International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 6, December 2021, pp. 5477~5487
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i6.pp5477-5487

Improved credit scoring model using XGBoost with Bayesian hyper-parameter optimization

Wirot Yotsawat, Pakaket Wattuya, Anongnart Srivihok
Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok, Thailand

Article Info

Article history:

Received Jan 11, 2021
Revised Apr 18, 2021
Accepted Apr 26, 2021

Keywords:

Bayesian optimization
Classification
Credit scoring
Creditworthiness assessment
XGBoost

ABSTRACT

Several credit-scoring models have been developed using ensemble classifiers in order to improve the accuracy of assessment. However, among the ensemble models, little consideration has been focused on the hyper-parameters tuning of base learners, although these are crucial to constructing ensemble models. This study proposes an improved credit scoring model based on the extreme gradient boosting (XGB) classifier using Bayesian hyper-parameters optimization (XGB-BO). The model comprises two steps. Firstly, data pre-processing is utilized to handle missing values and scale the data. Secondly, Bayesian hyper-parameter optimization is applied to tune the hyper-parameters of the XGB classifier and used to train the model. The model is evaluated on four widely public datasets, i.e., the German, Australia, Lending club, and Polish datasets. Several state-of-the-art classification algorithms are implemented for predictive comparison with the proposed method. The results of the proposed model showed promising results, with an improvement in accuracy of 4.10%, 3.03%, and 2.76% on the German, Lending club, and Australian datasets, respectively. The proposed model outperformed commonly used techniques, e.g., decision tree, support vector machine, neural network, logistic regression, random forest, and bagging, according to the evaluation results. The experimental results confirmed that the XGB-BO model is suitable for assessing the creditworthiness of applicants.

Corresponding Author:

Wirot Yotsawat
Department of Computer Science
Faculty of Science, Kasetsart University
50 Ngam Wong Wan Road, Lat Yao, Chatuchak, Bangkok, 10900, Thailand
Email: wirot.yo@ku.th

1. INTRODUCTION

Lending is a major source of income generation for most banks and other loan businesses. In the last few decades, credit scoring models have been developed for credit granting decisions. These models have traditionally implemented regression techniques that rely on several factors to characterize creditworthiness. At present, credit scoring models are utilized in modern artificial intelligence techniques through challenge environments. These models have become a popular and critical tool for financial institutions because it helps them to make profitable financial decisions and can prevent great losses due to poor decisions on loan granting, especially NPLs.

A large number of credit scoring models have been proposed as alternative choices for financial institutions. Generally, the development of a credit scoring model consists of two major techniques: statistical methods and machine learning methods. A decision tree method like C4.5 is a popular and powerful machine learning algorithm with high performance in credit scoring models [1]-[4]. Despite the plethora of

classification algorithms available within machine learning for credit scoring, the most suitable single classifier for enhancing a model is still not clear based on the current literature. In the last few decades, multiple classifier methods have been proposed and applied to manage the credit scoring problem [5]-[12]. Recently, extreme gradient boosting, or XGBoost (XGB), has shown excellent performance in many domains, including credit scoring [13]. The parameter settings of XGB often have an important effect on its performance. However, in previous studies, the hyper-parameter settings of credit scoring models have often been ignored [7], [14], [15], or grid search has been used with insufficient parameter space [13], [16].

Most classifiers, such as neural networks (NNs), support vector machine (SVM), and XGB, have a number of hyper-parameters that critically influence the efficiency of the model. Thus, the hyper-parameter tuning step should be carefully investigated. A popular hyper-parameter tuning approach is grid optimization, which thoroughly searches using the given hyper-parameter values [17]. However, grid optimization leads to other problems, including the curse of dimensionality, because its computational cost (such as time and memory space) increases dramatically with the number of given hyper-parameters. Thus, grid optimization is not suitable for classification algorithms with many hyper-parameters, especially XGB.

Li and Chen [16] proposed a comparative performance evaluation of several ensemble methods based on the Lending club dataset, and reported that RF showed the best performance. XGB was also included in the experiment. However, grid optimization was utilized to tune the XGB hyper-parameters, which did not allow enough hyper-parameter space. Munkhdalai *et al.* [13] compared the performance of machine learning methods on bank client credit assessments and found that the XGB algorithm achieved the highest accuracy. Li [15] suggested that XGB outperformed LR on credit risk prediction. Brown and Mues [18] proposed the comparison of credit scoring methods utilizing imbalanced datasets and reported that gradient boosting and RF classifiers performed very well. However, hyper-parameter tuning was not investigated. Our empirical study showed that the correct setting of hyper-parameters has a considerable impact on model performance.

Using public datasets, Malik and Hermawan [19] suggested that binary particle swarm optimization (BPSO) could improve the accuracy of the CART classifier. Xia *et al.* [12] found that the Bayesian hyper-parameter optimization technique outperformed other optimization techniques, e.g., grid search, manual search, and random search, in terms of accuracy. In our study, Bayesian hyper-parameter optimization is utilized by the tree-structured parzen estimator (TPE), which is a specific technique of Bayesian hyper-parameter optimization.

In this study, to improve the performance of the credit scoring model, XGB and Bayesian hyper-parameter tuning are combined. The XGB is used in the learning process. Bayesian hyper-parameter tuning is used to determine the hyper-parameters for obtaining the best model. The proposed model is compared against single classifiers that are widely used in credit scoring, such as DT, LR, KNN, SVM, NN, and Bagging. Four different public datasets have been included in the experiment for performance comparison. The remainder of the paper is structured as follows: Section 2 describes the materials and method that we use to develop the model. Section 3 presents the details of the experimental results and discussion. Section 4 draws conclusions and future research directions.

2. MATERIALS AND METHOD

2.1. Datasets

To evaluate the performance of XGB-BO, we used four datasets, namely the German, Polish, Australian, and Lending club datasets, which are widely used by researchers. The German, Polish, and Australian datasets are available in the UCI repository, and the Lending club dataset is provided in [12]. The datasets vary between 1 and 24.93 in terms of the imbalance ratio (IR). The summary of all datasets is depicted in Table 1.

2.2. Experimental setup

In this study, the credit scoring models were built using a common public dataset. The experiment was performed on a Windows 10 operating system with an Intel Core i7 7500 CPU and 8 GB of RAM. Python version 3.6 was used in the computer, along with other associated libraries. The Scikit-learn library version 0.20.0 was used to apply several well-known algorithms, such as SVM, LR, KNN, and DT, as well as other ensemble methods. XGBoost version 0.90 was used for the specific XGB classifier. As well, Hyperopt 0.2.4 was used for Bayesian hyper-parameter tuning.

2.3. Research frameworks

The conceptual framework of the proposed credit scoring model is illustrated in Figure 1. It comprises two steps, namely data preprocessing and model training and testing.

Table 1. Description of the datasets in the study

Dataset	Attributes	Instances	Good:Bad	IR	Source
German	24	1,000	700:300	2.33	UCI
Polish	64	7,027	6,756:271	24.93	UCI
Australian	14	690	383:307	1.25	UCI
Lending club	10	2,642	1,322:1,320	1.00	[12]

2.3.1. Data preprocessing

Data preprocessing is an important step to be taken before a model is constructed for data mining and machine learning tasks. It can improve the performance of the model in terms of accuracy and time complexity. To prepare the data for a classification algorithm, it is preprocessed so that it is representative and consistent. The data preprocessing steps are as follows.

- Data cleaning: The missing values are managed. The attributes having the same value for more than 99% of instances or having missing values for more than 30% instances are eliminated. The remaining missing values are replaced by the average or mode value of the entries, depending on the attribute's data type.
- Data transformation: Some classification algorithms, such as SVM and NN, require numerical data. Thus, one-hot encoding is applied to convert the categorical input features. Furthermore, different ranges of numerical attributes are normalized within a narrow, fixed range.

2.3.2. Model training and testing

According to the literature, XGB is a powerful classification algorithm. The XGB algorithm requires a number of hyper-parameters that should be carefully tuned. To train the XGB model, we adopted Bayesian search for hyper-parameter tuning. A grid search and default parameters were also implemented for comparing the performance of models. The hyper-parameter tuning was done within the given parameter space, as described in Table 2. The best parameter values for single classifiers were set for the base classifiers for Bagging. Excluding the parameter settings, the other parameters were set to default values with respect to the common mode in the literature. Other widely used classification algorithms were also developed to evaluate the proposed model.

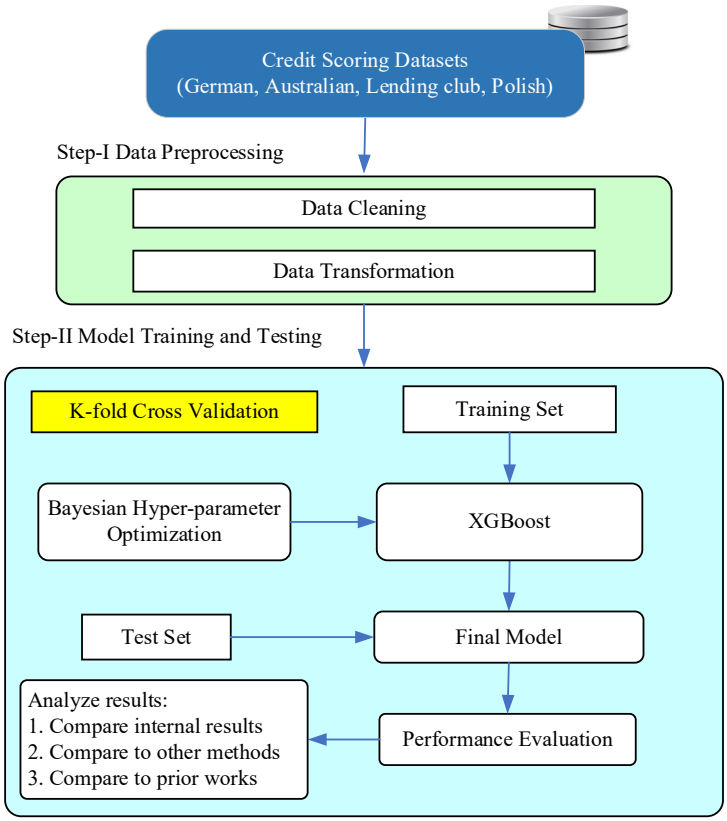

Figure 1. Conceptual framework for proposed XGB-BO

Table 2. Hyper-parameters space for optimization

Method	Parameters	Parameters space	
		Grid search	Bayesian search
SVM	gamma	0.0001, 0.001, 0.01, 0.1, 1.0	[0.0001, 1.0]
	C	1, 5, 10, 15, 50	[1, 50]
*K*NN	K	3, 5, 7, 9, 11, 13, 15, 20, 25, 30	[3, 30]
	metric	euclidean, manhattan, mahalanobis	euclidean, manhattan, mahalanobis
DT	Minimum sample split (min_split)	2, 3, 4, 5, 10, 15, 20	[2, 20]
	Minimum sample leaf (min_leaf)	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20	[1, 20]
	Maximum tree depth (max_depth)	3, 4, 5, 6, 7, 8, 9, 10, 15	[3, 30]
	Maximum features (max_features)	2,3,…,n_feature	[2, n_feature]
NN	Hidden layer sizes (hidden_layer)	n_feature/2, n_features, n_feature*2	[n_feature/2, n_feature*2]
	Batch size	8, 16, 32	[4, 64]
	Solver	adam, sgd	adam, sgd
	Initial learning rate (initial_lr)	0.001, 0.01, 0.1	[0.001, 0.1]
Bagging	Number of estimators (n_estimators)	3, 5, 7, 9, 10, 11	[3, 11]
	Subsample ratio (subsample)	0.8, 0.9, 1.0	[0.8, 1.0]
XGB	Maximum tree depth (max_depth)	4, 5, 6, 7, 8, 9	[3, 15]
	Subsample ratio (subsample)	0.8, 0.9 ,1.0	[0.8, 1.0]
	Learning rate	0.01, 0.1	[0.01, 0.1]
	Maximum delta step (max_delta_step)	0.0, 0.3, 0.6, 0.9, 1.0	[0.0, 1.0]
	Column subsample ratio (colsample_bytree)	0.8, 0.9, 1.0	[0.8, 1.0]
	Minimum child weight (min_child_weight)	0, 1, 2, 3, 4	[0, 5]
	Number of boosts (n_estimators)	100, 150, 200	[80, 200]
	gamma	0.0, 0.001, 0.01, 0.1	[0.0, 1.0]
RF	Minimum sample split (min_split)	2, 4, 5, 6, 8, 10, 20	[2, 25]
	Minimum sample leaf (min_leaf)	1, 3, 5, 10	[1, 25]
	Maximum features (max_features)	4, 5, 6, 7, 8, 9, 10	[4, 15]
	Maximum tree depth (max_depth)	4, 6, 8, 10	[4, 25]
	Number of trees (n_estimators)	100, 300, 500	[50, 300]

2.4. Performance measurement

Based on the confusion matrix, the performance indices were computed to evaluate the models. In the credit scoring application, true positive (TP) and true negative (TN) represent the numbers of correctly classified good (non-default) and bad (default) borrowers, respectively. False negative (FN) and false positive (FP) represent the numbers of misclassified good and bad borrowers, respectively.

- Accuracy (Acc) measures the overall true predicted value in all classes. However, the accuracy score alone cannot indicate model performance because some datasets are imbalanced. Thus, it only reflects the overall prediction accuracy of the dataset.

$$Acc = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

- Sensitivity (Sen) measures the proportion of actual positives that are correctly identified. In this case, it identifies the percentage of non-default loans that are correctly classified by the models as good applicants.

$$Sen = \frac{TP}{TP+FN} \quad (2)$$

- Specificity (Spec) measures the proportion of actual negatives that are correctly detected by the classifiers. In this case, it measures the percentage of default loans that are correctly detected as bad applicants.

$$Spec = \frac{TN}{TN+FP} \quad (3)$$

- Area under the ROC curve (AUC) measures the classification ability of the entire sample and the balance of classified samples simultaneously. Thus, it can be considered a more appropriate measure for imbalanced credit scoring [7]. The AUC is derived from the area under the ROC curve, which plots the TP rate (Sen) along the y-axis and the FP rate along the x-axis. The AUC score ranges from 0 to 1, where a value close to 1 indicates the model has high accuracy.
- The geometric mean (GM) is a comprehensive evaluation measurement computed by Sen and Spec. A high GM indicates the balance between classes is reasonable and good performance in a binary classification model. The GM is computed using (4).

$$GM = \sqrt{Sen \times Spec} \quad (4)$$

3. RESULTS AND DISCUSSION

presented in this section. This section is divided into two sub-sections. Firstly, the classification results show a comparison of the internal results and the results generated by other methods. Secondly, the model comparison describes the performance of the proposed XGB-BO compared with prior works.

3.1. Classification results

The approximate parameter values as determined by Grid and Bayesian optimization for the four datasets are listed in Tables 3 and 4, respectively. We found that Bayesian optimization optimized the hyper-parameters in the given range of available values. The optimal parameters determined by each optimizer are different, depending on the datasets. Thus, the prediction results should be comprehensively compared.

Tables 5 and 6 report the models' performance on the Australian, German, Lending club, and Polish datasets. It can be observed that for most models, performance is improved after the parameters are tuned. The main performance measurement and accuracy clearly indicates that the performance of the proposed ensemble model, XGB-BO, is generally better than that of the other models for the Australian, German, and Lending club datasets. It is also notable that performance is improved with the use of tuned parameters compared with using the default parameters. XGB-BO achieved a higher AUC than the other models for the four datasets. This indicates the reliability of the predictive model.

Table 3. The optimal hyper-parameters found by grid optimization over the four datasets

Method	Parameters	Optimal Parameters by Grid search			
		German	Australian	Polish	Lending club
SVM	gamma	0.01	0.0001	1.0	0.01
	C	5	15	50	5
*K*NN	K	15	9	9	30
	metric	manhattan	manhattan	euclidean	manhattan
DT	min_split	20	2	2	5
	min_leaf	9	6	4	3
	max_depth	9	4	5	4
	max_features	7	11	20	6
NN	hidden_layer	n_feature	n_feature*2	n_feature*2	n_feature/2
	Batch size	16	16	16	32
	Solver	adam	sgd	adam	sgd
	initial_lr	0.01	0.1	0.01	0.1
Bagging-DT	n_estimators	11	10	3	11
	subsample	1.0	0.8	0.9	0.9
Bagging-NN	n_estimators	11	11	9	10
	subsample	0.9	1.0	1.0	0.8
XGB	max_depth	4	6	4	5
	subsample	0.9	0.9	0.8	0.8
	Learning rate	0.1	0.1	0.1	0.01
	max_delta_step	1.0	0	1.0	1.0
	colsample_bytree	1.0	0.8	1.0	0.8
	min_child_weight	1	1	3	2
	n_estimators	150	150	150	150
	gamma	0.01	0.1	0.1	0.01
RF	min_split	2	2	2	5
	min_leaf	3	1	1	1
	max_features	9	5	10	5
	max_depth	10	10	8	8
	n_estimators	300	100	500	300

Table 5, based on the Australian dataset, shows that XGB-BO clearly outperforms the other methods on four of the five measures, including Acc, AUC, Spec, and GM. Our model outperforms the best single model (DT-GO) on accuracy by 1.16%, and it performs slightly better than the best ensemble (RF) on accuracy by 0.29%. The specificity is lowest for XGB-BO, meaning NPLs will be reduced for the dataset. SVM-GO provides the best sensitivity, meaning the profit gain generated by the loan interest will be increased. However, misclassifying good and bad applicants generated different costs, because errors in predicting bad applicants generated costs much higher than those associated with errors in predicting good applicants. Thus, XGB-BO was selected as the best model on the Australian dataset.

For the German dataset, XGB-BO shows better accuracy than the best single model (SVM-BO) and ensemble model (RF-BO) by 1.3% and 0.6%, respectively. KNN-BO showed the best sensitivity, which means that profit gain is increased. However, the specificity of KNN is quite low. Thus, KNN generates a large amount of NPLs. To measure the tradeoff between sensitivity and specificity, the GM is the most suitable. XGB-BO has the best GM, and it was selected as the best model on the German dataset because it performs best on three out of five measures.

Table 4. The optimal hyper-parameters found by Bayesian optimization over the four datasets

Method	Parameters	Optimal Parameters by Bayesian search			
		German	Australian	Polish	Lending club
SVM	gamma	0.00461689047116	0.00397383750477	0.08616475650820	0.00603628274295
	C	19.1932310081776	48.2499587098706	19.0295480251796	30.0026881399767
*K*NN	K	20	9	29	11
	metric	manhattan	manhattan	manhattan	euclidean
DT	min_split	8	5	2	17
	min_leaf	12	4	10	1
	max_depth	25	5	6	6
	max_features	7	17	21	5
NN	hidden_layer	60	66	103	5
	Batch size	6	10	53	17
	Solver	sgd	adam	adam	sgd
	initial_lr	0.02002453466554	0.00579494278262	0.08529809489715	0.05638280882043
Bagging-DT	n_estimators	10	11	9	10
	subsample	0.70439266598359	0.84871802929565	0.98811060975709	0.83804104392796
Bagging-NN	n_estimators	11	6	10	9
	subsample	0.88854405411826	0.96995421405780	0.99823909581760	0.96771414905683
XGB	max_depth	15	11	8	3
	subsample	0.89872245824250	0.90531574515555	0.85178000551936	0.84192396383936
	Learning rate	0.09050467528551	0.07254944643543	0.08639999779493	0.03391092419106
	max_delta_step	0.97589145195803	0.58917007517083	0.99116577350673	0.89950933568580
	colsample_bytree	0.97995886290936	0.82826159131223	0.99999282917556	0.99961275942745
	min_child_weight	4.27490478655498	4.49766738399845	3.19933608516842	3.84589275209607
	n_estimators	139	170	97	100
	gamma	0.04365358568541	0.56438492874988	0.05697412055871	0.05785146742594
RF	min_split	10	2	8	21
	min_leaf	4	1	1	22
	max_features	12	5	14	5
	max_depth	12	25	12	8
	n_estimators	123	179	160	60

Table 6, based on the Lending club dataset, shows that XGB-BO achieves the best accuracy, AUC, sensitivity, and GM by 67.86%, 72.48%, 73.41%, and 67.49%, respectively. NN-BO achieves the highest specificity of 66.74%, which is 4.42% higher than that of XGB-BO. However, NN-BO achieves a lower sensitivity than XGB-BO by 6.18%. In this case, most measurements indicate that XGB-BO performs better than other models.

For the Polish dataset, XGB-BO improves the accuracy of the best single model (DT-BO) by 0.27%. Although the default parameter value for XGB also shows the highest accuracy of 98.11%, it scores slightly lower than XGB-BO on AUC and sensitivity. The Polish dataset is quite imbalanced and consists of defaulters only 3.86% which XGB-BO still provide remarkably high performance. The KNN-BO and SVM models failed to predict defaulters. Thus, the models generate NPLs more than the other models.

Ensemble classifiers have enjoyed increasing popularity for credit scoring models. Malekipirbazari and Aksakalli [20] recommended RF as an effective algorithm for building a credit scoring model. RF has been used as a benchmark ensemble algorithm in many studies [20]-[22]. Compared with the RF model, our proposed XGB-BO achieves better performance for overall accuracy, AUC, sensitivity, as well as GM because the XGB-BO is constructed of a powerful classifier with careful hyper-parameter tuning.

In sum, the Bayesian and grid hyper-parameter optimization improved the performances of almost all base models for most measurements. Bayesian hyper-parameter optimization is more suitable than grid search in the case of a substantial number of hyper-parameters, because all possible combinations of parameters lead grid optimization to require a long computation period. Moreover, the heuristic nature of the Bayesian optimization mechanism produces the optimal parameters. Thus, the experimental results indicate that Bayesian hyper-parameter optimization provided better performance than grid optimization on the algorithms that have a substantial number of hyper-parameters, such as XGB.

Table 5. The performance comparison of various classifiers over Australian and German datasets

Methods	Australian					German				
	Acc	AUC	Sen	Spec	GM	Acc	AUC	Sen	Spec	GM
DT	79.42	79.37	79.89	78.85	79.18	68.20	63.38	75.43	51.33	62.04
DT-GO	87.54	92.07	88.76	86.01	87.28	76.00	76.56	89.57	44.33	62.34
DT-BO	86.67	91.48	86.67	86.65	86.47	75.70	76.44	88.14	46.67	63.81
LR	85.51	92.09	84.85	86.31	85.46	76.80	79.27	89.00	48.33	65.39
LR-GO	85.51	92.04	84.85	86.31	85.46	76.90	79.24	89.00	48.67	65.59
LR-BO	85.51	91.88	84.85	86.31	85.48	76.90	79.23	89.00	48.67	65.59
KNN	78.84	78.21	84.08	72.34	77.83	71.60	61.05	87.43	34.67	54.81
KNN-GO	83.04	82.23	89.82	74.65	81.64	73.30	59.60	93.86	25.33	47.80
KNN-BO	83.04	82.23	89.82	74.65	81.64	73.10	57.36	**96.71**	18.00	41.26
SVM	84.78	91.79	86.30	83.51	84.77	76.30	77.80	92.00	39.67	59.73
SVM-GO	86.38	92.39	**91.85**	81.95	86.67	78.10	79.54	90.14	50.00	66.89
SVM-BO	85.80	90.80	86.42	85.01	85.64	78.20	79.57	90.00	50.67	67.14
NN	84.78	90.43	86.41	82.75	84.40	75.10	76.00	85.29	51.33	65.73
NN-GO	86.23	91.60	88.24	83.73	85.82	77.50	79.49	88.86	51.00	66.58
NN-BO	86.52	89.92	87.46	85.35	86.29	77.50	79.80	88.14	52.67	67.80
RF	86.81	92.49	88.52	84.71	86.48	76.80	79.84	92.14	41.00	61.14
RF-GO	88.26	93.24	91.38	84.39	87.70	78.30	80.35	92.57	45.00	64.30
RF-BO	88.41	92.67	90.34	86.00	88.08	78.90	79.91	92.43	47.33	65.91
Bagging-DT	85.07	90.98	85.88	84.06	84.89	75.00	76.25	88.71	43.00	61.34
Bagging-DT-GO	85.94	91.13	88.24	83.08	85.54	75.30	77.29	86.14	50.00	65.37
Bagging-DT-BO	86.96	91.02	86.94	86.98	86.86	76.40	77.30	89.29	46.33	63.68
Bagging-NN	85.65	91.36	86.93	84.04	85.32	75.10	77.25	85.57	50.67	65.42
Bagging-NN-GO	86.38	91.86	86.67	85.99	86.19	77.80	79.87	89.00	51.67	67.68
Bagging-NN-BO	85.07	90.93	85.11	85.01	84.93	77.00	79.88	88.57	50.00	66.15
XGB	85.94	92.48	87.76	83.73	85.64	75.40	77.07	86.86	48.67	64.89
XGB-GO	87.39	93.03	88.26	86.32	87.23	78.89	80.16	89.14	**55.00**	69.82
XGB-BO	**88.70**	**93.25**	89.29	**87.96**	**88.59**	**79.50**	**80.50**	90.14	54.67	**69.95**

Table 6. The performance comparison of various classifiers over lending club and Polish dataset

Methods	Lending club					Polish				
	Acc	AUC	Sen	Spec	GM	Acc	AUC	Sen	Spec	GM
DT	57.97	57.98	58.00	57.95	57.84	95.62	76.12	97.25	54.99	72.29
DT-GO	64.98	68.27	71.96	58.02	64.33	97.54	84.59	99.66	44.63	65.55
DT-BO	65.28	67.76	70.52	60.06	64.97	97.84	87.02	99.82	48.33	68.70
LR	65.93	71.49	66.74	65.12	65.84	95.99	67.59	99.79	1.10	5.74
LR-GO	65.97	71.50	66.74	65.20	65.88	95.97	71.07	99.70	2.95	14.22
LR-BO	65.97	71.49	66.74	65.20	65.88	95.99	67.54	99.79	1.10	5.74
KNN	60.94	60.94	60.45	61.42	60.81	96.04	50.84	99.82	1.85	7.36
KNN-GO	65.71	65.71	71.74	59.68	65.30	96.14	50.18	99.99	0.37	1.92
KNN-BO	62.26	62.26	62.27	62.25	62.13	96.14	50.00	**100.00**	0.00	0.00
SVM	65.59	70.87	69.32	61.88	65.40	96.14	75.91	**100.00**	0.00	0.00
SVM-GO	67.29	71.85	71.59	63.01	67.06	96.68	79.12	99.04	38.03	60.62
SVM-BO	66.95	71.78	71.21	62.71	66.71	96.17	79.55	99.99	1.11	5.77
NN	64.76	70.34	65.68	63.84	64.67	97.15	87.85	99.50	38.78	61.10
NN-GO	66.58	71.79	68.75	64.39	66.40	96.85	84.38	99.41	33.24	56.08
NN-BO	66.99	71.70	67.23	**66.74**	66.85	97.04	84.24	99.60	33.33	50.73
RF	66.84	71.40	70.53	63.16	66.66	97.64	89.66	99.84	42.83	64.08
RF-GO	67.48	71.80	70.91	64.06	67.32	97.67	91.31	99.96	40.61	61.93
RF-BO	67.26	71.80	71.74	62.78	67.03	97.94	91.85	99.94	47.96	68.22
Bagging-DT	61.66	66.44	68.64	54.69	61.14	97.71	85.29	99.70	47.98	67.79
Bagging-DT-GO	63.48	67.21	63.26	63.69	63.40	97.69	88.13	99.87	43.52	64.88
Bagging-DT-BO	64.95	68.26	72.27	57.64	64.45	97.31	89.72	99.94	31.73	54.79
Bagging-NN	64.76	70.59	65.23	64.29	64.69	96.94	87.64	99.56	31.77	54.46
Bagging-NN-GO	66.80	71.70	69.39	64.22	66.67	96.91	88.87	99.66	28.45	51.45
Bagging-NN-BO	66.27	71.53	68.33	64.21	66.06	96.68	88.35	99.63	23.27	44.42
XGB	64.83	70.15	64.92	64.75	64.65	**98.11**	95.26	99.72	**57.94**	**75.53**
XGB-GO	67.07	72.03	71.74	62.40	66.78	98.09	94.64	99.94	52.04	71.60
XGB-BO	**67.86**	**72.48**	**73.41**	62.32	**67.49**	**98.11**	**95.32**	99.78	56.48	74.61

3.2. Statistical significance test

This section details the performance analysis using the non-parametric Wilcoxon statistical significance test suggested by Sun *et al.* [23]. DT, LR, and NN were selected because they are widely used as benchmark models for credit scoring tasks, while XGB is the base model for XGB-BO. The null hypothesis for the comprehensive comparison between model pairs is H0: there is no difference between the model pairs in the results. The H0 will be rejected at a significance level of 5% when the p-value is less than 0.05. This means the model pairs show a significant difference in their results. Otherwise, it is assumed that there is no

significant difference in the performance of the pair of models. The significant results of the nonparametric Wilcoxon test in terms of p-value are presented in Table 7. The "*" marks significance at 5%.

Based on the Lending club dataset, XGB-BO significantly outperforms DT, LR, NN, and XGB on Acc, AUC, and GM, while there was no significant difference between LR, NN, and XGB. For the Polish dataset, XGB-BO did not significantly improve the performance of XGB because the dataset had a high imbalance ratio. For the Australian dataset, XGB-BO significantly outperformed other models on Acc and GM, whereas there was no significant difference between XGB-BO and LR on AUC. For German credit, XGB-BO significantly outperformed other methods on GM, and there was no significant difference between XGB-BO and LR on Acc and AUC. In summary, XGB-BO significantly improved the performance of most models.

Table 7. The significance results of nonparametric Wilcoxon test for Acc, AUC and GM over the four datasets

Datasets	Methods	Acc				AUC				GM			
		LR	NN	XGB	XGB-GO	LR	NN	XGB	XGB-GO	LR	NN	XGB	XGB-GO
Australian	DT	.0125*	.0217*	.0142*	.0050*	.0051*	.0050*	.0051*	0.0051*	.0125*	.0367*	.0125*	.0051*
	LR	-	.3070	.8334	.0205*	-	.0077*	.6744	0.3863	-	.3270	.6465	.0284*
	NN	-	-	.5139	.0107*	-	-	.0926	0.0357*	-	-	.4413	.0166*
	XGB	-	-	-	.0123*	-	-	-	0.0218*	-	-	-	.0125*
German	DT	.0107*	.0141*	.0068*	.0049*	.0068*	.0051*	.0051*	0.0051*	.2411	.1688	.1141	.0125*
	LR	-	.5741	.2324	.0966	-	.0593	.0593	0.4446	-	.2845	.7989	.0169*
	NN	-	-	.9526	.0167*	-	-	.5073	0.0093*	-	-	.7213	.0205*
	XGB	-	-	-	.0169*	-	-	-	0.0050*	-	-	-	.0357*
Lending club	DT	.0051*	.0144*	.0051*	.0051*	.0051*	.0051*	.0051*	0.0051*	.0051*	.0051*	.0367*	.0050*
	LR	-	.2324	.1829	.0050*	-	.1141	.4413	0.0051*	-	.8785	.0926	.0049*
	NN	-	-	.4828	.0050*	-	-	.5751	0.0051*	-	-	.8785	.0051*
	XGB	-	-	-	.0169*	-	-	-	0.0051*	-	-	-	.0205*
Polish	DT	.0050*	.0051*	.0050*	.0051*	.0051*	.0051*	.0051*	0.0051*	.0051*	.0051*	.0217*	.0051*
	LR	-	.1829	.0202*	.0215*	-	.0284*	.0050*	0.0050*	-	.0284*	.0050*	.0050*
	NN	-	-	.8785	.0966	-	-	.9594	0.0125*	-	-	.8785	.0051*
	XGB	-	-	-	.0581	-	-	-	0.0926	-	-	-	.0506

3.3. Models comparison

From a cost perspective, a misclassification rate of 1% can result in a considerable loss to financial institutions [24]. Thus, the goal of this study is to improve the predictive capability of the model. The approximate performance comparisons between existing models and the proposed model are presented in Tables 8 to 11. Based on the empirical results, the proposed XGB-BO outperforms existing models on accuracy over the four datasets. Also, XGB-BO shows the best results in the terms of sensitivity over the Australian, Lending club, and Polish datasets, and the best result for GM on the Australian dataset. The sensitivity represents the proportion of applicants who received a loan and had the creditworthiness to get it. This means that XGB-BO approves good applicants and increases the number of borrowers having low risk. XGB-BO also shows good AUC over the four datasets. This indicates that the estimates are reliable and suitable for use in credit risk assessment.

Table 8. Performance comparison with other recent credit scoring models based on Australian dataset

Techniques	Year	Acc	AUC	Sen	Spec	GM
Decorate+LR [7]	2017	86.81	92.39	-	-	-
XGBoost-TPE [12]	2017	87.92	-	-	-	-
EBCA+PSO [11]	2018	-	93.40	-	-	86.29
CART+BPSO [19]	2018	87.53	90.34	86.97	87.99	-
Bstacking [25]	2018	88.28	92.80	-	-	-
NNBag [14]	2018	87.45	94.00	88.00	86.00	-
backflow XGB [22]	2019	87.20	**95.78**	-	-	-
MHS-RF [26]	2020	86.95	93.56	86.35	87.46	-
Overfitting-Cautious [27]	2020	86.89	93.76	-	-	-
MLPs+LR [28]	2020	82.60	91.10	-	-	-
BP-ANN-GWO [29]	2020	86.09	93.73	80.52	**91.21**	86.48
mg-GBDT [30]	2021	87.45	94.34	87.05	-	-
CS-NNE [31]	2021	84.93	91.31	83.06	86.40	84.63
XGB-BO	2021	**88.70**	93.25	**89.29**	87.96	**88.59**

Table 9. Performance comparison with other recent credit scoring models based on German dataset

Techniques	Year	Acc	AUC	Sen	Spec	GM
CSVM-RBF [32]	2015	77.10	69.23	-	-	-
Decorate+LR [7]	2017	77.40	79.37	-	-	-
XGBoost-TPE [12]	2017	77.34	-	-	-	-
EBCA+PSO [11]	2018	-	80.02	-	-	62.03
CART+BPSO [19]	2018	78.00	73.92	91.71	46.00	-
CS-Bagging-CS-CART [33]	2018	-	-	89.13	41.63	-
Bstacking [25]	2018	78.66	79.48	-	-	-
NNBag [14]	2018	76.70	77.00	86.00	51.00	-
MOPSO-CS [34]	2019	75.45	-	83.03	57.76	-
MHS-RF [26]	2020	75.60	**80.53**	**92.29**	36.67	-
Overfitting-Cautious [27]	2020	77.72	80.34	-	-	-
BP-ANN-PSO [29]	2020	76.60	80.04	66.86	**79.47**	63.57
mg-GBDT [30]	2021	77.15	79.29	91.86	-	-
CS-NNE [31]	2021	74.40	80.11	75.43	72.00	**73.63**
XGB-BO	2021	**79.50**	80.50	90.14	54.67	69.95

Table 10. Performance comparison with other recent credit scoring models based on Lending club dataset

Techniques	Year	Acc	AUC	Sen	Spec	GM
Bstacking [25]	2018	66.75	72.46	-	-	-
mg-GBDT [30]	2021	67.75	**73.98**	63.75	-	-
XGB-BO	2021	**67.86**	72.48	**73.41**	62.32	67.49

Table 11. Performance comparison with other recent credit scoring models based on Polish dataset

Techniques	Year	Acc	AUC	Sen	Spec	GM
backflow XGB [22]	2019	97.46	91.98	-	-	-
Bag-C4.5 [35]	2019	-	92.30	99.00	5.02	-
ABoost(C4.5) [35]	2019	-	87.00	99.70	53.10	-
V-GANs [36]	2019	-	73.79	-	52.05	-
CS-NNE [31]	2021	91.30	88.62	92.01	**73.80**	**82.14**
XGB-BO	2021	**98.11**	**95.32**	**99.78**	56.48	74.61

Xia *et al.* [12] also modeled credit scoring by utilizing XGB with Bayesian hyper-parameter optimization. The difference between the work in [12] and XGB-BO is in the preprocessing step. XGB-BO does not adopt any feature selection methods due to the DT-based learners, like CART and RF, and XGB algorithms can provide an attribute importance score that can be used as a matric for measuring the importance of an attribute. In other words, non-informative features will automatically be excluded from the model.

4. CONCLUSION

Credit scoring has recently become a powerful tool for financial institutions to use to assess the creditworthiness of applicants. Thus, the predictive performance of a credit scoring model is crucial to maximizing the profitability of most commercial banks or financial institutions. This research proposed an alternative credit scoring method by using XGB with Bayesian hyper-parameter optimization. We believe the proposed method could help financial institutions to improve the performance of their credit scoring model. To confirm the efficiency of the proposed method, individual and ensemble methods were implemented as benchmark models. Four public and widely used credit scoring datasets were utilized in this study.

The experimental results illustrated the superiority of the proposed method over the benchmark individual and ensemble methods in terms of overall accuracy and AUC over the four datasets. We found that by carefully tuning the hyper-parameters, the performance of most models increased. Compared with existing methods, the proposed XGB-BO also showed the best overall accuracy and particularly good AUC. Thus, the proposed model is suitable for assessing the creditworthiness of applicants and can be used as a technique for a credit scoring model. Finally, although we studied the XGB, there are some aspects that require further investigation, including the cost-sensitivity of XGB, the stacking ensemble, and the impact of imbalanced ratios on the datasets.

ACKNOWLEDGEMENTS

This work was supported by the Department of Computer Science, Faculty of Science, Kasetsart University, Thailand.

REFERENCES

[1] A. Chopra and P. Bhilare, "Application of Ensemble Models in Credit Scoring Models," *Business Perspectives and Research,* vol. 6, no. 2, pp. 129-141, 2018, doi: 10.1177/2278533718765531.

[2] A. De Caigny, K. Coussement, and K. W. De Bock, "A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees," *European Journal of Operational Research,* vol. 269, no. 2, pp. 760-772, 2018, doi: 10.1016/j.ejor.2018.02.009.

[3] Y.-C. Chang, K.-H. Chang, H.-H. Chu, and L.-I. Tong, "Establishing decision tree-based short-term default credit risk assessment models," *Communications in Statistics - Theory and Methods,* vol. 45, no. 23, pp. 6803-6815, 2016, doi: 10.1080/03610926.2014.968730.

[4] D. L. Olson, D. Delen, and Y. Meng, "Comparative analysis of data mining methods for bankruptcy prediction," *Decision Support Systems,* vol. 52, no. 2, pp. 464-473, 2012, doi: 10.1016/j.dss.2011.10.007.

[5] F. N. Koutanaei, H. Sajedi, and M. Khanbabaei, "A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring," *Journal of Retailing and Consumer Services,* vol. 27, pp. 11-23, 2015, doi: 10.1016/j.jretconser.2015.07.003.

[6] H. Xiao, Z. Xiao, and Y. Wang, "Ensemble classification based on supervised clustering for credit scoring," *Applied Soft Computing,* vol. 43, pp. 73-86, 2016, doi: 10.1016/j.asoc.2016.02.022.

[7] J. Abellán and J. G. Castellano, "A comparative study on base classifiers in ensemble methods for credit scoring," *Expert Systems with Applications,* vol. 73, pp. 1-10, 2017, doi: 10.1016/j.eswa.2016.12.020.

[8] M. Zięba, S. K. Tomczak, and J. M. Tomczak, "Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction," *Expert Systems with Applications,* vol. 58, pp. 93-101, 2016, doi: 10.1016/j.eswa.2016.04.001.

[9] C.-F. Tsai, "Combining cluster analysis with classifier ensembles to predict financial distress," *Information Fusion,* vol. 16, pp. 46-58, 2014, doi: 10.1016/j.inffus.2011.12.001.

[10] X. Feng, Z. Xiao, B. Zhong, J. Qiu, and Y. Dong, "Dynamic ensemble classification for credit scoring using soft probability," *Applied Soft Computing,* vol. 65, pp. 139-151, 2018, doi: 10.1016/j.asoc.2018.01.021.

[11] H. He, W. Zhang, and S. Zhang, "A novel ensemble method for credit scoring: Adaption of different imbalance ratios," *Expert Systems with Applications,* vol. 98, pp. 105-117, 2018, doi: 10.1016/j.eswa.2018.01.012.

[12] Y. Xia, C. Liu, Y. Li, and N. Liu, "A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring," *Expert Systems with Applications,* vol. 78, pp. 225-241, 2017, doi: 10.1016/j.eswa.2017.02.017.

[13] L. Munkhdalai, T. Munkhdalai, O.-E. Namsrai, J. Lee, and K. Ryu, "An Empirical Comparison of Machine-Learning Methods on Bank Client Credit Assessments," *Sustainability,* vol. 11, no. 3, p. 699, 2019, doi: 10.3390/su11030699.

[14] A. Dželihodžić, D. Đonko, and J. Kevrić, "Improved Credit Scoring Model Based on Bagging Neural Network," *International Journal of Information Technology and Decision Making,* vol. 17, no. 6, pp. 1725-1741, 2018, doi: 10.1142/S0219622018500293.

[15] Y. Li, "Credit Risk Prediction Based on Machine Learning Methods," *2019 14th International Conference on Computer Science and Education (ICCSE)*, 2019, pp. 1011-1013, doi: 10.1109/ICCSE.2019.8845444.

[16] Y. Li and W. Chen, "A Comparative Performance Assessment of Ensemble Learning for Credit Scoring," *Mathematics,* vol. 8, no. 10, 2020, doi: 10.3390/math8101756.

[17] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for nas," *arXiv preprint arXiv:1912.06059,* 2019.

[18] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets," *Expert Systems with Applications,* vol. 39, no. 3, pp. 3446-3453, 2012, doi: 10.1016/j.eswa.2011.09.033.

[19] R. F. Malik and H. Hermawan, "Credit Scoring Using Classification and Regression Tree (CART) Algorithm and Binary Particle Swarm Optimization," *International Journal of Electrical and Computer Engineering,* vol. 8, no. 6, pp. 5425-5431, 2018, doi: 10.11591/ijece.v8i6.pp5425-5431.

[20] M. Malekipirbazari and V. Aksakalli, "Risk assessment in social lending via random forests," *Expert Systems with Applications,* vol. 42, no. 10, pp. 4621-4631, 2015, doi: 10.1016/j.eswa.2015.02.001.

[21] D. Tripathi, D. R. Edla, R. Cheruku, and V. Kuppili, "A novel hybrid credit scoring model based on ensemble feature selection and multilayer ensemble classification," *Computational Intelligence,* vol. 35, no. 2, pp. 371-394, 2019, doi: 10.1111/coin.12200.

[22] S. Wei, D. Yang, W. Zhang and S. Zhang, "A Novel Noise-Adapted Two-Layer Ensemble Model for Credit Scoring Based on Backflow Learning," in *IEEE Access*, vol. 7, pp. 99217-99230, 2019, doi: 10.1109/ACCESS.2019.2930332.

[23] J. Sun, J. Lang, H. Fujita, and H. Li, "Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates," *Information Sciences,* vol. 425, pp. 76-91, 2018, doi: 10.1016/j.ins.2017.10.017.

[24] M. Ala'raj and M. F. Abbod, "A new hybrid ensemble credit scoring model based on classifiers consensus system approach," *Expert Systems with Applications,* vol. 64, pp. 36-55, 2016, doi: 10.1016/j.eswa.2016.07.017.

[25] Y. Xia, C. Liu, B. Da, and F. Xie, "A novel heterogeneous ensemble credit scoring model based on bstacking approach," *Expert Systems with Applications,* vol. 93, pp. 182-199, 2018, doi: 10.1016/j.eswa.2017.10.022.

[26] R. Y. Goh, L. S. Lee, H.-V. Seow, and K. Gopal, "Hybrid Harmony Search–Artificial Intelligence Models in Credit Scoring," *Entropy,* vol. 22, no. 9, 2020, doi: 10.3390/e22090989.

[27] Y. Xia, J. Zhao, L. He, Y. Li, and M. Niu, "A novel tree-based dynamic heterogeneous ensemble method for credit scoring," *Expert Systems with Applications,* vol. 159, 2020, doi: 10.1016/j.eswa.2020.113615.

[28] L. Munkhdalai, J. Y. Lee, and K. H. Ryu, "A Hybrid Credit Scoring Model Using Neural Networks and Logistic Regression,", in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, Springer Singapore, 2020, pp. 251-258, doi: 10.1007/978-981-13-9714-1_27.
[29] R. Zhang and Z. Qiu, "Optimizing hyper-parameters of neural networks with swarm intelligence: A novel framework for credit scoring," *PLoS One,* vol. 15, no. 6, p. e0234254, 2020, doi: 10.1371/journal.pone.0234254.
[30] W. Liu, H. Fan, and M. Xia, "Step-wise multi-grained augmented gradient boosting decision trees for credit scoring," *Engineering Applications of Artificial Intelligence,* vol. 97, 2021, doi: 10.1016/j.engappai.2020.104036.
[31] W. Yotsawat, P. Wattuya, and A. Srivihok, "A Novel Method for Credit Scoring Based on Cost-Sensitive Neural Network Ensemble," *IEEE Access*, vol. 9, pp. 78521-78537, 2021, doi: 10.1109/ACCESS.2021.3083490.
[32] T. Harris, "Credit scoring using the clustered support vector machine," *Expert Systems with Applications*, vol. 42, no. 2, pp. 741-750, 2015, doi: 10.1016/j.eswa.2014.08.029.
[33] M. Saidi, M. E. H. Daho, N. Settouti, and M. E. A. Bechar, "Comparaison of Ensemble Cost Sensitive Algorithms: Application to Credit Scoring Prediction," in *ICAASE,* pp. 56-61, 2018.
[34] Y. Guo, J. He, L. Xu, and W. Liu, "A novel multi-objective particle swarm optimization for comprehensible credit scoring," *Soft Computing*, vol. 23, no. 18, pp. 9009-9023, 2019, doi: 10.1007/s00500-018-3509-y.
[35] V. García, A. I. Marqués, and J. S. Sánchez, "Exploring the synergetic effects of sample types on the performance of ensembles for credit risk and corporate bankruptcy prediction," *Information Fusion*, vol. 47, pp. 88-101, 2019, doi: 10.1016/j.inffus.2018.07.004.
[36] H. Mansourifar, L. Chen and W. Shi, "Virtual Big Data for GAN Based Data Augmentation," *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 1478-1487, doi: 10.1109/BigData47090.2019.9006268.

BIOGRAPHIES OF AUTHORS

Wirot Yotsawat was born in Songkhla, Thailand in 1984. He received the B.Sc. degree in Computer Science from the School of Informatics, Walailak University, Thailand in 2007, and the M.Sc. degree in Computer Science from the Faculty of Science, Kasetsart University, Thailand in 2014. He is currently pursuing the Ph.D. degree in Computer Science with the Faculty of Science, Kasetsart University. His research interests include data mining, machine learning and image processing.

Pakaket Wattuya received the B.Sc. degree in Computer Science from the Faculty of Science, Kasetsart University, Thailand in 2000, the M.Eng. degree in Computer Engineering from the Faculty of Engineering, Kasetsart University, Thailand in 2004, and the Dr. rer. nat. degree in computer science from Westfälische Wilhelms-Universität Muenster, Germany, in 2010. She is currently an Assistant Professor with the Department of Computer Science, Kasetsart University. Her research interests include image processing, computer vision, machine learning, and deep learning.

Anongnart Srivihok received the B.Sc. degree in Microbiology from the Faculty of Science, Chulalongkorn University, Thailand, in 1978, the M.S. degree in Engineering Sci-Computer Science from the University of Mississippi, USA, in 1984, and the Ph.D. degree in Information Systems from Central Queensland University, Australia, in 1998. She has been an Associate Professor with the Department of Computer Science, Kasetsart University. Her research interests include data mining, machine learning, decision support systems, and knowledge management.