Adaptive Particle Swarm Optimized XGBoost Ensemble Algorithm for Online Credit Scoring

Tinofirei Museba (tmuseba@uj.ac.za)
University of Johannesburg

Research Article

Keywords: Credit scoring, machine learning, particle swarm optimization, concept drift

Posted Date: May 26th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1660274/v1

Abstract

Determining the credit worthiness of an individual for a loan approval is a challenging task for most financial institutions. To assess the credibility of an individual, a credit score model is often used and it encompasses factors such payment consistency of customers and their behaviors and their previous behaviors. Due to its easy interpretation and outstanding prediction performance for both static and time-varying, evolving domains, machine learning has gained a lot of reputation over statistical approaches. For this reason, machine learning has gained much attention in the automation of the design of effective and efficient credit scoring techniques over the years. Credit scoring task is also an ephemeral scenario as most of the variables are likely to drift overtime, making the use of stream mining suitable for detecting and adapting to changes in the underlying data distribution. Given a problem in such time-varying and evolving environment, credit scoring models that were once robust may become suboptimal if the behavior of customers change over time as their earnings get eroded by changes in interest rates, natural disasters and increases in pay as you earn, rendering the current credit score model outdated. In credit scoring, many of the variables tend to drift over time resulting in an ephemeral scenario that requires online machine learning algorithms tailored for learning in time-varying and evolving environments where the target concept changes. Since machine learning algorithms can learn incrementally, they are able to detect learn changes that occur in the data. In this paper, we propose an online Decision Trees based eXtreme Gradient Boosting (XGBoost) credit scoring model that is based on a heterogeneous adaptive particle swarm optimization where behaviors change at each iteration by randomly selecting new behaviors from the behavior pool. The prediction performance of our approach is evaluated first as a batch learning algorithm and secondly as a data steam learning algorithm using a variety of validation schemes for traditional batch learning and the Kolmogorov-Smirnov and Population Stability index metrics. Each behavior consists of a pair containing a position update and a velocity update. Decision Trees can easily sift through credit data characterized by a high dimensional curse and a complex correlation. The behavior of our proposed approach is evaluated for both static and dynamic domains since customer variables tend to drift over time. Empirical experiments conducted showed that our proposed approach performs comparatively well in both static and dynamic environments on four datasets.

1. Introduction

To mitigate information asymmetry and accurately manage credit risk, credit scoring models have emerged as efficient tools for most financial institutions [25]. The financial credibility of an individual serves as a determinant factor ast to whether to approve a loan or not for most financial institutions. An appropriate credit scoring model brings huge profits and reduce potential customer churn to a company if designed without loopholes but a poorly designed credit scoring model that cannot distinguish between credit worth applicants and unacceptable applicants can lead to huge financial losses. A credit score is often used nowadays to measure the financial credibility of an individual and it includes factors such as the customer's credit history on previous performances on debt payments, profile, monthly total income,

profession, place of stay and property owned such as cars, properties and census information in an effort to estimate the defaulting probability [1]. The design of credible risk assessment credit scoring models requires a lot of time and resources making them susceptible to abuse by perverted employees who may introduce biases into the models. When customers are able to borrow money easily from a financial institution, it enables a properly managed economy to efficiently function and boost the growth of the economy.

The rapid growth of credit economies has led to an increase in the number of problems for most financial services providers as credit defaulters such as overdue loans have increased greatly. In reality, the costs for defaults are often higher so financial institutions tend to emphasize the accuracy of default data [19]. Recently, there has been a growing need for robust scoring models based on the current data to evaluate the credit worthiness of a customer to support the bank's decision making process and allow customers with good credit records to get loan approval in a timely manner.

Machine learning algorithms have been used to perform the automation of credit scoring models as they have demonstrated excellent prediction capability for most complex non-linear data [2]. Machine learning approaches generalize well in both static and dynamic domains when compared to traditional statistical approaches. Machine learning approaches do not work under a stable distribution assumption, making them superior to statistical techniques in handling non-linear pattern classification [23]. Machine learning techniques can effectively handle big data in large, sparse and complex high dimensional samples [24], an attribute suitable for building credit scoring model in both static and dynamic financial data. Recent empirical experiments conducted have shown that machine learning approaches generalize well when compared to statistical financial data. Credit data has inherent characteristics such as changes in the target concept as many of the variables may drift over time as the behavior of customers change over time, credit data imbalance caused by a small number of default applications, a large number of credit features and complex feature relationships, a sparse matrix problem caused by many missing values and features and interpretability issues for feature decisions [3]. Traditional machine learning approaches assume that customer data is static but the behavior of customers change over time as many of the variables may drift over time.

For most imbalanced datasets, traditional customer credit scoring models tend to have a higher classification error rates for a small number of customers with bad credit when compared to the majority of customers with good credit. The misclassification of a customer with a poor credit record score can be detrimental to the business when compared to the misclassification of a customer with a good record [5]. A properly configured credit scoring that takes into account the inherent characteristics of credit data has the potential of bringing huge economic returns. This requires credit score models to have higher prediction accuracy for the task of classifying customers according to their credit worthiness to accurately address the inherent imbalanced ratios of credit data in more aspects.

XGBoost optimized by adaptive Particle Swarm Optimization facilitates the derivation of a customized objective function and performance evaluation indicators used to refine the model to handle inherent

characteristics of credit data such as drifting variables, missing values and class imbalance.

The performance of XGBoost depends hugely on properly configured hyper parameters. The hyperparameters of XGBoost have to be appropriately tuned in order to accurately minimize the objective function as manual approaches based on artificial experiences tend to consume a lot of time and require a lot of computer resources such as memory. Existing parameter optimization approaches such as grid search and random search have a tendency of overlooking the optimal value since the objective function is considered to be non-convex. The original version of Particle Swarm Optimization was proposed by Kennedy and Eberhart [4]. Particle Swarm Optimization is a computational intelligence technique used for optimizing complex engineering and financial models.

Particle Swarm Optimization is not restricted by constraints associated with the objective function such as continuity and differentiability, but finds the optimal solution through a number of iterations and has a fast convergence speed making it suitable for the optimization of hyper parameters to minimize the objective function for most applications.

This paper proposes an XGBoost ensemble optimized by the adaptive Particle Swarm Optimization for credit scoring. PSO has a tendency of falling into local optimum for most applications thereby causing premature convergence. A change in customer behavior may lead to an outdated credit scoring model and diversity may be lost as a result of swarm convergence and such problems need to be solved to allow the PSO to accurately minimize the objective function. To avoid the loss of diversity which may lead to premature convergence thereby hindering further exploration and exploitation, the swarm is allowed to re-diversify when the customers change their behaviors. For that, we adopt the use of the heterogeneous PSO that allows particles to assume different behaviors from each other as they adopt different velocity and position update rules. The adaptive credit scoring XGBoost optimized model is then used for credit scoring.

The rest of the paper is organized as follows. Section 2 provides a comprehensive review of machine learning approach for credit scoring. Section 3 provides a preview of the genesis and workings of Particle Swarm Optimizers (PSO) and the Extreme Gradient Boosting (XGBoost). Section 4 introduces our proposed methodology. Section 5 provides a description of datasets used and the empirical experiments conducted. In section 6, the conclusions and future work are provided.

2. Literature Review

Artificial intelligence techniques and machine learning have attracted the attention of researchers in the design of credit score models due to their ability to create adaptive models for both static and dynamic domains. Credit scoring is a challenging risk for most financial institutions and as a classification problem, it has now become an important field of research in the area of data mining. Recently, datasets related to loan applications from financial institutions has significantly increased and its now cumbersome to use traditional and statistical models for credit scoring tasks. To build robust and reliable credit scoring models, several issues are taken into consideration by researchers and this include the

extensiveness of the datasets in an effort to accurately model the planning horizon and the instigators of undesirable behavior [30].

A number of contemporary approaches to design credit score models based on machine learning algorithms capable of outperforming performances of existing models have been proposed. In [6], authors proposed a new hybrid ensemble credit scoring model. The model fuses five feature selection algorithms and uses three voting strategies. The approach generates eight ensemble models which are combined using the soft voting approach. No assumptions were made over the possible skewed distribution of the data and the possibility of customer variables drifting. Jian Xiao [7] proposed a semi-supervised cost sensitive learning, a data handling approach and an ensemble learning approach to create a customer credit scoring model. The approach introduces more noise as a result of incorrect labeling of some examples. In [8], the authors presented an overfitting cautious heterogeneous ensemble model for credit scoring. The approach uses tree based techniques to strike a balance between prediction and computational cost. The approach disregards the possibility of customer variables drifting over time and the imbalanced data problem. An analysis of the different aspects of the credit scoring applications using machine learning on static and streaming data was performed by Jean Paul Barddal et al [1]. In their empirical experiments conducted, they concluded that data streaming techniques yielded interesting discriminative rates and credit score models should take into account the possibility of customer variables drifting. Vincenzo Moscato [9] designed a benchmark for machine learning approaches for credit risk prediction for social lending platforms that also handles imbalanced datasets. The approach provides no suggestion of a possible customer variable drift. In [10], the authors suggested a deep learning ensemble credit risk evaluation model that also handles imbalanced credit data using an improved synthetic minority oversampling technique. The approach does not consider the creation of a credit score as a streaming classification problem where customer variables can drift. Salvatore Carta [11] proposed a proactive approach to the creation of a credit scoring model that consists of a combined entropy approach. The approach disregards the class imbalance problem and the streaming classification problem. Wenyu Zhang [12] proposed a novel multi-stage hybrid model which combines feature selection and classifier selection to obtain optimal feature subset and optimal classifier subset and uses the ensemble to optimize the prediction in credit scoring. An enhanced multi-population niche genetic algorithm is used to improve the ability of optimization effectively. In [13], the authors proposed a novel noise-adapted two-layer ensemble model for credit scoring based on backflow learning approach to address noise data and borderline data samples. The approach uses the synthetic minority oversampling technique (SMOTE) algorithm to address the imbalanced data problem. Xiaohong Chen [14] proposed a new heterogeneous ensemble model based on the generalized Shapley value and the Choquet integral to improve the predictive power of the classifier. The model firstly introduces the fuzzy measure to solve the problem where the base learners of the ensemble model may interact. To take into account the accuracy and the diversity of the base learners simultaneously, a linear programming model for determining the fuzzy measure is built with an accuracy and diversity based objective function. The approach disregards the possibility of customer variables drifting and imbalanced datasets. In [15], the authors proposed a novel multi-stage ensemble method with a hybrid genetic algorithm for credit scoring on imbalanced

data. The proposed multi-stage ensemble is computationally complex and disregards the possibility of a change of customer variables.

Chaoqun Wang [16] suggested an ensemble classifier optimized by PSO to assess the credit worthiness of customers in peer to peer lending platforms. The weights of the experts are optimized by the PSO. The authors did not consider the possibility of change of customer variables and imbalanced datasets.

An assertive credit score model based on machine learning is composed of information such as historical data that is veracious, relevant attributes of the customer related to the past loan payment profile and a prediction model that encompasses all the variables relevant to model the behavior of credit worthy and non-credit worthy creditors. Credit data for most financial institutions is characterized by credit data imbalance caused by a small number of default applications, a large number of credit features and complex feature relationships, a sparse matrix problem caused by many missing values and features and interpretability issues for feature decisions [3]. The credit scoring task is never addressed as a data stream classification problem as many of the customer variables may drift over time. The proposed approaches and other integrated learning methods addressed some of these problems but not all, making their credit score models inappropriate to classify customers accurately. In this study, we propose an adaptive Particle Swarm Optimized XGBoost Decision Tree ensemble that learns incrementally to handle customer variables that are drifting and employs the Synthetic Minority Oversampling Technique (SMOTE) algorithm to address the imbalanced customer data.

3. Particle Swarm Optimizers

Particle Swarm Optimizer (PSO) is a heuristic population based iterative, global and stochastic optimization algorithm inspired by the social behavior of bird flocking or fish schooling to conduct an intelligent search for the optimal solution [17]. PSO is derivative free as it does not require the optimization problem to be differentiable, therefore not requiring gradients making it applicable to a variety of problems such as those with discontinuous or non-convex and multimodal problems. Firstly, the algorithm is initialized with a set of agents called particles in random positions in the problem space. Each particle is assigned a random velocity at the beginning. For each particle, a fitness function is defined relating to its location. Efforts to find the best position of a particle entails solving an optimization problem by minimizing the fitness function. As the process iterates, the particle's fitness is evaluated by the algorithm, the velocity is updated and the new position is computed. A standard PSO algorithm assumes the existence of a dimensional search space, a swarm of m particles expressed as $\{x_i$,..., $x_m\}$ and the particles are expressed as $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. At iteration t, the information that characterizes a particle is provided as follows:

Position $X_i^t = [x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t]^T$, and velocity $V_{i1}^t = [v_{i1}^t, v_{i2}^t, \ldots, v_{iD}^t]^T$, personal best position, $[p_{i1}^t, p_{i2}^t, \ldots, p_{iD}^t]^T$ and global optimal position $p_g^t = [p_{g1}^t, p_{g2}^t, \ldots, p_{gD}^t]^T$. At iteration $t + 1$, the speed and position of the particle is updated as follows:

$$V_{id}^{t+1} = wv_{id}^{t} + c_1 r_1^t (p_{id}^t - x_{id}^t) + c_2 r_2^t (p_{gd}^t - x_{id}^t) \quad (1)$$

$$X_{id}^{t+1} = v_{id}^{t} + v_{id}^{t+1} \quad (2)$$

The coefficient ω represents the inertia weight used to control the balance between exploitation and exploration. The social learning factors, c_1 and c_2 changes the step length of the direction of movement to the position of the particles and the direction of the global best position.

3.1 Extreme Gradient Boosting (XGBoost)

The Extreme gradient boosting tree (XGBoost) was proposed by Chen and Guestrin [32] to solve classification and regression problems. Extreme Gradient Boosting (XGBoost) is a tree-based algorithm used for both classification and regression-related problems and is part of the family of ensemble machine learning-based class of algorithms. Machine learning is a form of artificial intelligence that allows programs to continuously self-improve using existing and new data. Machine learning algorithms parse past data, learn from it and make predictions about future data. Machine learning has the potential to improve credit risk modeling. The likelihood of a customer repaying a loan depends on many factors. Typically, statistical learning methods assume formal relationships between variables in the form of mathematical equations, while machine learning methods can learn from data without requiring any rules-based algorithms. Due to this flexibility, machine learning algorithms can better fit patterns in data for calculating credit risks. Machine learning algorithms can better capture non-linear relationships which are common to credit risk. XGBoost implements Gradient Boosting decision trees to harness higher speed and better performance. The algorithm minimizes prediction errors generated by prior models by adding more trees to the ensemble. For structured data, XGBoost has gained utmost popularity in applied machine learning.

XGBoost employs both first and second derivatives and the generated loss function is considered to be more accurate and thus enhancing its ability to precisely split the tree and create the suitability of the tree structure for financial data characteristics. XGboost approximates the loss function using the Taylor expansion and this creates a model that balances both bias and variance by the use of fewer decision trees to achieve superior generalization ability.

A regularization term is included in the cost function of the algorithm to control the learning model from getting complicated and reduces the chances of the model overfitting when the credit data to be learned by the model is of a small sample.

XGBoost automatically learns the segmentation direction of the sample in the event of missing eigenvalues. These attributes associated with XGBoost makes it achieve superior prediction in credit scoring [18]. The mechanism employed by the XGBoost algorithm makes it appropriate to learn credit data associated with missing values and sparse characteristics. XGBoost calculates the gain of samples in both the left and right subtrees to find the default direction of the missing data. By this, it learns how to

approximate or impute missing data. Given the vast advantages offered by XGBoost such as learning in time-varying, evolving environments as customer variables drift, the ability to address missing values of credit data, more accurate solutions of loss function, mechanisms to evade overfitting and the advantages of PSO in optimizing parameters in high dimensional continuous space, we build an XGBoost model that is optimized by a heterogeneous PSO to realize high precision credit scoring. XGBoost algorithm can handle sparse data, implement distributed and parallel computing flexibly [29]. Although XGBoost performs well in both static and dynamic environments, its excellent performances hinges on the proper hyper-parameters settings.

3.2 Parameter Optimization

To achieve high precision credit scoring results, the model must find appropriate parameters. For every learning model, there exists model parameters and hyper parameters. The model parameters are obtained by learning the distribution of training data, without the need for human experience. A hyper parameter is defined as a higher-level concept about the model such as its complexity or ability to learn. Generating a set of satisfactory hyper parameters improves the generalization performance of learning models so tuning them is important. However, hyper parameter tuning is subjective and relies on empirical judgement and trial and error approaches [19]. The process of optimization reduces the need to depend on manual search and trial and error. To appropriately optimize and generate an accurate subset of parameters and maintain the efficacy of XGBoost for classification tasks, our proposed approach uses an adaptive heterogeneous Particle Swarm Optimizer XGBoost (AHPSO-XGBoost).

The derived parameters of XGBoost are as follows:

Table 1
Derivation of hyper-parameters of XGBoost

Hyper-parameter	Derivation
Learning Rate	Weight value reduced and node robustness optimized
Tree depth maximum	Maximum tree depth generates a greater value, increases tree complexity and evades overfitting
Subsample ratio	Sample rate
Column subsample ratio	Features sampled to construct tree
Maximum child weight	Overfitting reduction via sum determination of minimum leaf nodes sample weights.
Maximum delta step	Tree weight change limit factor for unbalanced data
Gamma	Minimum loss function reduction necessary to evade overfitting
Number of estimators	The number of iterations of XGBoost

4. Adaptive Heterogeneous Particle Swarm Optimizer-xgboost

To get a better prediction accuracy of the model, the strategy employed to tune hyper parameters must be carefully selected. Diversity plays an important role in preventing the premature convergence of the PSO and improving the generalization performance of the learning algorithm. Most approaches adaptively introduce diversity in the swarm by spliting the swarm into different groups in relation to particle distribution and the search direction of two groups of particles is performed by different learning strategies. To perform the adaptive split of the swarm population, most studies have adopted the search clustering method [20] in an effort to improve diversity. The clustering approach automatically finds the class cluster center of the dataset samples. The objective is to create heterogeneous swarms. However, generating diversity of particles by clustering has its own drawbacks. Particles may fail to improve their personal best position over a window of recent iterations and if the personal best position does not change, it may be an indication of early stagnation. To overcome this drawback, in this paper, particles are instantiated at the individual level. Instantiation of the particles at individual level introduces heterogeneity among the swarm particles. Individual particle instantiation allows particles in a swarm to assume different search behaviors by randomly selecting velocity and position update rules from a behavior pool thereby creating a swarm that consists of explorative particles and exploitative particles, giving the optimization algorithm the ability to explore and exploit throughout the search process. The approach allows the swarm to manage its diversity dynamically. The particles in the swarm population update their positions and velocity over time, making the heterogeneity dynamic. Dynamic heterogeneity introduces adaptive particle swarms that are able to respond to changes in the environment. In credit score models, variables tend to drift with time. The objective of the PSO optimization algorithm in such time varying and evolving environments is no longer relegated to finding the global optimum but also to adapt to changes over time. To accurately adapt to a changing optimum, the PSO optimization algorithm has to solve the outdated memory problem as the environment changes, the particle's personal best position and its fitness value are no longer valid and diversity loss due to swarm convergence. Diversity loss occurs when the swarm converges to an optimum [21]. To arrest premature convergence, swarms are allowed to re-diversify as the environment changes. This enables the algorithm to simultaneously consider the relevance of both diversity and accuracy of experts in the ensemble by constructing an ensemble learning approach based on the accuracy and diversity objective function thereby producing a good balance between performance and efficiency.

4.1 Adaptive PSO

To introduce heterogeneity, the Adaptive Heterogeneous Particle Swarm Optimization XGBoost selects random behaviors for particles from a pool of behaviors. The personal best of the position of all particles is monitored in order to detect stagnation. If a particle's personal best position reflects no changes for a number of consecutive iterations, the assumption is that it has stagnated and is forced to randomly select a new behavior from the behavior pool. Each behavior is composed of a pair that consists of a position update and a velocity update. For this purpose of this study, we populate the behavior pool with the following five behaviors:

- **Standard PSO particle velocity and position update behavior.**

The particle updates its velocity and position using Eqs. (1) and (2). The value of the cognitive acceleration constant c_1 is set to 3 and decreases linearly to 0.5 the entire search process. The social acceleration constant c_2 is initially set to 0.5 and increases linearly to 2.5 during the entire search process. This behavior facilitates exploration near the start of the search process and exploitation towards the end of the search process as the iterations numbers increase.

- **Social PSO Behavior**

The social component is removed by the cognitive-only velocity update from Eq. (1). This makes the particle to be only attracted to the global best position. For position updates, Eq. (2) will still be used. The value the social acceleration constant is initially set to a constant value of 2.3. This behavior promotes exploitation since all particles tend to follow this behavior at specified time steps thereby becoming effectively a stochastic hill-climber.

- **Cognitive PSO Behavior**

The social component from Eq. (1) is removed by the cognitive-only velocity and a particle is only attracted to its own personal best position. For position updates, Eq. (2) continues to be used. A constant value of 2.5 is assigned to the cognitive acceleration constantc_1. This behavior facilitates exploration throughout the search process as all particles that assume this behavior become independent hill-climbers.

- **Bare bones PSO behavior**

This Bare bones PSO behavior was formulated by Kennedy [22]. The behavior replaces the velocity update equation with a novel Gaussian distribution such that:

$$v_{ij}(t+1) \sim N\left(\frac{y_{ij}+y_j(t)}{2}, \sigma^2\right) \quad (3)$$

The variance $\sigma^2 = |y_{i,j}(t) - y_j(t)|$ (4)

The equation of the position update is altered and becomes:

$x_i(t+1) = v_i(t+1)$ (5)

The velocity is now the particle's new position and is not added to the particle's current position. This behavior facilitates exploitation of the particle's midway point and the global best position. The particle's personal best positions are scattered throughout the entire search space allowing the behavior to promote exploration.

- **Modified Bare bones behavior**

Kennedy [22] suggested a modification of the bare bones PSO and Eq. (3) was changed to:

$$v_{ij}(t+1) = \left\{ \phi otherwise^{y_{ij}(t)\ if \cup (0,1) < 0.5} \right\} \text{ (6) where } \phi \sim N \left(\frac{y_{ij}(t) + y_{j(t)}}{2} \sigma^2 \right) \text{ (7)}$$

Most of the time, particles focus on their personal best positions enabling the behavior to associate itself with better initial exploration and better exploitation compared to the first version of the bare bones.

4.2 AHPSO-XGBoost Credit Scoring Model

The credit score model proposed employs XGBoost algorithm optimized by the Adaptive Heterogeneous Particle swarm Optimizer. To enhance the diversity of particles and help particles jump out of a local optimum, we use the idea of a heterogeneous particle swarm optimizer. Since hyperparameters control the complexity or regularization to refine the model, AHPSO-XGBoost allocates particles in a swarm different search behaviors by randomly selecting velocity and position update rules from a behavior pool to increase diversity. Although both accuracy and diversity of base learners are critical factors for constructing ensemble models, most existing models consider only one aspect and ignore the tradeoff between them [23]. For performance and efficiency of the learning algorithm, our proposed algorithm takes into account simultaneously the accuracy and diversity objective function. In credit scoring, more training data does not guarantee that the generalization performance of the learning model will improve when using traditional batch learning classifiers. More data may lead classifiers to over fit and the data from different months and years may exhibit drifting characteristics, requiring the application of incremental learning to obtain discriminative rates that are accurate.

4.3 Data Preprocessing

Data standardization is performed on the dataset as features normalization tend to make learning models to be more accurate and persuasive. The volume of features in the dataset often leads to overfitting for most machine learning algorithms. If the distance between variables of large-scale features and small-scale features is too large when features of different orders are computed, the result may lead to deviation in the process of training the model, resulting in poor generalization performance of the model, thereby falling short of the expected standard. A number of standardization methods exist and depending on the types variables, they are split into continuous variable standardization and discrete variable standardization. Two commonly used standardization methods for continuous variable normalization are the Min-max normalization and z-score normalization. Due to the inherent continuous characteristics of credit score datasets and their drifting features, this paper uses the min—max standardization. In the min—max standardization, the description of the training set is provided as D = {X,} where X = {x_1, x_2,..., x_m} representing the target value. If Y = 0, it is an indication of poor application whereas Y = 1 is an indication of good application. The objective of the min-max standardization is to indicate a minimum and maximum interval, map features to a given interval or convert the absolute value

of each feature to a unit size [26]. The values in the datasets are normalized. Given a feature labeled x, its target value is calculated using a 0–1 scaling. Given a value x of any attribute, the normalization or scaling is performed as follows:

$$x^{'} = \frac{x - \min(x)}{\max(x) - \min(x)^{'}}, \quad (8)$$

,where min(x) represents the minimum values of the attribute and max(x) represents the maximum value of the attribute and the standard value is expressed by $x^{'}$. Credit data variables may drift over time and is associated with missing values. The sparsity segmentation algorithm available in the XGBoost algorithm learns and approximates missing values and works well in modeling missing values when compared to traditional approaches for approximating missing values. The influence of outliers and noise in the credit data is smoothened using centralization and extreme values can be handled by centralization. To adapt to drifting variables, we adopt the incremental learning approach to detect and adapt to changes in the credit data distribution. To build the XGBoost model, feature selection chooses relevant features and discards irrelevant features. The feature selection helps to eliminate the curse of data dimensionality, enhance operational efficiency, simplify the process of learning and significantly reduce computational complexity. Features that make the learning task easy are selected by the model as they are favorable. The gain attribute is one of the measures of feature importance offered by XGBoost that is exploited in the feature selection process performed in this work. XGBoost adopts the stochastic fractal search (SFS) algorithm according to the rank of the importance of the features and adds features into the dataset to form subsets one by one [3]. Considering the existing default parameters of XGBoost, only the subset that reduces the logistical loss is chosen as the subset of features.

4.4 Training the Credit Score Model

The credit scoring problem is addressed using data stream mining approach as is associated with class imbalance and feature selection. In credit score datasets, the ratio between creditworthy and non-creditworthy customers is imbalanced due to variations in ratios of arriving instances [27]. The credit score data is imbalanced, that is, the number of non-defaults is far more than the number of defaults. For the prediction results not to be compromised, data imbalance has to be addressed. Due to the imbalance of the majority and minority class samples in most credit score datasets, the misclassification of some samples may occur, thereby affecting the model generalization performance. This paper uses the Synthetic Minority Oversampling technique (SMOTE) [28] as an augmentation to the features available in XGBoost to handle imbalanced data by firstly separating the defaults and the non-defaults in the training data. The Synthetic Minority Oversampling technique uses the k-nearest neighbor to produce new instances based on the distance between the minority data and some randomly selected nearest neighbor. The balanced data is then used for training the model constructed with the XGBoost classifier. SMOTE is applied to increase the minority class samples, thus further enhancing the robustness of the model. The test data is fed into the model with the same sampling ratios. The hyperparameters such as maximum tree depth, subsample ratio, column subsample ratio, minimum child weight, maximum delta step and gama delta in the XGBoost are the optimization targets. The position of each particle in the

search space is initialized. The behavior pool is populated with pairs of position and velocity from the five behavior models. Different pairs of position and velocity of the particles from the behavior pool are used are used to update the position of the particles' to avoid premature convergence and improve particle diversity. The fitness function of the particles is defined as the loss function on the credit score dataset. Credit scoring is a binary classification problem. The labels of the positive and negative samples of the credit data can be categorized as + 1/-1. The derivation of the logistic loss function can then be formulated as:

$L_{logistic}$ = log(1+exp(-yp)) (9)

The value of p represents prediction and y represents the actual value. If termination condition is reached, optimized values are obtained.

The description of the algorithm is as follows:

1. Partition the data into training and testing sets. Initizlize the adaptative heterogeneous PSO
2. Use Eq. (9), the logistic loss function, to calculate the fitness value of each particle. The XGBoost model is constructed with the obtained hyper-parameters derived by the best particle currently determined.
3. Use the behavior pool to update the position and velocity of the particles ascertained by the fitness values of the global optimal particles pbest and the local optimal particle gbest.
4. Update the position and velocity of the particles using the behavior pool.
5. Ascertain the termination criterion. Output the optimal values of the hyperparameters after the number of iterations n.
6. Build the XGBoost model using the optimal hyperparameters obtained.

The pseudocode starts with the initialization of the AHPSO-XGBoost algorithm and the position and velocity of the particles are initialized in the search space. Subsequently, the behavior pool is populated with pairs of position and velocity. Particles in a swarm assume different search behaviors by randomly selecting velocity and position update rules from a behavior pool. In all the experiments conducted, the behavior pool was populated with the same five behaviors used in [30]. Stagnation is detected by monitoring the personal best position of all particles. If a particle's personal best position does not change for a determined number of consecutive iterations, the assumption is that the particle has stagnated and is forced to randomly select a new behavior from the behavior pool. The current optimal values are used to determine the hyperparameters of the model. During the modeling process, the gain of each node is computed and the node with the highest score is split to generate a tree. When the modeling process is complete, the fitness value (loss function) is calculated and the particles are updated using different search behaviors by randomly selecting velocity and position update rules from a behavior pool.

5 Experimental Setup

The experiments were conducted using scikit learn, an important library for machine learning in Python for classification tasks. XGBoost is an open-source Python library that provides a gradient boosting framework. Gradient boosting is a machine learning approach used for classification, regression and clustering problems. XGBoost ensembles weak learners to create a strong learner. Models are introduced sequentially into the ensemble. The introduction of the next models sequentially enables the weak models' errors to be corrected thereby achieving an optimized solution.

5.1 Data Description

To validate the efficacy and generalization performance of our proposed model, we used four datasets that exhibit different imbalance ratios. We employ two commonly used credit scoring datasets and two real world datasets.

The Australian and German datasets are the most frequently used datasets for credit scoring by most researchers in related literature which allows us to perform a comparative study with previously conducted research in credit scoring. The datasets are freely available on the UCI Machine Learning Repository [31]. The Australian dataset consists of 690 instances of which 307 are fully paid and 383 are defaults. The Australian dataset is composed of eight numerical and six categorical features. The German dataset consists of 1000 instances with 700 samples indicating fully paid and 300 representing defaulters. The German dataset also consists of 13 categorical features and 7 numerical features. From the two datasets, the most frequently used financial and nonfinancial indicators or variables are selected.

The robustness of the Adaptive XGBoost is further examined using two real world Peer to Peer (P2P) credit scoring datasets to validate the suitability of the Adaptive XGBoost approach for real world problems since credit scoring is an ephemeral scenario since many of the variables may drift over time. This makes Adaptive XGBoost to be tailored for incremental learning environments and to detect and adapt to changes in the underlying data distribution.

The first real world dataset is the RRDai data sourced from a Chinese Internet finance enterprise called RenRenDai that consists of loan data for the year 2017. The dataset is publicly available on https://www.renrendai.com. The dataset was sourced via a web crawler. Most instances of the RRDai dataset represent the outstanding loan instances. The classification task is to ascertain whether a client defaults in his or her obligation to pay off the monthly repayment amount. The attributes related to the RRDai dataset include loan amount P, annualized yield rate *apr*, repayment period T, remaining repayment period T_r and actual outstanding amount L_u. Two repayment options exist and they are the average capital plus interest approach and the debt servicing approach.

The monthly repayment of loan for average capital plus interest method is expressed as:

$$L_e = P^* \frac{\frac{apr}{12} * \left(1 + \frac{apr}{12}\right) T}{\left(1 + \frac{apr}{12}\right) T - 1} \quad (10)$$

The gross interest for the debt servicing approach is calculated as:

$$R_t = P^* \frac{apr}{12} * T \quad (11)$$

The second real world dataset is called the LendingClubLoan data and is sourced from a Peer to Peer (P2P) lending company called LendingClub. The dataset is used to match borrowers with investors online. The dataset is available in Kaggle under the website https://www.kaggle.com/wendykan/lending-club-loan-data. The dataset consists of complete loan data from the year 2007 to 2015 and includes the current loan status and recent information regarding payments.

Credit scoring input variables of the datasets were split into the following subsets [33]: applicant assessment (grade,subgrade), loan characteristics (loan purpose and amount), applicant characteristics (annual income, housing situation), credit history length, delinquency) and applicant indebtness(loan amount to annual income, annual installment to income).

The imbalanced datasets are processed using the Synthetic Minority Technique (SMOTE). Previously proposed approaches for credit risk assessment models for financial institutions use imbalanced data whereby the number of non-default cases is usually much higher than the default cases. If the class imbalance problem is not taken into account and we use all data to build a classification model, a model that has high accuracy for the determination of non-defaults but extremely low accuracy for defaults is obtained. This study uses SMOTE to balance the number of cases from both defaults and non-defaults to minimize the effects of class imbalance problem on modeling. After identifying relevant variables, outliers and missing values are then solved. A random sampling of the data of 80% for the construction of a credit scoring model is performed. The remainder of 20% of the data is then used for back testing of the discriminating power of the completed model.

Table 2
Formulation of Credit Score data

Dataset	Instances	Features	Training Set	Test set	Good/bad
German	1000	24	800	200	700/300
Australia	690	14	552	138	307/383
RRDai Ren	1421	17	1137	284	1072/349
LendingClubLoan	2642	11	2114	528	1322/1320

5.2 Evaluation Measures

In credit scoring, the average accuracy is the commonly used performance metric and shows the overall generalization performance of the learning model. To better explore the suitability of the model to distinguish between nondefault applications and default applications and to accurately get a reliable and robust conclusion, we employ six evaluation metrics to empirically evaluate the prediction performance of our proposed approach. The six performance metrics are the accuracy (ACC), Brier score (BS), Area under the Rock Curve (AUC), the F1 score, and the type 1 and type11 errors of the confusion matrix to accurately distinguish between nondefault applications and default applications. The two types of errors are used to evaluate the models the models to predict their performance in much more detail. A type 1 error occurs when a default loan application is being wrongly classified as a nondefault loan application. Conversely, a type 11 error occurs when a nondefault is misclassified as default. In a confusion matrix, TP and TN represent the numbers of correctly classified good borrowers and bad borrowers respectively. The two variables, FP and FN represent the numbers of misclassified loan applications.

The performance metrics can be represented as follows:

The average accuracy (ACC)= $\frac{TP+TN}{TP+FP+TN+FN}$ (12)

The type 1 errors: $\frac{FP}{TN+FP}$ (13)

The type 11 errors: $\frac{FN}{TN+FP}$ (14)

The Brier score is a performance metric used to measure the accuracy of the predicted probability and the calibration of the prediction performance. The Brier score is within the interval 0 to 1 and the value of the interval represents probabilistic predictions from perfect to poor. A lower Brier score reflects better predictions.

The Brier score can be expressed as:

BS = $\frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)^2$ (15)

, where N is the number of samples andp_i

and y_i denote the probability prediction and the true label respectively of sample i.

The F1-score considers both precision and recall of classification models. It is the harmonic average of precision and recall and ranges within the interval of 0 to 1. F1 ca be expressed as:

F1 = 2.$\frac{precision.recall}{precision+recall}$ (16)

, where precision is the proportion of positive samples in positive cases and is defined as:

Precision = $\frac{FP}{TN+FP}$ (17)

Recall is expressed as the proportion of predicted positive cases in the total positive cases and is expressed as

Recall = $\frac{TP}{TP+FN}$ (18)

The Area Under the Roc Curve (AUC) performs a global assessment by computing the area under the Receiver Operating Characteristics Curve (ROC) according to the predicted scores which is plotted against the True Positive Rate (TPR) and False positive Rate (FPR).

5.3 Experimental Results

The prediction performance of the AHPSO-XGBoost is compared with nine other ensemble learning approaches. The prediction results of benchmark ensemble models across the four datasets and evaluation metrics are presented as baseline of our proposed method. Ten ensemble approaches namely AdaBoost, AdaBoost-NN, Bagging Decision Tree, Bagging NN, Random Forest, Decision Tree, Logistic Regression, Leveraging Bagging, Adaptive Random Forest and AHPSO-XGBoost are validated across the datasets as baselines for the learning process.

Table 3
Performance of benchmark models across datasets

Dataset	Classifier	ACC	F1	AUC	BS	TYPE 1	TYPE 11
Australia	AdaBoost	72.43	53.65	62.05	0.2268	56.49	12.65
	AdaBoost-NN	71.85	43.85	58.85	0.1678	53.66	16.46
	Bagging-DT	73.57	56.43	67.35	0.2465	50.42	13.68
	Bagging-NN	70.58	59.35	63.48	0.1874	65.48	10.48
	Random Forest	71.35	61.67	61.35	0.2247	56.34	11.56
	Decision Tree	74.45	54.32	59.85	0.1778	62.23	14.28
	Logistic Regression	72.85	62.28	60.59	0.1764	58.45	15.09
	Leverage Bagging	74.33	57.76	55.79	0.1743	63.54	17.36
	Adaptive Random Forest	72.85	54.34	61.75	0.1659	54.35	11.49
	AHPSO-XGBoost	**75.43**	**72.48**	**70.48**	**0.1604**	**53.05**	**10.43**
German	AdaBoost	66.34	58.58	63.46	0.2213	62.53	12.87
	AdaBoost-NN	69.34	62.87	67.82	0.2174	61.27	13.48
	Bagging DT	73.47	59.42	63.28	0.1783	59.74	15.85
	Bagging-NN	68.76	64.37	59.86	0.1674	56.46	11.78
	Random Forest	71.26	69.72	60.47	0.2034	60.74	12.35
	Decision Tree	74.64	67.48	58.73	0.2167	58.79	14.56
	Logistic Regression	72,58	65.39	65.48	0.1784	64.72	16.86
	Leverage Bagging	74.78	70.84	69.87	0.1674	68.43	11.38
	Adaptive Random Forest	68.85	69.87	64.89	0.1689	64.58	12.69
	AHPSO-XGBoost	**76.83**	**73.74**	**71.84**	**0.1567**	**69.79**	**11.21**
RRDai	AdaBoost	74.47	59.47	72.38	0.2217	0.1874	14.23
	AdaBoost-NN	69.83	61.35	69.85	0.2145	0.2167	16.78
	Bagging-DT	71.68	58.67	64.58	0.1673	0.2214	13.29
	Bagging-NN	68.73	63.48	62.87	0.1784	0.1894	15.63
	Random Forest	70.39	60.79	67.89	0.2242	0.1787	12.84

Dataset	Classifier	ACC	F1	AUC	BS	TYPE 1	TYPE 11
	Decision Tree	66.58	57.45	68.89	0.1783	0.2231	17.62
	Logistic Regression	68.38	70.34	70.38	0.1687	0.1875	12.31
	Leveraging Bagging	72.68	72.84	73.49	0.1602	0.2262	11.87
	Adaptive Random Forest	74.86	76.58	71.29	0.1687	0.1876	13.56
	AHPSO-XGBoost	**72.35**	**70.35**	**74.67**	**0.1583**	**0.1671**	**11.05**
LendingClubLoan	AdaBoost	69.47	66.89	71.86	0.1873	0.2217	16.81
	AdaBoost-NN	65.67	63.47	68.47	0.2241	0.1764	13.48
	Bagging-DT	63.28	59.83	66.89	0.1784	0.2137	12.75
	Bagging-NN	66.87	68.42	63.12	0.1843	0.1785	14.39
	Random Forest	69.84	71.89	67.84	0.1798	0.2236	13.89
	Decision Tree	67.38	69.84	69.54	0.2345	0.1706	12.34
	Logistic Regression	70.83	71.73	73.42	0.1865	0.1689	13.26
	Leveraging Bagging	73.48	74.86	74.89	0.1721	0.1654	12.19
	Adaptive Random Forest	71.89	72.49	72.36	0.1672	0.1702	11.87
	AHPSO-XGBoost	**77.58**	**76.69**	**74.85**	**0.1612**	**0.1603**	**10.89**

Table 3 shows the prediction performances of benchmark models using different indicators. Table 4 provides a summary of the results of XGBoost models that were optimized using different optimization algorithms. The AHPSO-XGBoost optimized using AHPSO achieved the best performance on the four credit datasets. For the Australia dataset, AHPSO-XGBoost achieved the highest accuracy and lowest type 1 error rate. A lower type 1 error is an indication that the model avoids misinterpreting a poor credit application with a higher probability of good credit. AHPSO-XGBoost has the lowest Brier score when compared to XGBoost-RS and XGBoost-TPE. The F1 score of AHPSO-XGBoost is the best among all the XGBoost optimized algorithms due the appropriate settings of AHPSO. The success is hugely attributed to the hyperparameter of XGBoost Maximum Delta step which resists the problem of imbalanced data to a certain extent. On the German dataset, AHPSO-XGBoost algorithm obtained the best prediction accuracy with a score of 75.43%, which is 4% higher than the default XGBoost model. The model achieves promising results for Type 11 error and Brier score. The F1 score obtained is quite promising and is a demonstration of the fact that the model can adequately learn the unbalanced data.

For the RRDai dataset, AHPSO-XGBoost obtains the best prediction performance when compared to other optimized credit scoring models for all indicators, a sign that the hyperparameters settings are more reasonable when compared to other model settings.

In the LendingClubLoan dataset, AHPSO-XGBoost performed well as shown by all indicators. The default non-optimized model Type 11 error is lower than the AHPSO-XGBoost Type 11 error due to the unbalanced data that has more labels associated with good credit. Parameter settings need to be improved to learn the imbalance accurately. AHPSO-XGBoost achieves the best ability of probability prediction as it achieves the highest Brier score.

On average, AHPSO-XGBoost achieves a better prediction as compared to other models. The results obtained indicate that AHPSO has the capacity to promote the alignment of the XGBoost algorithm with the characteristics of the credit data. In terms of prediction performance, the AHPSO-XGBoost algorithm performs better than PSO-XGBoost model due to the fact that AHPSO avoids premature convergence of the particle swarm and enables particles to obtain hyperparameters that render the algorithm to be more accurate.

Table 4
Results of the performances of XGBoost with various optimization methods on credit data

Dataset	Classifier	ACC	F1	AUC	BS	TYPE 1	TYPE 11
Australia	XGBoost	71.29	71.85	69.45	0.2016	63.37	13.59
	XGBoost-GS	68.83	69.59	66.84	0.1724	67.59	14.53
	XGBoost-RS	70.69	66.79	63.49	0.1836	61.47	12.68
	XGBoost TPE	72.53	68.74	68.86	0.1649	59.89	15.04
	PSO-XGBoost	73.48	70.28	67.57	0.1631	60.81	12.37
	AHPSO-XGBoost	**75.62**	**73.89**	**70.74**	**0.1514**	**58.84**	**11.19**
German	XGBoost	71.45	68.52	72.86	0.1784	66.89	16.59
	XGBoost-GS	68.56	71.49	70.49	0.2205	71.64	12.47
	XGBoost-RS	71.53	69.78	73.53	0.1687	69.42	14.65
	XGBoost TPE	69.59	70.89	71.63	0.1734	70.83	13.84
	PSO-XGBoost	70.37	72.18	74.76	0.1689	72.18	11.87
	AHPSO-XGBoost	**73.54**	**74.87**	**76.89**	**0.1582**	**74.27**	**11.16**
RRDai	XGBoost	74.64	69.73	71.43	0.1708	70.67	12.74
	XGBoost- GS	70.85	71.47	68.57	0.1636	69.42	13.56
	XGBoost-RS	72.58	70.83	69.43	0.2134	71.54	11.87
	XGBoost TPE	71.87	72.82	70.82	0.1672	68.71	12.25
	PSO-XGBoost	69.43	73.49	72.69	0.1587	73.47	11.34
	AHPSO-XGBoost	**76.49**	**75.58**	**73.45**	**0.1503**	**74.38**	**10.46**
LendingClubLoan	XGBoost	71.42	70.41	69.79	0.1504	59.87	13.27
	XGBoost-GS	73.19	69.86	72.69	0.1484	49.76	11.54
	XGBoost- RS	70.42	72.58	70.83	0.1253	53.28	12.47
	XGBoost TPE	72.39	71.39	71.59	0.1386	48.93	14.42
	PSO-XGBoost	73.87	74.83	73.82	0.1164	52.63	11.35
	AHPSO-XGBoost	**76.87**	**75.78**	**74.89**	**0.1056**	**49.89**	**11.21**

5.4 Kolmogorov-Smirnov Evaluation metric

The evaluation of the generalization performance of machine learning algorithms tailored for credit scoring requires evaluation metrics capable of highlighting how well creditworthy and non-creditworthy

customers are distinguished and ascertaining whether the learning model as changed or not. In credit scoring, the availability of more data does not guarantee that more training data will improve the generalization performance of the learning model when using traditional batch learning classifiers. More data may lead classifiers to overfit and the data may exhibit drifting characteristics. To carry out an evaluation of the capability of a machine learning model to accurately discriminate customers that will pay their debts in full or not and to evaluate whether the variables have drifted or not, we implement the Kolmogorov-Smirnov (KS) metric [34]. The population scores may exhibit concept drift [35] as population scores may evolve over time and some populations may remain stable. To evaluate whether the population scores have changed or not, the Population Stability Index (PSI) is adopted [36]. If the population Stability Index (PSI) evolves over time, it is an indication of the degradation of the prediction model and a sign that the population scores are changing a new prediction model needs to be built. The Kolmogorov-Smirnov (KS) statistic indicates the maximum distance between the cumulative probability distribution function (cdfs) obtained by customers that pay their debts in full and those who default [34]. Without loss of generality, if the number of customers to be scored is $(n+m)$, the probability that the ith customer will default is denoted as D_i = 1 and D_i=0 otherwise. The empirical cumulative distribution functions (cdf) of good and bad customers are expressed using Eqs. 1 and 2 respectively and n denotes the total number of good customers and m represents the number of good customers, L=min_{si} 1 ≤ i ≤ $(n+m)$ is the lower bounds of all the scores available and H= max_{si}1<i<$(n+m)$ is the upper bound.

$$F_{good(a)} = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} 1 ifs_{i\leq} a_{and} D_i = 1 \\ 0 otherwise \end{cases} with a \in [L, H]$$

(19)

$$F_{bad(a)} = \frac{1}{m}\sum_{i=1}^{m} \begin{cases} 1 ifs_i < a and D_i = 0 \\ o otherwise \end{cases} with a \in [L, H]$$

(20)

The Kolmogorov-Smirnov metric is given Eq. (21) and it expresses highest difference that exists between the cdfs that provides the description between the good and the bad customers.

KS= $max_{a \in [L, H]}$|F_{bad}(a)-F_{good}(a)| (21)

If the value of the KS statistic is zero, it's a sign that the two credit score distributions are the same and is an indication that the credit score cannot distinguish between defaulters and nondefaulters. If the KS value equals to 100, it is an indication that the credit score accurately distinguishes defaulters from nondefaulters. A KS score greater than 35% indicates a substantial discriminative power of the prediction model to distinguish te different types of customers. The Population Stability (PSI) shows the changes that happen in the population distribution of loan applicants. PSI reveals changes that occur in the environment that needs to be analyzed further by bank experts to determine whether any macroeconomic conditions or lending policies are affecting the model outcomes [36]. To get the PSI score, we compute

the probability distribution function (pdf) of the defaulting customers in two different time periods. Firstly, the pdfs of these distributions are calculated using a specific number of ranges so that each range has approximately the same number of defaulting customers. The variables n_i and m_i are counters of defaulting customers in the two samples at the ith bin and that $\sum n_i = N$ and $\sum m_i = M$. Given these counters, the PSI can be computed as:

$$PSI = \sum_{i=1}^{r} [(\frac{ni}{N} - \frac{mi}{M}) \times (\ln\frac{ni}{N} - \ln\frac{mi}{M})] \quad (22)$$

If PSI rates are below 10%, it is an indication that the population is quite stable and that the learning model is not unacceptably volatile.

5.5 AHPSO-XGBoost performance using the Kolmogorov-Smirnov metric

The use of batch learning to assess the prediction performance of AHPSO-XGBoost does not provide a proper indication of whether variables will drift over time. To evaluate the effectiveness of AHPSO-XGBoost on both static and dynamic domains for it to properly distinguish creditworthy and noncreditworthy customers in environments where the customer variables may drift or evolve with time, we use the two read world datasets, RRDai and the LendingClubLoan. The traditional credit scoring datasets used in the literature are too small in terms of features and instances available and unclear with regard to what each feature stands for. For the two real world datasets, the target is an indication of whether customers paid their debt in full or defaulted in the month being analyzed. The learning algorithms used to perform a comparative study analysis have different hyperparameters that may impact the generalization performance of the prediction models to be learned. For AHPSO-XGboost, the hyperparameters are optimized by the PSO algorithm and the rest of the algorithms are optimized using a grid search with ten fold cross-validation with the goal of maximizing the KS rates in the test data. To learn incrementally, the data stream learning algorithms are tuned so that the KS rates during the testing phase are optimized. To validate the prediction performance of AHPSO-XGBoost against other incremental learning algorithms, we adjusted the Holdout validation scheme together with the Test and Train approach since holdout validation alone is not tailored for streaming scenarios where data is streaming and often collected over an extended period of time. In the holdout validation scheme, the available data is split into a specific number of months for training and the residue is used for testing. The objective is to test if the created predictive model, AHPSO-XGBoost, KS and PSI generalize well to unseen instances or changing populations or not over time. In the event that the nature of the datais indeed ephemeral, the expectation is that both AHPSO-XGBoost and KS rates are to drop over time after the model is learned and if PSI rates would increase. In the event that AHPSO-XGboost, KS and PSI rates remain the same, incremental learning would not be implemented. To properly perform an informed analysis, different training and testing ratios of months are used. Different proportions of the dataset are tested in order to evaluate the impact of different volumes of data and to find out if the behavior of loan applications evolve or change over time. If the variables affecting loan applications are drifting, then a batch model would erroneously score the loan applications resulting in lower KS and AHPSO-XGBoost

rates and will generate a higher PSI. The monthly- test then train validation scheme uses each month to train just after the evaluation process. The objective is to make a comparison of the results generated by a learner that is constantly being updated with new data against a learning model that was learned to a predetermined point and the evaluation was performed later. The comparison of the results of the two different models enables us to determine whether the process of continuously updating the predictive learning models significantly improves the AHPSO-XGBoost and KS results without compromising the PSI rates.

5.6 Analysis of the results

Figure 1 shows the KS rates for the RRDai dataset. All the algorithms are learning incrementally and to evaluate their behavior, we use the KS rates

From the experiments conducted on the RRDai dataset, there is a positive trend as the KS rates grow over time for all the stream learning algorithms, an indication that the classifiers are capable of discerning between creditworthy and non-creditworthy customers although more training and testing data continues to arrive. The Friedman test [37] was used to compare the ranking performance of all the ten stream learning algorithms. The Friedman test is a nonparametric test that ranks the classifiers separately. The Friedman test revealed that there is no significant statistical differences that exist among the ten methods and the PSI rates of all the classifiers are low.

Figure 2 shows the PS rates obtained from the RRDai dataset. The PS rates are fluctuating over time, an indication that the variables are drifting and the prediction model has degraded and the population itself is changing and a new learning model needs to be built.

Figure 3 shows the KS rates on the LendingClubLoan dataset. All the data stream learning algorithms were able to learn a consistent prediction model capable of discerning between creditworthy and non-creditworthy customers.

The KS results obtained indicate that the availability of more data generated higher KS rates. The Friedman test applied indicate that there is no statistical differences between the KS rates observed and this is highly attributed to the face that the dataset is quite stable.

The PS rates for the LendingClubLoan are shown below. The population continues to change as in the RRDai dataset even though the dataset shows sign of stability. The PS rates continue to fluctuate and the model needs to be updated continuously.

5.6 Statistical Tests of Significance

To conclude the empirical experiments conducted, we perform a complete performance evaluation by implementing some hypothesis testing to evaluate if the experimental differences in the performances are statistically significant. To accomplish this, the Friedman test [37], is used to perform the comparison of the performance rankings of all the algorithms. The Friedman test is a nonparametric test and ranks the classifiers separately. The null hypothesis of the Friedman test states that all classifiers under

consideration perform identically and all differences are only random fluctuations. In the event that the null hypothesis of the Friedman test is rejected, then a post hoc test is conducted to find the particular pairwise comparison that generates significant differences. The post hoc test used in the significance test evaluation is the Nemenyi test [38]. In the Nemenyi test, the prediction performances of two learning algorithms is considered to be significantly different if their average ranks differ by at least the Critical Difference (CD). The Critical Difference (CD) is expressed as

CD = $q_a \sqrt{\frac{k(k+1)}{6n}}$, and q_a represents the critical value of the Tukey distribution and the variable k represents the number of machine learning prediction models that are to be compared and n stands for the number of datasets.

Table 5 shows the average rank of each model on each dataset.

Table 5
The average rank of each model on each dataset

Classifier	Australia	German	RRDai	LendingClubLoan	Average Rank
AdaBoost	2.25	3.45	3.42	2.85	2.99
AdaBoost-NN	3.14	2.78	3.78	4.00	3.43
Bagging DT	2.84	3.46	2.96	3.86	3.28
Bagging NN	1.94	3.58	2.86	4.68	3.27
Random Forest	1.86	2.74	3.48	3.94	3,01
Logistic Regression	2.72	3.82	2.98	3.84	3.34
Decision Tree	3.24	2.54	1.86	4.35	4.00
Leverage Bagging	1.35	1.89	2.34	2.18	1.94
Adaptive-Random Forest	1.48	2.34	2.00	2.54	2.09
AHPSO-XGBoost	1.00	1.35	2.26	2.45	1.77

The proposed AHPSO-XGBoost prediction model performs the best among the ten models with the average rank of 1.77. AHPSO-XGBoost is slightly better than Leverage Bagging with an average rank of 1.94 and this demonstrates the robustness and effectiveness of our approach in distinguishing between credit worthy and not creditworthy customers. Figure 5 shows the CD diagram of the results of the Nemenyi test. The average rankings of all the machine learning models are represented by the horizontal axis where the difference in average ranks is lower than the CD value, the models are connected by a black bar.

Figure 5 shows the CD diagram with the results of the Nemenyi test. In the Nemenyi diagram, all the ten learning model ranks are shown and our proposed approach, AHPSO-XGBoost model is significantly better than the other nine learning models.

6 Conclusion

Recently, machine learning approaches have been applied in the creation of credit scoring models due their promising ability to accurately classify customers as credit worthy and not credit worhy. To improve

the accuracy and ability of machine learning in accurately distinguishing between credit worthy and non-credit worthy customers, we proposed the Adaptive Heterogeneous Particle Swarm Optimized XGBoost capable of learning in both static and dynamic environments. The hyperparameters of XGBoost are optimized by randomly selecting selecting the values of the parameters from a behavior pool. This helped to avoid premature convergence of the swarm. In the first experiments conducted, we used batch learning to evaluate the behavior of our proposed approach and it performed comparatively well. In the second experiment conducted, we validated our approach against drifting variables and population changes and thus addressing the credit scoring task as a data stream classification problem. To effectively address the problem, we used the Kolmogorov-Smirnov metric and the Population Stability Index to check if the population of customers was changing. The data stream learning approach generated interesting discriminative rates in all the datasets. In future, we will address the problem of drifting variables and populations using stream data learning using a combination of different learning algorithms.

Declarations

Author's Contributions

Every section of this paper was written by I, Tinofirei Museba.

Declaration of Competing Interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval and consent to participate:

This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of data and material

The data sets used in this paper are available publicly and their sources are highlighted in the text.

Funding

No funding was received to conduct this research.

Consent for publication

The author gives the right to the journal to publish the work.

Acknowledgements

I thank the Editor in Chief and the reviewers for agreeing to review my work.

References

1. Jean Paul Barddal, Lucas Loezer, Fabricio Enembreck, Riccrado Lanzuolo, “Lessons learned from data stream classification applied to credit scoring”, Expert Systems with Applications 162, 2020.
2. D.S.C Nascimento, A.L.V Coelho and A.M.P Canuto, “Integrating complementary techniques for promoting diversity in classifier ensembles: a systematic study”, Neuro-computing, Volume 138, pages 347–357, 2014.
3. Chao Qin, Yunfeng Zhang, Fangxun Bao, Caiming Zhang, Peide Liu and Peipei Liu, “XGBoost Optimized by Adaptive Particle Swarm Optimization for Credit Scoring”, Hindawi, Mathematical Problems in Engineering, Volume 2021.
4. R.C Eberhart and J.Kennedy, “A New Optimizer using Particle Swarm Theory”, In Proceedings of the MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, October 1995.
5. W.Verbeke, D.Martens, C.Mues, B.Baesens, “Building comprehensible customer churn prediction models with adavanced rule induction techniques”, Expert Systems with Applications 38(3), pages 2354–2364, 2011.
6. Jasmina Nalic, Goran Martinovic, Diago Zagar, “ New Hybrid data mining model for credit scoring based on feature selection algorithm and ensemble classifiers”, Advanced Engineering Informatics 45, 2020.
7. Jin Xiao, Xu Zhou, Yu Zhong, Ling Xie, Xin Gu, Dunhu Liu, “Cost Sensitive semi-supervised selective ensemble model for customer credit scoring”, Knowledge-Based Systems 189, 2020.
8. Yufei Xia, Junhao Zhao, Lingyun He, Yinguo Li, Mengyi Niu, “A Novel Tree-based dynamic heterogeneous ensemble method for credit scoring”, Expert Systems with Applications 159, 2020.
9. Vincenzo Moscato, Antonio Picariello, Giancarlo Sperli, “A benchmark of machine learning approaches for credit score prediction”, Expert Systems with Applications 165, 2021.
10. Feng Shen, Xingachao Zhao, Gang Kou, Fawaz E. Alsaadi, “A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique”, Applied Soft Computing Journal 98, 2021.
11. Salvatore Carta, Anselmo Ferreira, Diego Reforgiato Recupero, Mario Saia, Roberto Saia, “A combined entropy –based approach for a proactive credit scoring”, Engineering Applications of Artificial

Intelligence 87, 2020

12. Wenyu Zhang, Hongliang He, Shuai Zhang, “A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring”, Expert Systems with Applications 121, pages 221–232, 2019.
13. Shuang Wei, Dongqi Yang, Wenyu Zhang, Shuai Zhang, “A novel noise adapted two-layer ensemble model for credit scoring based on Backflow learning’, IEE Access, Volume 7, pages 99230–99230, 2019.
14. Xiaohong Chen, Siwei Li, Yuanhua Xu, Fanyong Meng, Wenzhi Cao, “A Novel GSCI –based ensemble approach for credit scoring”, IEEE Access, Volume 8, pages 222449–222465, 2020.
15. Yilun Jin, Wenyu Zhang, Xin Wu, Yanan Liu, Zeqian Hu, “A novel multistage ensemble model with a hybrid genetic algorithm for credit scoring on imbalanced data”, IEE Access, Volume 9, pages 143593–143607, 2021
16. Chaoqan Wang, Zhoongyi Hu, Raymond Chiang, Sandeep Dhakal, Yuan Chen, Yukan Bao, “A PSO-Based ensemble model for per to peer credit scoring”, 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), July 2018.
17. Andries P. Engelbrecht “Computational Intelligence: An Introduction”, John Wiley and Sons, Chichester, December 2002
18. Y,C Chang, K.H Chang and G.J.Wu, “Applications of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions”, Applied Soft Computing, Volume 73, pages 914–920, 2018.
19. Y.C. Chang, K.H.Chang, H.H. Chu, L.I. Tong, “Establish decision tree-based short term default credit risk assessment models”, Communication in Statistics, Theory and Methods 45(23), pages 6803–6815, 2016.
20. Hu Wang, Li Zhi-Shu, “A simpler and more effective particle swarm optimization algorithm”, Journal of Software, Volume 18, Number 4, pages 861–868, 2007.
21. T. Blackwell, “Particle swarms and population diversity”, Soft Computing- A Fusion of Foundations, Methodologies and Applications, Volume 9, Issue 11, November 2005.
22. J.Kennedy, “Bare bones particle swarms”, Proceedings of the 2003 IEEE Swarm Intelligence Symposium SIS’03, pages 80–87, 2003.
23. H.Xiao, Z.Xiao and Y.Wang, “Ensemble classification based on supervised clustering for credit scoring”, Applied Soft Computing, Volume 43, pages 73–86, June 2016.
24. C.F Tsai, Y.F Hsu and D.C Yen, “A comparative study of classifier ensembles for bankruptcy prediction”, Applied Soft Computing, Volume 24, pages 977–984, November 2014.
25. T.T.Tang, “Information asymmetry and firms’ credit market access: Evidence from Moody’s credit rating format refinement”, Journal of Financial Economics, 93, pages 325–351, 2009.
26. J.Wu, Q.Zhang, Y.Zhao, “Radiomics analysis of iodine-based material decomposition images with dual-energy computed tomography imaging for preoperatively predicting microsatellite instability

status in colorectal cancer", Frontiers in oncology, Volume 9, pages 1250–1260, 2019.

27. A.Cano, B.Krawczyk, "Kappa updated ensemble for drifting data stream mining", Machine Learning 109, pages 175–218, 2020.
28. N.V Chawla, K.W Bowyer, L.O Hall and W.P Kegelmeyer, "SMOTE; Synthetic minority over-sampling technique", Journal of Artificial Intelligence, Volume 16, Number 1, pages 321–357, 2002.
29. S.X Wang, P.F Dong, Y.J Tian, "A novel method of statistical line loss estimation for distribution feeders based on feeder cluster and modified XGBoost", MDPI Energies, 10(27), 2067, 2017.
30. Maher Alaraj, Maysam F.Abbod, Munir Majdalawieh, Luay Jumia, "A deep learning model for behavioral credit scoring in banks", Neural Computing and Applications, pages 5839–5866, 2022, https://doi.org/10.1007/s00521-021-06695-z
31. A.Asuncion and D.Newman, "UCI Machine Learning Repository", Publishing, 2007.
32. T.Chen, C.Guestrin, "XGBoost: A scalable tree boosting system", In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, pages 785–794, 2016.
33. G.Wang, J.Hao, J.Ma and H.Wang, "A comparative assessment of ensemble learning for credit scoring", Expert Systems with Applications, Volume 38, Number 1, pages 223–230, January 2011, doi 10.1016/j.eswa.2010.06.048.
34. M.Rezac and F.Rezac, "How to measure the quality of credit scoring models", Czech Journal of credit scoring models", 61(5), pages 486–507, 2011
35. S.Priya and R.Annie Uthra, "Deep learning framework for handling concept drift and class imbalance complex decision making on streaming data", Complex and Intelligent Systems, 2021. https://doi.org/10.1007/s40747-021-00456-0
36. G.Karakoulas, "Empirical validation of retail credit-scoring models", The RMA Journal, 87(1), pages 56–60, 2004.
37. J.Abellan and J.G Castellano, "A comparative study on base classifiers in ensemble methods for credit scoring", Expert Systems with Applications, Volume 73, pages 1–10, May 2017.
38. Y.Xia, C.Liu, B.Da and F.Xie, "A novel heterogeneous ensemble credit scoring model based on bstacking approach", Expert Systems with Applications, Volume 93, pages 182–199, March 2018.

Figures

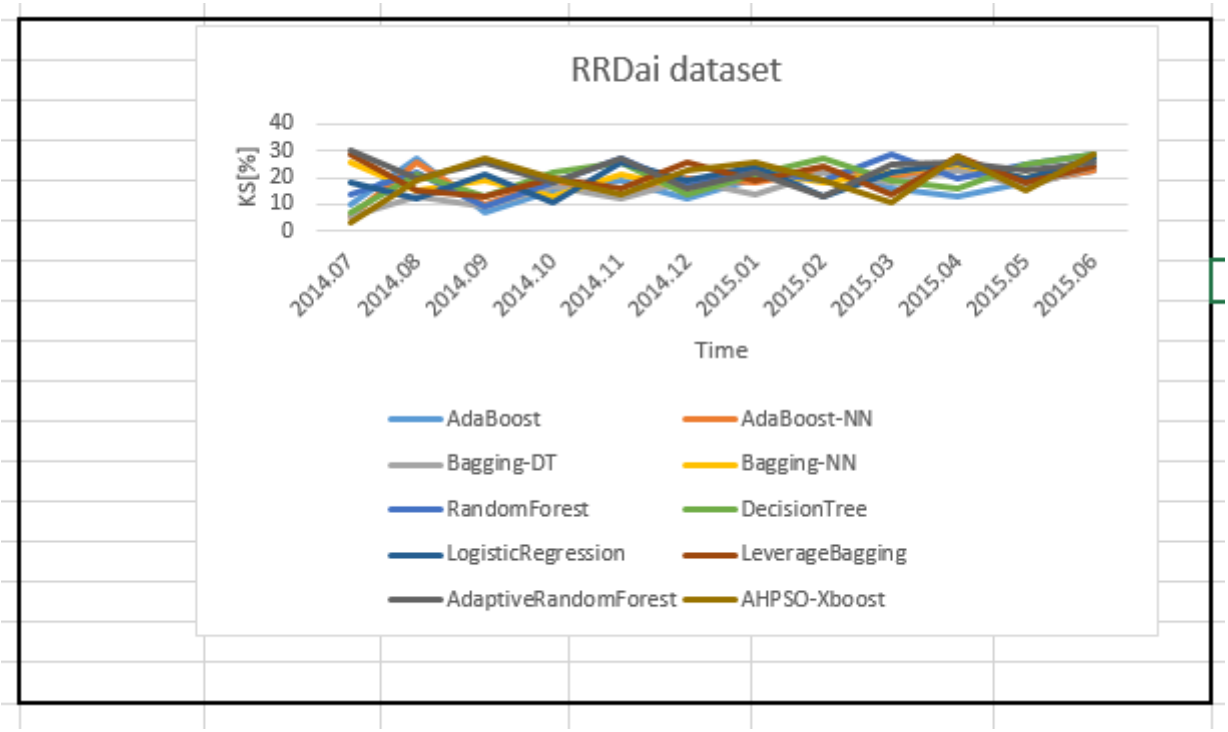

Figure 1

KS(%) rates obtained in RRDai dataset

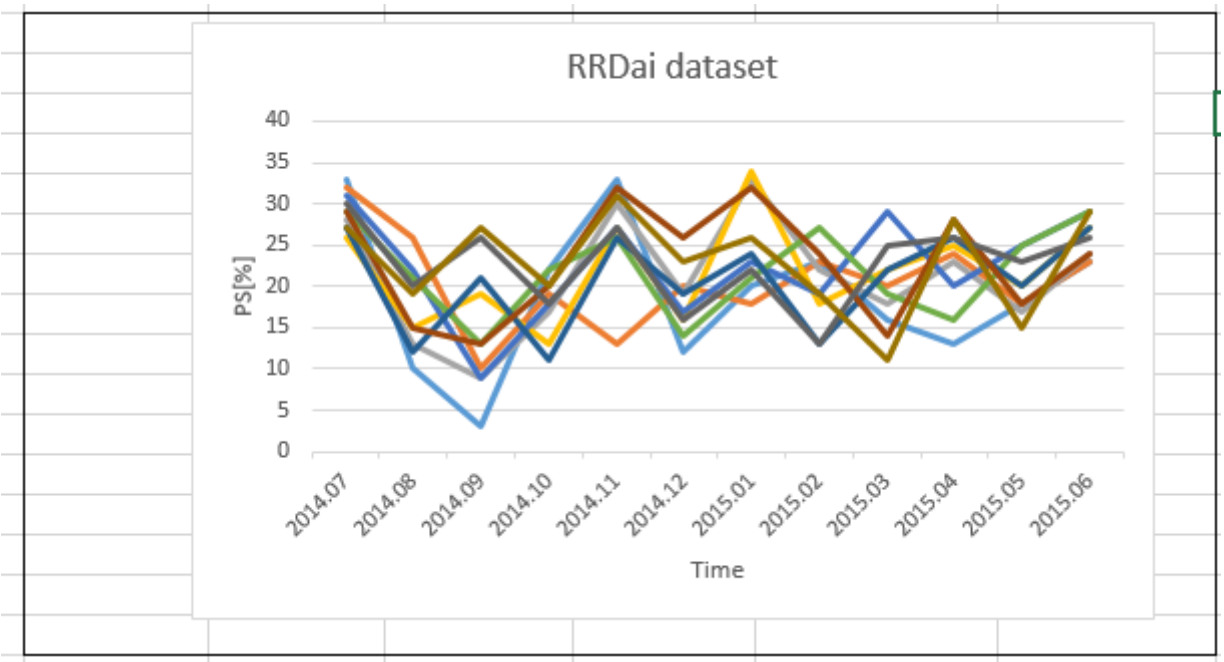

Figure 2

PS(%) rates obtained on RRDai dataset

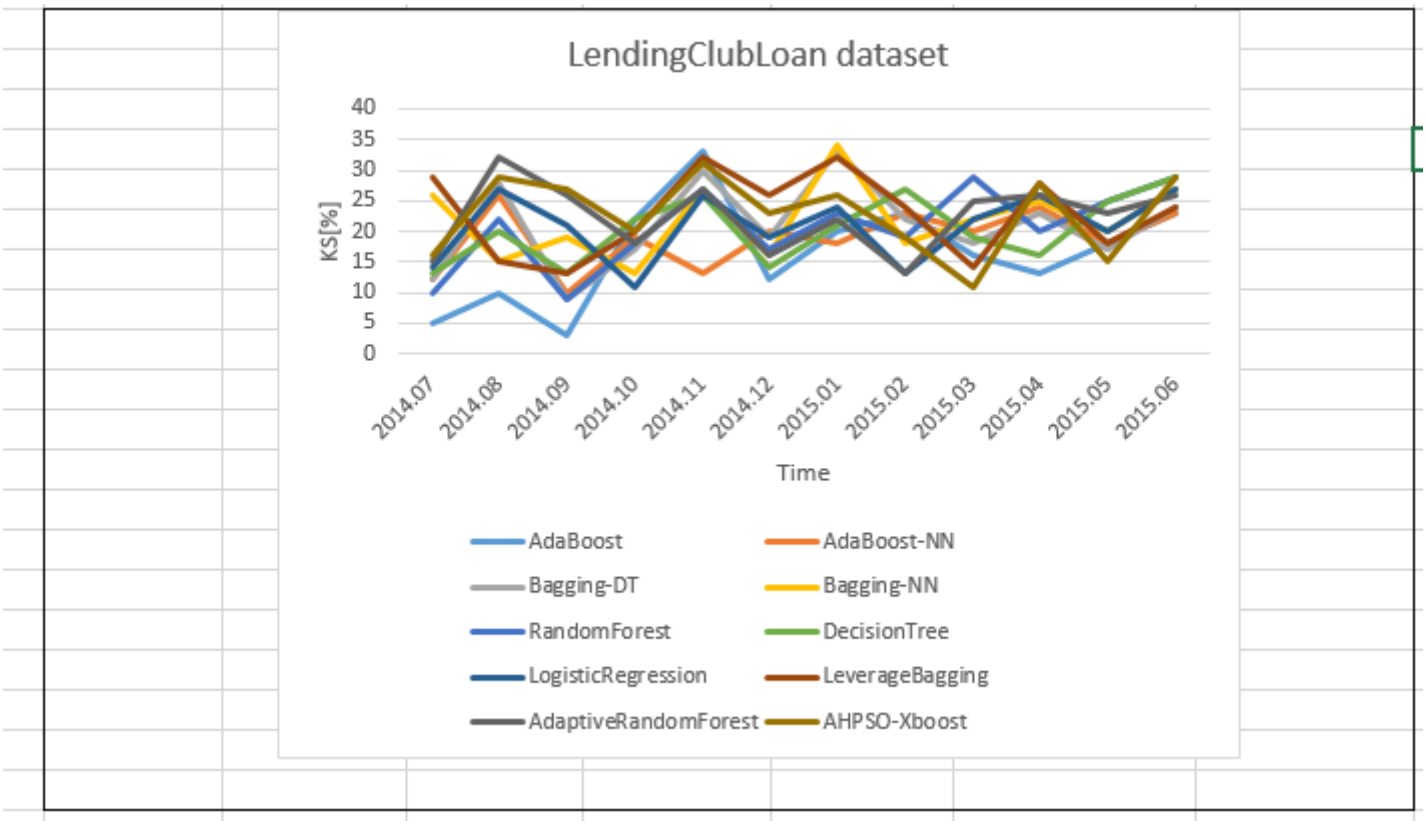

Figure 3

KS(%) rates obtained from the LendingClubLoan dataset

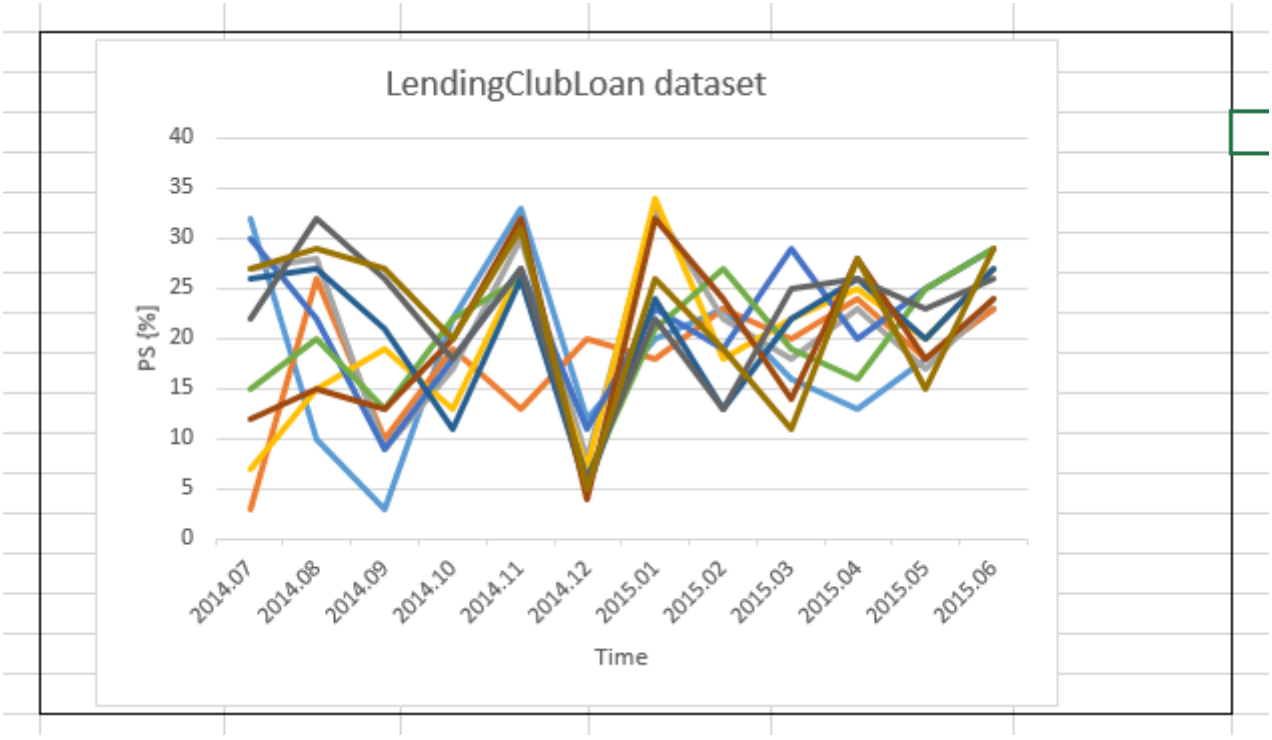

Figure 4

PS(%) rates obtained on the LendingClubLoan dataset

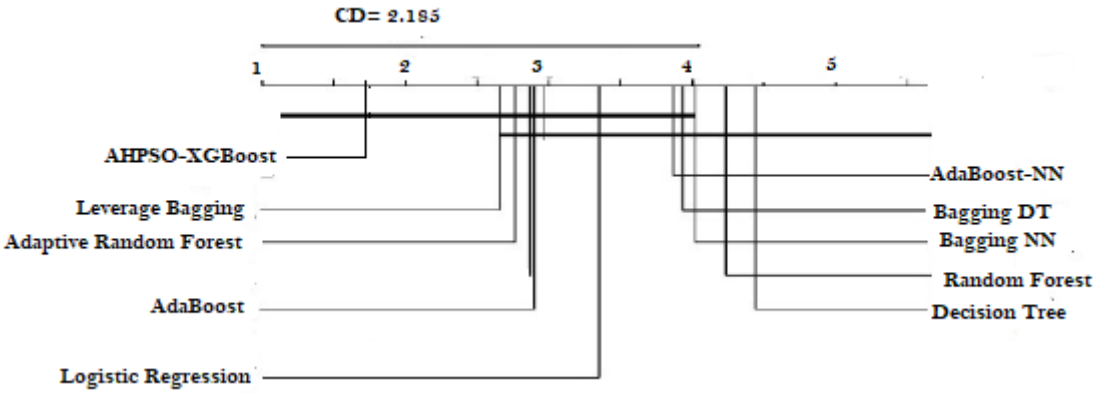

Figure 5

The CD diagram with the results of the Nemenyi test