Scientific African 24 (2024) e02223

Contents lists available at ScienceDirect

Scientific African

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

Sampling-based novel heterogeneous multi-layer stacking ensemble method for telecom customer churn prediction

Fatima E. Usman-Hamza [a], Abdullateef O. Balogun [b,*], Ramoni T. Amosa [a], Luiz Fernando Capretz [c], Hammed A. Mojeed [a,d], Shakirat A. Salihu [a], Abimbola G. Akintola [a], Modinat A. Mabayoje [a]

[a] Department of Computer Science, University of Ilorin, Ilorin 1515, Nigeria
[b] Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, Perak 32610, 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, Gdańsk 80-233, Poland

ARTICLE INFO

Editor by: DR B Gyampoh

Keywords:
Customer churn
Ensemble
Class imbalance
Telecommunication

ABSTRACT

In recent times, customer churn has become one of the most significant issues in business-oriented sectors with telecommunication being no exception. Maintaining current customers is particularly valuable due to the high degree of rivalry among telecommunication companies and the costs of acquiring new ones. The early prediction of churned customers may help telecommunication companies to identify the causes of churn and design industrial tactics to address or mitigate the churn problem. Controlling customer churn by developing efficient and reliable customer churn prediction (CCP) solutions is essential to achieving this objective. Findings from existing CCP studies have shown that numerous methods, such as rule-based and machine-learning (ML) mechanisms, have been devised to solve the CCP problem. Nonetheless, the problems of adaptability and the resilience of rule-based CCP solutions are its major weaknesses, and the skewed pattern of churn datasets (class imbalance) is detrimental to the prediction performances of conventional ML models in CCP. Hence, this research developed a robust heterogeneous multi-layer stacking ensemble method (HMSE) for effective CCP. Specifically, in the HMSE method, the prediction prowess of five ML classifiers (Random Forest (RF), Bayesian network (BN), Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Repeated Incremental Pruning to Produce Error Reduction (RIPPER)) with distinct computational characteristics are ensembled based on stacking and the resulting model is further enhanced using a forest penalizing attribute (FPA) model. The synthetic minority oversampling technique (SMOTE) is integrated with the proposed HMSE to balance the skewed class label present in the original experimental datasets. Extensive tests were carried out to determine the effectiveness of the proposed HMSE and S-HMSE on standard telecom CCP datasets. Observed findings from the experimental results showed that HMSE and S-HMSE can effectively predict churners even with the class imbalance (skewed datasets) problem. In addition, comparison studies demonstrated that the suggested S-HMSE offered improved prediction performance and optimum solutions for CCP in the telecom sector in comparison with baseline classifiers, homogeneous ensemble methods, and current CCP approaches.

* Corresponding author.
E-mail address: abdullateef.ob@utp.edu.my (A.O. Balogun).

https://doi.org/10.1016/j.sciaf.2024.e02223
Received 6 September 2023; Received in revised form 23 January 2024; Accepted 2 May 2024
Available online 3 May 2024

Introduction

In a sector populated by thriving and vibrant businesses, customers are seen as important stakeholders for any enterprise [1,2]. Customers in a such dynamic market where several vendors are vying for their business may swiftly jump ship if they are not satisfied with the service they are receiving [3]. This sudden switch, which can be referred to as customer churn may be brought on by service or product dissatisfaction, increasing prices, poor service or product quality, a dearth of functionalities, or other relative concerns [4]. Customer churning has been observed to have an immediate adverse impact on several businesses across various industries, such as banking services, airline services, and telecommunications [5]. The development and maintenance of life-long relationships with their existing customers is a rapidly expanding area of concern for these businesses. This trend has been noted, especially in the telecommunications industry.

Undoubtedly, the telecommunications industry's constant growth and development have greatly expanded the variety of businesses operating there, establishing a competitive market [6]. As a result of intense competition, cluttered businesses, a quick-changing sector, and the introduction of innovative and alluring deals, the telecommunications industry is undergoing substantial customer churn. It has become essential to maximize profits in this rapidly evolving industry [7,8]. To do this, many strategies have been recommended, including attracting new customers, retaining existing ones, and lengthening the retention time of existing subscribers. According to existing studies, businesses may find that acquiring new customers is more expensive than keeping their existing ones [9, 10]. Therefore, an appropriate solution to this problem depends on predicting the likelihood of customer churn [11,12].

A critical goal of customer churn prediction (CCP) is to assist in the development of customer retention strategies that will boost business revenue and gain industry recognition. Explicitly, the importance of CCP research in the telecommunications industry or any other relevant business-oriented sector stems from its ability to improve revenue, competitiveness, service quality, customer experience, resource allocation, and technology adaption [8,9]. Addressing this research issue is critical for the long-term profitability and expansion of telecommunications firms in a dynamic and competitive environment. However, businesses in the telecommunications industry now have access to a plethora of data on their customers such as call records, text messages, voice mail records, biodata, and more. This data is critical and essential as a tool for identifying which customers are close to churning. Thus, businesses must correctly predict customer behavior in advance [13,14]. Controlling customer churn may be done in two ways: proactively and reactively. If a firm operates using a reactive approach, it prepares for a client to depart before offering any rewards to keep them around [7,12]. Conversely, proactive approaches consider customers' propensity to leave and provide them with suitable benefits. The proactive method is sometimes expressed as a binary classification issue where churners and non-churners are distinguishable [14,15].

Many methods, such as those based on rules and those based on machine learning (ML), have been proposed to deal with CCP [16]. Regardless, a major flaw of rule-based CCP methods is that they are neither scalable nor robust in their operations [6,17]. For ML-based methods, many techniques have been implemented, with varying degrees of effectiveness. This is because traditional ML algorithms in CCP perform poorly due to the abnormally patterned churn datasets [18–20]. As a result, the success of an ML approach relies heavily on the intricacies of the dataset being used, making it crucial that these datasets be as clean and structured as possible for deployment in CCP. Building reliable ML models requires careful consideration of the proportion with which class labels appear in a dataset. This predisposition is called the "class imbalance phenomenon" [21]. A class imbalance occurs when there is a large disparity between the labels assigned to the various classes (Majority and Minority). When creating ML models, inaccuracies and difficulties are common because class labels are not always distributed uniformly [22–24]. To put it another way, CCP has an issue with class imbalance since there are more instances of non-churners (majority) than churners (minority). Consequently, the issue of class imbalance requires the development of efficient ML-based CCP models [15,21].

In this research work, the occurrence and effect of the class imbalance problem are closely monitored while generating ML-based CCP models that are scalable and robust with high predictive performances. The CCP employs adaptive intelligence models such as Random Forest (RF), Bayesian network (BN), Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Repeated Incremental Pruning to Produce Error Reduction (RIPPER)) classifiers with distinct computational characteristics which are then ensembled based on multiple stacking ensemble methods. The resulting model is further enhanced by a decision forest (DF) model such as the forest penalizing attribute (FPA) model. Specifically, the RF as an instance of the tree-based classifier, generates effective decision trees (DTs) by aggregating the prediction performances singly tree structures. This distinguishes RF from conventional DTs, which rely on a subset of the features [25]. The BN is a probabilistic-based model that deploys Bayesian inference in its computation. Unlike other Bayesian-based models such as Naïve Bayes (NB), BN takes advantage of conditional independence to generate a compact and factorized representation of the joint probability distribution [26]. As a function-based model, SVM categorizes data using a hyper-plane to separate the support vectors. That is, an optimal separating hyperplane between the two classes is deduced by maximizing the margin between the classes' closest points [20]. KNN is an instance-based model that categorizes datasets based on a similarity method. As a non-parametric model, KNN does not consider underlying assumptions with its function approximated locally and full computation is postponed until classification [27]. RIPPER is a rule-based model that uses the divide-and-conquer strategy to generate rules from a dataset. Thereafter, a subsequent optimization process considers each rule in the current set individually, after which a replacement rule and a revised rule are generated. Then, either the original rule, the replacement rule, or the revised rule is selected for the model depending on the lowest description length requirement [28].

Furthermore, multi-layer heterogeneous stacking ensemble approaches based on cross-validation and split-criteria are developed and an FPA model is deployed as a metalearner to further enhance the performance of the derived model. In this circumstance, the stacking ensemble techniques consider the effectiveness of each of the base classifiers (RF, BN, SVM, KNN, RIPPER) in developing an effective CCP model. The choice of stacking ensemble technique over other ensemble methods (hard voting and multischeme) is due to

its capacity to address ambiguity in its operations and in the generation of the final model [29]. It is proposed that a multi-layer stacking ensemble approach can be used to enhance the prediction powers of the heterogeneous baseline models to produce CCP models that are both dependable and generalizable. Moreover, the synthetic minority over-sampling technique (SMOTE) is used as a practical solution to the inherent class imbalance issue in experimental datasets and it is integrated with the proposed method to form an enhanced SMOTE-based HMSE (S-HMSE) for CCP.

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

1. A Heterogeneous Multi-Layer Stacking Ensemble Method (HMSE) is developed for CCP.
2. An enhanced HMSE with SMOTE data sampling (S-HMSE) method is developed for CCP.
3. The CCP performances of HMSE and S-HMSE are empirically evaluated and validated with baseline, homogeneous, 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 proposed HMSE and S-HMSE models.

1. How effective are the proposed HMSE and S-HMSE models in comparison with baseline and homogeneous ensemble methods in CCP?
2. How effective are the proposed HMSE and S-HMSE models against the state-of-the-art existing rule, ML, and DL-based CCP solutions?

The remainder of the paper is organized as follows. Related works section examines existing related CCP studies. The research methodology and experimental setup are analyzed in Methodology section. The experimental results and their discussion are presented in Results and Discussion section. The threat to the validity of this research work is presented in Threat to validity section while Conclusions and Future Works section concludes the research work and highlights future work.

Related works

This section explores and analyses the various ML-based techniques used by pre-existing CCP approaches.

A significant degree of research effort has been put into developing CCP solutions that are based on ML algorithms. Commonly used ML classifiers for CCP have been the basis of most of these research efforts. For instance, Brandusoiu and Toderean [18] deployed SVM with four separate kernel functions (Linear Kernel (LK), Polynomial Kernel (PK), Radial Basis Function Kernel (RBFK), and Sigmoid Kernel (SK)). Observed outcomes from their experimental results showed that the SVM with PK had the best performance in comparison to other implemented models. Nonetheless, the authors investigated only SVM and its variations for CCP, and their models were not compared with other baseline ML approaches to see how well the implemented SVMs performed. In a similar study, the applicability of SVM for CCP was explored by Hossain and Miah [20] on a proprietary dataset. SVM based on the LK function performed best, while other kernel functions were also explored. Mohammad, Ismail [30] investigated the deployment and suitability of Artificial Neural Networks (ANN), logistic regression (LR), and RF for CCP. Their experimental results showed that LR outperformed both ANN and RF. Also, Kirui, Hong [31] used Bayesian-based models for CCP. Specifically, the predictive performance of Naïve Bayes (NB) and BN for CCP was investigated. To better train NB and BN, new features were created and utilized using call log information and client profile records. The performance of the Bayesian-based models (NB and BN) was superior to the tree-based models such as the decision tree (DT). In their study, Abbasimehr, Setak [3] deployed a neuro-fuzzy approach for CCP. The authors compared the effectiveness of an adaptive network-based fuzzy inference system (ANFIS) to that of DT and RIPPER. Based on the results, it was deduced that ANFIS performs just as well as DT and RIPPER but generates lesser number rules. Aside from the relative prediction performances of the conventional ML models in CCP, there is still a need for more effective methods since data quality issues particularly the class imbalance problem weakens their respective predictive performances.

Some researchers have used feature selection (FS) procedures to choose pertinent features for CCP to improve the predictive performances of baseline ML models in CCP. For instance, Arowolo, Abdulsalam [32] merged the RelieFf FS approach with Classification and Regression Trees (CART) and ANN. Similarly, Zhang, Li [13] used the Affinity Propagation (AP) approach to pick features on RF for use in CCP. Lalwani, Mishra [33] utilized a metaheuristic FS method (gravitational search) for selection of relevant features and generated various standard ML models for CCP. Brânduşoiu, Toderean [19] used a dimensionality reduction method (principal component analysis (PCA)), on multilayer perceptron (MLP) and BN ML classifiers for CCP. From their experimental results, it observed that there was an enhancement in the prediction performances of the experimented models which can be attributed to the deployment of the PCA. However, issues like the filter rank selection problem may arise if the wrong FS approach is used for CCP. Furthermore, some of the most often used metaheuristic-based FS techniques are stochastic while other forms of dimensionality reduction such as PCA provides a new representation of the features that may be inappropriate.

In the quest for better CCP solutions, several recent studies have shifted to the application of deep learning (DL) techniques such as Deep Neural Networks (DNNs), Stacked Auto-Encoders (SAEs), Recurrent Neural Networks (RNNs), Deep Belief Networks (DBNs), and Convolution Neural Networks [4,34–39]. In a similar effort, Wael Fujo, Subramanian [39] proposed a Deep-BP-ANN model for CCP. The suggested approach used the Lasso Regularization (Lasso) and Variance Thresholding techniques to choose useful and key features. After the two FS techniques, the Random Over-Sampling strategy was used to fix the problem of class imbalance. Various hyperparameter values provide the basis for the development of Deep-BP-ANN. Their findings demonstrated that by carefully selecting

Table 1
Analyses on the model, findings, and limitations of existing studies on CCP.

Authors	Model	Class Imbalance	Findings	Limitations
Brandusoiu and Toderean [18]	SVM with 4 different Kernel Functions (Radial Basis Function, Linear, Polynomial, Sigmoid)	Not addressed	SVM based on Polynomial Function performed best.	The results lack generalization as the scope of the research was limited to SVM only. Also, class imbalance was addressed which could impact the performance of the experimented models
Hossain and Miah [20]	SVM with 6 different Kernel Functions (Anova RBF, Linear, Polynomial, Gaussian, Sigmoid, Laplacian)		SVM based on Linear Function performed best.	
Mohammad, Ismail [30]	LR, ANN, and RF		LR performed best.	The scope of the research is limited as only dataset was used and the class imbalance issue was not addressed.
Kirui, Hong [31]	NB, BN, and DT		NB and BN outperformed DT	
Abbasimehr, Setak [3]	ANFIS, DT, and RIPPER		ANFIS outperformed DT and RIPPER	ANFIS performance can be easily affected by data quality issues such as class imbalance. Moreover, it is heavily dependent on the location of a membership function
Arowolo, Abdulsalam [32]	ReliefFS+ANN and ReliefFS+CART		ReliefFS+ANN has superior performance	
Lalwani, Mishra [33]	LR, DT, KNN, RF, NB, SVM (Linear and Polynomial), XGBoost, CatBoost, Adaboost (LR, DT, KNN, RF, NB, SVM) with Gravitational Search Algorithm as FS method		Ababoost and XGBoost had highest accuracy values	The class imbalance issue was not addressed, and the number of features by the deployed FS method was not explicitly stated.
Brândușoiu, Toderean [19]	PCA+MLP, PCA+SVM, PCA+BN		PCA+SVM	
Wael Fujo, Subramanian [39]	Deep-BP-ANN	Random Oversampling (ROS)	ROS improved the predictive performance of Deep-BP-ANN	High computational cost and the lack of realism of ROS can impact the generalizability of the proposed model.
Agrawal, Das [4]	Multi-layered ANN		5-layered ANN	The class imbalance issue was not addressed, and high computational cost and hyperparameter tunning of the ANN model is a drawback
Cao, Liu [34]	SAE+LR Network		SAE+LR Network	
Cenggoro, Wirastari [35]	Vector Embedding Network Model		Batch Normalization layer	The proposed models still have the problem of inadequate accuracy, which can be optimized by adjusting parameters continuously.
Dalli [36]	various hyperparameter configurations on the performance of a NN		NN based on RemsProp optimizer	
Domingos, Ojeme [37]	various hyperparameter configurations on the performance of a DNN		DNN performed better than the MLP when a rectifier function was used for activation in the hidden layers and a sigmoid function was used in the output layer. CNN performed comparably to MLP.	
Karanovic, Popovac [38]	CNN with Grid Search			The class imbalance issue was not addressed, and the proposed models still have the problem of inadequate accuracy.
Shabankareh, Shabankareh [29]	Stacking (SVM+DT, SVM+CHAID, SVM+NN, SVM+KNN, SVM+C&RTree		SVM+CHAID outperformed other methods	
Mishra and Reddy [40]	Bagging, Boosting, RF, SVM, DT, NB, CART, ANN		RF outperformed other methods	
Xu, Ma [41]	Feature Grouping + Stacking with Soft Voting based on XGBoost and (DT, LR, NB)		feature grouping improved the prediction accuracy compared to the original customer-churn dataset.	The scope of the research is limited as only dataset was used and the class imbalance issue was not addressed.
Saghir, Bibi [42]	Bagging, Boosting, Majority Voting of (DL, NN, MLP)		BaggingMLP outperformed other experimented models	The class imbalance issue was not addressed, and the proposed models still have the problem of inadequate accuracy.
Bilal, Almazroi [43]	Clustering + Ensemble (Bagging, Boosting, Stacking, Voting)		K-med+GBT+DT+DL +Adaboost performed best.	High overhead computational cost due to several layers on ML processes

the optimal features, epochs, and the number of neurons, Deep-BP-ANN outperformed other experimented baseline ML models. Karanovic, Popovac [38] investigated how well CNN works for CCP. Compared to Multi-Layer Perceptron (MLP), the suggested CNN achieved 98% accuracy. In another study, Agrawal, Das [4] reported comparable results on the predictive performance of multi-layered ANN for CCP. Cao, Liu [34] utilized SAE for feature extraction and LR for prediction in their suggested CCP method. The retrieved features are first categorized using LR, and then SAE is pretrained based on parameter settings generated by Backward Propagation (BP). The proposed technique was observed to have comparable CCP performance; nevertheless, more refinement is possible, especially concerning the choice of parameters. While research shows that DL techniques are gaining popularity and can even beat more conventional ML methods in certain cases, they are still subject to major limitations due to issues like system (hardware) stability and hyper-parameter tuning.

Furthermore, major efforts have been shown to enhance the prediction performances of conventional ML models by employing ensemble techniques. Shabankareh, Shabankareh [29] presented stacked ensemble approaches that include DT, chi-square automated interaction detection (CHAID), MLP, and KNN with SVM in pairs. According to their results, the proposed stack ensembles outperformed the individual baseline models. Mishra and Reddy [40] proposed an ensemble method based on selected baseline classifiers with diverse computational characteristics. In terms of overall performance, their research showed that ensemble approaches were superior to the experimented baseline classifiers. Xu, Ma [41] implemented multiple ensemble techniques for CCP. The authors first used a feature clustering method based on an arbitrary measure to increase the size of the sample and uncover previously unknown information within the data. The proposed ensemble approach was based on DT, LR, and NB baseline classifiers. Their findings provide more evidence that ensemble approaches are superior to individual classifiers when it comes to CCP. The work of Saghir, Bibi [42] proposed the use of ensembles of NN-based models for CCP. Specifically, MLP, ANN, and CNN were combined using Bagging, Adaboost, and Majority Voting ensemble techniques. Their results indicated that ensemble-based NN approaches outperform other implemented models in most situations. Although there was no correlation between the efficacy of the suggested approaches and that of conventional ML models, the suggested method performed comparably well. In a different setting, Bilal, Almazroi [43] integrated clustering and classification methods for CCP with positive results. Specifically, seven classifiers were combined with four clustering approaches using multiple ensemble methods. The results indicated that, although ensemble approaches perform better overall, the classification models were superior to clustering models in terms of prediction performances even though clustering methods do not need model training. While it is true that ensemble methods can work with imbalanced data, this does not make them a realistic option. Table 1 presents a summary of the findings and limitations of existing models in CCP.

In conclusion, multiple CCP models and methodologies, including conventional ML models to sophisticated models that rely on DL, ensemble, and neuro-fuzzy approaches, have been suggested. Due to CCP's significance in CRM and other areas of business growth, there is a persistent push to discover novel approaches for CCP. The prediction performance of current CCP solutions may also be hindered, according to previous research, by the class imbalance issue. Consequently, this study proposed robust HMSE and S-HMSE methods for CCP.

Methodology

The research process and methodology are presented in this section. Specifically, information on the proposed HMSE method is explicitly explained. In addition, the CCP datasets, predictive evaluation metrics, and the experimental procedure are presented and discussed.

Random forest (RF) algorithm

Random forest (RF) works by first constructing binary subtrees using training bootstrap samples of the training dataset S, and thereafter making random selections of Y at each node. The RF model considers all classifications and then chooses the one that receives the highest support. Bootstrap agglomeration and random selection are two key principles used to characterize the operations of RF. When a dataset is bootstrapped B times, approximately two-thirds of its original size is selected at random. The remainder of the instances will be considered an "*out-of-bag*" dataset, and these instances are not considered for the generation of the subtrees but for error prediction. Every node in the tree represents a potential decision point, and these nodes are generated by randomly selecting attributes. The magnitude of the attribute being examined at each split is typically equal to i or $i/2$ where i is the total number of features [44]. All the subtrees are maximum trees since no trimming is performed. Each subtree learns about RF throughout its phase or existence. A subtree classifier is developed based on the predictions of its instances, and all the subtree classifiers developed throughout the numerous iterations are combined to produce the ultimate subtree. Each subtree classifier selects the category that an instance can belong to the class with the highest votes is used to name such an instance. Each subtree in RF is built from its own randomly generated replica of the original input dataset [45]. The bootstrapping idea is useful because it reduces the variation in an otherwise unbiased learner, like a decision tree (DT), to an acceptable level [46].

Support vector machine (SVM) algorithm

The Support Vector Machine (SVM) is a function-based supervised ML technique for both classification (linear and non-linear) and regression processes. To provide a computationally efficient ML model, SVM uses hyperplane separation in a high-dimensional feature space. Several different hyperplanes may exist for separating datasets [20]. The hyperplane with the largest margin is the best option. The margin is the maximum distance a border may extend before it encroaches on an instance. For this reason, the support vectors can

be viewed as elevated data points. Thus, the purpose of the SVM is to choose the best hyperplane for classifying target vectors into available class labels. The decision boundaries that assist in the classification of the data points are called hyperplanes. The data points that lie on either side of the hyperplane are each capable of being assigned to a distinct category [18]. In addition to this, the size of the hyperplane is determined by the total number of attributes. If there are just two input attributes, then the hyperplane is nothing more than a straight line. When there are just three input attributes, the hyperplane transforms into a plane that is only two dimensions deep. When there are more than three distinct attributes, it becomes difficult to conceptualize. Removing or altering the support vectors will change the orientation of the hyperplanes. In the case of a non-linear task, estimating the margins is done using a variety of different kernel functions. The maximization of the margins across hyperplanes is the primary goal of various kernel functions (i.e., linear, polynomial, radial basis, and sigmoid), among others [47].

K nearest neighbour (KNN) algorithm

K Nearest Neighbor (KNN) is a non-parametric instance-based ML model and it is one of the basic classification techniques that may be used in situations when there is minimal or no information about the dataset. Like SVM, KNN can be used for both classification and regression tasks. However, KNN is used for classification processes based on the presumption that similar instances are always around one another. The necessity to do discriminant analysis led to the development of the KNN model, which is used in situations in which credible parameterized estimations of probability densities are either uncertain or impossible to obtain. KNN can also be regarded as a kind of lazy learning, in which the function is simply estimated locally and actual processing is postponed till it is time to classify the data [27]. An instance's neighbourhood space determines how that instance is categorized. The neighbors are chosen from among a group of instances for which the appropriate categorization is already established. When there is a requirement to categorize a new instance, the instance in question's k closest neighbors from the training data are utilized to determine the class of that instance's copy in the test set [48]. KNN is based on the Euclidean distance between a test dataset and the training datasets. That is, distances between a new instance and other instances must be computed to identify the nearest data points to a new instance. When used with other distance measures, decision boundaries can be used to divide the space occupied by instances into manageable components. The number of neighbors that will be used to decide on a new instance classification is set by the k variable in the KNN model. For instance, if $k = 1$, such an instance will be placed in the same category as the instance that is spatially closest to it. Different values of k might cause overfitting or underfitting, thus finding the right one can be tricky. Smaller values of k may have more variance but lower bias, and bigger values of k will have higher bias but lower variance. Data with more outliers or noise would do better with greater values of k, hence this parameter is data-dependent. In general, it is better to choose an odd positive integer for k to prevent classification tie-breakers, and cross-validation methods can be used to get the optimal k value.

Bayesian network (BN) algorithm

Bayesian network (BN) is a Bayesian or probability-based model for displaying the joint probability distribution of a group of random variables that may be linked causally. Each node in the network has a conditional probability distribution, and the edges between them show the causal link between the random variables they represent. In other words, each node in a BN represents a random variable, and each edge reflects the conditional chance that two or more nodes in the network have some common feature [49]. To determine the likelihood of a certain event, BN uses directed cyclic graphs (DCGs) in conjunction with a table of conditional probabilities. The graph is acyclic, which means that there is no direct route that can be taken from one node to another. On the other hand, the table of probabilities illustrates the possibility that a random variable would assume values based on the information presented. One of the primary goals of the BN approach is to simulate the conditional probability distribution of outcome (often causative) variables considering new data [50].

Repeated incremental pruning to produce error reduction (RIPPER) algorithm

Repeated Incremental Pruning to Produce Error Reduction (RIPPER) is one of the most effective and widely used rule-based ML techniques [28]. RIPPER uses the "divide and conquer" technique for rule induction and Incremental Reduced Error Pruning (IREP) to create a base set of rules for each class in a dataset. The rules in the present set are then evaluated one by one in a second optimization process, which then generates a replacement rule and a revised rule. Then, the model is evaluated against the minimal description length requirement to determine whether the original rule, the replacement rule, or the revised rule should be kept. Findings from several studies have shown the suitability of RIPPER for class-skewed and noisy datasets as it uses such data as a validation set to avoid model overfitting. Initially, RIPPER selects the majority class label as the default class label and generates rules for those set of instances that belong to the default class label. It moves from a general into a specific method for rule induction by starting from an empty rule set before adding the best conjunct to the rule antecedent [51].

Forest penalizing attribute (FPA) algorithm

FPA is a decision forest model that encourages substantial variety by considering several factors associated with weight. These weight-related constructs are based on the distribution and increasing of weights. FPA is an approach that, in most cases, builds a series of DTs that are very effective by making use of the potency that is present in all non-class attributes that are available in each dataset [25]. In other words, to mitigate the negative impact of keeping weights that were not assessed in the most recent DT, FPA increases the

weights of the attribute overall. This is done by maintaining weights that are missing in the most recent DT. The rationale for this is that a feature at a lower level may have an influence on a greater number of logic rules than a feature at a higher level. So, to uncover a varied set of logic rules, it is preferable to assess qualities at higher nodes than at lower nodes while creating a prospective DT. Furthermore, the FPA picks an attribute's weight at random within the weight range (WR) that has been set for that particular property level [44]. This helps to increase the possibility that different weights will be applied to features that are on the same level. Eq. (1) presents the depiction of WR.

$$\boldsymbol{WR^{\alpha}} = \begin{cases} \left[0.0000,\ \boldsymbol{e}^{-1/\alpha}\right],\ \alpha = 1 \\ \left[\boldsymbol{e}^{-1/\alpha-1} + \beta,\ \boldsymbol{e}^{-1/\alpha}\right],\ \alpha > 1 \end{cases} \tag{1}$$

Where $\boldsymbol{\alpha}$ represents the attribute level and β addresses the non-overlapping of WR for various levels.

In contrast to other DF models such as the random subspace (RS), and the extremely randomized trees (ERT), the FPA seeks to guarantee that relevant attributes are identified or retained on the resulting DT. The primary foundation of RS, and ERT is the random feature weight mechanism. This mechanism produces and distributes weights in a haphazard manner, which often results in a disparity between the weights of different attributes. Also, RS and ERT have no assurance that relevant attributes will be consistently chosen or retained. Nonetheless, to maximize the effect that the weights have, FPA adds an exponent (p-value) to the value of each weight [50]. Also, FPA does not use the subspacing of attributes, in contrast to RS and ERT. This is crucial because low and high-dimensional datasets typically exhibit inconsistencies in performance due to attribute subspacing. These analyses served as basis for the selection of FPA for CCP procedures [52].

Heterogeneous multi-layer stacking ensemble method (HMSE)

The HMSE model's primary goal is to combine the CCP performance of several baseline models with different computational capacities to get optimal CCP performance overall. Fig. 1 shows HMSE's pseudocode. Due to the HMSE model being formed by collective induction ML models with varying degrees of computational capabilities, stacking is an efficient integrated ensemble learning approach, but from the viewpoint of the baseline model, it is heterogeneous. The primary concept is to improve the efficacy of the produced CCP model by decreasing the generalization error by using several functional and varied baseline models and employing a meta-classifier to aggregate the outcomes of baseline model predictions. In this research, the representative algorithms of baseline classifiers from tree-based, function-based, instance-based, probability-based, and rule-based ML algorithms and a DF model are used to amplify their aggregated prediction performance.

HMSE generalization capacity can be enhanced by the heterogeneous integration of several kinds of basic learners, and HMSE fitting ability may be enhanced by meta-learners to combine the outcomes of previous predictions. As presented in Fig. 1, the input

Algorithm 1. Heterogeneous Multi-Layer Stacking Ensemble (HMSE) Method

Input: Training set $A = \{b_i, c_i\}, i = 1 \ldots m, y_i \in Y, Y = \{c_1, c_2, \ldots, c_k\}, c_k$ is the class label.
Rounds of Iterations $I = 100$.
Baseline Classifiers D_i = {RF, KNN, BN, SVM, RIPPER}
Meta Learner E = FPA
Output = HMSE
Step 1: Train each baseline classifier of D_i on dataset A
for i = 1 to D
    $F_i = D_i$ (A)
end for
Step 2: Develop a new dataset of Predictions A'
for j = 1 to n do
    for i = 1 to D do
        $z_{ij} = F_i(b_j)$
    end for
    A' = $\{Z_j, c_j\}$, where $Z_j = \{z_{ij}, z_{1j}, \ldots, z_{nj}\}$
end for
Step 3: Train a Meta Learner E
HMSE = E (A')
return HMSE

Fig. 1. The pseudocode for the proposed HMSE.

datasets of the HMSE are selected and preprocessed to clean the datasets. The HMSE consists of two layers. The first layer is based on heterogeneous baseline classifiers: RF, KNN, SVM, BN, and RIPPER, and a simple stacking model is produced. The second layer is the deployment of the FPA model, which is an instance of the DF meta-classifier to perform the final classification process. In the early stage of the experiment, to obtain a baseline model with better generalizability and accuracy, the CCP performance of the selected baseline classifiers and their homogeneous ensemble methods (Bagging, Boosting, Cascade, Rotation Forest, and Dagging) were investigated to deduce a wide range of combined experiments. A variant of HMSE (S-HMSE) is also designed based on the introduction of a data sampling method (SMOTE) as an additional data preprocessing step. This is due to the latent and inherent imbalance characteristic of the investigated customer churn datasets based on the skewness of their class labels. In summary, HMSE and S-HMSE use the initially produced CCP model via stacking of the primary baseline models (RF, KNN, SVM, BN, and RIPPER) as the input into a DF model (FPA algorithm) to generate an improved CCP model. Therefore, the S-HMSE model variant is expected to have a reduced variance, improved efficacy, and its CCP performance enhanced. Table 2 presents the parameters of the baseline classifiers and the DF model as used in this research work.

Experimental procedure

A schematic representation of the experiments conducted in this research work is shown in Fig. 2. The procedure discussed here is essential and meant to provide empirical evidence for the effectiveness of the proposed and implemented CCP models. Specifically, we developed and examined a two-stage experimental design, and we evaluated the prediction performances of the resultant CCP models consistently and objectively.

At first, an ensemble of the heterogeneous baseline classifiers (RF, BN, SVM, KNN, RIPPER) based on the stacking method is implemented and the resulting model is further improved using the FPA model. Also, each of the individual baseline classifiers and their respective homogeneous variants (Bagging, Boosting, Cascade, Rotation Forest, and Dagging) on the original customer churn datasets are implemented. The essence of this evaluation is to determine and validate the efficacy of the proposed HMSE for CCP in comparison with the baseline and homogeneous ensemble methods. In addition, the evaluation is conducted to determine the effect of the class imbalance problem in customer churn datasets on the CCP performances of the proposed HMSE and other implemented CCP models.

Furthermore, an enhanced HMSE is designed to address the inherent class imbalance issue in the experimental datasets. Specifically, HMSE is further improved by adding SMOTE data sampling method as a preprocessing stage before the development of the model. SMOTE is a well-known data sampling approach that has been used to address the issue of class imbalance in ML tasks [53,54]. Consequently, the new variant (S-HMSE) is compared with baseline and homogeneous CCP models on the SMOTE-balanced datasets. The experimental results will illustrate the effect of the SMOTE technique on HMSE (S-HMSE) and indicate whether the CCP models under investigation are effective when applied to balanced CCP datasets.

The empirical results of the experiments with observations and inferences drawn from those experiments are analyzed to provide answers to the research questions presented in Introduction section. Regarding the model development technique, the datasets were initially divided into 80% (training dataset) −20% (testing) based on stratified sampling. This is done to ensure and maintain the distribution of classes in the training and testing datasets is like the original dataset. Thereafter, the investigated CCP models are

Table 2
Parameter setting of the implemented classifiers.

Classifiers	Classifier Type	Parameter Setting
RF		BagSise=100; breakTiesrandomly=False; calculateOutofBag=False; numInterations=100; numExecutionSlots=1; numDecimalPlaces=2
DS	Tree-based	numDecimalPlaces=2
FT		binSplit=False; errorOnProbabilities=False; ModelType=FT; numBoostongIterations=15, useAIC=False; numDecimalPlaces=2
BN	Bayesian/Probability-	Estimator = SimpleEstimator(alpha(0.5)); SearchAlgorithm = HillClimbing; ScoreType = Bayes; UseADT=False
NB	based	UseKernelEstimator=True; UseSupervisedDiscretization=False
SVM	Function-based	SVMType=C-SVC; KernelType=RadialBasisFunction; loss=0.1; eps= 0.001; cost=0.1; nu=0.5; shrinking=True; gamma=0.0
RIPPER	Rule-based	Folds=3; minNo=2.0; usePrunning=True; optimization=2
KNN	Instance-based	K = 1; distanceWeighting=False; NearestNeighbourSearchAlgorithm = LinearNNSearch; DistanceFunction=EuclideanDistance
FPA	Decision Forest Model	NumberOfTrees=10; SimpleCartMinimumRecords=2; simpleCartPruningFolds=2
Boosting	Homogeneous Ensemble	Classifier= {RF, KNN, BN, SVM, RIPPER}; resume=False; useResampling=False; weightThreshold=100; numIterations=10
Bagging	Homogeneous Ensemble	Classifier= {RF, KNN, BN, SVM, RIPPER}; CalcOutOfBag=False; storeOutOfBagPredictions=False; bagSizePercent=100; numIterations=10
Rotation Forest	Homogeneous Ensemble	Classifier= {RF, KNN, BN, SVM, RIPPER}; projectionFilter=PCA; removePercentage=50; MaxGroup=3; MinGroup=3; batchSize=100; numIterations=10
Cascade	Homogeneous Ensemble	Classifier= {RF, KNN, BN, SVM, RIPPER}; ContatenatePredictions=True; KeepOriginal=True; meta=PCT; numIterations=10
Dagging	Homogeneous Ensemble	Classifier= {RF, KNN, BN, SVM, RIPPER}; verbose=False; seed=1; batchSize=100; numIterations=10

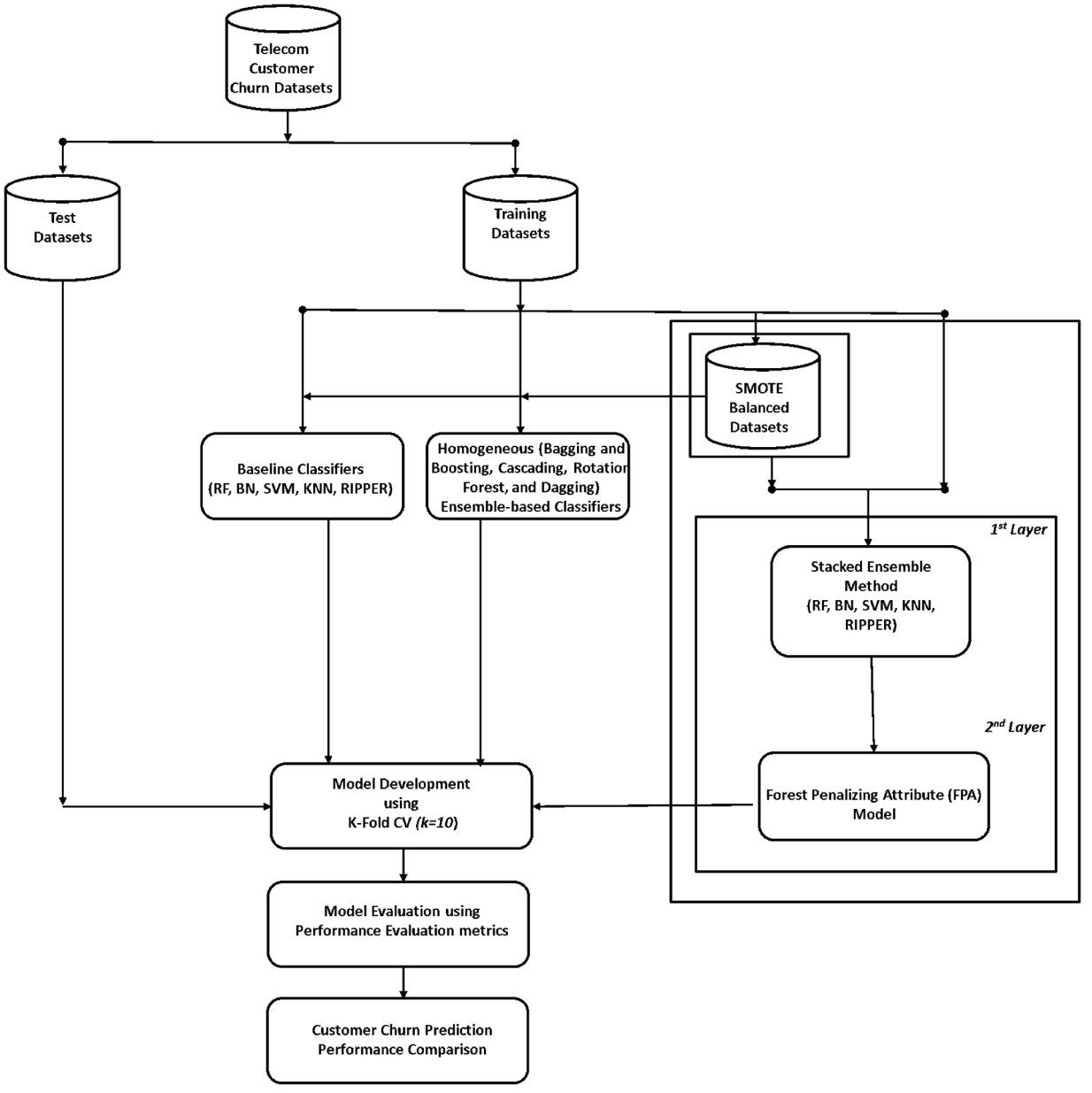

Fig. 2. Experimental Framework.

developed using the cross-validation (CV) technique for a more robust evaluation by averaging the results over multiple splits. In this research, *k*-fold (where $k = 10$) is utilized to develop the CCP models. The preference for the CV technique is based on its resilience to data quality problems that could lead to model overfitting [55–59]. For fairness, each experiment was conducted ten times to ensure consistency in the performance of the CCP models we investigated. In the end, the mean values of the produced performance metric are used for the evaluation of the implemented CCP models. The Waikato Environment for Knowledge Analysis (WEKA) machine learning library [60] and the R programming language [61] were used for the tests on an Intel(R) CoreTM computer that had 16 gigabytes of random access memory and an i7–6700 CPU that ran at 3.4 gigahertz.

Experimented datasets

For its investigations, this research relied on preexisting customer churn datasets collected from different telecommunications providers, which included demographic data and details about the services their customers had purchased. This research focused on customers who are likely to churn for distinct reasons. Two datasets from Kaggle (Dataset A) and UCI (Dataset B) repositories are utilized in this research. These customer churn datasets are freely available and widely utilized in previous CCP research [15,21,36,39,

43]. Specifically, Dataset A is sourced from the IBM business analytics community and portrays a telecommunications provider that offered phone and broadband services to its clients. It has 3333 instances with 2850 and 483 instances of non-churners (NC) and churners (C) respectively. Likewise, Dataset B has 5000 instances, but only 4493 of them are NC. Additional details regarding the churn rate and the degree of imbalance ratio are shown in Table 3.

Performance evaluation metrics

In this study, we used prominent evaluation metrics such as accuracy, f-measure, area under the curve (AUC), and Mathew's Correlation Coefficient (MCC) to compare the predictive performances of different CCP models. These performance indicators were selected due to their extensive and repeated use in previous research to evaluate ML-based CCP models [62–64]. Moreover, MCC is dependable since it considers all regions of the confusion matrix generated for every new model [54,65].

Results and discussion

This section presents and analyzes the empirical results of the various experiments conducted based on the experimental procedure described in Experimental Procedure section. The CCP performances of the proposed HMSE and S-HMSE are evaluated using the selected performance evaluation metrics as mentioned in Performance Evaluation Metrics section. Thereafter, the CCP performances of the proposed methods will be compared with the baseline classifiers and their respective homogeneous (bagging, boosting, cascade, rotation forest and dagging) variants on both original and balanced (SMOTE) datasets. In the end, the comparison of the proposed methods with existing CCP models with diverse computational characteristics is presented on each of the customer churn datasets.

CCP performances of proposed HMSE and S-HMSE methods

Tables 4 and 5 present the CCP performances of the proposed HMSE and S-HMSE methods on Datasets A and B, respectively. From Table 4, it can be observed that the CCP performances of HMSE on Dataset A are remarkable with an accuracy value of 95.02%, an AUC value of 0.903, an F-measure value of 0.948 and an MCC value of 0.879. The fact that HMSE has an accuracy value, AUC value and F-measure value above 90% on Dataset A indicates its effectiveness and efficacy for CCP. In the case of experimental results of S-HMSE on the same Dataset A, improved CCP performances were observed as shown in Table 4. Specifically, S-HMSE recorded α + 2.34%, +9.52%, +2.53% and +7.51% increment in accuracy, AUC, F-measure and MCC values respectively over HMSE on Dataset A.

Similarly, Table 5 presents the CCP performances of HMSE and S-HMSE methods on Dataset B. Like the observed CCP performance on Dataset A, HMSE recorded an accuracy value of 90.80%, an AUC value of 0.700, an F-measure value of 0.848 and an MCC value of 0.801. As observed, the CCP performances of HMSE on Dataset B are not as good as those of Dataset A. This may be due to the higher imbalance ratio in Dataset B (8.86) as compared to Dataset A (5.9). Furthermore, studies have shown that a high imbalance ratio due to the class imbalance problem reduces the prediction performances of ML models. However, the deployment of the S-HMSE model on the same Dataset B had better CCP performances and significant positive increments in its performance metrics. Specifically, S-HMSE had α + 5.46%, +21.63%, +12.97%, and +14.36% increment in accuracy, AUC, F-measure and MCC values over HMSE on Dataset B.

The superior CCP performances of S-HMSE over HMSE on both Datasets A and B can be attributed to the deployment of SMOTE data sampling method in its procedure which is absent in the HMSE method. SMOTE addressed the latent class imbalance problem in Datasets A and B by equalizing the imbalance ratio value (See Table 3), that is balancing the skewed class labels. Figs. 3 and 4 present the graphical illustrations and comparisons of HMSE and S-HMSE on Datasets A and B for better depictions.

In summary, the remarkable CCP performances of HMSE and S-HMSE on Datasets A and B, specifically the high accuracy (above 90%), AUC (above 80%), F-measure (above 80%), and MCC (above 80%) values indicate that the proposed methods can perform well for CCP even with the occurrence of latent data quality problems such as the class imbalance problem. For generalization, the CCP performances of HMSE and S-HMSE methods are further compared with high-performing baseline ML models and ensemble variants in succeeding sections.

CCP performance of HMSE and S-HMSE against baseline and homogeneous ensemble methods

In this section, the CCP performances of HMSE and S-HMSE are compared with their respective baseline classifiers (RF, KNN, SVM, RIPPER, BN), selected prominent classifiers (NB, DS, FT), and the homogeneous ensemble variants (Bagging, Boosting, Cascade, Rotation Forest, and Dagging) of the baseline classifiers on both original and balanced Datasets A and B. The essence of the comparison is to ascertain and validate the performance of the proposed methods alongside prominent baseline classifiers and homogeneous ensemble methods as used in existing studies.

For clarity, the result analysis is presented and discussed in the form of two scenarios. The first scenario presents the CCP

Table 3
Description of CCP datasets.

Dataset	Features	Instances	Churners	Non-Churner	Churn Rate	Imbalance Ratio
Dataset A	21	3333	483	2850	14.49%	5.9
Dataset B	18	5000	507	4493	10.14%	8.86

Table 4
The CCP performance of HMSE and S-HMSE on Dataset A.

CCP Models	Accuracy (%)	AUC	F-measure	MCC
HMSE	95.02	0.903	0.948	0.879
S-HMSE	97.24	0.989	0.972	0.945

Table 5
The CCP performance of HMSE and S-HMSE on Dataset B.

CCP Models	Accuracy (%)	AUC	F-measure	MCC
HMSE	90.80	0.800	0.848	0.801
S-HMSE	95.76	0.973	0.958	0.916

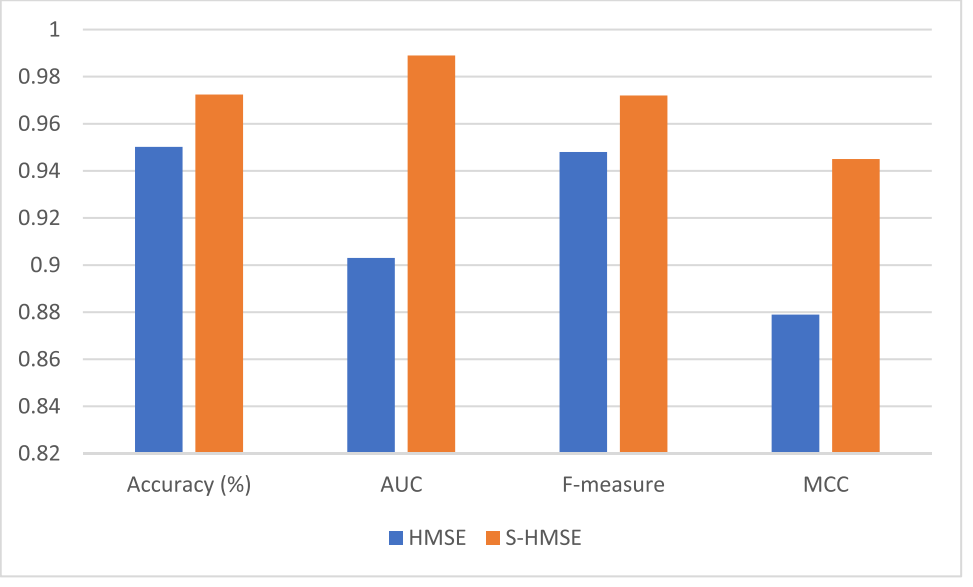

Fig. 3. Graphical representation and Comparison of CCP Performance of HMSE and S-HME on Dataset A.

performances of HMSE, the baseline ML methods and their respective homogeneous ensemble variants on original Datasets A and B. For the second scenario, the CCP performances of S-HMSE are compared with baseline ML methods and the same homogeneous ensemble variants but this time on the balanced (SMOTE) Datasets A and B.

Scenario 1: CCP performance comparison of HMSE against baseline and homogeneous ensemble models on original datasets A and B

As presented in Table 6, the CCP performances of HMSE were compared with the individual baseline classifiers such as RF, SVM, KNN, BN, and RIPPER on original Dataset A. As observed, HMSE had an accuracy value of 95.02% which is a + 4.51%, +11.12%, +13.96%, +8.87%, and +1.85% increment over the respective accuracy values of RF (90.97%), SVM (85.51%), KNN (83.38%), BN (87.38%) and RIPPER (93.29%). In addition, regarding other evaluation metrics, HMSE outperformed the baseline classifiers with superior AUC (0.903), F-measure (0.948) and MCC (0.789) values. A similar occurrence was observed with the CCP performances of HMSE on Dataset B. Specifically, Table 7 presents the CCP performance comparison of HMSE against baseline classifiers on original Dataset B.

As presented, HMSE recorded an accuracy value of 90.80% which reflects a + 1.49%, +1.05%, +10.87%, +1.05%, and +1.05% increment over the respective accuracy values of RF (89.46%), SVM (89.86%), KNN (81.90%), BN (89.86%) and RIPPER (89.86%). In addition, based on other evaluation metrics, HMSE was still superior in performance to the baseline classifiers with better AUC (0.800), F-measure (0.848) and MCC (0.801) values. The experimental results (Tables 6 and 7) demonstrate the superiority of HMSE over the baseline classifiers on the original Datasets A and B over the evaluation metrics can be observed. This finding is in line with the reports on the effectiveness of ensemble methods over individual baseline classifiers in existing studies. It is also worth noting that RIPPER (a rule-based classifier) had better CCP performances on the studied datasets than other baseline classifiers with diverse computational characteristics.

Figs. 5 and 6 present the graphical representation of the CCP performance of HSME against the baseline classifiers on Datasets A and B, respectively. As a result of these findings, further experiments were conducted to determine the effectiveness of HMSE against the

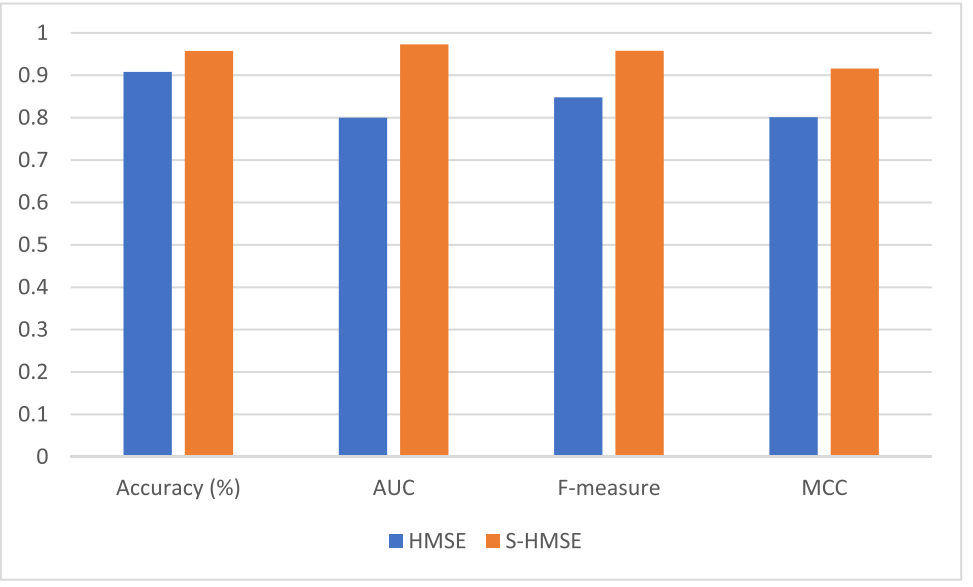

Fig. 4. Graphical representation and Comparison of CCP Performance of HMSE and S-HME on Dataset B.

Table 6
The CCP performance comparison of HMSE against baseline classifiers on Original Dataset A.

	Accuracy (%)	AUC	F-measure	MCC
HMSE	95.02	0.903	0.948	0.789
RF	90.97	0.896	0.895	0.581
SVM	85.51	0.500	?	?
KNN	83.38	0.603	0.821	0.237
BN	87.28	0.834	0.863	0.424
RIPPER	93.29	0.875	0.921	0.730

Table 7
The CCP performance comparison of HMSE against baseline classifiers on Original Dataset B.

	Accuracy (%)	AUC	F-measure	MCC
HMSE	90.80	0.800	0.848	0.801
RF	89.46	0.513	0.850	0.003
SVM	89.86	0.500	?	?
KNN	81.90	0.510	0.820	0.020
BN	89.86	0.501	?	?
RIPPER	89.86	0.498	?	?

homogenous ensemble variants of the baseline classifiers. The essence of the evaluation is to further validate the performance of HMSE with diverse ensemble variants (Bagging, Boosting, Cascade, Rotation Forest, and Dagging) of the baseline classifiers. Tables 8 and 9 present the CCP performance of HMSE against homogeneous ensemble classifiers on Original Datasets A and B.

As shown in Table 8, HMSE recorded the highest accuracy value over the homogeneous ensemble variants on original Dataset A. Although the homogeneous ensemble (Bagging, Boosting, Cascade, Rotation Forest, and Dagging) variants based on the RIPPER classifier recorded comparable accuracy values which were better than other homogeneous ensemble variants of other experimented baseline classifiers, HMSE still had the highest accuracy value. Specifically, HMSE had α + 1.56. +1.58%, +1.26%, +1.34%, and +3.6% increment in accuracy value as compared with Bagged_RIPPER, Boosted_RIPPER, RotationForest_RIPPER, Cascade_RIPPER and Dagged_RIPPER, respectively. Concerning the AUC, F-measure, and MCC values, similar findings were observed as the AUC, F-measure, and MCC values of HMSE were superior to the homogeneous ensemble variants of the baseline classifiers. In addition, as presented in Table 9, in most cases, HMSE outperformed the homogeneous ensemble variants and in some cases performed comparably to the homogeneous ensemble variants of the baseline classifiers on original Dataset B. Based on the accuracy values, differences in the value of HMSE and other homogeneous ensemble variants are insignificant. However, based on AUC and the MCC values, HMSE significantly outperformed the experimented homogeneous ensemble variants on original Dataset B. The high accuracy, F-measure, and low AUC and MCC values of the homogeneous ensemble variants is an indicator of the over-fitting tendency of the homogeneous

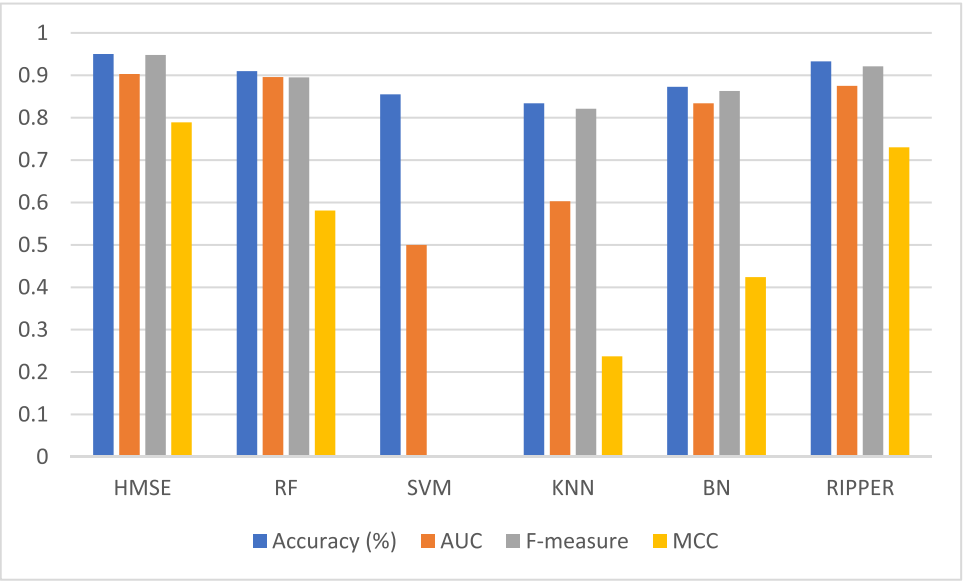

Fig. 5. Graphical representation of the CCP Performances of HMSE and baseline classifiers on Dataset A.

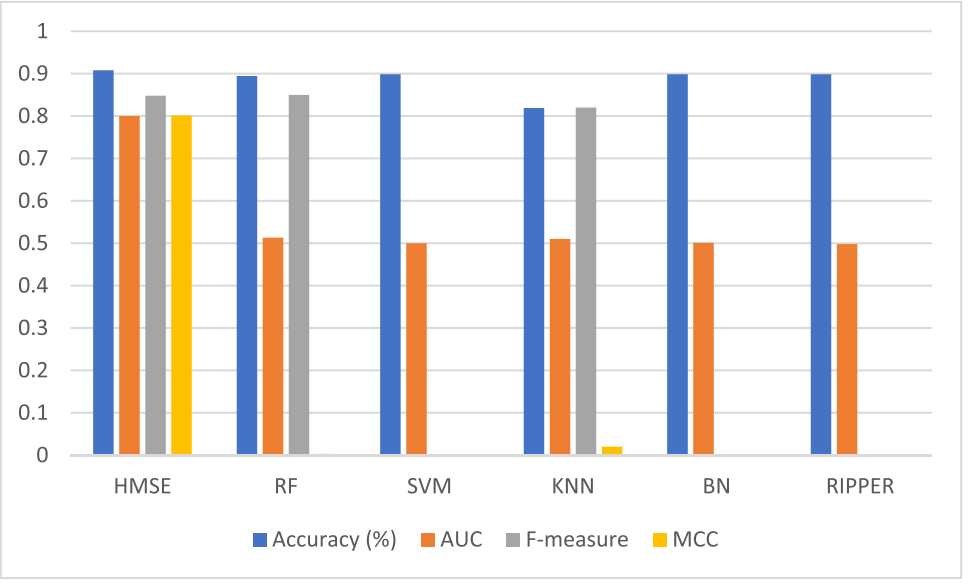

Fig. 6. Graphical representation of the CCP Performances of HMSE and baseline classifiers on Dataset B.

ensemble variant models. Moreover, the high IR of the original Dataset B could also be a factor for the poor CCP performance when compared to HMSE.

From Tables 8 and 9, it can be observed that the proposed HMSE outperformed the homogeneous ensemble variants of the baseline classifiers on original Datasets A and B. This finding further consolidates the CCP performances of the proposed HMSE model on the experimental datasets. However, it is worth noting that the CCP performances of the homogeneous ensemble variants of the base classifiers were better than the CCP performances of the individual baseline classifiers. This observation is in line with the reported findings that ensemble methods are better than individual classifiers. That is, ensemble methods can still accommodate data quality problems such as class imbalance, but they cannot be used as an explicit solution to data quality problems. Moreover, among the homogeneous ensemble methods, Cascade and Boosting ensemble methods were better than other methods on original Datasets A and B.

For a generalized CCP performance, 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 imbalance problem.

Table 8
The CCP performances of HMSE against homogeneous ensemble classifiers on Original Dataset A.

		Accuracy (%)	AUC	F-measure	MCC
	HMSE	95.02	0.903	0.948	0.789
Bagging	RF	90.16	0.858	0.881	0.533
	SVM	85.50	0.500	0.822	?
	KNN	84.22	0.676	0.827	0.257
	BN	87.88	0.845	0.866	0.434
	RIPPER	93.56	0.860	0.854	0.711
Boosting	RF	91.36	0.899	0.900	0.602
	SVM	85.51	0.501	?	?
	KNN	83.38	0.603	0.821	0.237
	BN	87.19	0.813	0.800	0.424
	RIPPER	93.54	0.896	0.843	0.736
Cascade	RF	91.09	0.804	0.897	0.587
	SVM	85.51	0.500	?	?
	KNN	83.38	0.603	0.820	0.232
	BN	87.28	0.834	0.863	0.424
	RIPPER	93.84	0.863	0.846	0.731
Rotation Forest	RF	88.90	0.888	0.860	0.453
	SVM	86.77	0.634	0.822	0.272
	KNN	83.44	0.661	0.811	0.176
	BN	89.41	0.854	0.885	0.517
	RIPPER	93.76	0.809	0.832	0.727
Dagging	RF	85.90	0.880	0.798	0.152
	SVM	85.51	0.500	?	?
	KNN	86.20	0.757	0.809	0.202
	BN	86.83	0.843	0.849	0.354
	RIPPER	91.72	0.867	0.807	0.623

Table 9
The CCP performances of HMSE against homogeneous ensemble classifiers on Original Dataset B.

		Accuracy (%)	AUC	F-measure	MCC
	HMSE	90.80	0.800	0.848	0.801
Bagging	RF	89.72	0.521	0.830	0.001
	SVM	89.86	0.500	?	?
	KNN	82.80	0.494	0.824	0.011
	BN	89.86	0.501	?	?
	RIPPER	89.86	0.502	?	?
Boosting	RF	88.72	0.516	0.839	0.016
	SVM	89.86	0.486	?	?
	KNN	81.90	0.510	0.820	0.020
	BN	89.86	0.488	?	?
	RIPPER	89.86	0.493	?	?
Cascade	RF	89.56	0.507	0.830	0.009
	SVM	89.86	0.500	?	?
	KNN	81.86	0.498	0.818	−0.003
	BN	89.86	0.494	?	?
	RIPPER	89.86	0.499	?	?
Rotation Forest	RF	89.80	0.520	0.830	−0.008
	SVM	89.86	0.500	?	?
	KNN	84.96	0.527	0.837	0.039
	BN	89.86	0.491	?	?
	RIPPER	89.86	0.498	?	?
Dagging	RF	89.86	0.489	?	?
	SVM	89.86	0.502	?	?
	KNN	89.78	0.510	0.830	−0.010
	BN	89.86	0.493	?	?
	RIPPER	89.86	0.497	?	?

Specifically, a data over-sampling method (SMOTE) is utilized to balance the frequency of the minority and majority class labels in the experimented datasets. Hence, Scenario 2 showcases the CCP performances of S-HMSE against the baseline and homogeneous ensemble variants of the baseline classifiers on a balanced (SMOTE) Datasets A and B.

Scenario 2: CCP performance of S-HMSE against baseline and homogeneous ensemble models on balanced (SMOTE) datasets A and B

Tables 10 and 11 present the CCP performances of HMSE and the individual baseline classifiers (RF, SVM, KNN, BN, and RIPPER) on the balanced (SMOTE) Datasets A and B, respectively. As shown in Table 10, S-HMSE had an accuracy value of 97.24%, an AUC value of 0.989, an F-measure value of 0.972, and an MCC value of 0.945 which represent +2.34%, +9.52%, +2.53%, and +19.77% increment over HMSE (Accuracy (95.02%), AUC (0.903), F-measure (0.948), and MCC (0.789)) on the same Dataset A. This finding indicates that S-HMSE is more effective for CCP than HMSE on Dataset A. In addition, S-HMSE had α + 5.54%, +24.62%, +10.16%, +5.06%, and +2.05% increment over the respective accuracy values of RF (92.14%), SVM (78.03%), KNN (88.27%), BN (92.56%) and RIPPER (95.29%). A similar superior CCP performance outcome was observed with the AUC (0.989), F-measure (0.972) and MCC (0.945) values of S-HMSE over the baseline classifiers on balanced Dataset A. Likewise, on balanced Dataset B, similar performance occurrences were observed with the CCP performances of S-HMSE. As presented in Table 11, the CCP performances of S-HMSE against baseline classifiers on balanced Dataset B were analyzed.

According to Table 10, S-HMSE obtained accuracy values of 95.76%, 0.973 for AUC, 0.958 for F-measure, and 0.916 for MCC values, which are +5.46%, +21.63%, +12.97%, and +14.36% higher than HMSE's CCP performances (accuracy values of 90.80%, 0.800 for AUC, 0.848 for F-measure, and 0.801 for MCC) on Dataset B. According to this result, S-HMSE shows better CCP performances on Dataset B than HMSE. Furthermore, compared to the corresponding accuracy values of RF (90.32%), SVM (79.30%), KNN (83.91%), BN (93.02%), and RIPPER (89.56%), S-HMSE had α + 6.02%, +20.76%, +14.12%, +2.95%, and +6.92% increase. With AUC (0.989), F-measure (0.972), and MCC (0.945) values of S-HMSE above the investigated baseline classifiers on balanced Dataset A, a comparable improved CCP performance outcome was observed.

Figs. 7 and 8 present the graphical representation of the CCP performance of S-HSME against the baseline classifiers on balanced (SMOTE) Datasets A and B, respectively. Further experiments were done to compare the effectiveness of the homogeneous ensemble variations of the baseline classifiers to these observations. The main goal of the assessment is to further test the effectiveness of S-HMSE using several homogeneous ensemble variations of the base classifiers (Bagging, Boosting, Cascade, Rotation Forest, and Dagging). The CCP performances of S-HMSE with homogeneous ensemble classifiers on balanced (SMOTE) Datasets A and B are shown in Tables 12 and 13.

As presented in Table 12, the proposed S-HMSE model outperformed the homogeneous ensemble variants on balanced Dataset A based on the evaluation metrics studied. Also in this case, the homogeneous ensemble variants (Bagging, Boosting, Cascade, Rotation Forest, and Dagging) based on the RIPPER classifier had comparable CCP performances which were better than other homogeneous ensemble variants of other baseline classifiers. However, S-HMSE still had better CCP performances on the balanced Dataset A. It was also observed that the CCP performances of the homogeneous ensemble variants improved on balanced Dataset A. Specifically, the AUC and MCC values of the homogeneous ensemble variants are more than 90% in most cases. On the balanced Dataset B, S-HMSE still had better CCP performances than the homogeneous ensemble variants of the baseline classifiers. Similar experimental results were observed on the enhanced CCP performances of the S-HMSE and the homogeneous ensemble variants were noticed. This observation supports the initial findings on Dataset A that the alleviation of the class imbalance issue can further improve the CCP performances of the models we investigated. It can be observed from Table 13 that the homogeneous ensemble methods are also comparable in CCP performances to the proposed S-HMSE and that the introduction of SMOTE enhanced their respective CCP performance as with S-HMSE. That is, the deployed SMOTE data sampling method was able to balance the class labels thereby making the IR value 0. Consequently, the homogeneous ensemble variants were able to avoid the overfitting tendency. Nonetheless, S-HMSE had a superior CCP performance on the balanced Dataset B than other methods.

In summary, based on the observed findings from Tables 12 and 13, the suggested S-HMSE outperformed the homogeneous ensemble variants of the baseline classifiers on the balanced Datasets A and B. This observation further consolidates the CCP performances of the proposed S-HMSE on the experimented datasets. Similarly, it was observed that the CCP performances of the homogeneous ensemble variants of the base classifiers were better than the CCP performances of the individual baseline classifiers on the balanced datasets. Also, it was deduced that the CCP performances of the homogeneous ensemble variants of the base classifiers on the balanced datasets were better than the CCP performances of homogeneous ensemble variants of the base classifiers on the original datasets. In other words, ensemble methods with appropriate data sampling can address data quality problems such as class imbalance and generate effective ML models with high prediction performances. Hence, it is recommended that researchers consider using ensemble methods with appropriate data sampling approaches for ML tasks when encountering data quality problems.

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.

Table 10
The CCP performance of S-HMSE against baseline classifiers on balanced (SMOTE) Dataset A.

	Accuracy (%)	AUC	F-measure	MCC
S-HMSE	97.24	0.989	0.972	0.945
RF	92.14	0.843	0.769	0.843
SVM	78.03	0.780	0.769	0.624
KNN	88.27	0.881	0.883	0.767
BN	92.56	0.931	0.925	0.858
RIPPER	95.29	0.934	0.933	0.926

Table 11
The CCP performance of S-HMSE against baseline classifiers on balanced (SMOTE) Dataset B.

	Accuracy (%)	AUC	F-measure	MCC
S-HMSE	95.76	0.973	0.958	0.916
RF	90.32	0.946	0.903	0.806
SVM	79.30	0.793	0.793	0.586
KNN	83.91	0.839	0.839	0.678
BN	93.02	0.935	0.910	0.865
RIPPER	89.56	0.908	0.895	0.796

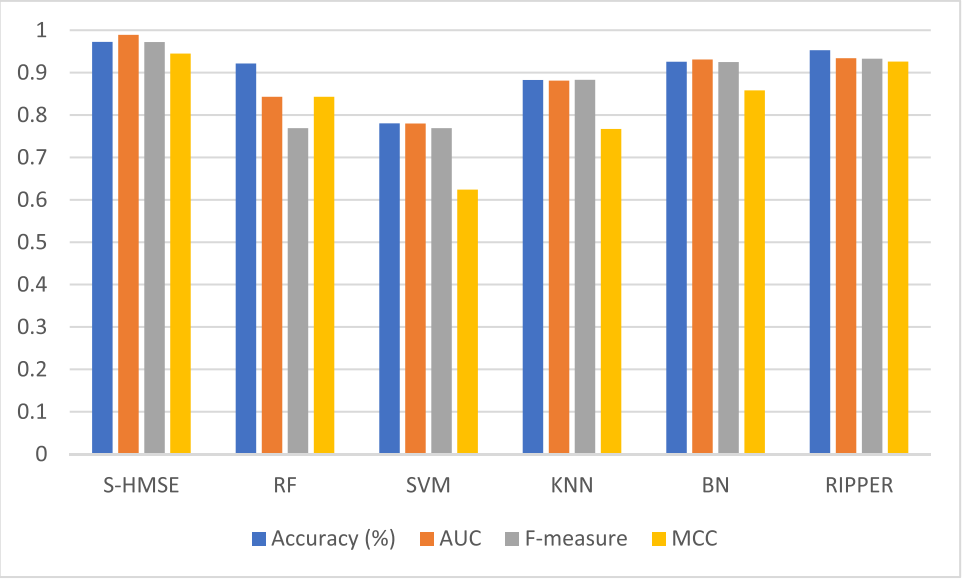

Fig. 7. Graphical representation of the CCP Performances of S-HMSE and baseline classifiers on Balanced (SMOTE) Dataset A.

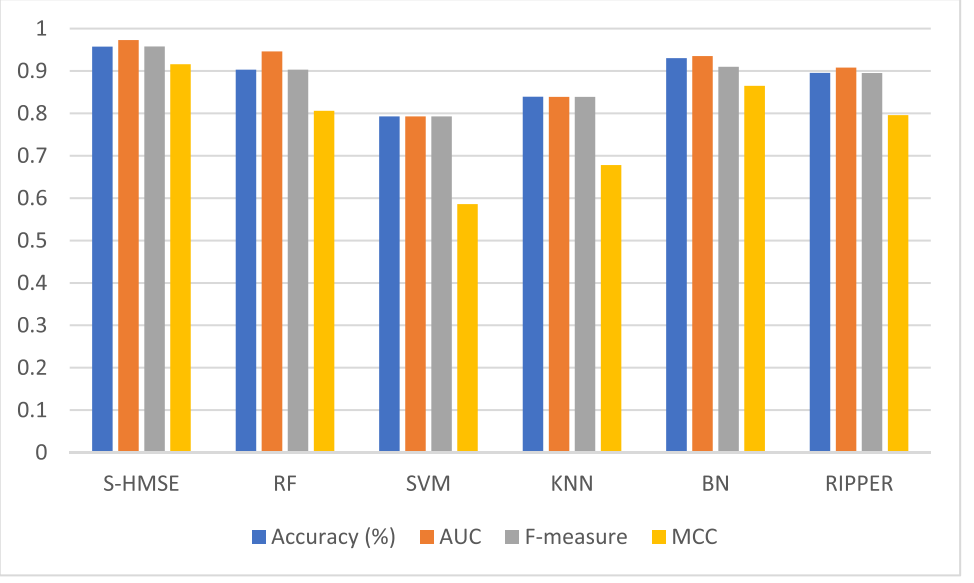

Fig. 8. Graphical representation of the CCP Performances of S-HMSE and baseline classifiers on Balanced (SMOTE) Dataset B.

Table 12
The CCP performances of HMSE against homogeneous ensemble classifiers on Balanced (SMOTE) Dataset A.

		Accuracy (%)	AUC	F-measure	MCC
	S-HMSE	97.24	0.989	0.972	0.945
Bagging	RF	92.30	0.971	0.923	0.846
	SVM	76.49	0.818	0.751	0.600
	KNN	88.06	0.931	0.881	0.762
	BN	92.63	0.963	0.926	0.860
	RIPPER	95.64	0.966	0.936	0.903
Boosting	RF	92.30	0.973	0.923	0.847
	SVM	79.51	0.792	0.786	0.647
	KNN	88.27	0.881	0.883	0.767
	BN	91.33	0.966	0.913	0.827
	RIPPER	95.18	0.964	0.932	0.904
Cascade	RF	92.70	0.974	0.927	0.854
	SVM	78.35	0.783	0.773	0.629
	KNN	86.65	0.865	0.866	0.735
	BN	92.56	0.961	0.925	0.858
	RIPPER	95.11	0.965	0.931	0.903
Rotation Forest	RF	92.70	0.974	0.927	0.855
	SVM	88.31	0.909	0.883	0.766
	KNN	88.67	0.921	0.887	0.774
	BN	91.58	0.965	0.916	0.832
	RIPPER	94.81	0.964	0.948	0.897
Dagging	RF	87.90	0.949	0.878	0.757
	SVM	51.76	0.859	0.370	0.129
	KNN	84.82	0.928	0.848	0.700
	BN	90.52	0.956	0.905	0.812
	RIPPER	92.07	0.968	0.921	0.842

Table 13
The CCP performances of S-HMSE against homogeneous ensemble classifiers on Balanced (SMOTE) Dataset B.

		Accuracy (%)	AUC	F-measure	MCC
	S-HMSE	95.76	0.973	0.958	0.916
Bagging	RF	90.06	0.945	0.901	0.801
	SVM	79.31	0.805	0.793	0.586
	KNN	84.44	0.902	0.844	0.689
	BN	93.44	0.946	0.934	0.873
	RIPPER	91.67	0.948	0.917	0.836
Boosting	RF	89.97	0.930	0.900	0.799
	SVM	79.32	0.852	0.793	0.587
	KNN	83.91	0.839	0.839	0.678
	BN	93.00	0.939	0.930	0.865
	RIPPER	89.48	0.938	0.895	0.790
Cascade	RF	90.19	0.943	0.902	0.804
	SVM	79.30	0.793	0.793	0.586
	KNN	82.38	0.824	0.824	0.648
	BN	93.19	0.944	0.932	0.868
	RIPPER	89.30	0.909	0.893	0.791
Rotation Forest	RF	91.82	0.951	0.850	0.837
	SVM	79.30	0.795	0.793	0.586
	KNN	87.91	0.918	0.879	0.760
	BN	89.86	0.517	0.518	0.536
	RIPPER	90.91	0.942	0.909	0.820
Dagging	RF	85.21	0.916	0.852	0.706
	SVM	79.21	0.830	0.792	0.584
	KNN	84.27	0.914	0.842	0.691
	BN	80.26	0.864	0.803	0.606
	RIPPER	76.73	0.831	0.765	0.545

CCP performance of HMSE and S-HMSE against existing CCP models

The CCP performances of the proposed HMSE and S-HMSE models as well as the current CCP solutions are shown in Tables 14 and 15, respectively, for Datasets A and B.

As shown in Table 14, the CCP performance of HMSE and S-HMSE models are compared to existing CCP models such as [5,36,42,43,66–70] on Dataset A. In contrast, current CCP solutions consist of ensembles, hybrids, and advanced DL models. As an example, [66] 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, [42] developed a BaggedMLP for CCP performance with accuracy and F-measure values of 94.15% and 0.874 respectively. Also, [68] suggested the amplification of logistic regressing with a boosting ensemble method for CCP. Regardless, the CCP performances of HMSE and S-HMSE models outperformed these ensemble-based CCP models. Using hybrid methods, [5] hybridized Social Network Analysis (SNA) and XGBoost for CCP. Similarly, [68] 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 HMSE and S-HMSE. Some existing methods are based on the combination of clustering and classification models like in the case of [70,43]. A probabilistic-based fuzzy local information c-means (PFLICM) for CCP was suggested by [70,43] 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 [43] are high. Furthermore, the performance of HMSE and S-HMSE is compared with current CCP models such as [36,69] that were implemented on sophisticated DL techniques. Specifically, [36] hyper-parameterized DL+RMSProp while [69] 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 HMSE and S-HMSE models outperformed the current CCP models that were investigated using various computational methods on Dataset A.

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 (Introduction section).

RQ1. How effective are the proposed HMSE and S-HMSE models in comparison with baseline and homogeneous ensemble methods in CCP?

The proposed HMSE and S-HMSE models outperformed the individual baseline models (RF, SVM, RIPPER, BN, and KNN) as well as the homogeneous (Bagging, Boosting, Cascade, Rotation Forest, and Dagging) ensemble variations of the baseline models on both the original and balanced (SMOTE) CCP datasets (Datasets A and B). The baseline classifiers are prominent ML classifiers with diverse computational characteristics that have been widely used for CCP and other ML tasks. In addition, the CCP performances of the recommended S-HMSE model were enhanced by the SMOTE approach we adopted to resolve the innate class imbalance issue present in the CCP datasets.

RQ2. How effective are the proposed HMSE and S-HMSE models against the state-of-the-art existing rule, ML, and DL-based CCP solutions?

The proposed HMSE and S-HMSE models outperformed some existing CCP solutions with diverse computational methodologies and capabilities. Specifically, HMSE and S-HMSE outperformed some current ensemble, hybrid, and DL-based CCP models on Datasets A and B.

Table 14
The CCP performance of HMSE and S-HMSE against Existing CCP models on Dataset A.

CCP Models	Accuracy (%)	AUC	F-measure	MCC
HMSE	95.02	0.903	0.948	0.789
S-HMSE	97.24	0.989	0.972	0.945
Tavassoli and Koosha [66] (BNNGA)	86.81	–	0.688	–
Ahmad, Jafar [5] (SNA + XGBOOST)	–	0.933	–	–
Jain, Khunteta [67] (CNN+VAE)	90.00	–	0.930	–
Saghir, Bibi [42] (BaggedMLP)	94.15	–	0.874	–
Jain, Khunteta [68] (LogitBoost)	85.24	0.717	0.810	0.160
Jeyakarthic and Venkatesh [69] (P-AGBPNN)	91.71	–	0.951	
Praseeda and Shivakumar [70] (PFLICM)	95.41	–	–	-
Dalli [36] (Hyper-Parameterized DL with RMSProp)	86.50	–	–	–
Bilal, Almazroi [43] (K-med+GBT+DT+DL+Voting)	92.40	–	0.662	–
Bilal, Almazroi [43] (K-med+GBT+DT+DL+Stacking)	92.40	–	0.717	–
Bilal, Almazroi [43] (K-med+GBT+DT+DL+Adaboost)	92.43	–	0.718	–
Bilal, Almazroi [43] (K-med+GBT+DT+DL+Bagging)	92.41	–	0.664	–

Table 15
The CCP performance of HMSE and S-HMSE against Existing CCP models on Dataset B.

CCP Models	Accuracy (%)	AUC	F-measure	MCC
HMSE	90.80	0.700	0.848	0.801
S-HMSE	95.76	0.973	0.958	0.916
Tavassoli and Koosha [66](BBNGA)	77.50	–	0.773	–
Saghir, Bibi [42] (Bagging)	80.80	–	0.784	–
Shaaban, Helmy [71] (SVM)	83.70	–	–	–
Bilal, Almazroi [43] (KMed+GBT+DL+DL+Voting)	94.06	–	0.745	–
Bilal, Almazroi [43] (KMed+GBT+DL+DL+Stacking)	94.65	–	0.796	–
Bilal, Almazroi [43] (KMed+GBT+DL+DL+Adaboost)	94.70	–	0.806	–
Bilal, Almazroi [43] (KMed+GBT+DL+DL+Bagging)	94.12	–	0.746	–
Kumar and Kumar [72]	84.26	–	0.900	–
Long Short Term Memory (LSTM) [43]	90.04	–	0.821	–
Gated Recurrent Unit (GRU) [43]	90.10	–	0.846	–
CNN[43]	89.80	–	0.826	–

Threat to validity

The evaluation and mitigation of risks to the reliability of experimental findings is a crucial part of every empirical investigation [15,44]. The following threats to the validity of this research work as observed are presented thus:

External validity: The reliability of scientific investigations relies heavily on their transferability to real-world settings. Insights from experiments may not apply to other situations for a variety of reasons depending on the nature and quantity of the datasets deployed. This has led to the utilization of two widely-used CCP datasets with a wide variety of attributes in this research work. These public datasets are used extensively for the development and evaluation of CCP models, and they are freely available to the public. In addition, this research offered a comprehensive evaluation of the experimental technique, which may improve the repeatability and validity of its methodological methods across a range of CCP datasets.

Internal validity: This paradigm represents the relevance and coherence of data, experimented models, and experimental investigation. Therefore, this research work employs some well-known ML techniques implemented and applied in past studies. We chose these ML methods because they represent a wide range of approaches and are proven to be successful in ML tasks. In addition, the CV approach was used to meticulously train the examined CCP models on the selected CCP datasets, and each experiment was run 10 times for validity. This strategy helped reduce the likelihood of unanticipated inconsistencies in empirical results. On the other hand, more research may look at other strategies and methods for evaluating models in the future.

Construct validity: This problem is associated with the standards used to judge the success of the CCP models we studied. This research work applied several statistical performance metrics such as accuracy, AUC, f-measure, and MCC. These measures provided a comprehensive empirical assessment of the CCP models employed in the research. The churning process and its state were also essential considerations in the design of the CCP models.

Conclusions and future works

This research work proposed a data-sampling-based heterogeneous multi-layer stacking ensemble method (S-HMSE) for CCP. Categorically, the prediction capabilities of five heterogeneous ML (RF, BN, SVM, KNN, and RIPPER) classifiers with distinct computational characteristics were ensembled based on the stacking method and the resulting model is further improved using the FPA model as a meta-learner on the original (imbalanced) and balanced (SMOTE) telecommunication customer churn datasets. The viability and effectiveness of the proposed S-HMSE models were tested via experiments. The experimental findings observed on the studied CCP datasets (original and balanced) indicated the superiority of the S-HMSE over baseline ML models. In addition, S-HMSE outperformed homogeneous ensemble methods (bagging and boosting) on both original and balanced CCP datasets. These observed findings validate the applicability and efficacy of S-HMSE for CCP. In an extended evaluation, the prediction performances of S-HMSE in most cases outperformed existing rule, ML, and DL-based CCP solutions on the publicly available Kaggle and UCI CCP datasets. Consequently, this research work recommends the deployment of the proposed HMSE and S-HMSE models for CCP.

As a continuation of this research, the effect of data quality problems such as outliers and extreme values on predictive performance of CCP models will be investigated. Outliers and extreme values in a dataset can have a major impact on a classifier's predictive abilities. They have the tendency to distort the dataset, creating an imbalanced and tainted dataset, as well as making the predictive model less dependable, which leads to erroneous projections. Besides, findings from current research have shown that predictive models react and adapt variably to these data quality issues. Outliers and extreme values, for example, may have a major effect on distance-based and, in most cases, linear ML models such as kNN, LR (linear), and SVM (linear and non-linear), because these ML models critical rely on the spread or distance between data points from formulating decision boundaries in their respective classification processes. However, there are no studies on the extensive investigation of these data quality issues singly, collectively, or in conjunction to other notable data quality issues such as the class imbalance problem and high dimensionality. Additionally, the projected customer turnover behavior will be explored, as customers having a low churn propensity could prove beneficial in the future. This knowledge may assist organizations in making informed and strategic decisions about client retention which are susceptible to churning. Nonetheless, further research on the concepts is anticipated.

Availability of data and materials

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

CRediT authorship contribution statement

Fatima E. Usman-Hamza: Supervision, Conceptualization, Methodology, Writing – original draft. **Abdullateef O. Balogun:** Conceptualization, Methodology, Software, Investigation, Writing – original draft. **Ramoni T. Amosa:** 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.

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.

Acknowledgements

This research/paper was fully supported by Universiti Teknologi PETRONAS, under the STIRF Research Grant Scheme (015LA0-049).

References

[1] K. Su, Y. Zhao, Y. Wang, Customer concentration and corporate financialization: evidence from non-financial firms in China, Res. Int. Bus. Financ. 68 (2024) 102159.
[2] S. Sukrat, A. Leeraphong, A digital business transformation maturity model for micro enterprises in developing countries, Glob. Bus. Organ. Excell. 43 (2) (2024) 149–175.
[3] H. Abbasimehr, M. Setak, M. Tarokh, A neuro-fuzzy classifier for customer churn prediction, Int. J. Comput. Appl. 19 (8) (2011) 35–41.
[4] S. Agrawal, et al., Customer churn prediction modelling based on behavioural patterns analysis using deep learning, in: Proceedings of the International Conference on Smart Computing and Electronic Enterprise (ICSCEE), IEEE, 2018.
[5] 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.
[6] A. Amin, et al., Customer churn prediction in the telecommunication sector using a rough set approach, Neurocomputing 237 (2017) 242–254.
[7] R. Soltani, et al., Competitive pricing of complementary telecommunication services with subscriber churn in a duopoly, Expert Syst. Appl. 237 (2024) 121447.
[8] A.C. Louro, C.G. Pugirá, R.S. Murari, A scoping review for churn prediction: step-by-step tutorial and reproducible R code, Int. J. Bus. Forecast. Mark. Intell. 9 (2) (2024) 160–178.
[9] U.A. Bhale, H.S. Bedi, Customer churn construct: literature review and bibliometric study, Manag. Dyn. 24 (1) (2024) 1.
[10] H. Ribeiro, et al., Determinants of churn in telecommunication services: a systematic literature review, Manag. Rev. Q. (2023) 1–38, https://doi.org/10.1007/s11301-023-00335-7.
[11] Y. Beeharry, R. Tsokizep Fokone, Hybrid approach using machine learning algorithms for customers' churn prediction in the telecommunications industry, Concurr. Comput. Pract. Exp. 34 (4) (2022) e6627.
[12] S. Saha, et al., ChurnNet: Deep Learning Enhanced Customer Churn Prediction in Telecommunication Industry, IEEE Access, 2024.
[13] C. Zhang, et al., Customer churn model based on complementarity measure and random forest, in: Proceedings of the International Conference on Computer, Blockchain and Financial Development (CBFD), IEEE, 2021.
[14] T. Zhang, S. Moro, R.F. Ramos, A data-driven approach to improve customer churn prediction based on telecom customer segmentation, Future Internet 14 (3) (2022) 94.
[15] F.E. Usman-Hamza, et al., Intelligent decision forest models for customer churn prediction, Appl. Sci. 12 (16) (2022) 8270.
[16] F.E. Usman-Hamza, et al., Empirical analysis of tree-based classification models for customer churn prediction, Sci. Afr. (2023) e02054.
[17] Y. Huang, B. Huang, M.T. Kechadi, A rule-based method for customer churn prediction in telecommunication services, in: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2011.
[18] I. Brandusoiu, G. Toderean, Churn prediction in the telecommunications sector using support vector machines, Margin 1 (2013) x1.
[19] I. Brânduşoiu, G. Toderean, H. Beleiu, Methods for churn prediction in the pre-paid mobile telecommunications industry, in: Proceedings of the International Conference on Communications (COMM), IEEE, 2016.
[20] M.M. Hossain, M.S. Miah, Evaluation of different SVM kernels for predicting customer churn, in: Proceedings of the 18th International Conference on Computer and Information Technology (ICCIT), IEEE, 2015.
[21] I. AlShourbaji, et al., Anovel HEOMGA approach for class imbalance problem in the application of customer churn prediction, SN Comput. Sci. 2 (6) (2021) 1–12.
[22] A.O. Balogun, et al., Software defect prediction: analysis of class imbalance and performance stability, J. Eng. Sci. Technol. 14 (6) (2019) 3294–3308.
[23] J.L. Leevy, et al., A survey on addressing high-class imbalance in big data, J. Big Data 5 (1) (2018) 1–30.
[24] L. Wang, et al., Addressing class imbalance in federated learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021.
[25] Y.A. Alsariera, A.V. Elijah, A.O. Balogun, Phishing website detection: forest by penalizing attributes algorithm and its enhanced variations, Arab. J. Sci. Eng. 45 (12) (2020) 10459–10470.
[26] Y.A. Alsariera, et al., Intelligent tree-based ensemble approaches for phishing website detection, J. Eng. Sci. Technol 17 (2022) 563–582.
[27] M.A. Mabayoje, et al., Parameter tuning in KNN for software defect prediction: an empirical analysis, J. Teknol. Sist. Komput. 7 (4) (2019) 121–126.
[28] S. Asadi, Evolutionary fuzzification of RIPPER for regression: case study of stock prediction, Neurocomputing 331 (2019) 121–137.
[29] M.J. Shabankareh, et al., A stacking-based data mining solution to customer churn prediction, J. Relationsh. Mark. 21 (2) (2022) 124–147.
[30] N.I. Mohammad, et al., Customer churn prediction in telecommunication industry using machine learning classifiers, in: Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, 2019.
[31] C. Kirui, et al., Predicting customer churn in mobile telephony industry using probabilistic classifiers in data mining, Int. J. Comput. Sci. Issues (IJCSI) 10 (2) (2013) 165. Part 1.
[32] M.O. Arowolo, et al., Customer churn prediction in telecommunication industry using decision tree and artificial neural network algorithms, Indones. J. Electr. Eng. Inform. (IJEEI) 10 (2) (2022).

[33] P. Lalwani, et al., Customer churn prediction system: a machine learning approach, Computing 104 (2) (2022) 271–294.
[34] S. Cao, et al., Deep learning based customer churn analysis, in: Proceedings of the 11th International Conference on Wireless Communications and Signal Processing (WCSP), IEEE, 2019.
[35] T.W. Cenggoro, et al., Deep learning as a vector embedding model for customer churn, Procedia Comput. Sci. 179 (2021) 624–631.
[36] A. Dalli, Impact of hyperparameters on deep learning model for customer churn prediction in telecommunication sector, Math. Probl. Eng. 2022 (2022).
[37] 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.
[38] M. Karanovic, et al., Telecommunication services churn prediction-deep learning approach, in: Proceedings of the 26th Telecommunications Forum (TELFOR), IEEE, 2018.
[39] S. Wael Fujo, S. Subramanian, M.A. Khder, Customer churn prediction in telecommunication industry using deep learning, Inf. Sci. Lett. 11 (1) (2022) 24.
[40] A. Mishra, U.S. Reddy, A comparative study of customer churn prediction in telecom industry using ensemble based classifiers, in: Proceedings of the International Conference on Inventive Computing and Informatics (ICICI, IEEE, 2017.
[41] T. Xu, Y. Ma, K. Kim, Telecom churn prediction system based on ensemble learning using feature grouping, Appl. Sci. 11 (11) (2021) 4742.
[42] M. Saghir, et al., Churn prediction using neural network based individual and ensemble models, in: Proceedings of the 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), IEEE, 2019.
[43] S.F. Bilal, et al., 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.
[44] A.G. Akintola, et al., Empirical analysis of forest penalizing attribute and its enhanced variations for android malware detection, Appl. Sci. 12 (9) (2022) 4664.
[45] A. Cutler, D.R. Cutler, J.R. Stevens, Random forests. Ensemble Machine Learning, Springer, 2012, pp. 157–175.
[46] A. Antoniadis, S. Lambert-Lacroix, J.M. Poggi, Random forests for global sensitivity analysis: a selective review, Reliab. Eng. Syst. Saf. 206 (2021) 107312.
[47] M. Mohammadi, et al., A comprehensive survey and taxonomy of the SVM-based intrusion detection systems, J. Netw. Comput. Appl. 178 (2021) 102983.
[48] S. Zhang, et al., A novel kNN algorithm with data-driven k parameter computation, Pattern Recognit. Lett. 109 (2018) 44–54.
[49] S.Y. Yerima, et al., A new android malware detection approach using bayesian classification, in: Proceedings of the IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), IEEE, 2013.
[50] D. Atienza, C. Bielza, P. Larrañaga, PyBNesian: an extensible python package for Bayesian networks, Neurocomputing 504 (2022) 204–209.
[51] P. Xu, Z. Ding, M. Pan, A hybrid interpretable credit card users default prediction model based on RIPPER, Concurr. Comput. Pract. Exp. 30 (23) (2018) e4445.
[52] T. Van Phong, et al., Landslide susceptibility mapping using Forest by Penalizing Attributes (FPA) algorithm based machine learning approach, Vietnam J. Earth Sci. 42 (3) (2020) 237–246.
[53] A.O. Balogun, et al., SMOTE-based homogeneous ensemble methods for software defect prediction, in: Proceedings of the International Conference on Computational Science and its Applications, Springer, 2020.
[54] A.O. Balogun, et al., Empirical analysis of data sampling-based ensemble methods in software defect prediction, in: Proceedings of the International Conference on Computational Science and Its Applications, Springer, 2022.
[55] A.O. Balogun, et al., Cascade generalization based functional tree for website phishing detection, in: Proceedings of the International Conference on Advances in Cyber Security, Springer, 2021.
[56] A.O. Balogun, et al., Improving the phishing website detection using empirical analysis of function tree and its variants, Heliyon 7 (7) (2021) e07437.
[57] A.O. Balogun, et al., Software defect prediction using ensemble learning: an ANP based evaluation method, FUOYE J. Eng. Technol. 3 (2) (2018) 50–55.
[58] A.O. Balogun, et al., Search-based wrapper feature selection methods in software defect prediction: an empirical analysis, in: Proceedings of the Computer Science On-line Conference, Springer, 2020.
[59] A.O. Balogun, et al., Optimized decision forest for website phishing detection, in: Proceedings of the Computational Methods in Systems and Software, Springer, 2021.
[60] M. Hall, et al., The WEKA data mining software: an update, ACM SIGKDD Explor. Newsl. 11 (1) (2009) 10–18.
[61] M.J. Crawley, The R Book, John Wiley & Sons, 2012.
[62] V.E. Adeyemo, et al., Ensemble-based logistic model trees for website phishing detection, in: Proceedings of the International Conference on Advances in Cyber Security, Springer, 2020.
[63] R. Jimoh, et al., A PROMETHEE based evaluation of software defect predictors, J. Comput. Sci. Appl. 25 (1) (2018) 106–119.
[64] B.J. Odejide, et al., An empirical study on data sampling methods in addressing class imbalance problem in software defect prediction, in: Proceedings of the Computer Science On-line Conference, Springer, 2022.
[65] A.G. Akintola, et al., Performance analysis of machine learning methods with class imbalance problem in android malware detection, Int. J. Interact. Mob. Technol. 16 (2022) 140–162.
[66] S. Tavassoli, H. Koosha, Hybrid ensemble learning approaches to customer churn prediction, Kybernetes 51 (3) (2022) 1062–1088.
[67] Jain, H., A. Khunteta, and S.P. Shrivastav, Telecom churn prediction using seven machine learning experiments integrating features engineering and normalization. 2021.
[68] H. Jain, A. Khunteta, S. Srivastava, Churn prediction in telecommunication using logistic regression and logit boost, Procedia Comput. Sci. 167 (2020) 101–112.
[69] M. Jeyakarthic, S. Venkatesh, An effective customer churn prediction model using adaptive gain with back propagation neural network in cloud computing environment, J. Res. Lepid. 51 (1) (2020) 386–399.
[70] 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.
[71] E. Shaaban, et al., A proposed churn prediction model, Int. J. Eng. Res. Appl. 2 (4) (2012) 693–697.
[72] S. Kumar, M. Kumar, Predicting customer churn using artificial neural network, in: Proceedings of the International Conference on Engineering Applications of Neural Networks, Springer, 2019.