer base, boosting its bottom line, and establishing itself as an industry leader. Customers in the telecommunications sector may be less loyal to their current service providers than they formerly were because of the proliferation of comparable businesses, incentives, and service-related packages. Thus, decision-makers in the telecom business focus on customer retention rather than customer acquisition to save costs. Losing customers is bad for business. Because of this, CCP is vital not just for the company's image but also for keeping existing customers at a cheaper cost.

The impacts of CCP methods such as rule-based, ML, data mining, and hybrid (ML and data mining) approaches have been investigated in much research [16,30–32]. These techniques aid decision-makers in the telecommunications sector in categorizing and projecting customer churn and keeping loyal customers. DT, a popular ML method for CCP in the telecommunications sector, are the most typical approach described in the literature for dealing with CCP issue [7,26,27,33]. Nonetheless, [34] showed how DTs are limited by intricate nonlinear linkages between attributes. Another research, [35] looked at whether pruning helps DTs function better. However, several research [9,36] also noted some benefits of the DTs in the context of CCP such that they can process both categorical and numerical data of customers, can easily visualize and construct the classification model and can employ a nonparametric method that does not require prior assumptions. DTs are useful, but they lack the coverage rate of a classification system (Balogun [37]). Also, [26] examined the CCP performance of DT and logistic regression (LR). From their result, they noted the usefulness of DT for CCP because of its high accuracy and interpretability. However, DTs have trouble dealing with non-linear correlations between features, and LR has trouble with interaction effects between variables. As such, other forms of tree-based classifiers such as FTs and CS-Forest are known to have better coverage rates and correlation between features as compared with conventional DTs and LR.

In an effort for effective CCP models, [9] sought to combine numerous ML approaches, including DT, RF, gradient-boosted tree, and XGBoost. The resulting AUC value of 93.3 % indicates that this method is effective at identifying at-risk customers. Also, [34] combined multiple ML algorithms (DT, artificial neural network (ANN), support vector machine (SVM), naive Bayes (NB), and LR) for CCP. Specifically, they compared different boosting versions of ML approaches for CCP. From the results, it was stated that combining SVMPoly with AdaBoost had the best CCP performance. However, the parameter tuning, and the black-box nature of the deployed SVM, ANN, LR, and ensemble method are of concern [38].

Based on hybrid approaches, [39] used a neural network-based method for CCP in the telecommunications sector. Also, [40] utilized NN with a Self-Organized map (SOM) to create a hybrid strategy for CCP. The NN feature was deployed for selecting relevant and irredundant features from the training set. The output from the prior phase was used as input into the SOM model builder, which resulted in the CCP. Their result showed that the combination of NN methods, rather than just one form of NN, had better prediction results. However, the issue of data loss is of primary concern in their approach. Similarly, [41] utilized a hybrid model consisting of LR and voted perceptron to examine CCP in the Asian mobile operator communications dataset. Their results showed that compared to using a single ML technique, the hybrid model is more accurate with a ROC value of 0.721. [30] presented a hybrid learning approach for CCP in the telecommunications sector by combining weighted k-means clustering and rule induction techniques. With demonstrations of enhanced accuracy, ROC, and AUC, the authors demonstrate the superior performance of their suggested hybrid learning approach. These reported findings showed the superiority of hybrid methods over single methods in CCP. However, the overhead computational cost is another drawback.

To mitigate the problems of hybrid approaches, ensemble methods are being used for ML tasks. Several studies have investigated the use of ensemble methods or models in CCP [32,35]. Xu et al., [32] highlighted that ensemble methods simply aggregate the performance of multiple single classifiers into a single model. De Bock and Van den Poel [42] developed an ensemble approach for CCP by fusing the rotation forest with the Rotboost algorithm. Rotation forest (RotF) ML is used for feature extraction, while Rotboost is based on AdaBoost and rotation forest to customer churn. The results showed that Rotboost was more accurate than the conventional rotation forest, although the forest had a better AUC value. Nevertheless, the main drawback of this research was the incomprehensibility of the elements that led to CCP. Also, [43] proposed the integration of the Generalized Additive Models (GAM) method inside an ensemble approach. In summary, their CCP performance is superior to that of individual classifiers trained by logistic regression and the GAM technique. However, data quality problems such as class imbalance can still affect the CCP performances of ensemble models.

Findings from existing studies have indicated that the presence of latent class imbalance problems in a dataset may lead to subpar performances in ML classifiers [44]. Finding a balance (minimal values) in a model's bias and error rate is crucial to developing an effective prediction model. That is, ML models are expected to correctly categorize class labels present in a dataset irrespective of their amount (minority or majority). For instance, [45] discovered that conventional ML algorithms fare well on imbalanced datasets, where the classifier tends to classify the majority class accurately only. This is because most telecommunications customers fall into the majority class, while only a small percentage of samples represent the minority class. In contrast, class imbalance poses threats during the training phase of traditional ML models, leading to subpar classification performance [46,47].

One of the viable ways of addressing the class imbalance problem is the data sampling approach. Six popular oversampling techniques used in the telecommunications industry have been studied in depth and analyzed in a meta-analysis of prospective observational studies [48,49]; these include the Synthetic Minority Oversampling TEchnique (SMOTE), the Mega trend diffusion function (MTDF), the Adaptive Synthetic Sampling Approach (Adasyn), the Couple topN reverse k-nearest neighbour (CTRKNN), the Majority Weighted Minority Over (ICOTE). Results showed that when compared to the other five oversampling strategies and rule generation ML strategies (learning from example module version 2 (LEM2), covering, and exhaustive), the CCP model performance using MTDF and genetic algorithm-based rules extraction for churn classification performed the best. But neither over- nor under-sampling is required to boost efficiency. However, [50] found that neither the oversampling of minority class samples nor the undersampling of majority class samples substantially aided in solving the classification issue. Particle swarm optimization (PSO) under-sampling techniques were studied in another research with mRMR data reduction and KNN, RF, and RotF ensemble ML techniques [51]. According to their findings, the performance of the CCP in the telecommunications sector may be enhanced by paying attention to the subject data's high dimensionality during the data pre-processing stage. Relatedly, [52] presented the CCP model using an SVM and a random sampling strategy to rectify the asymmetry between the classes in the data. Whilst it attempted to achieve a more even distribution of classes, the CCP model did not boost prediction accuracy. Another research [53] looked at the use of a weighted RF ML model for CCP, despite the widespread criticism that this method is black-box and hard to explain.

Feature engineering, the process of creating meaningful features from existing data, has been found to play a significant role in the construction of promising CCP models. [54] used the PSO approach for feature selection, then had a domain expert manually estimate the weights for each feature using a subjective weight process. However, the assignment of the subjective weight is time-consuming and expensive, and the authors only explored the effect of features on different weights rather than feature reduction. The suggested method was reported to be outperformed and reached 93 % accuracy, as opposed to 90 % accuracy for the DT, 89 % accuracy for the KNN, and 89 % for the NB classifier [55]. on the other hand, employed a DL fmethod to forego human feature engineering and instead created a hybrid deep neural network (DNN) architecture for CCP by analyzing the Crowd Analytix and Cell2Cell telecom datasets. As compared to more conventional ML models like SVM and RF, the authors state that DNN eliminates the need for human feature selection. In addition, they highlighted substantial issues with feature engineering, such as time constraints, subjective feature ranking, and generalizability problems. To address the drawbacks of feature engineering in CCP, [56] looked at feature selection using the Kmeans (KM) and single value decomposition (SVD) algorithms, as well as the feature filtering approach, before developing the CCP model. Based on their findings, occurred that the suggested method has the potential to vastly enhance accuracy while simultaneously lowering the number of misclassifications.

In summary, many CCP models and approaches from conventional ML classifiers to sophisticated models based on DL, ensemble, and neuro-fuzzy approaches on CCP have been investigated. Regardless, due to the impact of CCP across industrial sectors, particularly in CRM, there is a continuous need for novel and robust CCP solutions. Hence, this research aims to conduct a comprehensive empirical analysis of the effectiveness of tree-based ML classifiers with various computational characteristics in the presence of a class imbalance problem for CCP.

Methodology

This section describes the steps and techniques used to conduct the research. In particular, the investigated tree-based classifiers are described in detail. Also, the experimental approach, the performance assessment metrics, and the CCP datasets are provided and discussed.

Tree-based classification algorithms

Tree-based classification algorithms are a form of ML technique that utilizes tree structure to build classification models for categorizing data. Tree-based classifiers are a unique form of model that may recursively segment the data into subsets based on the significance of certain parameters or data features. Through each node of the tree, a choice is taken based on the significance of a feature. This decision indicates the path of the tree construction. In addition, tree-based classification algorithms can handle both numerical and categorical data, which makes them adaptable and may be used for a broad variety of different use cases. Besides, tree-based structure and operation mechanisms are easy to comprehend, which makes them a valuable medium to develop interpretable and explainable classification models. Consequently, this study aims to investigate the deployment of tree-based classification algorithms with varying computational characteristics. In particular, the CCP performances of RF, DT, LMT, FT, CDT, CS-Forest, DS, HT, RT, REPTree, ADT, SysFor, and LADTree are evaluated. Also, due to their computational diversity, the tree-based classification algorithms are classified into the single, ensemble, and hybrid tree-based classifiers. The following subsections present each of these groupings of tree-based classifiers alongside their respective parameter settings as utilized in the experimentation processes of this research.

Single tree classification algorithms

Single tree classification algorithms are tree-based classifiers that are built primarily on one sole tree structure. The operation of single tree classification algorithms involves splitting the input data into similar subsets, with each subset being determined by the results of a sequence of binary analyses that are performed on each feature of the data. At each node of the tree, the test that offers the most effective division of the data into the two groups is to be used for further processing. This method is then applied recursively to each subset of the data until it reaches the leaf nodes of the tree, which stand for the ultimate categorization of the data that was supplied. DT, DS, are RT are examples of single-tree classification algorithms that have been used in several ML processes. In a DT, each node represents a feature or attribute, and each edge represents a possible value or outcome of that feature. The root node represents the most important feature, and subsequent nodes represent the next most important features. The leaf nodes represent the outcome or decision. To build a DT, the algorithm analyzes the data and decides which feature to split on at each node, based on criteria such as information gain or Gini impurity. The algorithm continues recursively until all the data is separated into its final decision or outcome. In the case of DS, a single decision node and two or more leaf nodes are considered for tree construction and decision-making. Unlike the DT, a DS only considers one feature or attribute to decide. That is, the decision node evaluates a single feature and splits the data into two subsets based on a threshold value. It is a very simple model that is often used as a building block for more complex models. An RT is a type of binary search tree that allows for more efficient operations by performing rotations on the tree to maintain balance. RT is a very effective type of tree classifier and has been used in ML tasks. An ADT is a type of single tree that is used for both classification and regression problems. It is like a standard DT, but instead of making a single split at each node, an ADT makes two splits at each node, one for each class or group being considered. At each node of the ADT, the algorithm alternates between two classes or groups and generates a pair of decision rules, one for each class. Each decision rule is generated using a statistical test that maximizes the separation between the two classes or groups, based on the available features. The ADT algorithm continues to recursively split the data until the stopping criterion is met. The stopping criterion is typically the maximum depth of the tree, or the minimum number of samples required in a node. ADTs are particularly useful in situations where the classes or groups are not separated, and the decision boundary is complex. They are also useful in situations where the data is imbalanced, i.e., class imbalance problem which is inherent in CCP.

Ensemble tree classification algorithms

Ensemble tree classification algorithms are based on the combinations of multiple single tree models to enhance the prediction performances of the classification process. Specifically, ensemble tree classifiers are built by utilizing diverse subsets of the training data based on different randomization strategies to leverage their collective performance and reduced error rates. Compared to standalone single tree models, ensemble tree classification models are better in terms of avoiding model overfitting as they can deal with numerous sources of variance and provide resilience against noisy and missing values. Also, ensemble tree classification models can adapt and work well with high-dimensional feature spaces, unlike single tree classification models that only work well with low-dimensional feature spaces. Ensemble tree classification models carefully select the most useful features at each split and can circumvent the splitting constraint and solve the issue in fewer dimensions [7,57–59]. RF, CS-Forest, SysFor, and LADTree are prime examples of ensemble tree classification algorithms and are utilized in this research work as ensemble tree classification models. RF builds multiple decision trees independently and combines their predictions by majority voting. In RF, a huge number of decision trees are created separately using distinct subsets of the training data and random feature subsets. Each decision tree in the forest is trained on a random subset of the available features at each split. This approach generates a diversified collection of decision trees that are less prone to overfitting and more resistant to data noise. In the case of CS-Forest, every single tree in a CS-Forest is trained to minimize a loss function that accounts for the costs associated with misclassifying data. The single trees are constructed using a variant of the DT method that considers the misclassification costs and utilizes the information gain ratio as the splitting criteria. Each DT in CS-Forest predicts the class of testing data by processing the data and making a judgement about it. After evaluating each tree's performance on a validation set and their respective misclassification costs, the final prediction is made using a weighted average of the trees' predictions. SysFor can build many trees even from a low-dimensional data set. Another strength of the technique is that instead of building multiple trees using any attribute (good or bad) it uses only those attributes that have high classification capabilities. For LADTree, combines the principles of DTs and boosting to create a powerful predictive model. LADTree builds a set of ADTs, where each tree is trained to predict the residual error of the previous tree. This approach allows the algorithm to capture complex nonlinear relationships between the input variables and the target variable. It incorporates the concept of boosting, where each tree is trained on a weighted sample of the data, where the weights are adjusted to give more emphasis to the misclassified instances in the previous tree.

This approach helps to improve the overall accuracy of the model.

Hybrid tree classification algorithms

Hybrid tree classification algorithms are different from single and ensemble tree classification algorithms as they are based on the improvement of some operations (leaf selection or node splitting) of a single tree model to enhance its prediction performances. Specifically, hybrid tree classifiers are built by addressing some limitations in single tree models by introducing a combination of tree structures or sophisticated functions to improve their efficacy and ultimately prediction performances. Hybrid tree algorithms are particularly useful when dealing with high-dimensional data or data with complex relationships between features as they can capture a wider range of feature interactions and reduce the risk of overfitting to noisy or irrelevant features [60–64]. CDT, HT, REPTree, FT, and LMT are some prominent hybrid tree classification algorithms. CDT is a type of hybrid tree that is used in decision-making under uncertainty. Unlike a traditional DT, which assumes that all the information available is certain, a CDT represents the available information using sets of probabilities or imprecise probabilities, also known as sets of probability distributions. Also, HT is a type of hybrid tree that is used in most cases for stream mining applications. Unlike conventional single trees, which require the entire dataset to be available before they can be built, HT can incrementally learn from data in real time, without requiring access to the entire data at once. HT is designed to handle large datasets where the data may arrive in a continuous stream, and where the underlying distribution of the data may change over time. They use a statistical test called the Hoeffding bound to determine when a decision can be made with a high level of confidence, even with a small sample size. The Hoeffding bound is a mathematical formula that determines the number of samples required to accurately estimate the probability distribution of a random variable with a given level of confidence. By using the Hoeffding bound, HT can dynamically adjust the size of the tree as new data arrives and only expand the tree when there is sufficient evidence to do so. REPTree as a hybrid tree is based on using a pruning technique to reduce overfitting in its constituent trees. Overfitting occurs when a tree is too complex, and it captures the noise in the training data, leading to poor performance on new data. In REPTree, a large tree is first built using the training data. Then, the tree is pruned by removing some of the branches that do not contribute significantly to the overall accuracy of the tree. This pruning process helps to simplify the tree and reduce its complexity, thereby reducing the risk of overfitting. To determine which branches to prune, a validation set is used. The validation set is a portion of the training data that is not used during the tree-building process but is instead used to evaluate the performance of the tree. The pruning process involves iteratively removing branches from the tree and evaluating the resulting tree's accuracy on the validation set. The branches that lead to the least reduction in accuracy are removed, and the process is repeated until further pruning would decrease accuracy on the validation set. According to Gama's [37] proposal, Functional Trees (FT) are created by using constructive induction to combine multivariate DTs and discriminant functions. FT is also known as a multivariate tree generalization. In the leaf nodes and decision nodes, FT incorporates features. For creating classification trees, FT sometimes contains features at both the nodes and the leaves, so that decision nodes are built based on the classification tree's growth and functional leaves are built when the tree is pruned [37]. For applications requiring prediction, FT can be used to foretell the value of class variables for a specific dataset. The dataset specifically traverses the tree from the root node to a leaf, expanding its set of characteristics utilizing constructor methods inserted into each decision node. The node's decision test is then used to determine the direction in which the dataset will travel. The dataset is finally labelled as a leaf using either the constructor function based on the leaf or the leaf-related constant [37, 38]. Classical single tree algorithms divide the input data into tree nodes by comparing the value with a constant of specific input attributes, whereas FT uses logistic regression (LR) functions for internal node splitting (referred to as oblique split) and leaf prediction. This is the primary difference between conventional single-tree algorithms and FT. In the case of LMT, linear logistic regression (LLR) is combined with the DT technique. LMT is a powerful predictive model that can provide an interpretable model with excellent prediction accuracy. LMT

Table 1
Parameter setting of the implemented tree-based classifiers.

Classifiers	Tree Type	Parameter Setting
DT	Single Trees	BinarySplit=False; CollapseTree=True; ConfidenceFactor=0.25; numDecimalPlaces=2; minNumObj=2; reduceErrorPrunig=False; subTreeRaising=True; useMDLCorrection=True
DS		BatchSize= 100; numDecimalPlaces=2
RT		KValue=0; allowUnclassifiedInstances=False; breakTiesRandomly=False; minVarianceProp=0.001;
ADTree		numOfBoostingIterations=10; searchPath=ExpandAllPaths; resume=False; batchSize=100
RF	Ensemble Trees	BagSise=100; breakTiesrandomly=False; calculateOutofBag=False; numInterations=100; numExecutionSlots=1; numDecimalPlaces=2
CS-Forest		batchSize=100; confidence=0.25; costGoodness=0.2; costMatrix=2×2cost matrix; minRecLeaf=10; numberTrees=60; separation =0.3;
SysFor		Batchsize=100; confidence=0.25; minRecLeaf=10; numberTrees=60; separation =0.3; minRecLeaf=10; goodness=0.3;
LADTree		batchSize=100; numOfBoostingIterations=10; resume=False
HT	Hybrid Trees	BatchSize=100; gracePeriod=200; hoeffdingTieThreshold=0.05; leafPredictionstrategy=adaptive Naïve Bayes; minimumFractionofWeightingInfoGain=0.01; NaiveBayesPredictionThreshold=0.0; splitConfidence=1.0E-7; splitCriterion=info gain split
CDT		KTHRootAttribute=1; SValue=1.0; initialCount=0.0; maxDepth=-1; minNum=2.0; minVarianceProp=0.001; noPruning=False; spreadInitiaCount=False
REPTree		initialCount=0.0; maxDepth=-1; minNum=2.0; mainVarianceProp=0.001; spreadInitialCount=False
LMT		doNotMakeSplitPointActualValue=False; fastRegression=True; minNumInstances=15; numBoostingOteartions=-1; SplitOnResiduals=False; useAIC=False; weightTrimBeta=0.0
FT		binSplit=False; errorOnProbabilities=False; ModelType=FT; numBoostongIterations=15, useAIC=False; numDecimalPlaces=2

consists of a single parent root, many branches, leaves, and nodes. It constructs a standard DT but inserts an LLR from the leaf nodes down to the vertices. In deciding how to divide, it considers the information gain ratio [26, 27]. Because of its special characteristics, LMT is considered a hybrid tree classification algorithm in this study.

Overall, the effectiveness and efficiency of the tree-based classification algorithms will be investigated. Table 1 presents the categorization of the tree-based classification algorithms and their respective parameter settings as used in this research.

Experimental procedure

Fig. 1 is a schematic depiction of the experiments performed for this study. The method outlined here is critical since it is designed to provide empirical data on the performance of the investigated CCP models. Particularly, we established and analyzed a two-stage experimental design, and we uniformly and objectively assessed the prediction capabilities of the resulting CCP models.

At first, the respective tree-based classification algorithms are implemented and evaluated on the CCP datasets with the class imbalance problem. That is, each of the RF, DT, LMT, FT, CDT, CS-Forest, DS, HT, RT, REPTree, ADT, SysFor, and LADTree are deployed on the original customer churn datasets. The essence of this evaluation is to empirically assess the prediction performances of diverse tree-based on the original customer churn datasets. Specifically, the differences in the prediction performances within each variant and across the variants of the tree-based classifiers will be investigated. Findings from these analyses will validate the effectiveness and performance of tree-based classifiers in CCP. Thereafter, the CCP performances of the tree-based classifiers will be further evaluated on balanced customer churn datasets. The SMOTE technique will be deployed to resolve the inherent class imbalance problem in the original customer churn datasets. SMOTE is a well-known data sampling approach that has been used to address the issue of class imbalance in ML tasks [65,66]. Besides, the CCP performances of these tree-based classifiers will be compared with prominent linear, function, and instance-based classifiers with or without the class imbalance problem (that is, on the original and balanced datasets).

Secondly, the CCP performances of the tree-based classification algorithms are further amplified with the use of homogeneous ensemble methods. Particularly, six homogeneous ensemble methods (Boosting, Bagging, Dagging, Rotation Forest, Cascade Generalization, and RealAdaboost) will be used to amplify the performances of the tree-based classifiers. The ensemble variant models will be tested with both original and balanced customer churn datasets to ascertain their respective efficacy in the presence of the CIP. Table 2 presents the homogeneous ensemble methods and their respective parameter settings as used in this research. Observed findings from these evaluations will provide insight and answers to the RQs presented in the introduction section. Also, high-performing tree-based models will be compared with existing state-of-the-art CCP models.

Investigated CCP models are generated utilizing the cross-validation (CV) approach as part of the CCP model development

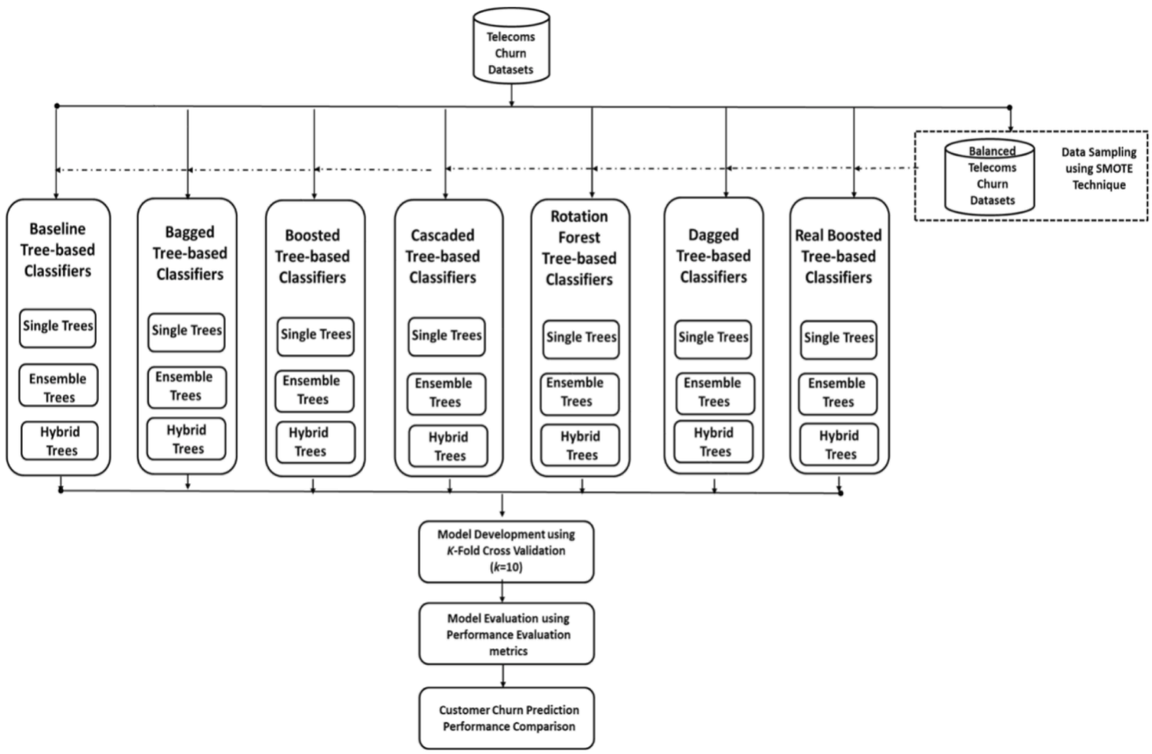

Fig. 1. Experimental framework.

Table 2
Parameter setting of the implemented ensemble methods.

Classifiers	Parameter Setting
Boosting	Classifier= { RF, DT, LMT, FT, CDT, CS-Forest, DS, HT, RT, REPTree, ADT, SysFor, and LADTree}; resume=False; useResampling=False; weightThreshold=100; numIterations=10
Bagging	Classifier= { RF, DT, LMT, FT, CDT, CS-Forest, DS, HT, RT, REPTree, ADT, SysFor, and LADTree}; CalcOutOfBag=False; storeOutOfBagPredictions=False; bagSizePercent=100; numIterations=10
Rotation Forest	Classifier= { RF, DT, LMT, FT, CDT, CS-Forest, DS, HT, RT, REPTree, ADT, SysFor, and LADTree}; projectionFilter=PCA; removePercentage=50; MaxGroup=3; MinGroup=3; batchSize=100; numIterations=10
Cascade	Classifier= { RF, DT, LMT, FT, CDT, CS-Forest, DS, HT, RT, REPTree, ADT, SysFor, and LADTree}; ContatenatePredictions=True; KeepOriginal=True; meta=PCT; numIterations=10
Dagging	Classifier= { RF, DT, LMT, FT, CDT, CS-Forest, DS, HT, RT, REPTree, ADT, SysFor, and LADTree}; verbose=False; seed=1; batchSize=100; numIterations=10
RealBoost	Classifier= { RF, DT, LMT, FT, CDT, CS-Forest, DS, HT, RT, REPTree, ADT, SysFor, and LADTree}; verbose=False; seed=1; batchSize=100; numIterations=10

procedure. The CCP models in this study were developed using k-folding (with k set to 10). The CV method is preferred due to its resilience to data quality issues that might cause model overfitting [46,67–70]. For this reason, we shuffled both the training and testing sets at random to exclude the possibility of any repetitive patterns or values. Each experiment was repeated 10 times to obtain reliable results among all CCP models tested. Finally, the deployed CCP models are graded according to their mean values on the selected performance indicator. The experiments were carried out on an Intel(R) CoreTM i7-6700 processor with 16 GB of RAM and 3.4 GHz of CPU speed, using the WEKA machine learning library [71] and the R programming language [72].

Experimented datasets

This study used existing customer churn datasets compiled from several telecommunications providers to get insight into customer behaviour. These datasets contained information such as consumer demographics and the services acquired by these customers. Several types of clients who are inclined to churn were the subject of this study. This study makes use of two datasets, one from the Kaggle repository (Dataset A) and another from the UCI repository (Dataset B). Previous studies of CCP have made extensive use of these publicly accessible customer churn records [7,10,13,73,74]. Dataset A depicts a telecommunications company that provides phone and internet services to its customers, and it was gathered from the IBM business analytics community. The ratio of non-churners (NC) to churners is 2850 to 483. There are a total of 3333 cases. Much like Dataset A, Dataset B includes 5000 occurrences, but only 4493 of them are NC. Table 3 displays more data on the churn rate and magnitude of the imbalance ratio. Furthermore, a tabular description of the features in Datasets A and B are presented in Appendix A and B.

Performance evaluation metrics

In this research, we compared the prediction performance of several CCP models using well-established assessment criteria including the area under the curve (AUC) and Matthews Correlation Coefficient (MCC). These metrics were selected since they have been used extensively in the evaluation of ML-based CCP models in the literature [75–77]. Besides, AUC calibrates the trade-off between sensitivity and specificity at the best-chosen threshold while MCC considers all areas of the confusion matrix produced by each new model [66,78].

Results and discussions

This section presents and analyzes the empirical results of the various experiments conducted based on the experimental procedure described in Section 3.2. The CCP performances of the investigated tree-based classification algorithms are evaluated using the selected performance evaluation metrics as mentioned in Section 3.4. Thereafter, the CCP performances of the tree-based classifiers will be compared with the prominent baseline classifiers and their respective homogeneous (bagging, boosting, cascade, rotation forest, dagging, and real boosting) variants on both original and balanced (SMOTE) customer churn datasets. In the end, the comparison of the best-performing tree-based classification algorithms with existing CCP models with diverse computational characteristics is presented on each of the deployed customer churn datasets.

Table 3
Description of CCP datasets.

Dataset	Features	Instances	Churners	Non-Churner	Churn Rate	Imbalance Ratio
Dataset A	21	3,333	483	2,850	14.49%	5.9
Dataset B	18	5,000	507	4,493	10.14%	8.86

CCP performances of tree-based classification algorithms

In this sub-section, the CCP performances of tree-based classification algorithms on both the original and balanced customer churn datasets are presented and compared. The comparison is to ascertain the efficacy and resilience of the tree-based classifiers with and without the latent class imbalance problem in the customer churn datasets. The performance comparisons are done based on the categorization of the tree-based classification algorithms. That is, tree classifiers from the same category/variants are compared within themselves and the best performer from each category are further analyzed. Also, the CCP performances of the tree-based classifiers are further compared with prominent ML classifiers with diverse computational features on both original and balanced customer churn datasets. This analysis will validate the performance of investigated tree-based classifiers against known baseline classification algorithms as utilized in existing CCP studies.

For clarity, the result analysis is presented and discussed in the form of two scenarios. The first scenario presents the CCP performances of tree-based classifiers and the baseline ML classifiers on original Datasets A and B. For the second scenario, the CCP performances of tree-based classifiers are compared with the same baseline ML classifiers but on the balanced (SMOTE) Datasets A and B.

Scenario 1: CCP performance comparison of tree-based classifiers against baseline ML classifiers on original datasets A and B

As presented in Table 4, the CCP performances of tree-based classifiers were compared among themselves and with the prominent classifiers such as SVM, KNN, NB, and MLP on original Dataset A. As observed, the tree-based classifiers had relatively high and comparable AUC and MCC values on original Dataset A. Specifically, SysFor had the highest AUC value of 0.914 followed by CS-Forest (0.906), LMT (0.905), and FT (0.905). Based on the categorization, among the single trees, ADTree had the highest AUC value of 0.890 followed by DT (0.876), RT (0.709), and DStump (0.603). In terms of the MCC values, DT (0.742) had the highest value followed by ADTree (0.697), RT (0.413), and DStump (0.317). These results showed that there are other single-tree classifiers such as ADTree that are comparable in performance to DT. In the case of the ensemble tree classifiers, SysFor outperformed other classifiers with AUC and MCC values of 0.914 and 0.772, respectively. Likewise, the CS-Forest also had a remarkable CCP performance on Dataset A with high AUC (0.906) and MCC (0.718) values as compared with RF which had an AUC and MCC values of 0.896 and 0.581, respectively. It is worth noting that the RF had a good CCP performance, however, SysFor and CS-Forest were better in this case. For the hybrid tree classifiers, LMT and FT recorded proportionate results with the same AUC value of 0.905 and slightly comparable MCC values of 0.777 and 0.763, respectively. Other hybrid tree classifiers such as CDT, HT, and REPTree also had comparable CCP performances. In comparison to the baseline classifiers such as SVM, KNN, NB, and MLP, most of the experimented tree-based classifiers were superior in CCP performance to the baseline classifiers on the original Dataset A. Table 4 presents tabulated experimental results and Fig. 2 illustrates the graphical representation of the CCP performance comparison of tree-based classifiers and the baseline classifiers on original Dataset A.

Furthermore, Table 5 shows the CCP performances of tree-based classifiers and prominent baseline classifiers on original Dataset B. However, the CCP results of the tree-based classifiers and baseline classifiers based on Dataset B are somewhat low with average AUC values ($<=0.500$) and in most cases negative MCC values. These results indicated that the experimented classifiers generated poor models, and this can be attributed to the quality of Dataset B. As shown in Table 3, Dataset 2 has an IR value of 8.86 which indicates the presence and high degree of class imbalance phenomenon in Dataset B. This observed finding further corroborates the existing knowledge that data quality problems such as the class imbalance problem can affect the predictive capabilities of ML classifiers. Fig. 3 presents the graphical representation of the CCP performance comparison of tree-based classifiers and the baseline classifiers on original Dataset B.

In summary, it can be observed that the CCP performances of the tree-based classifiers are in most cases superior to that of the

Table 4
The CCP performance comparison of tree-based classifiers against baseline ML classifiers on Original Dataset A.

	Dataset A	AUC	MCC
Single Trees	DT	0.876	0.742
	DStump	0.603	0.317
	RT	0.709	0.413
	ADTree	0.890	0.697
Ensemble Trees	RF	0.896	0.581
	CS-Forest	0.906	0.718
	SysFor	**0.914**	0.772
	LADTree	0.887	0.700
Hybrid Trees	HT	0.836	0.472
	CDT	0.893	0.762
	REPTree	0.831	0.500
	LMT	0.905	**0.777**
	FT	0.905	0.763
Base Classifiers	SVM	0.500	?
	KNN	0.603	0.237
	NB	0.834	0.465
	MLP	0.775	0.347

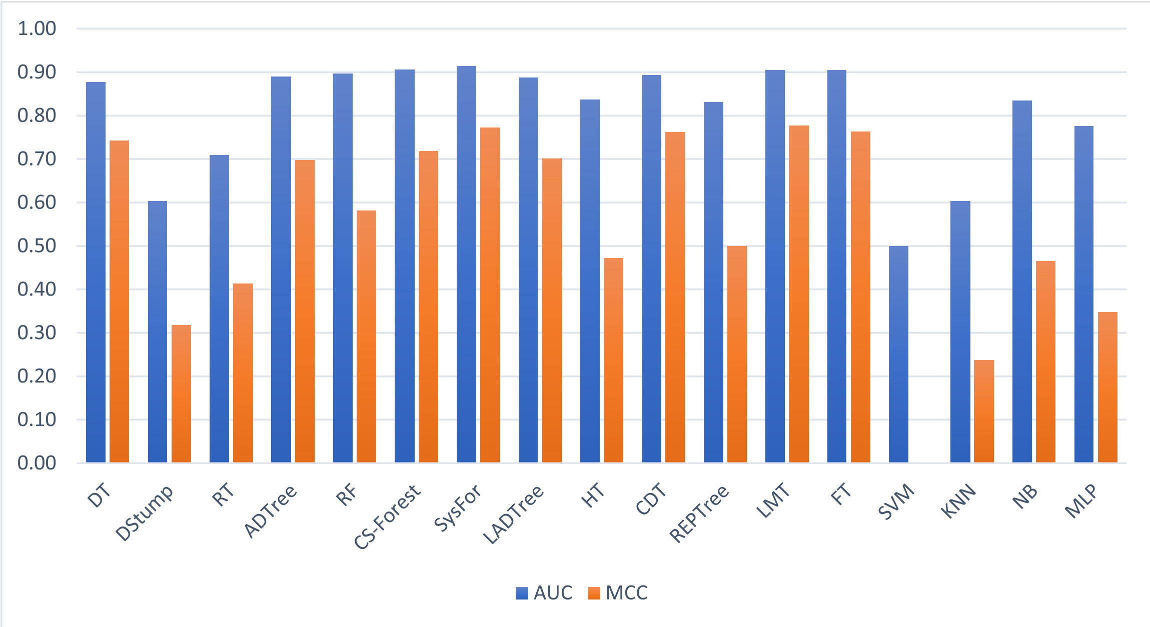

Fig. 2. Graphical representation and comparison of CCP performance of tree-based classifiers with prominent baseline classifiers on Dataset A.

Table 5
The CCP performance comparison of tree-based classifiers against baseline ML classifiers on Original Dataset B.

	Dataset B	AUC	MCC
	DT	0.498	?
Single Trees	DStump	0.496	?
	RT	0.508	0.009
	ADTree	0.506	?
	RF	0.513	0.003
Ensemble Trees	CS-Forest	0.525	0.009
	SysFor	0.499	?
	LADTree	0.507	-0.016
	HT	0.505	-0.015
	CDT	0.492	-0.005
Hybrid Trees	REPTree	0.498	-0.015
	LMT	0.500	?
	FT	0.500	?
	SVM	0.500	?
Base Classifiers	KNN	0.510	0.02
	NB	0.502	?
	MLP	0.489	-0.016

selected prominent ML classifiers such as SVM, KNN, NB, and MLP. This finding further supports the use of tree-based classifiers in CCP and ML tasks. Also, it was observed that there are other types of tree-based classifiers other than the DT and RF that can perform comparably well or better than the popularly used DT and RF models. Specifically, among the single-tree classifiers, the ADTree proved better than DT on Dataset A. For ensemble tree classifiers, SysFor and CSForest outperformed the famous RF model on Dataset A. By extension, the hybrid tree classifiers, particularly LMT and FT also had better CCP performances than DT and RF on Dataset A. From studies, these high-performing types of tree-based classifiers (ADTree, SysFor, CS-Forest, LMT, FT) are rare and not often used in CCP or ML tasks. However, these observed findings are based on the original churn datasets that are affected by the class imbalance problem. Perhaps if the class imbalance issue is addressed the CCP performances of the tree-based classifiers may change (enhanced). The following sub-section presents a scenario for the CCP performance comparison of tree-based classifiers against baseline ML classifiers on Balanced Datasets A and B.

Scenario 2: CCP performance comparison of tree-based classifiers against baseline ML classifiers on balanced datasets A and B

In this scenario, the CCP performances of the tree-based classifiers and the baseline classifiers on SMOTE-balanced customer churn datasets (Datasets A and B) are presented and analyzed. Specifically, the SMOTE technique (a data sampling method) is deployed to alleviate the latent class imbalance problem. As reported in existing studies, data sampling is a viable and feasible solution for the class

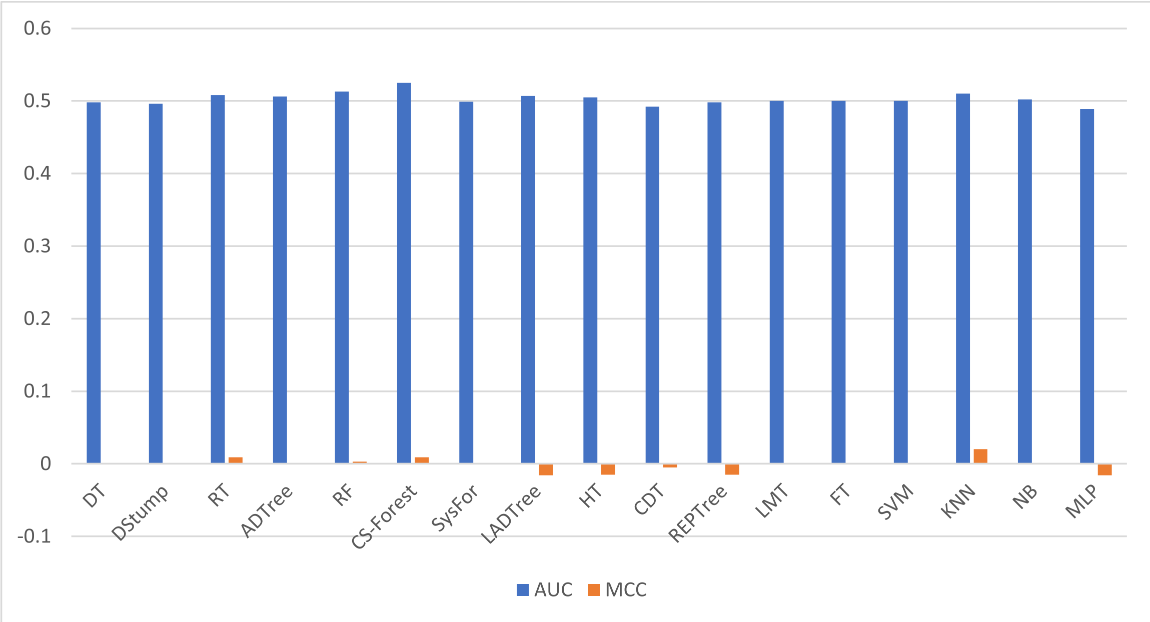

Fig. 3. Graphical representation and comparison of CCP performance of tree-based classifiers with prominent baseline classifiers on Dataset A.

imbalance problem. The deployed SMOTE is utilized on the customer churn datasets (Datasets A and B) to balance the frequency of the minority and majority class labels in each of the datasets. Tables 6 and 7 present the CCP performance comparison of tree-based classifiers against baseline ML classifiers on Balanced Datasets A and B, respectively.

From Table 6, it can be observed that there are improvements in the CCP performances of the tree-based classifiers and baseline classifiers on the balanced Dataset A. For instance, in comparison to the experimental results in Table 4, DT and RT had a (+2.74 % and +19.18 %) and (+14.82 % and +67.07 %) increment in their respective AUC and MCC values. A similar trend was observed in the CCP performances of the ensemble and hybrid tree-based classifiers on balanced Dataset A as SysFor (+7.11 % and +13.99 %), CS-Forest (+1.98 % and +10.03 %), LMT (+7.29 % and +12.22 %), FT (+7.73 % and +17.56 %), REPTree (+5.17 % and +34.40 %) recorded significant increments in their respective AUC and MCC values. Also, the CCP performances of the baseline classifiers improved with SVM (+56 % and +62.4 %) and KNN (+46.1 % and +223.6 %) recording significant increments in the AUC and MCC values. These observed findings indicated the performances of the investigated models improved on the balanced Dataset A.

As observed, the tree-based classifiers had high (not less than 0.8) and superior AUC and MCC values on balanced Dataset A. Specifically, SysFor still had the highest AUC value of 0.979 followed by FT (0.975), LMT (0.971), and CS-Forest (0.924). Concerning the categorization of the tree-based classifiers, among the single trees, ADTree had the highest AUC value of 0.915 followed by DT (0.900), RT (0.845), and DStump (0.650). In terms of the MCC values, ADTree (0.888) had the highest value followed by DT (0.852), RT (0.690), and DStump (0.346). Also, for ensemble tree classifiers, SysFor still outperformed other classifiers with AUC and MCC

Table 6
The CCP performance comparison of tree-based classifiers against baseline ML classifiers on Balanced Dataset A.

	Dataset A	AUC	MCC
Single Trees	DT	0.900	0.852
	DStump	0.650	0.346
	RT	0.845	0.690
	ADTree	0.915	0.888
Ensemble Trees	RF	0.843	0.843
	CS-Forest	0.924	0.790
	SysFor	0.979	0.880
	LADTree	0.886	0.691
Hybrid Trees	HT	0.887	0.683
	CDT	0.875	0.674
	REPTree	0.874	0.672
	LMT	0.971	0.872
	FT	0.975	0.897
Base Classifiers	SVM	0.780	0.624
	KNN	0.881	0.767
	NB	0.866	0.567
	MLP	0.824	0.654

Table 7
The CCP performance comparison of tree-based classifiers against baseline ML classifiers on Balanced Dataset B.

	Dataset B	AUC	MCC
Single Trees	DT	0.498	?
	DStump	0.571	0.257
	RT	0.888	0.721
	ADTree	0.799	0.542
Ensemble Trees	RF	0.846	0.606
	CS-Forest	0.916	0.610
	SysFor	0.911	0.694
	LADTree	0.794	0.534
Hybrid Trees	HT	0.813	0.594
	CDT	0.890	0.728
	REPTree	0.894	0.732
	LMT	0.896	0.752
	FT	0.973	0.883
Base Classifiers	SVM	0.793	0.586
	KNN	0.839	0.678
	NB	0.825	0.545
	MLP	0.868	0.659

values of 0.979 and 0.880, respectively. Similarly, CS-Forest recorded a comparable CCP performance with AUC and MCC values of 0.924 and 0.790, respectively. Both RF and LADTree had comparable performances but SysFor and CS-Forest were superior on the balanced Dataset A. Concerning the hybrid tree classifiers, LMT and FT recorded are still superior to REPTree, HT, and CDT in CCP performance. Although the trio REPTree, HT, and CDT gained significant enhancement in performance on the balanced dataset. In comparison to the baseline classifiers such as SVM, KNN, NB, and MLP, most of the experimented tree-based classifiers were superior in CCP performance to the baseline classifiers on the original Dataset A. Fig. 4 depicts the graphical representation of the CCP performance comparison of tree-based classifiers and the baseline classifiers on the balanced Dataset A.

In addition, Table 7 presents the CCP performances of the tree-based classifiers and the selected baseline classifiers on the balanced Dataset B. This analysis is essential since the initial CCP performances of these classifiers on the original Dataset B produced were average, which can be attributed to the data quality problem in Dataset B. Similar to the findings from Table 6, there were improvements in CCP performances of the investigated classifiers on the balanced Dataset B as presented in Table 7. Specifically, in comparison to the experimental results in Table 5, ADTree and RT had a (+57.9 % and +74.8 %) in AUC values and over an +100 % increment in their respective MCC values. A similar trend was observed in the CCP performances of the ensemble and hybrid tree-based classifiers on balanced Dataset B as SysFor, CS-Forest, LMT, FT, and REPTree recorded significant (over +100 %) increments in their respective AUC and MCC values. Furthermore, the CCP performances of the baseline classifiers improved with MLP and KNN recording significant increments in the AUC and MCC values. These observed findings indicated the performances of the investigated models

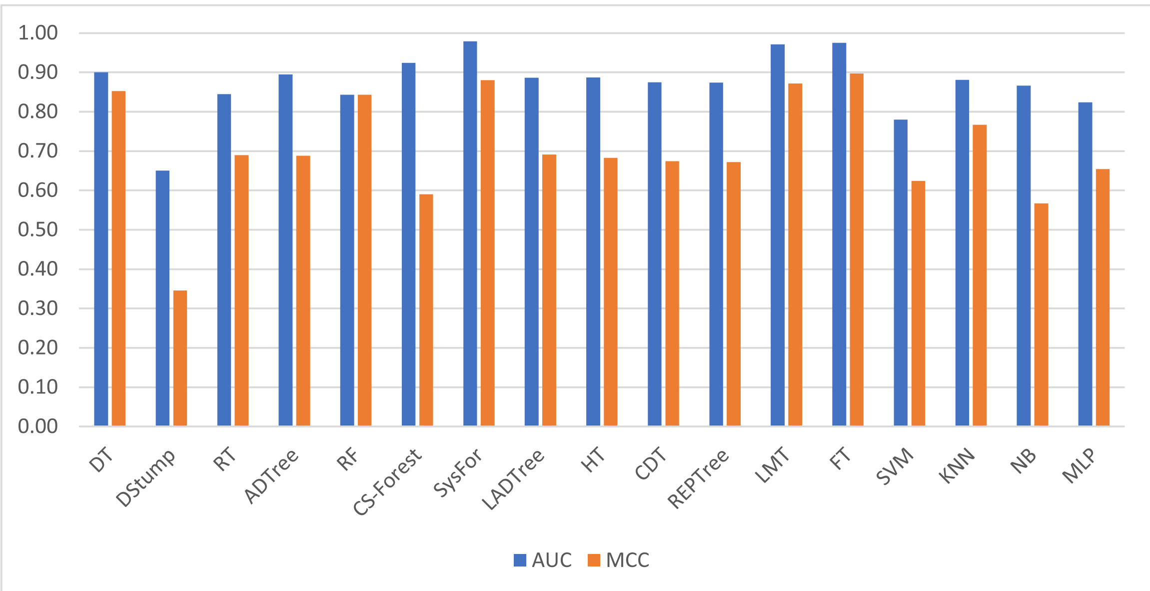

Fig. 4. Graphical representation and comparison of CCP performance of tree-based classifiers with prominent baseline classifiers on balanced Dataset A.

improved on the balanced Dataset B. These findings indicate that the datasets used for the training of models should be free of latent issues such as imbalance features and dimensionality as they negatively deteriorate the predictive performances of ML classifiers.

In terms of the CCP performance of the investigated classifiers, the tree-based classifiers recorded high (not less than 0.8) and superior AUC and MCC values on balanced Dataset B except for DT (AUC:0.498; MCC: ?) and DS (AUC: 0.571; MCC: 0.257). In this case, FT had the highest AUC value of 0.973 followed by CS-Forest (0.916), SysFor (0.911), and LMT (0.894). Concerning the categorization of the tree-based classifiers, among the single trees, RT had the highest AUC value of 0.888 followed by ADTree (0.799), DStump (0.571), and DT (0.498). In terms of the MCC values, RT (0.721) had the highest value followed by ADTree (0.542), and DStump (0.257). Also, for ensemble tree classifiers, CS-Forest was superior to other classifiers with AUC and MCC values of 0.916 and 0.610, respectively. Similarly, SysFor recorded a relatively comparable CCP performance to CS-Forest with AUC and MCC values of 0.911 and 0.694, respectively. Both RF and LADTree recorded competitive CCP performances but SysFor and CS-Forest were superior on the balanced Dataset B. Regarding the hybrid tree classifiers, LMT and FT recorded are still superior to REPTree, HT, and CDT in CCP performance. Although the trio REPTree, HT, and CDT gained significant enhancement in performance on the balanced Dataset B. In comparison to the baseline classifiers such as SVM, KNN, NB, and MLP, most of the experimented tree-based classifiers were superior in CCP performance to the baseline classifiers on the balanced Dataset B. Fig. 5 presents the graphical representation of the CCP performance comparison of tree-based classifiers and the baseline classifiers on the balanced Dataset B.

In summary, the observed empirical-based findings on the experimental scenarios have shown that other possible tree-based classifications algorithms such as CS-Forest, SysFor, LMT, FT, ADTree, HT, and REPTree can in most cases produce better CCP performances as compared to the popularly used DT and RF. Also, the use of single tree-based classifiers may not be appropriate as there are other forms of tree-based classifiers based on ensemble and hybrid concepts that can generate better results and still maintain the interpretability and simplicity characteristics of tree-based classifiers. It is worth noting that relying on the implicit operations of tree-based classifiers to handle noisy or imbalanced datasets may not be sufficient. Hence, appropriate data preprocessing methods should be used to improve the quality of the datasets to be used for ML processes. In this study, the use of SMOTE to address the latent class imbalance problem in experimented datasets (Datasets A and B) produced balanced datasets which subsequently generated high-performing tree-based classifier models.

However, it has been reported in some studies that the explicit use of ensemble methods with base classifiers not only amplifies the predictive performances of the base classifiers but also accommodates datasets with data quality problems [65,66,75,79]. Hence, the CCP performances of ensembled tree-based classifiers on both original and balanced customer churn datasets are presented in the subsequent sub-sections.

CCP performances of ensembled tree-based classification algorithms

In this sub-section, the CCP performances of ensembled tree-based classification algorithms on both the original and balanced customer churn datasets are presented and compared. Specifically, the CCP performances of Bagging, Boosting, Cascade, Rotation Forest (RotForest), Dagging, and RealBoosting (RBoost) homogeneous ensemble variations of each of the tree-based classification

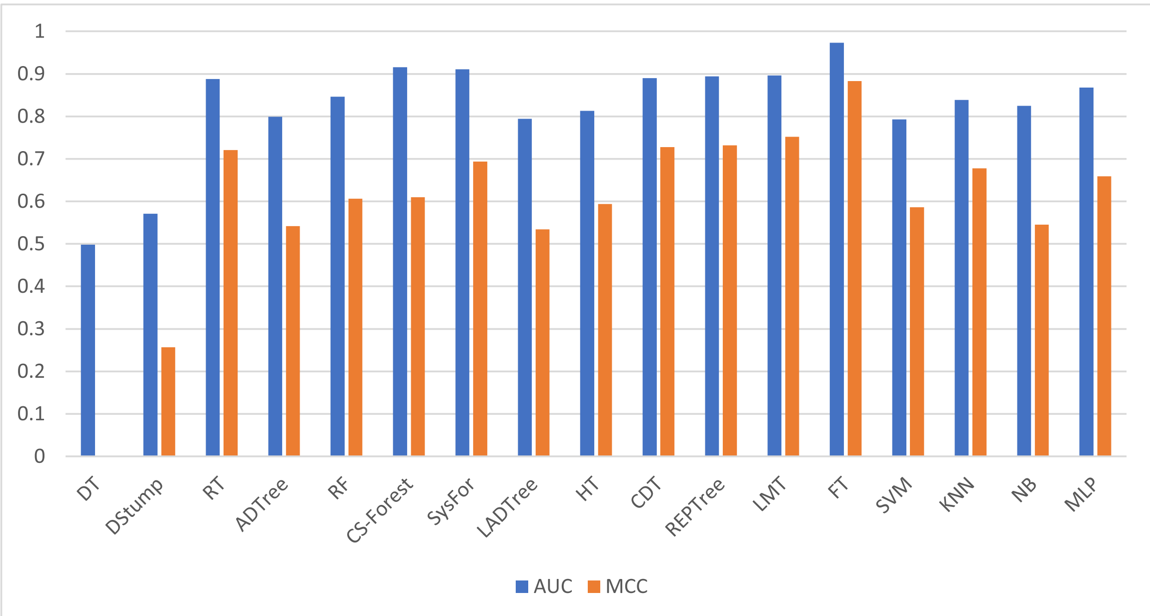

Fig. 5. Graphical representation and comparison of CCP Performance of tree-based classifiers with prominent baseline classifiers on Balanced Dataset B.

Table 8
The CCP performance comparison of ensembled tree-based classifiers against baseline ML classifiers on Original Dataset A.

	Dataset A	AUC							MCC						
		NE	Bagging	Boost	Cascade	RotForest	Dagging	RBoost	NE	Bagging	Boost	Cascade	RotForest	Dagging	RBoost
Single Trees	**DT**	0.876	0.913	0.903	0.901	0.913	0.897	0.898	0.742	0.786	0.736	0.748	0.748	0.537	0.722
	DStump	0.603	0.761	0.841	0.603	0.818	0.802	0.865	0.317	0.145	0.366	0.317	0.193	0.140	0.382
	RT	0.709	0.890	0.729	0.693	0.859	0.789	0.844	0.413	0.808	0.428	0.374	0.443	0.247	0.533
	ADTree	0.890	0.908	0.893	0.890	0.904	0.899	0.909	0.697	0.716	0.711	0.697	0.589	0.545	0.754
Ensemble Trees	RF	0.896	0.898	0.899	0.904	0.898	0.880	0.898	0.581	0.533	0.602	0.587	0.453	0.152	0.606
	CS-Forest	0.906	0.927	0.788	0.906	0.910	0.877	0.778	0.718	0.668	0.420	0.719	0.735	0.533	0.355
	SysFor	0.904	0.916	0.902	0.914	0.917	0.892	0.906	0.772	0.793	0.743	0.772	0.743	0.291	0.751
	LADTree	0.887	0.906	0.898	0.887	0.909	0.908	0.900	0.700	0.730	0.734	0.700	0.598	0.710	0.730
Hybrid Trees	HT	0.836	0.837	0.704	0.836	0.501	0.814	0.609	0.472	0.474	0.331	0.472	?	0.389	0.125
	CDT	0.893	0.912	0.863	0.891	0.909	0.884	0.815	0.762	0.789	0.631	0.756	0.670	0.528	0.476
	REPTree	0.831	0.851	0.789	0.824	0.908	0.558	0.802	0.500	0.470	0.426	0.534	0.671	?	0.424
	LMT	0.905	0.914	0.911	0.905	0.915	0.892	0.867	0.777	0.801	0.756	0.776	0.761	0.458	0.549
	FT	0.905	0.918	0.903	0.898	0.914	0.880	0.868	0.763	0.807	0.721	0.758	0.747	0.476	0.647
Base Classifiers	SVM	0.500	0.500	0.501	0.500	0.634	0.500	0.498	?	?	?	?	0.272	?	?
	KNN	0.603	0.676	0.603	0.603	0.661	0.757	0.697	0.237	0.257	0.237	0.232	0.176	0.202	0.267
	NB	0.834	0.833	0.778	0.887	0.602	0.804	0.756	0.465	0.470	0.465	0.749	0.201	0.677	0.286
	MLP	0.775	0.805	0.794	0.780	0.809	0.786	0.775	0.347	0.325	0.388	0.347	0.348	0.284	0.363

algorithms with or without the class imbalance problem are investigated. The analysis is to determine the degree of resilience of ensemble methods to the class imbalance problem. Also, to determine the effectiveness of ensembled tree-based classifiers with and without the class imbalance problem in CCP. In a similar pattern to the preceding section, the experimental result analysis is presented and discussed in the form of two scenarios. The first scenario presents the CCP performances of ensembled tree-based classifiers and the ensembled baseline ML classifiers on original Datasets A and B. For the second scenario, the CCP performances of ensembled tree-based classifiers are compared with the same ensembled baseline ML classifiers but on the balanced (SMOTE) Datasets A and B.

Scenario 1: CCP performance comparison of ensembled tree-based classifiers against ensembled baseline ML classifiers on original datasets A and B

As presented in Table 8, the CCP performances of investigated ensembled tree-based classifiers were compared among themselves and with the ensembled variants of the selected baseline classifiers on original Dataset A. From the results, the ensembled tree-based classifiers had relatively high and comparable AUC and MCC values on original Dataset A in most cases. That is, the AUC and MCC values of the ensembled tree-based models were superior in CCP performances compared to the tree-based models on Dataset A. Based on the AUC value, BaggedCS-Forest had the highest AUC value of 0.927 which is $a + 2.32$ % increment over CS-Forest only on the same Dataset A. Also, BaggedFT had a similar AUC increment of +1.44 % as compared with the AUC value of FT only. Similar increments in AUC values were observed with other ensembled tree-based models such as BaggedSysFor (+1.32 %), CascadeSysFor (+1.11 %), RotForestSysFor (+1.44 %), and RotForestFT (+1.1 %). In terms of the MCC values, BaggedRT (0.808) had the highest value followed by BaggedFT (0.807), BaggedLMT (0.801), and BaggedSysFor (0.793). However, it was observed that the MCC values of some of the ensembled tree-based models are below average even with a corresponding high AUC value. This disparity might be due to the inherent data quality problem present in the original Dataset A. For further analysis, the CCP performances of the tree based on their respective categories were analyzed. For the single ensemble tree-based classifiers, BaggedDT and RotForestDT had the highest AUC and MCC values, respectively. Other single-based classifiers such as BaggedADTree and RBoostADTree had comparable CCP performances in terms of the AUC and MCC values. In the case of the ensemble tree classifiers, BaggedCS-Forest outperformed other ensemble tree classifiers with an AUC value of 0.927. Likewise, BaggedSysFor, CascadeSysFor, and RotForestSysFor models also had a remarkable CCP performance on Dataset A with superior AUC and MCC values as compared with ensembled RF models. It is worth noting that the CCP performances of RF, CS-Forest, SysFor, and LADTree classifiers are high but their ensembled variants are in most cases superior in terms of their AUC values. For the hybrid tree classifiers, BaggedLMT and BaggedFT recorded comparable results with AUC values of 0.914 and 0.918 and MCC values of 0.801 and 0.807, respectively. Other ensembled hybrid tree classifiers such as BaggedCDT and RotForestREPTree also had comparable CCP performances on Dataset A. In comparison to the baseline classifiers such as SVM, KNN, NB, and MLP, most of the experimented ensemble tree-based classifiers were superior in CCP performance to the baseline classifiers on the original Dataset A.

Figs. 6 and 7 present the graphical representation of the AUC and MCC values comparison of ensembled tree-based classifiers and the baseline classifiers on original Dataset A.

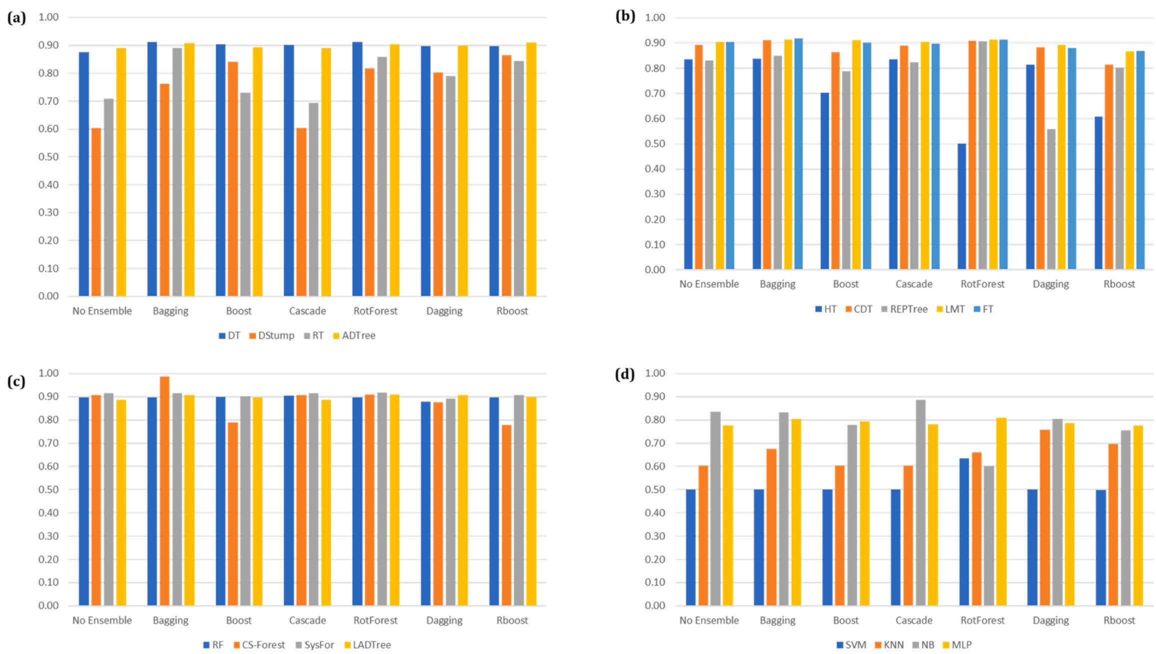

Fig. 6. AUC values of ensembled variants of investigated classifiers on original Dataset A (a) Single Tree Classifiers (b) Hybrid Tree Classifiers (c) Ensemble Tree Classifiers (d) Baseline Classifiers.

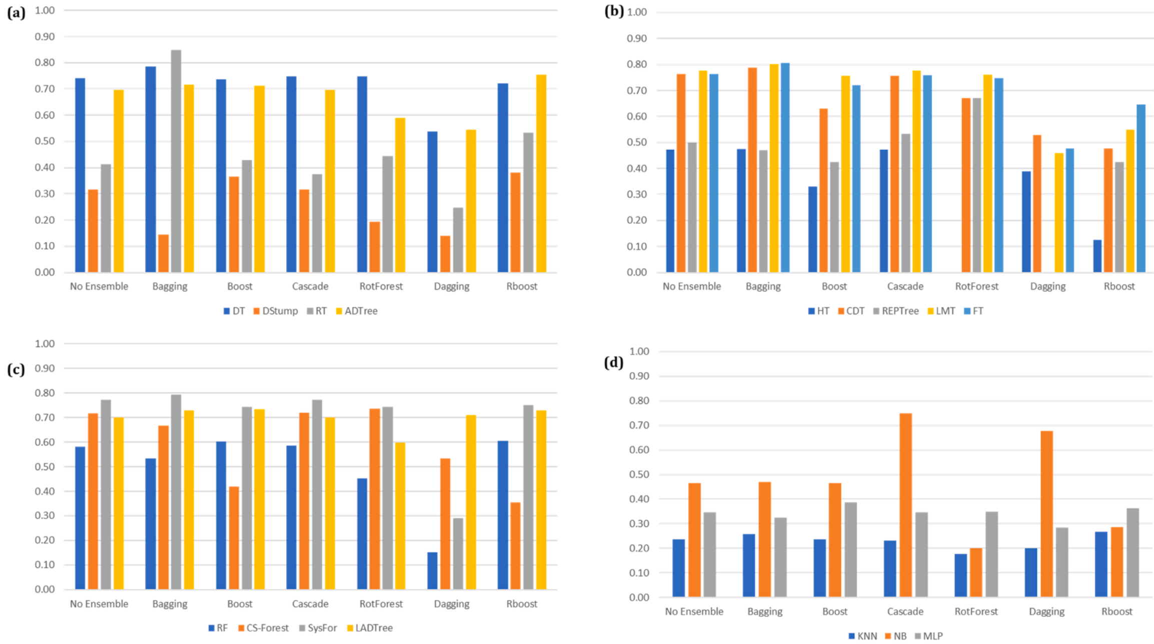

Fig. 7. MCC values of ensembled variants of investigated classifiers on original Dataset A (a) Single Tree Classifiers (b) Hybrid Tree Classifiers (c) Ensemble Tree Classifiers (d) Baseline Classifiers.

Furthermore, Table 9 presents the CCP performances of investigated ensembled tree-based classifiers and the ensembled variants of the selected baseline classifiers on original Dataset B. On average, the CCP performances of ensembled tree-based models were still poor and not significantly better than the ordinary tree-based models on Dataset B. This finding is related to the initial observation of poor CCP performances of ML models on Datasets B with the data quality problem such as the class imbalance playing a major role.

The low AUC and MCC values of the ensembled tree-based models indicated that the tree-models recorded poor performances. For instance, RBoostREPTree had the highest AUC and MCC values of 0.545 and 0.023 while BoostedFT recorded AUC and MCC values of 0.534 and 0.016, respectively. This observation further showed that the deployment of ensemble methods as a solution for the class imbalance may not necessarily be true for all cases as suggested in existing studies. A similar outcome was observed in the CCP performances of the ensembled selected baseline ML method. RotForestKNN had the highest AUC value of 0.511 which is an indicator that the model is not able to properly perform the CCP task. Figs. 8 and 9 present the graphical representation of the AUC and MCC values comparison of ensembled tree-based classifiers and the baseline classifiers on original Dataset B.

In summary, the experimental results as presented in Tables 8 and 9 have shown that the ensemble methods have little or no resilience for the class imbalance problem and may not necessarily be used as a viable solution for the class imbalance problem. Although the investigated ensembled tree-based classifiers had good CCP performances on Dataset A, the same cannot be said for Dataset B. The observed findings from Dataset A (Table 8) still further support our research aim that tree-based classifiers are appropriate for CCP tasks and there exist tree-based classifiers that are better than the popularly used DT and RF applicable for CCP tasks. However, the class imbalance problem deteriorates the CCP performances of the investigated tree-based classifiers and deploying only ensemble methods may not necessarily be applicable as a solution to the class imbalance problem in most cases. However, the combination of a data sampling method (in this case SMOTE technique) and ensemble methods may produce CCP models that not only have superior predictive performances but can also address the latent class imbalance problems in the experimented customer churn datasets. Hence, the succeeding sub-section showcases a scenario for the CCP performance comparison of ensembled tree-based classifiers and ensembled baseline ML classifiers on Balanced (SMOTE) Datasets A and B.

Scenario 2: CCP performance comparison of ensembled tree-based classifiers and ensembled baseline ML classifiers on Balanced Datasets A and B

In this scenario, the CCP performances of the ensembled tree-based classifiers and ensembled baseline classifiers on SMOTE-balanced customer churn datasets (Datasets A and B) are presented and analyzed. For the class imbalance problem, the SMOTE technique was deployed, being a known solution. Specifically, the deployed SMOTE is utilized on the customer churn datasets (Datasets A and B) to balance the frequency of the minority and majority class labels in each of the datasets. Tables 10 and 11 present the CCP performance comparison of the ensembled tree-based classifiers and the ensembled baseline ML classifiers on Balanced Datasets A and B, respectively. Findings from these analyses will validate the applicability of ensemble and data sampling methods as a solution for the class imbalance problem.

As shown in Table 10, it can be observed that there are improvements in the CCP performances of the ensembled tree-based classifiers and ensembled baseline classifiers on the balanced Dataset A in most of the cases. Overall, BaggedLMT and RotForestFT

Table 9
The CCP performance comparison of tree-based classifiers against baseline ML classifiers on Original Dataset B.

	Dataset A	AUC							MCC						
		NE	Bagging	Boost	Cascade	RotForest	Dagging	RBoost	NE	Bagging	Boost	Cascade	RotForest	Dagging	RBoost
Single Trees	**DT**	0.498	0.512	0.536	0.499	0.510	0.498	0.522	**?**	?	0.004	?	?	?	-0.003
	DStump	0.496	0.507	0.509	0.487	0.492	0.531	0.504	?	?	?	?	?	?	?
	RT	0.508	0.516	0.518	0.499	0.510	0.497	0.503	0.009	0.011	0.007	-0.010	-0.017	?	-0.026
	ADTree	0.506	0.497	0.512	0.501	0.507	0.518	0.504	?	?	?	-0.010	?	?	-0.008
Ensemble Trees	RF	0.513	0.521	0.516	0.507	0.520	0.489	0.521	0.003	0.001	0.016	0.009	-0.008	?	-0.015
	CS-Forest	0.525	0.511	0.531	0.515	0.499	0.499	0.513	0.009	-0.002	0.009	0.008	-0.019	?	0.002
	SysFor	0.499	0.493	0.504	0.499	0.500	0.501	0.494	?	?	-0.014	?	?	?	0.003
	LADTree	0.507	0.502	0.511	0.500	0.497	0.518	0.497	-0.016	?	0.002	-0.013	?	?	0.012
Hybrid Trees	HT	0.505	0.488	0.518	0.504	0.493	0.520	0.492	-0.015	?	-0.012	0.002	?	?	0.002
	CDT	0.492	0.527	0.510	0.509	0.493	0.489	0.525	-0.005	?	0.005	-0.005	?	?	-0.016
	REPTree	0.498	0.505	0.522	0.500	0.492	0.505	0.542	-0.015	?	-0.001	-0.010	?	?	0.023
	LMT	0.500	0.515	0.523	0.500	0.500	0.506	0.519	?	0.012	0.007	?	?	0.012	0.026
	FT	0.500	0.497	0.534	0.498	0.500	0.495	0.518	?	?	0.016	?	?	?	0.026
Base Classifiers	SVM	0.500	0.500	0.486	0.500	0.500	0.502	0.505	?	?	?	?	?	?	0.007
	KNN	0.510	0.494	0.510	0.498	0.527	0.510	0.496	0.020	0.011	0.020	-0.003	0.039	-0.010	-0.004
	NB	0.502	0.511	0.483	0.484	0.499	0.520	0.481	?	?	?	-0.011	0.011	?	-0.016
	MLP	0.489	0.506	0.517	0.482	0.471	0.502	0.513	-0.016	?	-0.015	?	?	?	0.027

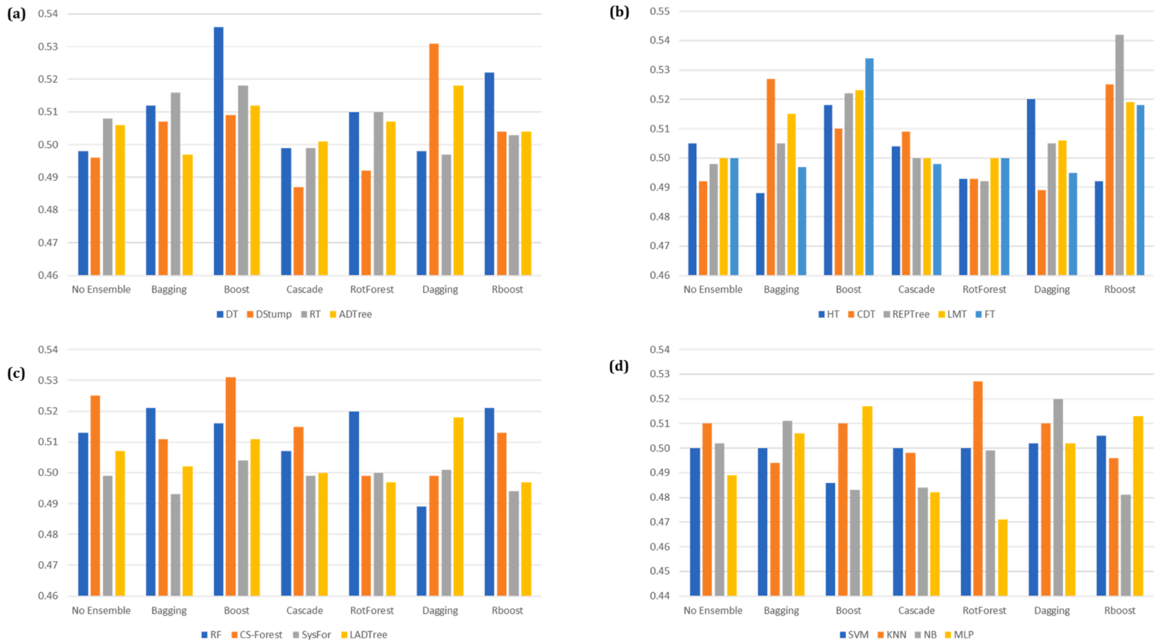

Fig. 8. AUC values of ensembled variants of investigated classifiers on original Dataset B (a) Single Tree Classifiers (b) Hybrid Tree Classifiers (c) Ensemble Tree Classifiers (d) Baseline Classifiers.

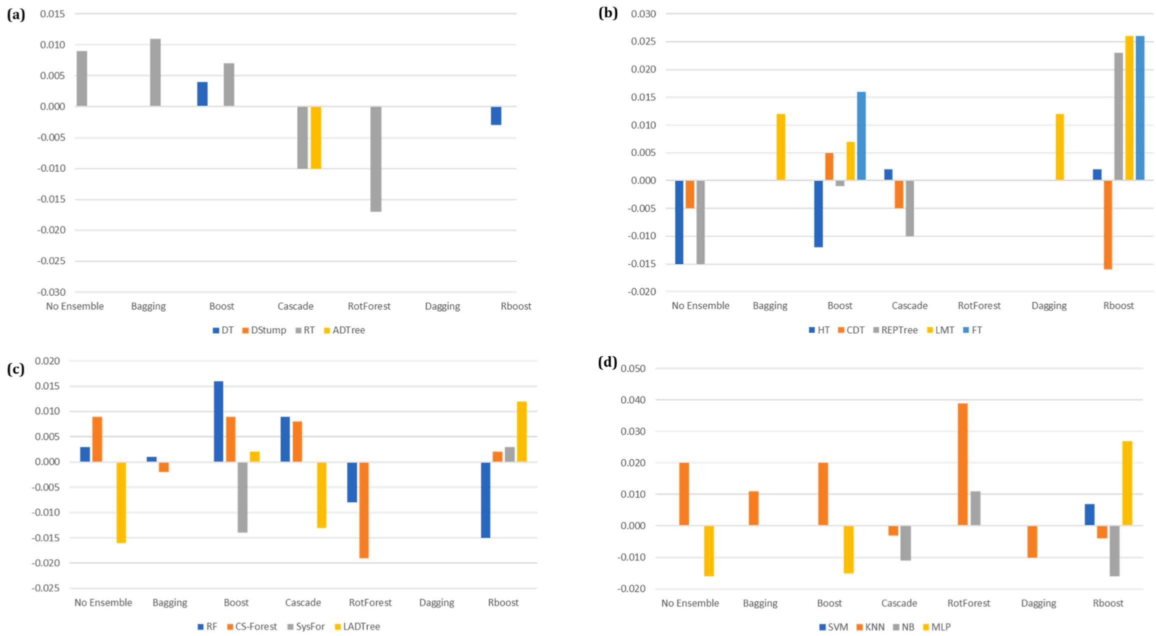

Fig. 9. MCC values of ensembled variants of investigated classifiers on original Dataset B (a) Single Tree Classifiers (b) Hybrid Tree Classifiers (c) Ensemble Tree Classifiers (d) Baseline Classifiers.

on Balanced Dataset A had the highest AUC values of 0.986 and MCC values of 0.910 and 0.908, respectively. Specifically, BaggedLMT and RotForestFT on Balanced Dataset A had a (+1.54 % and +4.36 %) and (+1.13 % and +1.023 %) increment in their respective AUC and MCC values as compared to their respective performances on the Original Dataset A. This is an indicator that the combination of the ensemble and data sampling methods for the class imbalance problem is viable as the CCP performances of most of the investigated ensembled tree-based and ensembled baseline classifiers improved. In terms of the tree categories, for the single tree classifiers, BoostedADTree (0.979) and RBoostADTree (0.924) had the highest AUC and MCC values, respectively. Also, the CCP performances of

Table 10
The CCP performance comparison of ensembled tree-based classifiers and ensembled baseline ML classifiers on Balanced Dataset A.

	Dataset A	AUC							MCC						
		NE	Bagging	Boost	Cascade	RotForest	Dagging	RBoost	NE	Bagging	Boost	Cascade	RotForest	Dagging	Rboost
Single Trees	**DT**	0.950	0.960	0.963	0.960	0.965	0.925	0.898	0.852	0.900	0.909	0.889	0.907	0.669	0.912
	Dstump	0.650	0.715	0.855	0.650	0.890	0.772	0.865	0.346	0.356	0.555	0.346	0.614	0.401	0.674
	RT	0.845	0.962	0.859	0.854	0.967	0.931	0.844	0.690	0.821	0.713	0.704	0.849	0.725	0.830
	ADTree	0.895	0.944	0.979	0.895	0.974	0.950	0.909	0.688	0.759	0.883	0.688	0.855	0.758	0.924
Ensemble Trees	RF	0.843	0.971	0.973	0.974	0.974	0.949	0.898	0.843	0.846	0.847	0.854	0.855	0.757	0.875
	CS-Forest	0.924	0.926	0.948	0.924	0.980	0.958	0.978	0.590	0.625	0.763	0.581	0.837	0.431	0.745
	SysFor	0.979	0.982	0.985	0.979	0.985	0.932	0.906	0.880	0.898	0.921	0.876	0.908	0.625	0.909
	LADTree	0.886	0.941	0.976	0.886	0.972	0.949	0.900	0.691	0.750	0.871	0.691	0.844	0.758	0.907
Hybrid Trees	HT	0.887	0.904	0.900	0.885	0.882	0.862	0.609	0.683	0.694	0.693	0.681	0.495	0.556	0.548
	CDT	0.875	0.921	0.953	0.866	0.982	0.870	0.815	0.674	0.707	0.774	0.659	0.896	0.583	0.810
	REPTree	0.874	0.924	0.941	0.870	0.982	0.817	0.802	0.672	0.729	0.758	0.675	0.894	0.805	0.762
	LMT	0.971	0.986	0.984	0.969	0.985	0.952	0.867	0.872	0.910	0.909	0.873	0.906	0.763	0.836
	FT	0.975	0.985	0.984	0.972	0.986	0.963	0.868	0.897	0.919	0.907	0.887	0.908	0.811	0.742
Base Classifiers	SVM	0.780	0.818	0.792	0.783	0.909	0.859	0.498	0.624	0.600	0.647	0.629	0.766	0.629	0.533
	KNN	0.881	0.931	0.881	0.865	0.921	0.928	0.697	0.767	0.762	0.767	0.735	0.774	0.700	0.767
	NB	0.866	0.866	0.873	0.866	0.837	0.862	0.866	0.567	0.567	0.586	0.567	0.457	0.557	0.567
	MLP	0.824	0.909	0.926	0.886	0.920	0.894	0.812	0.654	0.665	0.711	0.650	0.698	0.634	0.707

Table 11
The CCP performance comparison of tree-based classifiers against baseline ML classifiers on Balanced Dataset B.

	Dataset B	AUC NE	Bagging	Boost	Cascade	RotForest	Dagging	RBoost	MCC NE	Bagging	Boost	Cascade	RotForest	Dagging	RBoost
Single Trees	**DT**	0.498	0.910	0.826	0.787	0.850	0.890	0.917	?	0.767	0.755	0.748	0.734	0.656	0.742
	DStump	0.496	0.577	0.720	0.575	0.857	0.635	0.801	0.257	0.257	0.261	0.257	0.572	0.264	0.528
	RT	0.508	0.939	0.918	0.879	0.950	0.914	0.911	0.721	0.792	0.753	0.713	0.838	0.721	0.733
	ADTree	0.506	0.922	0.895	0.799	0.930	0.834	0.921	0.542	0.546	0.663	0.535	0.731	0.524	0.724
Ensemble Trees	RF	0.513	0.945	0.930	0.943	0.941	0.916	0.943	0.806	0.801	0.799	0.804	0.737	0.706	0.802
	CS-Forest	0.525	0.927	0.908	0.913	0.947	0.856	0.887	0.610	0.690	0.660	0.603	0.728	0.413	0.587
	SysFor	0.499	0.927	0.921	0.910	0.947	0.842	0.902	0.694	0.752	0.740	0.725	0.830	0.569	0.667
	LADTree	0.507	0.806	0.854	0.794	0.923	0.826	0.868	0.534	0.545	0.593	0.530	0.714	0.545	0.626
Hybrid Trees	HT	0.505	0.845	0.863	0.810	0.864	0.530	0.729	0.594	0.619	0.638	0.590	0.588	0.604	0.323
	CDT	0.492	0.926	0.929	0.887	0.945	0.859	0.920	0.728	0.761	0.763	0.713	0.817	0.587	0.744
	REPTree	0.498	0.928	0.930	0.887	0.947	0.860	0.920	0.732	0.767	0.763	0.713	0.814	0.588	0.741
	LMT	0.500	0.934	0.932	0.895	0.948	0.865	0.867	0.752	0.771	0.760	0.742	0.828	0.602	0.653
	FT	0.500	0.928	0.934	0.883	0.947	0.842	0.877	0.883	0.771	0.766	0.735	0.821	0.567	0.716
Base Classifiers	SVM	0.500	0.805	0.852	0.793	0.795	0.830	0.794	0.586	0.586	0.587	0.586	0.586	0.584	0.586
	KNN	0.510	0.902	0.839	0.824	0.918	0.914	0.835	0.678	0.689	0.678	0.648	0.760	0.691	0.674
	NB	0.825	0.827	0.827	0.826	0.865	0.827	0.826	0.545	0.541	0.546	0.548	0.584	0.531	0.545
	MLP	0.868	0.887	0.904	0.868	0.886	0.880	0.911	0.659	0.665	0.711	0.646	0.656	0.650	0.725

the DT were better than RT and Dstump in most cases. This can be attributed to the deployment of the ensemble methods and the data sampling technique. Likewise, a similar trend was observed in the CCP performances of the ensemble tree-based classifiers on balanced Dataset A as RotForestSysFor (0.985) and BaggedSysFor (0.982) recorded the highest AUC values. In addition, the ensembled variants LADTree, CS-Forest, and SysFor all had improvements in their respective AUC values that in most cases superior to the AUC value of ensembled RF. As for the hybrid tree-based classifiers, their initial (no ensemble variations) AUC and MCC values on balanced Dataset A showed good predictive performances. However, the deployment of their respective homogeneous ensemble variants further improved their CCP performances with LMT (AUC: 0.986; MCC: 0.910) and FT (AUC: 0.986; MCC: 0.908) superior to other hybrid tree models. Also, the CCP performances of the ensembled baseline classifiers improved in most cases except for the RBoost variants. Specifically, BaggedKNN and BoostMLP are among the top performers with significant increments in the AUC and MCC values. Figs. 10 and 11 present the graphical representation of the AUC and MCC values comparison of ensembled tree-based classifiers and the ensembled baseline classifiers on balanced Dataset A.

Furthermore, Table 11 presents the CCP performances of the ensembled tree-based classifiers and ensembled baseline classifiers on the balanced Dataset B. Like the observed findings in Table 9, there were improvements in the CCP performances of the ensembled tree-based classifiers and ensembled baseline classifiers on the balanced Dataset B. Overall, RotForestRT on Balanced Dataset B had the highest AUC values of 0.950 and MCC value of 0.838. This is followed by RotForestLMT, RotForestFT, and RotForestREPTree with AUC values of 0.948, 0.947, and 0.947, respectively. This observed finding validates the outcome of the combination of the ensemble and data sampling methods for the class imbalance problem and this is due to the improvements in the CCP performances of most of the investigated ensembled tree-based classifiers. For the single tree classifiers, RotForestRT had the highest AUC (0.950) and MCC (0.838) values, respectively. Also, the CCP performances of the ensemble variants of ADTree were better than the ensemble variants of DT. However, ensemble variants of Dstump recorded the least performances when compared with other investigated single tree-based classifiers. In the case of the CCP performance of ensemble tree-based classifiers on balanced Dataset B, RotForestSysFor (0.947) and RotForestCS-Forest (0.947) recorded the highest AUC values. In addition, the ensembled variants of LADTree and RF all had improvements in their respective AUC values that in most cases. As for the hybrid tree-based classifiers, the homogeneous ensemble variants of these hybrid tree models on balanced Dataset B showed remarkable CCP performances with RotForestLMT, RotForestFT, and RotForestREPTree recording superior AUC values to other homogeneous ensemble hybrid tree models. Also, the baseline classifier specifically homogeneous ensemble variants SVM and KNN had improved CCP performances on the balanced Dataset B. The graphical representations of the CCP performances of the ensembled tree-based classifiers and the ensembled baseline classifiers on balanced Dataset B are depicted in Figs. 12 and 13.

In summary, the experimental results as presented in Tables 10 and 11 have shown that the combination of ensemble methods with a data sampling method can be a solution to the class imbalance problem and may it can also improve the CCP performances of the applied ML classifier. In this scenario, the investigated ensembled tree-based classifiers had competitive CCP performances on both balanced Datasets A and B. As for the tree-based classifiers, the experimental findings from balanced Datasets A and B (Tables 10 and 11) still further support our research aim that tree-based classifiers are appropriate for CCP tasks and there exist tree-based classifiers that are better than the popularly used DT and RF applicable for CCP tasks. Similarly, it was observed that the CCP performances of the

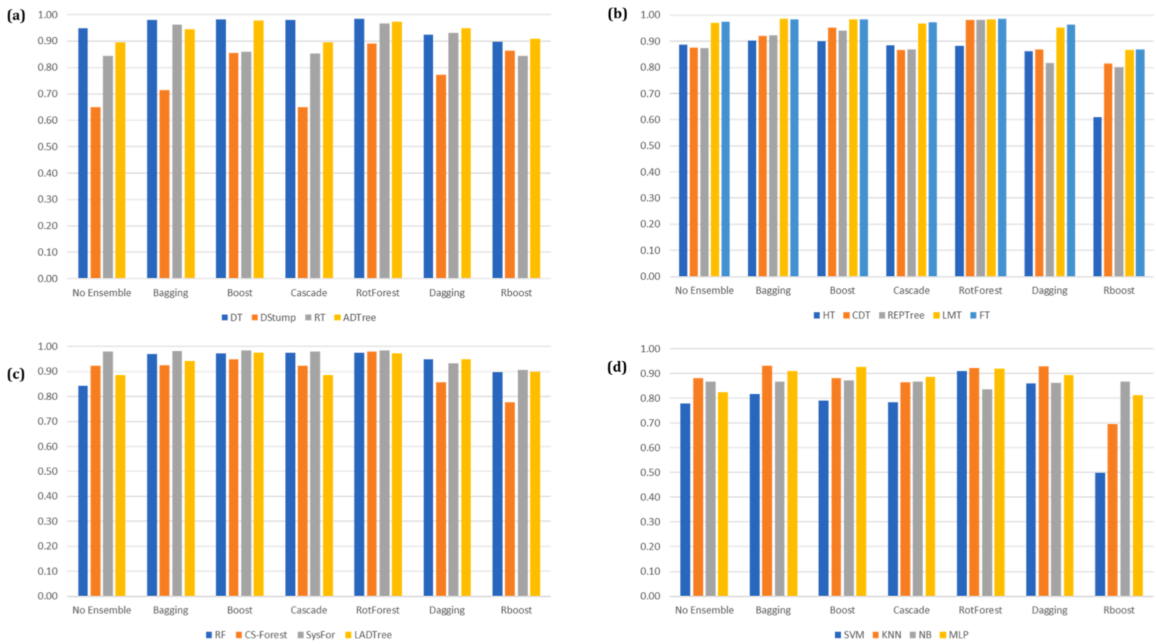

Fig. 10. AUC values of ensembled variants of investigated classifiers on Balanced Dataset A (a) Single Tree Classifiers (b) Hybrid Tree Classifiers (c) Ensemble Tree Classifiers (d) Baseline Classifiers.

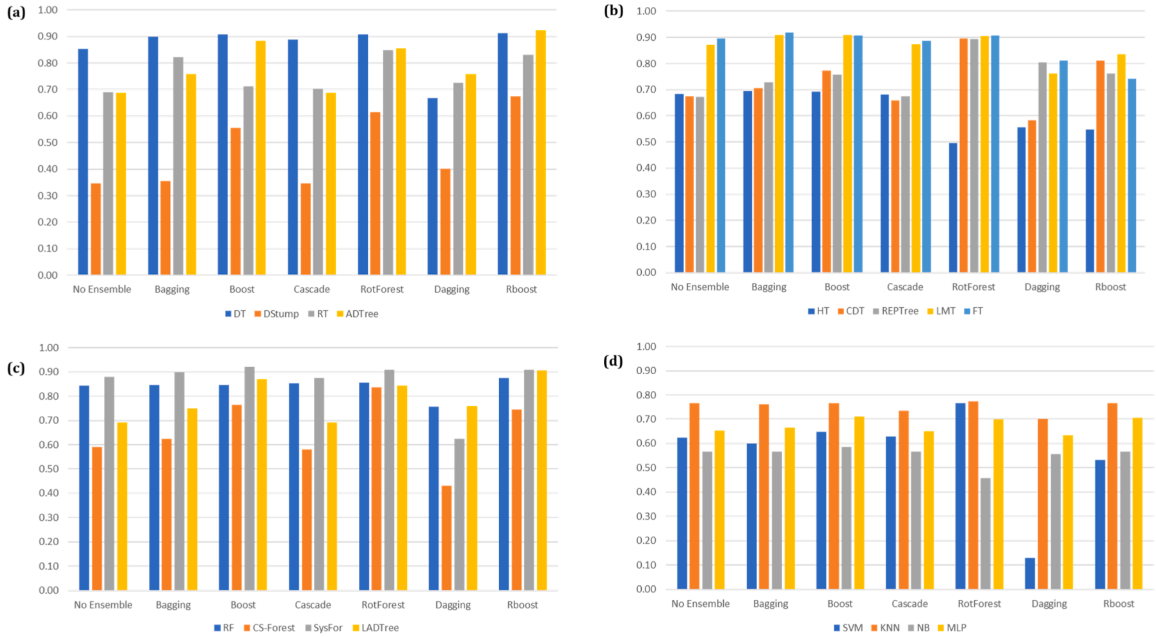

Fig. 11. MCC values of ensembled variants of investigated classifiers on Balanced Dataset A (a) Single Tree Classifiers (b) Hybrid Tree Classifiers (c) Ensemble Tree Classifiers (d) Baseline Classifiers.

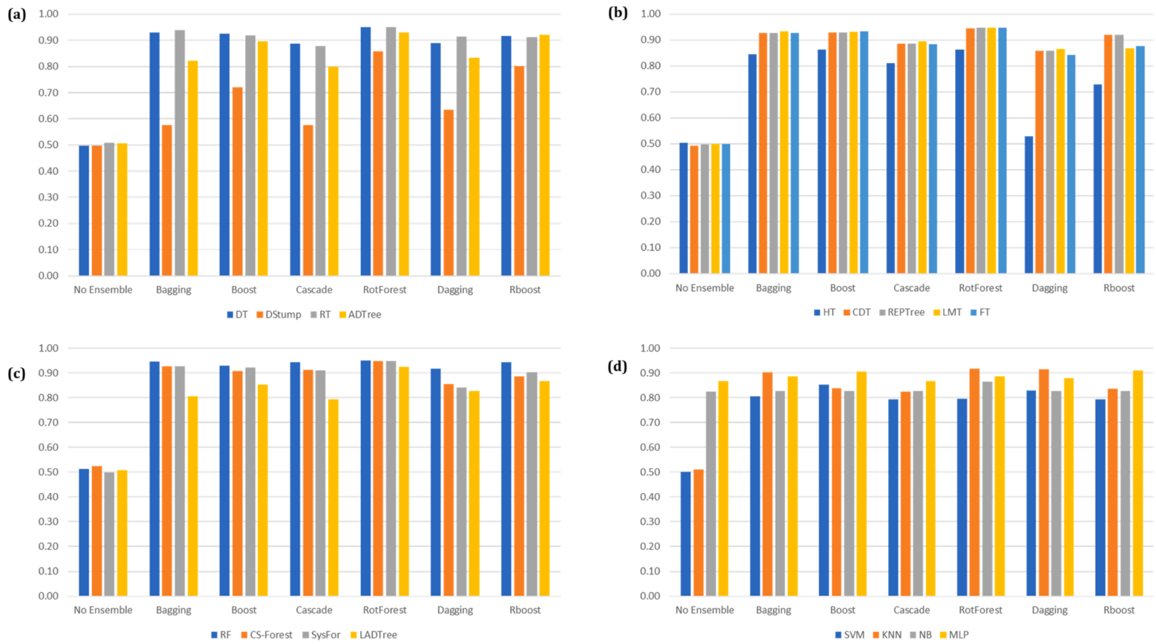

Fig. 12. AUC values of ensembled variants of investigated classifiers on Balanced Dataset B (a) Single Tree Classifiers (b) Hybrid Tree Classifiers (c) Ensemble Tree Classifiers (d) Baseline Classifiers.

homogeneous ensemble variants of the tree-based classifiers were better than the CCP performances of the individual tree-based classifiers on the balanced Datasets A and B. In addition, it was also observed that the CCP performances of the homogeneous ensemble variants of the tree-based classifiers on the balanced Datasets A and B were better than the CCP performances of homogeneous ensemble variants of the tree-based classifiers on the original Datasets A and B. Specifically, the selection of ensemble methods with applicable data sampling method can alleviate the class imbalance problem and produce efficient ML models with high prediction performances. Using individual ensemble methods or data sampling techniques for addressing the class imbalance problem may not be

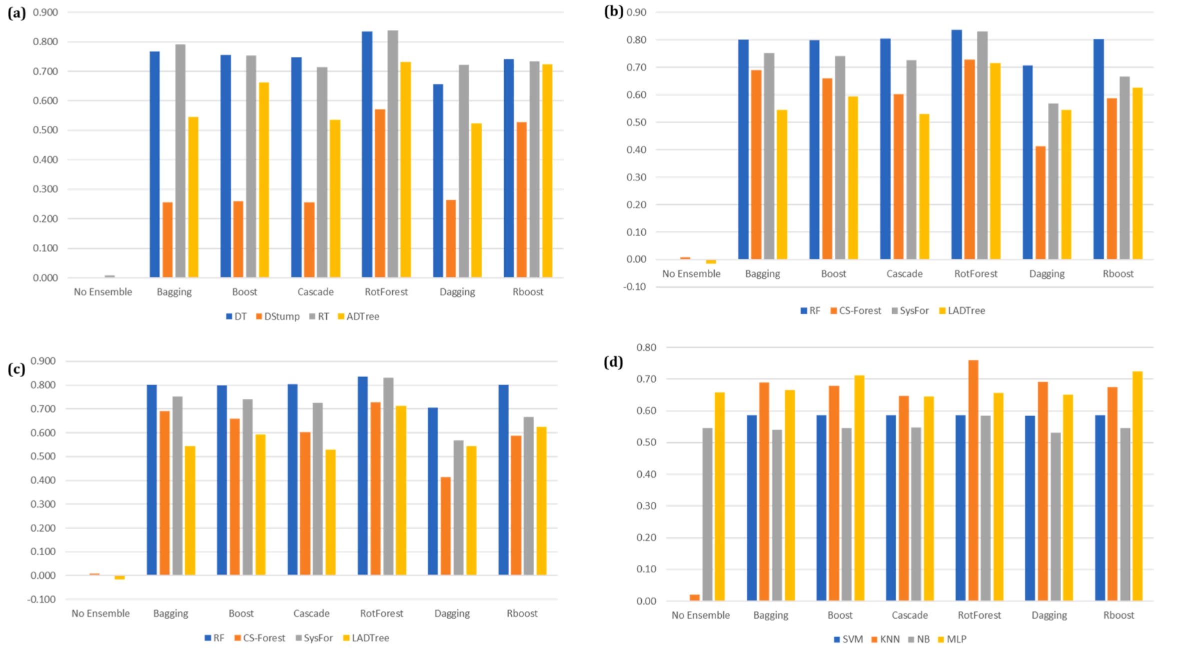

Fig. 13. AUC values of ensembled variants of investigated classifiers on Balanced Dataset B (a) Single Tree Classifiers (b) Hybrid Tree Classifiers (c) Ensemble Tree Classifiers (d) Baseline Classifiers.

sufficient and applicable in all cases. Therefore, it is suggested that the combination of ensemble methods with appropriate data sampling approaches should be deployed when encountering data quality problems in ML tasks.

For a standard CCP performance evaluation, the CCP performances of HMSE and S-HMSE are contrasted with those of the current CCP models with varied computational characteristics on the same Datasets A and B. That is, the CCP performances of the proposed methods are compared with the current ensemble, hybrid, and advanced DL-based models on the same dataset

For a generalizable CCP performance assessment, in the succeeding section, the CCP performances of top performers among the investigated tree-based classifiers and recent CCP models with varying computational features on Datasets A and B are compared. The essence of the comparison is to ascertain and validate the effectiveness of the top-performing tree-based classifiers with the existing sophisticated ensemble, hybrid, and DL-based models on the same customer churn datasets.

CCP performances of best performing tree-based classification algorithms against existing CCP models

The CCP performances of the top-performing investigated tree-based models as well as the recent CCP ML and DL-based model on Datasets A and B are presented in Tables 12 and 13, respectively.

From the preceding experimental results, it was observed that SMOTE-BaggedLMT and SMOTE-RotForestFT are the top performers among the investigated tree-based models on Dataset A while SMOTE-RotForestRT and SMOTE-RotForestLMT are the top performers for Dataset B. Consequently, the CCP performances of these two models will be compared with performances of the existing rule,

Table 12
The CCP performance of HMSE and S-HMSE against Existing CCP models on Dataset A.

CCP Models	Accuracy (%)	AUC	F-measure	MCC
SMOTE-BaggedLMT	95.50	0.986	0.955	0.910
SMOTE-RotForestFT	95.41	0.986	0.948	0.908
Tavassoli and Koosha [80] (BNNGA)	86.81	-	0.688	-
Ahmad et al. [9] (SNA + XGBOOST)	-	0.933	-	-
Jain et al. [15] (CNN+VAE)	90.00	-	0.930	-
Saghir et al. [81] (BaggedMLP)	94.15	-	0.874	-
Jain et al. [20] (LogitBoost)	85.24	0.717	0.810	0.160
Jeyakarthic and Venkatesh [82] (P-AGBPNN)	91.71	-	0.951	
Praseeda and Shivakumar [83] (PFLICM)	95.41	-	-	-
Dalli [13] (Hyper-Parameterized DL with RMSProp)	86.50	-	-	-
Bilal et al. [73] (K-med+GBT+DT+DL+Voting)	92.40	-	0.662	-
Bilal et al. [73] (K-med+GBT+DT+DL+Stacking)	92.40	-	0.717	-
Bilal et al. [73] (K-med+GBT+DT+DL+Adaboost)	92.43	-	0.718	-
Bilal et al. [73] (K-med+GBT+DT+DL+Bagging)	92.41	-	0.664	-

Table 13
The CCP performance of HMSE and S-HMSE against Existing CCP models on Dataset B.

CCP Models	Accuracy (%)	AUC	F-measure	MCC
SMOTE-RotForestRT	91.88	0.950	0.919	0.838
SMOTE-RotForestLMT	91.39	0.948	0.914	0.828
Tavassoli and Koosha [80](BBNGA)	77.50	-	0.773	-
Saghir et al. [81] (Bagging)	80.80	-	0.784	-
Shaaban et al. [84] (SVM)	83.70	-	-	-
Kumar and Kumar [39]	84.26	-	0.900	-
Long Short Term Memory (LSTM) [73]	90.04	-	0.821	-
Gated Recurrent Unit (GRU) [73]	90.10	-	0.846	-
CNN [73]	89.80	-	0.826	-

ensemble, hybrid, and DL models on same Dataset A.

As shown in Table 12, the CCP performance of SMOTE-BaggedLMT and SMOTE-RotForestFT models are compared to existing CCP models such as [9,13,15,20,80–83], and [73] on Dataset A. In contrast, current CCP solutions consist of ensembles, hybrids, and advanced DL models. As an example, [80] designed a hybrid ensemble (BNNGA) method for CCP that recorded a prediction accuracy value of 86.81 % and an F-measure value of 0.688. Likewise, [81] developed a BaggedMLP for CCP performance with accuracy and F-measure values of 94.15 % and 0.874, respectively. Also, [20] suggested the amplification of logistic regressing with a boosting ensemble method for CCP. Regardless, the CCP performances of SMOTE-BaggedLMT and SMOTE-RotForestFT models outperformed these ensemble-based CCP models. Using hybrid methods, [9] hybridized Social Network Analysis (SNA) and XGBoost for CCP. Similarly, [20] combined CNN with a Variable Auto-Encoder (VAE). Even with the comparable AUC value of SNA+XGBoost and the competitive accuracy and F-measure values of CNN+VAE, their respective CCP performances are still inferior to that of the suggested SMOTE-BaggedLMT and SMOTE-RotForestFT. Some existing methods are based on the combination of clustering and classification models like in the case of [83] and [73]. A probabilistic-based fuzzy local information c-means (PFLICM) for CCP was suggested by [83] and [73] combined K-medoid, gradient boosted tree (GBT), DT, and DL models using multiple ensemble methods such as Voting, Stacking, Bagging and Boosting. These suggested advanced methods performed well but they are still outperformed by HMSE and S-HMSE. In addition, the computation time for the methods suggested by [73] are high. Furthermore, the performances of SMOTE-BaggedLMT and SMOTE-RotForestFT are compared with current CCP models such as [13] and [82] that were implemented on sophisticated DL techniques. Specifically, [13] hyper-parameterized DL+RMSProp while [82] developed a generative adaptive network (GAN) with Back Propagation Neural Networks (P-AGBPNN) for CCP. These DL methods recorded competitive CCP performances based on prediction accuracy and F-measure values. In conclusion, the recommended SMOTE-BaggedLMT and SMOTE-RotForestFT models outperformed the current CCP models that were investigated using various computational methods on Dataset A.

The CCP performances of SMOTE-RotForestRT and SMOTE-RotForestLMT were also compared with current CCP models exhibiting diverse computational characteristics on Dataset B. As shown in Table 13, the CCP performances of SMOTE-RotForestRT and SMOTE-RotForestLMT methods were correlated with some well-known ML and DL techniques such as SVM (with different kernel functions) [84] and LSTM, GRU, and CNN [73,84] studied various forms of SVM with different kernel functions and the result comparison showed that the suggested methods (SMOTE-RotForestRT and SMOTE-RotForestLMT) outperformed the DL techniques which showed comparatively poor CCP results when compared to those of the suggested SMOTE-RotForestRT and SMOTE-RotForestLMT methods. This may be a consequence of the inability of individual studies to address the issue of class imbalance. In the quest for better CCP models, advanced ensemble approaches were developed by [80] and [81] for CCP. Although the CCP performances of these advanced ensemble methods CCP performances were comparable to those of the suggested models, their significant computational cost (time) is a cause for concern. Nevertheless, SMOTE-RotForestRT and SMOTE-RotForestLMT outperformed the current CCP models assessed on Dataset B in most cases. Hence, the observed experimental findings are used to answer the RQs as highlighted in introductory section.

Answers to research questions

Based on the investigations and experimental observations, the following findings were obtained to answer the RQs posed in the introductory section (Section 1).

RQ1: How effective are the tree-based CCP models in comparison with baseline methods in CCP with and without the CIP?

From the experimental results, it was observed that the tree-based CCP models in most cases are superior to prominent baseline classifiers (SVM, NB, MLP, and KNN). Although the differences in the CCP performances of the single tree-based classifiers (DT, Dstump, ADTree, and RT) are in some cases comparable to that of the baseline classifiers. However, ensemble (CS-Forest, SysFor, and LADTree) and hybrid tree (LMT, FT, REPTree, CDT, and HT) based classifiers are better than the baseline classifiers on the experimented original and SMOTE-balanced datasets (with and without CIP), respectively.

RQ2: What are the intra and inter-tree-based CCP performances with and without the CIP with and without the CIP?

For the intra-performance evaluations of the tree-based classifiers, for the single tree-based classifiers, it was observed that ADTree and RT and comparable in performance to DT. As for the ensemble-based tree classifiers, CS-Forest and SysFor are better predictive models compared to RF and LADTree while LMT and FT and the best-performing hybrid tree classification algorithms. Although the predictive performances of each of these tree-based classifiers are affected by CIP, however, the deployment of data sampling (SMOTE) and homogeneous ensemble methods (Bagging, Boosting, Cascade, RotForest, and RBoost) largely improved the prediction performances of the investigated tree-based classifiers. As for the intra-performance evaluations of the tree-based classifiers, the hybrid tree-based classifiers are superior to other experimented tree-based classifiers with and without the CIP and on both customer churn datasets.

RQ3: How effective are the homogeneous ensemble variants of the tree-based classifiers with and without CIP?

In comparison to the ordinary tree-based classifiers, the homogeneous ensemble variants of the classifiers are in most cases superior in CCP performances. The enhancement in the CCP performances of the ensemble variants of the tree-based classifiers on the balanced customer churn datasets is significant as compared with their CCP performances on the original datasets. This indicates that ensemble methods can not always accommodate the CIP and the combination of a data sampling method with an ensemble can lead to effective CCP models.

RQ4: How effective are the tree-based CCP models in comparison with the state-of-the-art existing rule, ML, and DL-based CCP solutions?

Not all the investigated tree-based classifiers are comparable in CCP performances as existing ML and DL models. However, the respective top-performing tree-based classifiers for each customer churn dataset, are in most cases superior to some of the existing rule, ensemble, hybrid, and advanced DL CCP models.

Threat to validity

All scientific research relies heavily on assessing and reducing potential threats to the credibility of experimental findings [7,85]. The following issues have been identified as threats to the validity of this research and are presented as follows:

External validity: Evidence from scientific studies is most convincing when it can be reliably applied in practical situations. Depending on the kind and size of the datasets used, the results of studies may not generalize to different settings. Thus, this study makes use of two popular CCP datasets that provide a broad range of features. To create and assess CCP models, several publicly accessible datasets are used extensively. This study also provided an in-depth analysis of the experimental procedure, which might increase its reliability and validity in future applications across a variety of CCP datasets.

Internal validity: This paradigm exemplifies the significance and consistency of facts, tried-and-tested theories, and empirical research. As a result, this study makes use of established ML methods which have been used before. These ML techniques were selected because they are representative of many different approaches and are effective in ML tasks. Furthermore, the analyzed CCP models were trained (handcrafted) using the CV strategy on the chosen CCP datasets, and each experiment was conducted 10 times to ensure reliability. This method helps lessen the possibility of discovering contradictions in the results. Yet, future studies may concentrate on other tactics and techniques for assessing models.

Construct validity: This issue stems from the criteria used to evaluate the effective CCP models investigated. Accuracy, AUC, F-measure, and MCC were only a few of the statistical performance indicators used in this study. The research's CCP models were empirically evaluated in depth using these metrics. The CCP models were developed with careful attention paid to the churning process and its status.

Conclusions and future works

This research work conducted an empirical analysis of tree-based classification models for CCP. Categorically, the prediction capabilities of thirteen tree-based classifiers with distinct computational characteristics were evaluated. In addition, the efficacy and resilience of these tree-based classification models and their respective homogeneous variants models to CIP were evaluated and analyzed on two publicly available telecommunication customer churn datasets from Kaggle and UCI repositories. The viability and effectiveness of the tree-based classification models were tested via experiments. The experimental findings observed on the studied CCP datasets (original and balanced) indicated the effectiveness of the tree-based classifiers over baseline ML models. In addition, it was observed that the CIP can deteriorate the CCP performances of the tree-based classifiers, but the deployment and combination of data sampling (SMOTE) and the homogeneous ensemble methods will not only resolve the CIP but also enhance the CCP performances of the tree based classification models. The hybrid tree classifiers were in most cases superior in CCP performances to single, and ensemble tree-based classifiers with or without the CIP. Consequently, this research work recommends the use of sophisticated tree-based classifiers such as SysFor, CS-Forest, LMT, FT, and ADTree for CCP and other ML tasks as they are superior to the DT and RF and are relatively resilient to data quality problems such as the CIP. Also, the combination of a data sampling method and an ensemble method to address CIP is recommended against the sole use of the ensemble or data sampling method as reported in some existing ML studies.

The next step of this study will examine how outliers and extreme values influence the correctness with which CCP models make predictions. The presence of outliers and extreme values in a dataset may have a significant influence on a classifier's prediction ability. Both outliers and extreme values have the potential to skew the dataset leading to an imbalance and high variability dataset, and make the model less reliable, resulting in inaccurate predictions. Also, studies have shown that classifiers respond differently to outliers and extreme values. For instance, outliers and extreme values may have a significant influence on distance-based and in most cases linear classification algorithms such as kNN, LR (linear) and SVM (linear and non-linear) since these algorithms depend on the distance between data points, and outliers might introduce significant distances, which may be critical to the decision boundaries of such models. That is, a single outlier can significantly affect the placement and orientation of the decision boundary. This, in turn, can lead to misclassification of other data points. Hence, the effect of outliers and extreme values must be investigated and understood for data preparation and model selection. Furthermore, the expected customer churn pattern will be investigated since consumers with a low churn probability may be more valuable in the long run, this information may help businesses make intelligent and strategic decisions regarding the retention of customers who are prone to churn. Regardless, more investigation into the topics is planned.

CRediT authorship contribution statement

Fatima E. Usman-Hamza: Supervision, Conceptualization, Methodology, Writing – original draft. **Abdullateef O. Balogun:** Conceptualization, Methodology, Software, Investigation, Writing – original draft. **Salahdeen K. Nasiru:** Conceptualization, Methodology, Writing – original draft. **Luiz Fernando Capretz:** Supervision, Writing – review & editing. **Hammed A. Mojeed:** Software, Validation, Data curation, Visualization. **Shakirat A. Salihu:** Data curation, Visualization, Investigation. **Abimbola G. Akintola:** Data curation, Visualization, Investigation. **Modinat A. Mabayoje:** Software, Validation, Data curation, Visualization. **Joseph B. Awotunde:** Visualization, Investigation, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

A. Description of Dataset A (Orange Company Dataset)

No.	Feature	Feature Description	Feature Type
1	State	States in the United States of America	Categorical
2	Account Length	Active duration of accounts	Numeric
3	Area Code	Costs of areas	Categorical
4	International Plan	1=Active, 2= Inactive	Categorical
5	Voice Mail Plan	1=Active, 2= Inactive	Categorical
6	Number of VMail Messages	Amount of voice mail messages	Numeric
7	Total day minutes	Total day call minutes clients have used	Numeric
8	Total day calls	Total number of day calls	Numeric
9	Total day charge	Today day fee	Numeric
10	Total eve minutes	Total call minutes clients have used in the evening	Numeric
11	Total eve calls	Total amount of evening calls	Numeric
12	Total eve charge	Toal evening fee	Numeric
13	Total night minutes	Total night call minutes clients have used	Numeric
14	Total night calls	Total amount of night calls	Numeric
15	Total night charge	Total night fee	Numeric
16	Total international minutes	Total call minutes clients have used for international calls	Numeric
17	Total International calls	Total amount of international call	Numeric
18	Total international charge	Total international fee	Numeric
19	Customer Service Calls	Amount of customer service calls made	Numeric
20	Churn	1=Churner, 0= Non-Churner	Categorical

B. Description of Dataset A (UCI Dataset)

No.	Feature	Feature Description	Feature Type
1	Age	Client's Age	Numeric
2	Gender	Client's Gender	Categorical
3	MarStatus	Marital Status	Categorical
4	AvgInMinUse	Average Minute of In calls with the same service provider	Numeric
5	AvgOutMinUse	Average Minute of Outcalls with the same Service Provider	Numeric
6	AvgAllMinUse	Average Minute of All calls with the same service provider	Numeric
7	AvgInMinUseOSP	Average Minute of In calls with another service provider	Numeric
8	AvgOutMinUseOSP	Average Minute of Outcalls with another Service Provider	Numeric
9	AvgAllMinUseSOSP	Average Minute of All Calls with other service provider	Numeric
10	AvgInMinUseSOSP	Average Minute of In calls from same and other service provider	Numeric
11	AvgOutMinUseSOSP	Average Minute of Out calls from same and other service provider	Numeric
12	MeanAllMinUseSOSP	Average Minute of All calls from same and other service provider	Numeric
13	ChangeMinUsed	Change in call rate (decreased, normal, increased)	Nominal
14	AvgSMS	Average Short Message Services	Numeric

(*continued on next page*)

(continued)

No.	Feature	Feature Description	Feature Type
15	AvgMonthRev	Average Monthly Revenue	Numeric
16	AssocProduct	Associated Product (Yes or No)	Nominal
17	AssocService	Associated Service (Yes or No)	Nominal
18	AvgCustCareCall	Average Customer Care Calls	Numeric
19	AvgDropCalls	Average Drop Calls	Numeric
20	AvgComplaints	Average Complaints	Numeric
21	NoChnageTarrifPlan	Number of changes of tariff	Numeric
22	MaxCallDistance	Maximum Time distance between calls	Numeric
23	Churn	1=Churner, 0= Non-Churner	Categorical

References

[1] L. Broccardo, A. Zicari, F. Jabeen, Z.A. Bhatti, How digitalization supports a sustainable business model: a literature review, Technol. Forecast Soc. Change 187 (2023), 122146, https://doi.org/10.1016/j.techfore.2022.122146.

[2] L. Saha, H.K. Tripathy, S.R. Nayak, A.K. Bhoi, P. Barsocchi, Amalgamation of customer relationship management and data analytics in different business sectors—a systematic literature review, Sustainability 13 (9) (2021) 5279, https://doi.org/10.3390/su13095279.

[3] A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, S. Anwar, Customer churn prediction in telecommunication industry using data certainty, J. Bus. Res. 94 (2019) 290–301, https://doi.org/10.1016/j.jbusres.2018.03.003.

[4] K. Oztoprak, Y.K. Tuncel, I. Butun, Technological transformation of telco operators towards seamless IoT Edge-cloud continuum, Sensors 23 (2) (2023) 1004, https://doi.org/10.3390/s23021004.

[5] A.A. Hamzah, M.F. Shamsudin, Why customer satisfaction is important to business? J. Undergraduate Soc. Sci. Technol. 1 (1) (2020).

[6] A.S. Arifin, Telecommunication service subscriber churn likelihood prediction analysis using diverse machine learning model, in: 2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT), 2020.

[7] F.E. Usman-Hamza, A.O. Balogun, L.F. Capretz, H.A. Mojeed, S. Mahamad, S.A. Salihu, A.G. Akintola, S. Basri, R.T. Amosa, N.K. Salahdeen, Intelligent decision forest models for customer churn prediction, Appl. Sci. 12 (16) (2022) 8270.

[8] S. Agrawal, A. Das, A. Gaikwad, S. Dhage, Customer churn prediction modelling based on behavioural patterns analysis using deep learning, in: 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE), 2018.

[9] A.K. Ahmad, A. Jafar, K. Aljoumaa, Customer churn prediction in telecom using machine learning in big data platform, J. Big Data 6 (1) (2019) 1–24, https://doi.org/10.1186/s40537-019-0191-6.

[10] I. AlShourbaji, N. Helian, Y. Sun, M. Alhameed, Anovel HEOMGA approach for class imbalance problem in the application of customer churn prediction, SN Comput. Sci. 2 (6) (2021) 1–12, https://doi.org/10.1007/s42979-021-00850-y.

[11] K. Ljubičić, A. Merćep, Z. Kostanjčar, Churn prediction methods based on mutual customer interdependence, J. Comput. Sci. 101940 (2023), https://doi.org/10.1016/j.jocs.2022.101940.

[12] P.P. Šimović, C.Y. Chen, E.W. Sun, Classifying the variety of customers' online engagement for churn prediction with a mixed-penalty logistic regression, Comput. Econ. 61 (1) (2023) 451–485, https://doi.org/10.1007/s10614-022-10275-1.

[13] A. Dalli, Impact of hyperparameters on deep learning model for customer churn prediction in telecommunication sector, Math.Problems Eng. 2022 (2022), https://doi.org/10.1155/2022/4720539.

[14] E. Domingos, B. Ojeme, O. Daramola, Experimental analysis of hyperparameters for deep learning-based churn prediction in the banking sector, Computation 9 (3) (2021) 34, https://doi.org/10.3390/computation9030034.

[15] Jain, H., Khunteta, A., & Shrivastav, S.P. (2021). Telecom churn prediction using seven machine learning experiments integrating features engineering and normalization. 10.21203/rs.3.rs-239201/v1.

[16] A. Amin, S. Anwar, A. Adnan, M. Nawaz, K. Alawfi, A. Hussain, K. Huang, Customer churn prediction in the telecommunication sector using a rough set approach, Neurocomputing 237 (2017) 242–254, https://doi.org/10.1016/j.neucom.2016.12.009.

[17] A. Amin, S. Shehzad, C. Khan, I. Ali, S. Anwar, Churn prediction in telecommunication industry using rough set approach, New Trends Comput. Collect. Intell. (2015) 83–95, https://doi.org/10.1007/978-3-319-10774-5_8.

[18] I. Brandusoiu, G. Toderean, Churn prediction in the telecommunications sector using support vector machines, Margin 1 (2013) x1.

[19] M.M. Hossain, M.S. Miah, Evaluation of different SVM kernels for predicting customer churn, in: 2015 18th International Conference on Computer and Information Technology (ICCIT), 2015.

[20] H. Jain, A. Khunteta, S. Srivastava, Churn prediction in telecommunication using logistic regression and logit boost, Proc. Comput. Sci. 167 (2020) 101–112, https://doi.org/10.1016/j.procs.2020.03.187.

[21] N.I. Mohammad, S.A. Ismail, M.N. Kama, O.M. Yusop, A. Azmi, Customer churn prediction in telecommunication industry using machine learning classifiers, in: Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, 2019.

[22] A. Amin, B. Shah, A.M. Khattak, F.J.L. Moreira, G. Ali, A. Rocha, S. Anwar, Cross-company customer churn prediction in telecommunication: a comparison of data transformation methods, Int. J. Inf. Manage. 46 (2019) 304–319, https://doi.org/10.1016/j.ijinfomgt.2018.08.015.

[23] M.A. Mabayoje, A.O. Balogun, H.A. Jibril, J.O. Atoyebi, H.A. Mojeed, V.E. Adeyemo, Parameter tuning in KNN for software defect prediction: an empirical analysis, Jurnal Teknologi dan Sistem Komputer 7 (4) (2019) 121–126, https://doi.org/10.14710/jtsiskom.7.4.2019.121-126.

[24] V. Fortuin, Priors in bayesian deep learning: a review, Int. Stat. Rev. 90 (3) (2022) 563–591, https://doi.org/10.1111/insr.12502.

[25] A. Xu, M. Raginsky, Minimum excess risk in bayesian learning, IEEE Trans. Inf. Theory 68 (12) (2022) 7935–7955, https://doi.org/10.1109/TIT.2022.3176056.

[26] A. De Caigny, K. Coussement, K.W De Bock, A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees, Eur. J. Oper. Res. 269 (2) (2018) 760–772, https://doi.org/10.1016/j.ejor.2018.02.009.

[27] S. Höppner, E. Stripling, B. Baesens, S. vanden Broucke, T. Verdonck, Profit driven decision trees for churn prediction, Eur. J. Oper. Res. 284 (3) (2020) 920–933, https://doi.org/10.1016/j.ejor.2018.11.072.

[28] I. Ullah, B. Raza, A.K. Malik, M. Imran, S.U. Islam, S.W. Kim, A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector, IEEE Access 7 (2019) 60134–60149, https://doi.org/10.1109/ACCESS.2019.2914999.

[29] J. Zhang, Customer churn prediction based on a novelty hybrid random forest algorithm, in: Third International Conference on Computer Vision and Data Mining (ICCVDM 2022), 2023.

[30] Y. Huang, B. Huang, M.T. Kechadi, A rule-based method for customer churn prediction in telecommunication services, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2011.

[31] R. Vadakattu, B. Panda, S. Narayan, H. Godhia, Enterprise subscription churn prediction, in: 2015 IEEE International Conference on Big Data (Big Data), 2015.

[32] T. Xu, Y. Ma, K. Kim, Telecom churn prediction system based on ensemble learning using feature grouping, Appl. Sci. 11 (11) (2021) 4742, https://doi.org/10.3390/app11114742.

[33] M.O. Arowolo, S.O. Abdulsalam, Y.K. Saheed, J.O. Afolayan, Customer churn prediction in telecommunication industry using classification and regression trees and artificial neural network algorithms, Indonesian J. Electr. Eng. Inf. 10 (2) (2022), https://doi.org/10.52549/ijeei.v10i2.2985.
[34] T. Vafeiadis, K.I. Diamantaras, G. Sarigiannidis, K.C. Chatzisavvas, A comparison of machine learning techniques for customer churn prediction, Simul. Modell. Pract. Theory 55 (2015) 1–9, https://doi.org/10.1016/j.simpat.2015.03.003.
[35] A. Idris, A. Khan, Ensemble based efficient churn prediction model for telecom, in: 2014 12th International Conference on Frontiers of Information Technology, 2014.
[36] N. Saini, G.K. Monika, K. Garg, Churn prediction in telecommunication industry using decision tree, Int. J. Eng. Res. Technol. 6 (2017) 439–443.
[37] O. Balogun Abdullateef, A. Mabayoje Modinat, A.S.A. Salihu Shakirat, Enhanced classification via clustering techniques using decision tree for feature selection, Int. J. Appl. Inf. Syst. 9 (6) (2015), https://doi.org/10.5120/ijais2015451425.
[38] H. Lee, S. Kim, Black-box classifier interpretation using decision tree and fuzzy logic-based classifier implementation, Int. J. Fuzzy Logic Intell. Syst. 16 (1) (2016) 27–35, https://doi.org/10.5391/IJFIS.2016.16.1.27.
[39] S. Kumar, M. Kumar, Predicting customer churn using artificial neural network, in: International Conference on Engineering Applications of Neural Networks, 2019.
[40] C.F. Tsai, Y.H. Lu, Customer churn prediction by hybrid neural networks, Expert Syst. Appl. 36 (10) (2009) 12547–12553, https://doi.org/10.1016/j.eswa.2009.05.032.
[41] G.D.O. Olle, S. Cai, A hybrid churn prediction model in mobile telecommunication industry, Int. J. e-Educ. e-Bus. e-Manag. e-Learn. 4 (1) (2014) 55, https://doi.org/10.7763/IJEEEE.2014.V4.302.
[42] K.W. De Bock, D. Van den Poel, An empirical evaluation of rotation-based ensemble classifiers for customer churn prediction, Expert Syst. Appl. 38 (10) (2011) 12293–12301, https://doi.org/10.1016/j.eswa.2011.04.007.
[43] K.W. De Bock, K. Coussement, D. Van den Poel, Ensemble classification based on generalized additive models, Comput. Stat. Data Anal. 54 (6) (2010) 1535–1546, https://doi.org/10.1016/j.csda.2009.12.013.
[44] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Trans. Syst. Man Cybernetics Part C (Applic. Rev.) 42 (4) (2011) 463–484, https://doi.org/10.1109/TSMCC.2011.2161285.
[45] M.A.H. Farquad, V. Ravi, S.B. Raju, Churn prediction using comprehensible support vector machine: an analytical CRM application, Appl. Soft Comput. 19 (2014) 31–40, https://doi.org/10.1016/j.asoc.2014.01.031.
[46] A.O. Balogun, S. Basri, S.J. Abdulkadir, V.E. Adeyemo, A.A. Imam, A.O. Bajeh, Software defect prediction: analysis of class imbalance and performance stability, J. Eng. Sci. Technol. 14 (6) (2019) 3294–3308.
[47] Q. Yu, S. Jiang, Y. Zhang, The performance stability of defect prediction models with class imbalance: an empirical study, IEICE Trans. Inf. Syst. 100 (2) (2017) 265–272, https://doi.org/10.1587/transinf.2016EDP7204.
[48] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, A. Hussain, Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study, IEEE Access 4 (2016) 7940–7957, https://doi.org/10.1109/ACCESS.2016.2619719.
[49] A. Amin, F. Rahim, I. Ali, C. Khan, S. Anwar, A comparison of two oversampling techniques (smote vs mtdf) for handling class imbalance problem: a case study of customer churn prediction. New Contributions in Information Systems and Technologies: Volume 1, 2015.
[50] C.X. Ling, C. Li, Data mining for direct marketing: problems and solutions, in: KDD'98: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, 1998.
[51] J. Vijaya, E. Sivasankar, An efficient system for customer churn prediction through particle swarm optimization based feature selection model with simulated annealing, Cluster Comput. 22 (2019) 10757–10768, https://doi.org/10.1007/s10586-017-1172-1.
[52] B. He, Y. Shi, Q. Wan, X. Zhao, Prediction of customer attrition of commercial banks based on SVM model, Proc. Comput. Sci. 31 (2014) 423–430, https://doi.org/10.1016/j.procs.2014.05.286.
[53] J. Burez, D. Van den Poel, Handling class imbalance in customer churn prediction, Expert Syst. Appl. 36 (3) (2009) 4626–4636, https://doi.org/10.1016/j.eswa.2008.05.027.
[54] S. Kanwal, I. Younas, M. Bashir, Evolving convolutional autoencoders using multi-objective particle swarm optimization, Comput. Electr. Eng. 91 (2021), 107108, https://doi.org/10.1016/j.compeleceng.2021.107108.
[55] V. Umayaparvathi, K. Iyakutti, Automated feature selection and churn prediction using deep learning models, Int. Res. J. Eng. Technol. 4 (3) (2017) 1846–1854.
[56] B. Luo, S. Lin, Research on the anonymous customer segmentation model of telecom, in: 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), 2020.
[57] O.N. Akande, O. Gbenle, O.C. Abikoye, R.G. Jimoh, H.B. Akande, A.O. Balogun, A. Fatokun, SMSPROTECT: an automatic smishing detection mobile application, ICT Express 9 (2) (2023) 168–176, https://doi.org/10.1016/j.icte.2022.05.009.
[58] L. Rokach, Decision forest: twenty years of research, Inf. Fus. 27 (2016) 111–125, https://doi.org/10.1016/j.inffus.2015.06.005.
[59] T. Van Phong, H.B. Ly, P.T. Trinh, I. Prakash, P. Btjvjoes, Landslide susceptibility mapping using Forest by Penalizing Attributes (FPA) algorithm based machine learning approach, Vietnam J. Earth Sci. 42 (3) (2020) 237–246, https://doi.org/10.15625/0866-7187/42/3/15047.
[60] R. Ajin, S. Saha, A. Saha, A. Biju, R. Costache, S.L. Kuriakose, Enhancing the accuracy of the REPTree by integrating the hybrid ensemble meta-classifiers for modelling the landslide susceptibility of Idukki district, South-western India, J. Indian Soc. Remote Sens. 50 (11) (2022) 2245–2265, https://doi.org/10.1007/s12524-022-01599-4.
[61] S.M. Fati, A. Muneer, A. Alwadain, A.O. Balogun, Cyberbullying detection on twitter using deep learning-based attention mechanisms and continuous bag of words feature extraction, Mathematics 11 (16) (2023) 3567, https://doi.org/10.3390/math11163567.
[62] H. Mamman, A.O. Balogun, S. Basri, L.F. Capretz, V.E. Adeyemo, A.A. Imam, G. Kumar, Software requirement risk prediction using enhanced fuzzy induction models, Electronics 12 (18) (2023) 3805, https://doi.org/10.3390/electronics12183805.
[63] P.-T.T. Ngo, T.D. Pham, V.H. Nhu, T.T. Le, D.A. Tran, D.C. Phan, P.V. Hoa, J.L. Amaro-Mellado, D.T. Bui, A novel hybrid quantum-PSO and credal decision tree ensemble for tropical cyclone induced flash flood susceptibility mapping with geospatial data, J. Hydrol. 596 (2021), 125682, https://doi.org/10.1016/j.jhydrol.2020.125682.
[64] B.T. Pham, M. Avand, S. Janizadeh, T.V. Phong, N. Al-Ansari, L.S. Ho, S. Das, H.V. Le, A. Amini, S.K. Bozchaloei, GIS based hybrid computational approaches for flash flood susceptibility assessment, Water 12 (3) (2020) 683, https://doi.org/10.3390/w12030683.
[65] A.O. Balogun, F.B. Lafenwa-Balogun, H.A. Mojeed, V.E. Adeyemo, O.N. Akande, A.G. Akintola, A.O. Bajeh, F.E. Usman-Hamza, SMOTE-based homogeneous ensemble methods for software defect prediction, in: International Conference on Computational Science and its Applications, 2020.
[66] A.O. Balogun, B.J. Odejide, A.O. Bajeh, Z.O. Alanamu, F.E. Usman-Hamza, H.O. Adeleke, M.A. Mabayoje, S.R. Yusuff, Empirical analysis of data sampling-based ensemble methods in software defect prediction, in: International Conference on Computational Science and Its Applications, 2022.
[67] A.O. Balogun, K.S. Adewole, A.O. Bajeh, R.G. Jimoh, in: Cascade generalization based functional tree for website phishing detection International Conference on Advances in Cyber Security, 2021.
[68] A.O. Balogun, K.S. Adewole, M.O. Raheem, O.N. Akande, F.E. Usman-Hamza, M.A. Mabayoje, A.G. Akintola, A.W. Asaju-Gbolagade, M.K. Jimoh, R.G. Jimoh, Improving the phishing website detection using empirical analysis of function tree and its variants, Heliyon 7 (7) (2021) e07437.
[69] A.O. Balogun, S. Basri, S.A. Jadid, S. Mahamad, M.A. Al-momani, A.O. Bajeh, A.K. Alazzawi, Search-Based Wrapper Feature Selection Methods in Software Defect Prediction: An Empirical Analysis, Computer Science On-line Conference, 2020.
[70] A.O. Balogun, H.A. Mojeed, K.S. Adewole, A.G. Akintola, S.A. Salihu, A.O. Bajeh, R.G. Jimoh, Optimized decision forest for website phishing detection, in: Proceedings of the Computational Methods in Systems and Software, 2021.
[71] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor. Newsletter 11 (1) (2009) 10–18, https://doi.org/10.1145/1656274.1656278.
[72] M.J. Crawley, The R Book, John Wiley & Sons, 2012.

[73] S.F. Bilal, A.A. Almazroi, S. Bashir, F.H. Khan, A.A. Almazroi, An ensemble based approach using a combination of clustering and classification algorithms to enhance customer churn prediction in telecom industry, PeerJ Comput. Sci. 8 (2022) e854, https://doi.org/10.7717/peerj-cs.854.
[74] S. Wael Fujo, S. Subramanian, M. Ahmad Khder, Customer churn prediction in telecommunication industry using deep learning, Inf. Sci. Lett. 11 (1) (2022) 24, https://doi.org/10.18576/isl/110120.
[75] V.E. Adeyemo, A.O. Balogun, H.A. Mojeed, N.O. Akande, K.S. Adewole, Ensemble-based logistic model trees for website phishing detection, in: International Conference on Advances in Cyber Security, 2020.
[76] R. Jimoh, A. Balogun, A. Bajeh, S. Ajayi, A PROMETHEE based evaluation of software defect predictors, J. Comput. Sci. Applic. 25 (1) (2018) 106–119.
[77] B.J. Odejide, A.O. Bajeh, A.O. Balogun, Z.O. Alanamu, K.S. Adewole, A.G. Akintola, S.A. Salihu, F.E. Usman-Hamza, H.A. Mojeed, An empirical study on data sampling methods in addressing class imbalance problem in software defect prediction, in: Computer Science On-line Conference, 2022.
[78] A.G. Akintola, A.O. Balogun, H.A. Mojeed, F. Usman-Hamza, S.A. Salihu, K.S. Adewole, G.B. Balogun, P.O. Sadiku, Performance analysis of machine learning methods with class imbalance problem in android malware detection, Int. J. Inter. Mobile Technol. 16 (2022) 140–162.
[79] Y.A. Alsariera, A.O. Balogun, V.E. Adeyemo, O.H. Tarawneh, H.A. Mojeed, Intelligent tree-based ensemble approaches for phishing website detection, J. Eng. Sci. Technol 17 (2022) 563–582.
[80] S. Tavassoli, H. Koosha, Hybrid ensemble learning approaches to customer churn prediction, Kybernetes (2021), https://doi.org/10.1108/K-04-2020-0214.
[81] M. Saghir, Z. Bibi, S. Bashir, F.H. Khan, Churn prediction using neural network based individual and ensemble models, in: 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), 2019.
[82] M. Jeyakarthic, S. Venkatesh, An effective customer churn prediction model using adaptive gain with back propagation neural network in cloud computing environment, J. Res. Lepidoptera 51 (1) (2020) 386–399, https://doi.org/10.36872/LEPI/V51I1/301034.
[83] C. Praseeda, B. Shivakumar, Fuzzy particle swarm optimization (FPSO) based feature selection and hybrid kernel distance based possibilistic fuzzy local information C-means (HKD-PFLICM) clustering for churn prediction in telecom industry, SN Appl. Sci. 3 (6) (2021) 1–18, https://doi.org/10.1007/s42452-021-04576-7.
[84] E. Shaaban, Y. Helmy, A. Khedr, M. Nasr, A proposed churn prediction model, Int. J. Eng. Res. Appl. 2 (4) (2012) 693–697.
[85] A.G. Akintola, A.O. Balogun, L.F. Capretz, H.A. Mojeed, S. Basri, S.A. Salihu, F.E. Usman-Hamza, P.O. Sadiku, G.B. Balogun, Z.O. Alanamu, Empirical analysis of forest penalizing attribute and its enhanced variations for android malware detection, Appl. Sci. 12 (9) (2022) 4664.