ICIC Express Letters
Volume **16**, Number **8**, August **2022**

ICIC International ©2022 ISSN 1881-803X
pp. **819–826**

EXPLORING THE APPLICATION OF CLASSIFIERS TO TAX PREDICTION

Garebangwe Victoria Mabe-Madisa

Department of Decision Sciences
University of South Africa
P.O. Box 392, Unisa 0003, South Africa
mabemgv@unisa.ac.za

Received October 2021; accepted January 2022

Abstract. *In this study, the application of classifiers to tax prediction is investigated. Five supervised learning classifiers are considered: artificial neural network (ANN), Naïve Bayes classifier (NBC), decision trees (DT), logistic regression (LgR) and Rules. The performance of four ensemble classifiers: Vote, Stacking (stacked generalization), Adaboost (adaptive boosting) and Bagging (bootstrap aggregating) is also demonstrated and evaluated in terms of their ability to correctly predict or classify. Examining the performance of the base classifiers and ensemble classifiers showed Rules and Bagging to have the highest accuracies. The results further show the homogeneous ensemble classifiers to significantly outperform the heterogeneous ensemble classifiers. Vote and Stacking are substantially inferior while Bagging and Adaboost represent a superior approach to classifying data. The statistical significance of the results is confirmed by the Analysis of Variance (ANOVA) test.*
Keywords: Ensemble classifiers, Experiments, Classification, Performance measures

1. **Introduction.** The idea of building a predictive model that integrates multiple models has been investigated by several authors [1-3]. [1] developed a robust classification procedure based on ensembles of classifiers, with each classifier constructed from a different set of predictors determined by a random partition of the entire set of predictors. The proposed methods combined the results of multiple classifiers to achieve a substantially improved prediction compared to the optimal single classifier. [2] developed an integrated intrusion detection system by combining SVM with Adaboost. A few years later [3] expanded tree-based classifiers using a meta-algorithm called "LogitBoost" in the mining process. [4] suggested an ensemble of configured neural networks to improve the predictive performance of a single neural network.

Some researchers [5-7] start with a set of selected classifiers as constituent classifiers. Four different classifiers, linear discrimination, logistic regression and two different neural networks, were considered for combination [6]. Again, [7] suggested an ensemble methodology, which builds a classification model by integrating multiple classifiers, to improve prediction performance.

[8] used an ensemble of support vector machines (SVMs) to build a binary radiation induced lung injury (radiation pneumonitis (RP) risk model) from clinical and dosimetric parameters. Patient treatment data were partitioned into balanced subsets to prevent model bias. Forward feature selection, maximizing the area under the curve (AUC) for a cross-validated ROC curve, was performed on each subset. Model parameter selection and construction occurred concurrently via alternating SVM and gradient descent steps to minimize estimated generalization error. They showed that an ensemble classifier with a mean fusion function, five component SVMs, and a limit of five features per classifier

DOI: 10.24507/icicel.16.08.819

exhibited a mean AUC of 0.818. This was an improvement over previous SVM models of RP risk.

A computer-based method for differentiating normal and pathological larynges on the basis of high-speed video-endoscopy (HSV) was applied. HSV recordings were collected from 101 patients with normal larynges, leukoplakia, nodules or polyps. After preprocessing, samples were assessed for the number of glottal regions present during the open phase, the symmetry of the glottal area, the convex nature of the vocal folds and the ratio of the minimal to maximal glottal area. A decision-tree based method with SVMs at the tree nodes was used to separate samples. Normal samples were differentiated from pathological samples with a sensitivity of 91.1% and a specificity of 81.8%. When samples were divided into normal, nodule, polyp and leukoplakia groups, samples were correctly separated 70.3% of the time. The combination of SVM and decision tree improved the differentiating capabilities of the parameters employed [9].

[10] also conducted a comparative study of various ensemble methods with perspective taxonomy. These methods included Bagging, Boosting, Random Trees, Random Forest, Random Subspace Stacking, and Voting. They compared these ensemble methods to a single classifier Naïve Bayes. A series of benchmarking experiments on public domain datasets, showed that applying classifier ensemble methods to predicting software defect could achieve a better performance than using a single classifier. A program which ran experiments on public-domain data, and which could be used by any researcher was developed by [11]. In all seven ensemble methods, Voting and Random Forest had obvious performance superiority over other methods, and Stacking had a better generalization ability.

A classifier ensemble is generated by training multiple learners for the same task and then combining their predictions. There are different ways in which ensembles can be generated with the resulting output from each classifier and then combined to classify new instances. The popular approaches to creating ensembles include changing the instances used for training through techniques such as Bagging [12], Boosting [13], Stacking [14], changing the features used in training [15], and introducing randomness in the classifier itself [16]. The method used in this study is changing the instances used for training.

The contribution of the study is to show the accuracy of five classifiers for income tax prediction and show how the use of ensemble classifiers improves pattern prediction and further show that homogeneous ensemble classifiers are better predictive models than heterogeneous ones. The base classifier systems that were evaluated are artificial neural network (ANN), Naïve Bayes classifier (NBC), decision trees (DT), logistic regression (LgR) and Rules and the ensemble classifier systems that were evaluated are Vote, Stacking (stacked generalization), Adaboost (adaptive boosting) and Bagging (bootstrap aggregating). The paper is organized as follows. Section 2 discusses the research methodology. Experiments and results are presented in Sections 3 and 4 respectively, while Section 5 discusses the conclusions.

2. **Research Methods.**

2.1. **Same algorithm, different training examples.**

Statistics has primarily focused on constructing multiple models by manipulating the training data [17]. The same algorithm is run several times, and each time with a different set of training examples. Methods like Bagging [12], and Boosting [13] construct multiple classifiers by applying a single learning algorithm to different samples of a single dataset. Manipulating this dataset, two methods are employed: random sampling with replacement, or bootstrap sampling in Bagging and weighting of the misclassified training samples in Boosting. Techniques for combining the predictions obtained from single classifiers occur in different ways. These combinations can be done in several ways. One can use expectation, or product.

Bagging, is a collection of similar homogeneous classifiers built on re-sampled training data and held together by a combination method. Adaboost, a collection of similar homogeneous classifiers as well, on the other hand, takes a different approach to building the ensemble. It constructs a layered classifier. Each round of Adaboost chooses a new classifier from a set of potential classifiers constructed from training data weighted, or re-sampled, according to the mis-classifications of the previous round. The new classifier is selected so as to minimize the total ensemble error. In the early stages of ensemble construction, Adaboost has few weak classifiers and each is focused on different areas of the training space; the effect of this is to primarily reduce bias. As the ensemble size grows, the scope for bias reduction diminishes and error from variance is improved.

2.2. **Different algorithms, same training example.** The other approach used, is similar to the method of [18], which uses a single dataset to generate classifiers by applying different learning algorithms with heterogeneous model representations. Stacking combines multiple classifiers to induce a higher-level classifier with improved performance. A learning algorithm is used to determine how the outputs of the classifiers should be combined. The original dataset constitutes the level one data and all the base classifiers run at this level. The level two data are the outputs of the base classifiers and another learning process occurs using as input the level two data and as output the final prediction as depicted in Figure 1.

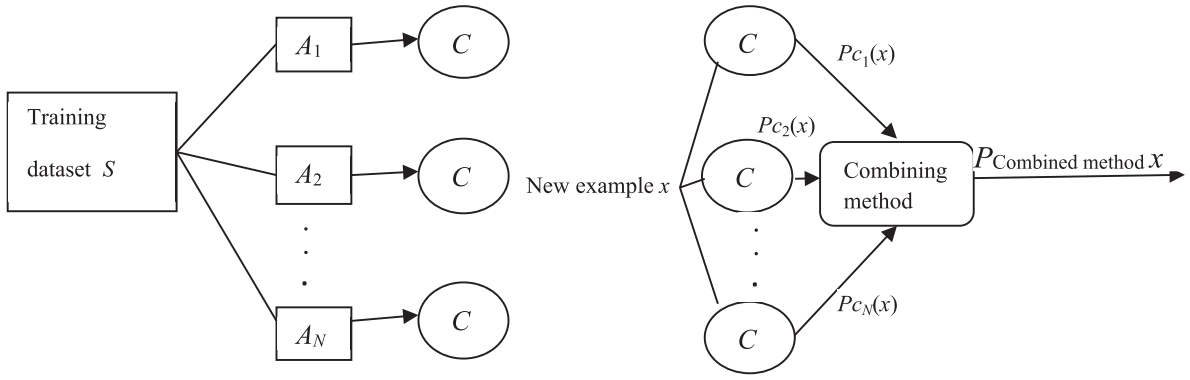


FIGURE 1. Parallel architecture

In the Voting framework for combining classifiers, the predictions of the learning algorithms are combined according to a static Voting scheme, which does not change with training dataset. The Voting scheme remains the same for all the different training sets and sets of learning algorithms (or base-level classifiers). The simplest Voting scheme is the plurality vote. According to this Voting scheme, each learning algorithm casts a vote for its prediction. The example is classified in the class that collects the most votes.

In Figure 1, a set $C = \{C_1, C_2, \ldots, C_N\}$ of learning algorithms or classifiers is generated by applying the learning algorithms $A_1, A_2, \ldots, A_N$ to a single training dataset S. It is assumed that each of the learning algorithms from C predicts a probability distribution over the possible class values. Thus, the prediction of the single classifier C when applied to example x is a probability distribution vector: $PC(x) = (PC(c_1|x), PC(c_2|x), \ldots, PC(c_k|x))$, where $\{c_1, c_2, \ldots, c_k\}$ is a set of possible class values and $PC(c_i|x)$ denotes the probability that example x belongs to class c_i as predicted by classifier C. The class c_j with the highest class probability $PC(c_j|x)$ is predicted by classifier C.

2.3. **Multiple classifier systems (MCSs).** MCSs can be classified [19] into one of three architectural types: Static Parallel (SP), Multi-Stage (MS) and Dynamic Classifier Selection (DCS). The outputs from each classifier are combined to deliver a final classification decision. A large number of combination functions are available. These include

voting methods (simple majority vote, weighted majority vote, the product or sum of model outputs also known as the product rule, the minimum rule, the maximum rule); rank based methods (borda count); probabilistic methods (Bayesian methods). This study used static parallel method. Figure 1 is probably the most popular architecture and it is where two or more classifiers are developed independently in parallel [20].

2.4. **Performance measures.** The following performance measures were considered.

2.4.1. *Accuracy and computation time.* To measure the performance of classifiers, the training set/test set methodology is employed. For each run, each dataset is split randomly into 80% training set and 20% testing or validating set. The performance of each classifier is then assessed by the misclassification rate (i.e., the percent of misclassified instances out of the total instances in the validation data) and the computation time.

2.4.2. *The receiver operating characteristic curve.* The ROC curve, in Figure 2, is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. It is created by plotting the fraction of true positives out of the total actual positives (TP = true positive rate) versus the fraction of false positives out of the total actual negatives (FP = false positive rate), at various threshold settings. TP is also known as sensitivity (recall), and FP is one minus the specificity or true negative rate (TN).

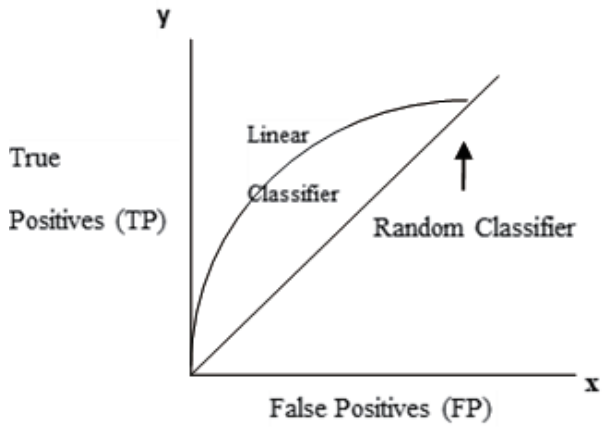


FIGURE 2. The ROC curve

A particular classifier is represented by a pair of TP and FP in the ROC space. A perfect classifier will have TP being 100% and FP being 0% [21]. Areas under the ROC curves are used to evaluate the performance of the classifier. Higher value means better performance.

2.4.3. *Metrics.* The metrices such as precision, recall, error rate, kappa statistics, root mean squared error and F-value were used to measure the performance. Error rate measures the number of incorrectly classified instances. Kappa statistic (K) is similar to the correlation coefficient. It measures the agreement or relation between the classifiers. Root mean squared error (RMSE) is the square root of the variance of the residuals and measures how far the data are from the model's predicted values. In statistical analysis of binary classification, the F-value or F-measure is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F-value can be interpreted as a weighted average of the precision and recall, where an F-value reaches its best value at 1 and worst at 0. The traditional F-value is the harmonic mean of precision and recall.

2.5. **Classifier ensemble.** A generalized classifier ensemble algorithm is summarized in the following steps [22].

1) Partition original dataset into n training datasets, TR_1, TR_2, ..., TR_n.

2) Construct n individual models $(M_1, M_2, \ldots, M_n)$ with the different training datasets TR_1, TR_2, ..., TR_n to obtain n individual classifiers (ensemble members).

3) Select m the de-correlated classifiers from n classifiers using de-correlation maximization algorithm [23].

4) Using step 3, obtain m classifier output values (misclassification error rates) of an unknown instance.

5) Transforming output value to reliability degrees of positive class and negative class, given the imbalance of some datasets.

6) Fuse the multiple classifiers into aggregate output.

3. **Experimental Set-up.** For simulation, five base classifiers which generate different models: linear models, density estimation, trees and networks were chosen. Each classifier uses a different form of parametric learning. The classifiers also come from two diverse research communities: statistics and machine learning. Single classifiers were constructed using each method.

To select the appropriate number of ensemble members, the de-correlation maximization method by [23] was utilized. The four classification techniques were applied to the test sample and the cross validation sample.

Classifiers were constructed using a cross-validation method as the sampling method. The ensemble classification methods were each applied to the training, validation and the test samples. Classifiers were constructed using the Waikato Environment for Knowledge Analysis (WEKA) free and open software that uses the Java^{TM} language [24] and a 2.60GHz CPU microcomputer. The WEKA freeware software is a collection of algorithms for data mining tasks. All algorithms were implemented in WEKA Release 3.6.9.

The data collected and used is the personal income tax data and the sample consists of 7 890 tax returns. All statistical tests were conducted using the statistical analysis software (SAS). Analyses of variance, using the general linear model procedure were used to examine the main effects and their respective interactions.

4. **Experimental Results.** The results are summarized in Figures 3-6 in terms of misclassification error, computation time and metrics. Rules achieved the highest accuracy rate, with an error rate of 5.2471%. It is followed by DT with an error rate of 5.7161%, and in the third place ANN with an error rate of 5.8935%. LgR and NBC have error rates of 6.5779% and 7.976%, respectively.

The Bagging achieved the highest accuracy with an error rate of 0.0507%, as shown in Figure 4. It is followed by Adaboost with an error rate of 0.0634%. Vote and Stacking both have an error rate of 5.6527%. Vote outperformed all ensembles with computation time of 0.03, as shown in Figure 5, and Bagging is the second best in computation time. The performances between most of the ensemble classifiers were found to differ significantly at the 5% level.

The statistical significance of the homogeneous ensemble classifiers is better than that of heterogeneous ensemble classifiers. This follows from the value of their "Kappa statistic", 0.9941 and 0.9953, as shown in Figure 6, which indicate the existence of moderate statistical dependence. Another metric is the "receiver operating characteristic (ROC) area". If its value is near 0.5, it indicates the lack of any statistical dependence [5]. The ROC areas of Vote, Stacking, Adaboost and Bagging are 0.5, 0.5, 1, and 1, respectively. The homogeneous ensemble classifiers displayed better performance. The heterogeneous ensemble classifiers both have the area under the curve equal to 0.5. They failed to distinguish between high risk and low risk but managed to identify the true positives.

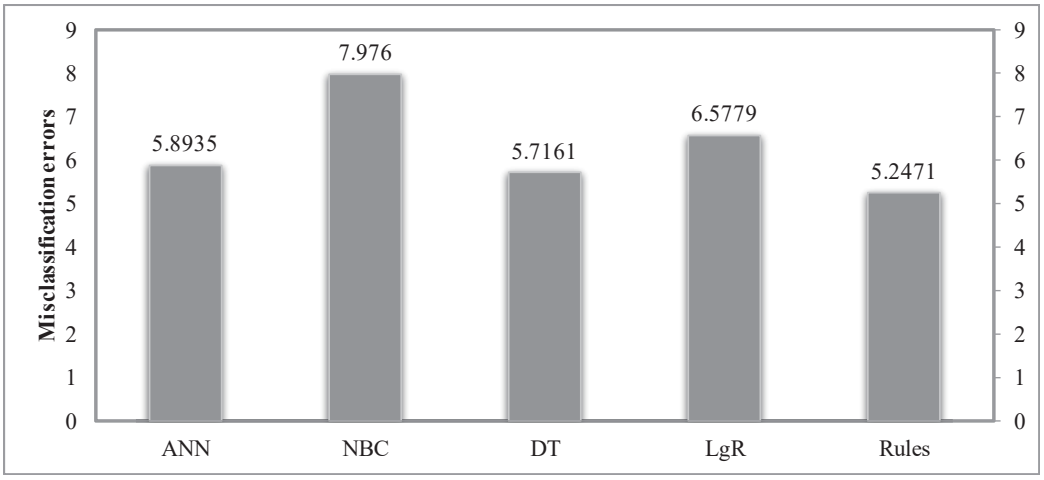


FIGURE 3. Misclassification errors of base classifiers

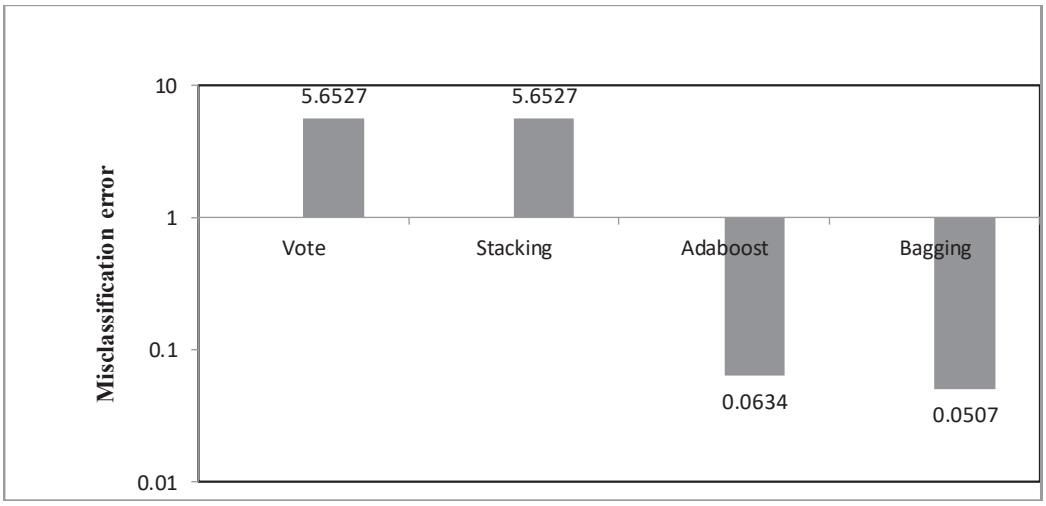


FIGURE 4. Misclassification error of ensemble classifiers

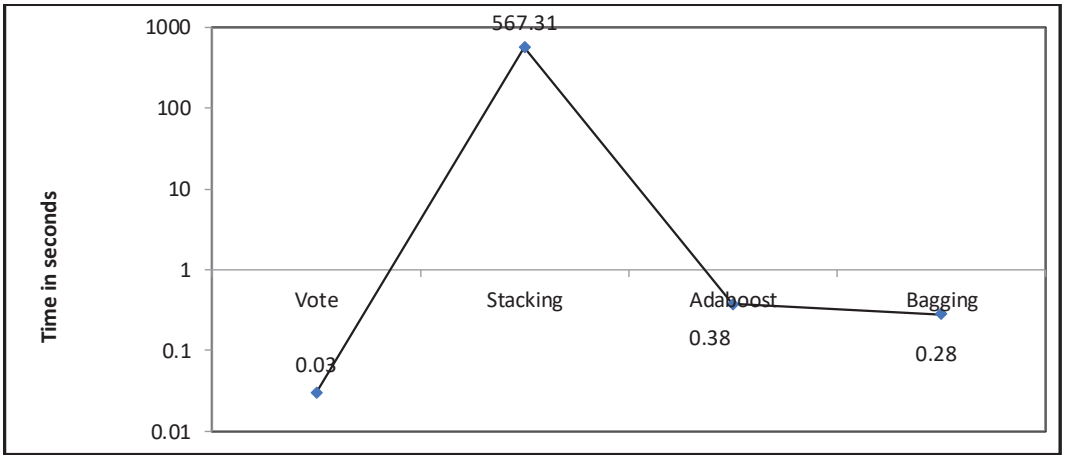
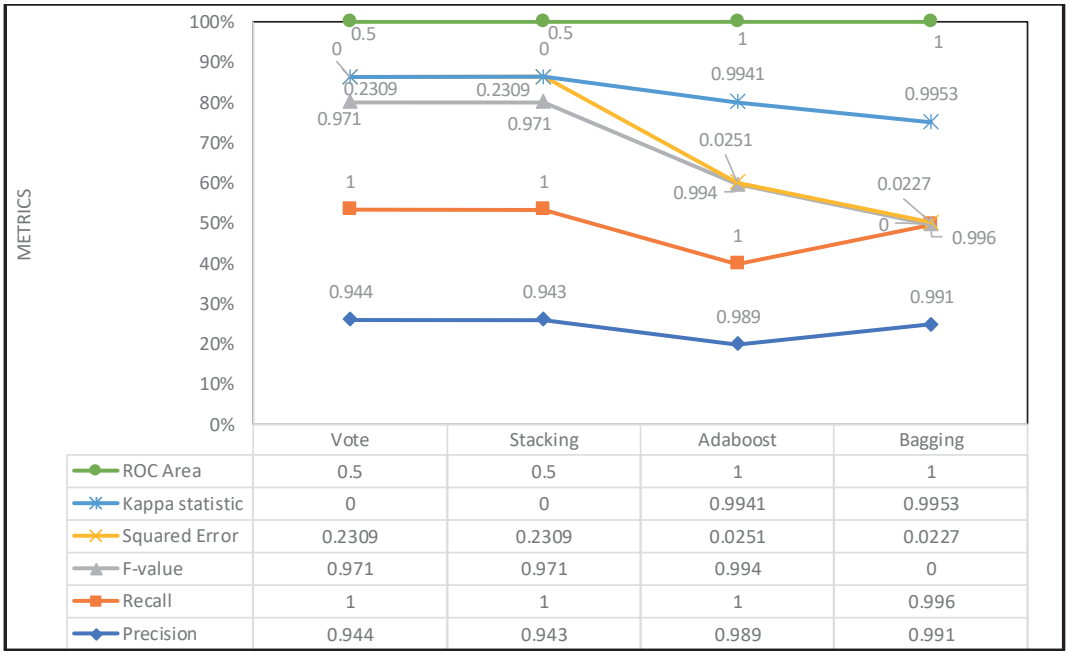


FIGURE 5. Computation time of ensemble classifiers

5. **Conclusions.** It has been found that the combination of multiple classifiers can enhance the classification and identification accuracy to a great extent. The ensemble classifiers outperformed the single classifiers in performance. The homogeneous classifiers, in return, outperformed the heterogeneous classifiers in performance as well.



	Vote	Stacking	Adaboost	Bagging
ROC Area	0.5	0.5	1	1
Kappa statistic	0	0	0.9941	0.9953
Squared Error	0.2309	0.2309	0.0251	0.0227
F-value	0.971	0.971	0.994	0
Recall	1	1	1	0.996
Precision	0.944	0.943	0.989	0.991

FIGURE 6. The performance in metrics of ensemble classifiers

Different designs of a classifier provide complementary information about the patterns to be classified, and these could be integrated to improve the performance. The unstable nature of the component classifiers (output predictions change due to small changes in training samples) makes it appropriate for an ensemble. An unstable predictor depends strongly on its training data and exhibits high variance. A stable predictor does not exhibit high dependency on the training data and has a low variance [12]. Unstable classifiers are therefore chosen over stable ones for more promising results. [4] maintains that an ensemble classifier is more accurate if and only if constituent classifiers are accurate and diverse, and this makes the accuracy and diversity of classifiers much more important. They define an accurate model as the one that has an error rate better than random guessing, and two models making different errors on new data points are regarded as diverse. The condition of being diverse is the foundation for the reason the combined models work better since the errors that are made by individual uncorrelated models can be removed by combining [25].

There are different approaches to selecting best models for combination into a better one. None of these approaches is a winner. No single model works for all types of datasets, also known as the "no free lunch" theorem [26]. Characteristics such as data size, normality, linearity and correlation do have an impact on the performance of a model [27].

The condition of accuracy is the requirement for every model. The condition of diversity is the reason why combined models perform better than individual ones as confirmed by [25]. These are all research-based opinions of different researchers, there is no single winner here, and everything depends upon the domain and what is going to be done.

Acknowledgment. This work was supported by the University of South Africa, College of Economic and Management Sciences research department.

REFERENCES

[1] H. Ahn, H. Moon, M. J. Fazzari, N. Lim, J. J. Chen and R. L. Kodell, Classification by ensembles from random partitions of high-dimensional data, *Computational Statistics & Data Analysis*, vol.51, no.12, pp.6166-6179, 2007.

[2] Y. Ren, An integrated intrusion detection system by combining SVM with Adaboost, *J. Softw. Eng. Appl.*, vol.7, no.12, pp.1031-1038, 2014.
[3] Y. Yamasari, S. M. S. Nugroho, K. Yoshimoto, H. Takahashi and M. H. Purnomo, Expanding tree-based classifiers using meta-algorithm approach: An application for identifying students' cognitive level, *International Journal of Innovative Computing, Information and Control*, vol.15, no.6, pp.2085-2107, 2019.
[4] L. Hansen and T. Salamon, Neural networks ensembles, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.12, no.10, pp.993-1001, 1990.
[5] Y. Kim, J. Kim and J. Jongwoo, Convex hull machine for regression and classification, *IEEE Conference on Data Mining*, pp.243-253, 2002.
[6] H. Zhu, P. A. Beling and G. A. Overstreet, A Bayesian framework for the combination of classifier outputs, *Journal of the Operation Research Society*, vol.53, pp.719-727, 2002.
[7] L. Rokach, Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography, *Computational Statistics & Data Analysis*, vol.53, no.12, pp.4046-4072, https://doi.org/10.1016/j.csda.2009.07.017, 2009.
[8] S. Chen, A. Wiliem, C. Sanderson and B. Lovell, *Matching Image Sets via Adaptive Multi Convex Hull*, University of Queensland, https://espace.library.uq.edu.au/data/UQ_330285/UQ330285_OA.pdf, 2014.
[9] Y. Gao, J. Jack, M. D. Jiang and A. Sprecher, *Support Vector Machine Based Decision Tree to Classify Voice Pathologies Using High-Speed Videoendoscopy*, https://www.semanti-cscholar.org, Accessed in July 2021.
[10] Z. Liu, H. Shi, T. Wang and W. Li, Software defect prediction based on classifier ensembles, *Journal of Information and Computational Science*, vol.8, no.16, pp.4241-4254, 2011.
[11] G. Boetticher, T. Menzies and T. Ostrand, *Promise Repository of Empirical Software Engineering Data*, Department of Computer Science, West Virginia University, http://promisedata.org/repository, 2007.
[12] L. Breiman, Bagging predictors, *Machine Learning,* vol.24, no.2, pp.123-140, 1996.
[13] Y. Yamasari, S. M. S. Nugroho, D. F. Suyatno and M. H. Purnomo, Meta-algorithm adaptive boosting to improve the classification performance of students' achievement, *J. Nas. Tek. Elektrodan Teknol. Inf.*, vol.6, no.3, 2017.
[14] A. Ghorbani and K. Owrangh, Stacked generalization in neural networks: Generalization on statistically neural problems, *International Joint Conference on Neural Networks*, 2001, https://ieeexplore.ieee.org/document/938420, Accessed in August 2021.
[15] T. K. Ho, Random decision forests, *Proc. of the 3rd Int'l Conf. on Document Analysis and Recognition*, pp.278-282, 1995.
[16] T. Dietterich, Ensemble methods in machine learning, *Proc. of the 1st International Workshop on Multiple Classifier Systems*, pp.1-5, 2000.
[17] L. Breiman, Arcing classifiers, *The Annals of Statistics*, vol.26, no.3, pp.49-64, 1998.
[18] T. T. Khuat and M. Le, Evaluation of sampling-based ensembles of classifiers on imbalanced data for software defect prediction problems, *SN Computer Science*, vol.1, no.108, https://doi.org/10.1007/s42979-020-0119-4, 2020.
[19] P. Molale, S. Seeletse and B. Twala, Fingerprint prediction using classifier ensembles, *The 53rd Annual Conference of SASA*, 2011, http://hdl.handle.net/10204/5379, Accessed in June 2021.
[20] H. Zhu, P. A. Beling and G. A. Overstreet, A study in the combination of two consumer credit scores, *Journal of the Operation Research Society*, vol.52, pp.2543-2559, 2001.
[21] M. Niranjan, R. W. Prager and M. J. J. Scott, Realizable classifiers: Improving operating performance on variable cost problems, *The 9th British Machine Vision Conference*, vol.1, pp.304-315, 1998.
[22] B. Twala, Multiple classifier application to credit risk assessment, *Expert Systems and Applications*, vol.37, pp.3326-3336, 2010.
[23] I. Jolliffe, *Principal Component Analysis*, Wiley StatsRef: Statistics Reference Online, 2014, https://doi.org/10.1002/9781118445112.stat06472, Accessed in July 2021.
[24] E. Frank, I. H. Witten and C. Pal, *Data Mining, Practical Machine Learning Tools and Techniques*, 4th Edition, Hamilton, Morgan Kaufmann, 2016.
[25] T. Dietterich, Machine learning research: Four current directions, *AI Magazine*, vol.18, no.4, pp.97-136, 1998.
[26] G. Macready, G. William and H. Wolpert, No free lunch theorems for optimization, *Transactions on Evolutionary Computation*, vol.1, no.1, pp.67-82, 1997.
[27] Y. Kiang, A comparative assessment of classification methods, *Decision Support Systems*, vol.35, pp.441-454, 2003.