Journal of Theoretical and Applied Information Technology
31st July 2022. Vol.100. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CLASSIFICATION PERFORMANCE FOR CREDIT SCORING WITH ENSEMBLE METHOD APPROACH

ABBA SUGANDA GIRSANG [a]*, CHRISTOPHER EDMOND[b]

[a] Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480,
[b*] Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480,
Email : agirsang@binus.edu, christopher.edmond@binus.ac.id

ABSTRACT

Credit scoring is an important part in controlling risk in financial companies. With the high number of non-performing loans, the assessment of potential new customers in financial companies has become a major focus of the financial industry. High accuracy credit scoring system can give better predictions on new customers and can change the company's economic growth and for better capital. This study uses a real world dataset, where data is obtained directly from a financial company and will be used to feed a random forest model to differentiate between good and bad potential new customers. This contribution of this research is to improve single model with real dataset by using ensembled bagging, bootstrap aggregating. Two methods are implemented, random forest and neural network to see the performance of ensembled using bootstrap aggerating. The output accuracy from the final model of the ensemble methods that resulted from the voting will be compared with the original unmodified single model and another model with similar architecture.The result shows that the modified multiple model surpasses the unmodified single model in terms of accuracy with a tradeoff on duration in the process.

Keywords: *Bootstrap Aggregation, Credit Scoring, Ensemble Methods, Random Forest*

1. INTRODUCTION

In a financial perspective of a financial company, granting a credit to a new customer is mission critical. Evaluating and selecting the right customers is undoubtedly an important topic especially in financial companies, such as commercial banks and certain retailers, the ability to distinguish good customers from bad is very important [1] Because of this importance, In the last ten years, many researcher actively doing an analysis related to customer credit analysis using credit scoring. Many statistical model involved and had been developed to support credit risk using different formula and algorithms, such as Linear Regression, Logistic Regression, K nearest neighbors and Decision Trees. Leo Breiman introduced the random forests, an extension of algorithms creating an ensemble learning of multiple decision tree with bootstrap aggregating to improve accuracy and correct the overfitting habit on decision tree [2].

Random forest is widely used in many machine learning cases, especially in the field of financial sector, a credit scoring system because of its predictive regression and classification capabilities [3] . But in the recent years, there is a new model of mathematics and statistics called the ensemble methods has been introduced.[4]. Therefore the statement problem of this research is how to combine multiple classifier that solve a similar problem to get a more accurate model through voting or averaging, instead of using only a single classifier to improve overall performance results [5]. Is the ensemble method able to effectively reduce misclassification, also is able to obtain lower error, reduce overfitting and variance and is believed to have a better performance compared to a single classifier? The most commonly used ensemble methods are bagging, boosting, and stacking [6] , but from those most commonly used ensemble method, bagging simply known as bootstrap aggregating is the most high performance ensemble methods that works really well in a classification for credit scoring problem [7] [8].

The motivation of this research is to improve the single model likes random forest and neural network by ensembling using bagging in a credit scoring problem The performance of this research by evaluate and comparing to single classifier model.

2. STUDY LETERATURE

2.1 Random Forest

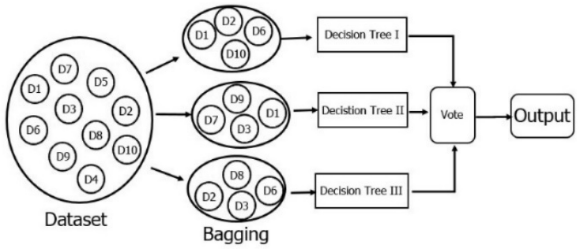

Figure 1. Random Forest Architecture

Random forest is a type of ensemble machine learning method called bootstrap aggregation or bagging. In Figure 1, the ensemble method combines several outputs created by various predictors to get better results [2]. Formally, the random forest (strong learner) was built as an ensemble of the decision tree (weak learner). The purpose of using the ensemble method is to find the average of individual prediction results by diversifying the set of predictors so as to reduce variance and form a reliable prediction model that can reduce overfitting. Bootstrap aggregation consists of taking a random sample subset of training data, fitting the model to that subset of data, and combining predictions. This method allows several samples to be used repeatedly in the training phase (random sampling with replacement). The bagging tree consists of a subset of the training dataset sample, fitting each decision tree, and aggregating the results. Random forest process is done through merging trees where the more trees the more it can affect the accuracy for a better prediction. The process starts from splitting the existing sample data into a random Decision Tree. The process of choosing the best tree is taken based on the voting results from the tree formed. Development of each tree is carried out by applying the random feature selection method to minimize errors. The advantages of using Random forest are being able to classify data that has incomplete attributes, can be used for classification and regression but not too good for regression, it is more suitable for data classification and can be used to handle large sample data [9]. In Random forest, there is a function called impurity, it is used to measure the quality of a split. There are several impurity methods, depending on the case, classification or regression. In classification case, the most common method used is the Gini impurity as shown in Equation 1, while Entropy as shown in Equation 2 is the alternative.

$$gini(N) = \frac{1}{2}\left(1 - \sum_{j} \rho(\omega_j)\right) \quad (1)$$

$$\text{Entropy} \sum_{i=1}^{c} - f_i \log \log (f_i) \quad (2)$$

2.2 Ensemble Method: Bootstrap Aggregation

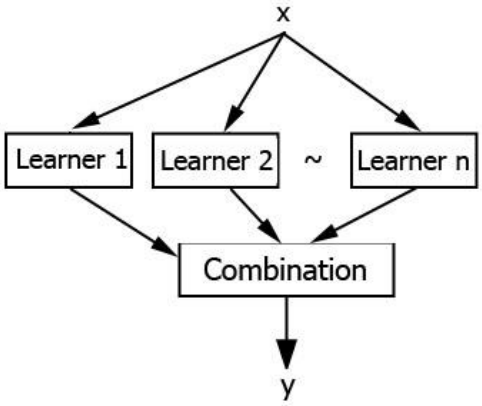

Figure 2. Common Ensemble Architecture

Ensemble is a method of aggregating a group of classifiers. It is believed that ensemble methods can increase overall results from a combination of learning models [10]. The ensemble method can effectively reduce misclassification, and is believed to have good performance compared to using a single classifier. The main idea of the ensemble method as shown in Figure 2 is to combine several sets of models that solve a similar problem to get a more accurate model through voting or averaging. An ensemble method is really useful since it can lower error, reduce overfitting and variance [6]. The most commonly used ensemble methods are bagging, boosting, and stacking.

The most commonly used ensemble methods are bagging, boosting, and stacking. While Bootstrap Aggregating or simply known as bagging follows the concept of majority of voting, where a subset of different training data is used randomly in training different learners or models in the same way. As shown in Figure 3, modelling the bagging method is done in a number of iterations. Each iteration, the model formed predicts each subset of data. At each bagging iteration method, the model formed has the same vote weight. The bagging method will choose the classification model with the most votes. The classification model produced by the bagging

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

method has better accuracy and is quite significant compared to the single (base) classification model.

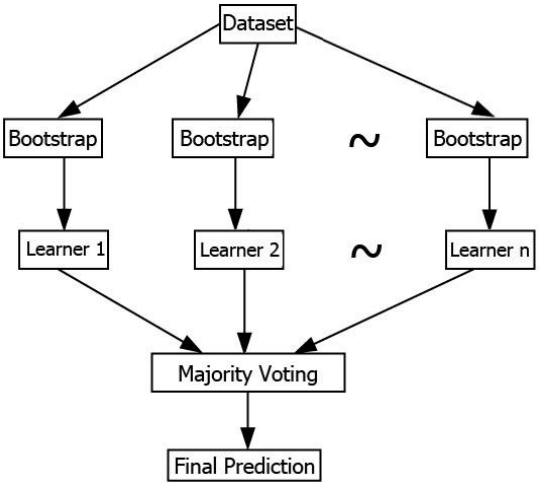

Figure 3. Bootstrap Aggregation Architecture

The increase in accuracy occurs because the combination of models can reduce the variance of a single grouping. As shown on Equation 3, a bagging has B which separates the bootstrapped samples from the training set and using those separated samples to feed the model [11].

$$\tilde{f}_{avg}(x) = \frac{1}{B}\sum_{b-1}^{B} \hat{f}^{b}(x) \quad (3)$$

3. PROPOSED METHOD

In this section, every stage that needs to be carried out to make the model will be discussed as described before. As shown in Figure 4, there will be 6 stages to be carried out. In this study, a real-world dataset directly from a financing company is used.

The first stage, "Variable Identification" is to identify each variable that will be used. The dataset contains 395.821 total cases with 236.686 of good cases and 159.135 of bad cases, with a total of 25 variables including:

- Customer's personal information (marital, house, education, dependency, etc.)
- Customer's item and credit information (type, brand, year, price, down payment, etc.)

In the second stage, "Data Pre-processing" is to prepare and process the data, so it can be ingested by the model in the next stage. Process in this stage is divided into two, cleaning and transforming. The data cleaning process is done by doing some analysis to check if there is data duplication, null values variable, and data with outlier values, then if there is any, there will be a pre-process to remove the duplication data, filling null values with modus, median, or mean based on the value of the specific attribute, and replacing some outliers or extreme value with the correct one.

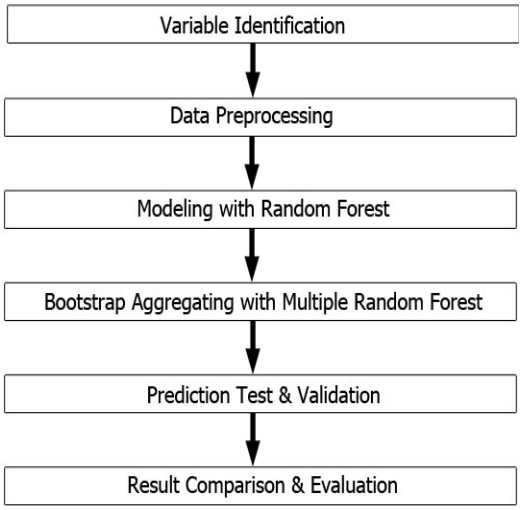

Figure 4. Research Method Stage

And the data transformation process is done by transforming the data according to the form of the data, whether it is qualitative or quantitative, where qualitative data is data in the form of words/sentences such as gender and marital status, while quantitative data is data in the form of numbers such as income and dependency. The process will transform these qualitative and quantitative into categorical data and ranged numbers. The last one is to transform the credit status attribute based on the payment overdue from each customer during his contract, if overdue between 0-90 days means 'good', while overdue >=90 days means 'bad'. This labelling purpose is for the 'supervised learning' process in the training stage.

In the third stage, random forest model will be built using python. In random forest architecture, we need to specify how many trees in the forest and the max depth of each tree. A larger forest is not guaranteed to give the best prediction output [12]. So in this study, author will also do trial and error to define the best number of trees, but for the start will be set with 2 to 5 decision trees in parallel and for the max depth of the trees will be set to 5, 10, 20, 30, 40 and unlimited where it will expand until it uses all the variables. The trees modelling process will be carried out by using the random feature selection method inside to minimize errors and for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the splitting measures will be using the Gini impurity method with 2 to 5 number of trees inside the forest.

In the fourth stage, the focus will be on ensembling using the random forests itself. As shown in Figure 5, showing the multiple random forests in parallel process in detail, this is similar to the architecture of a random forest itself but in a nested bagging, because random forests is based on decision tree with bagging [2]. The last best number of trees and maximum depth on the tree will used in the nested random forests architecture and it will be made based on the the highest accuracy output from the previous stage, then the model will be made in parallel using one of the ensemble methods called the bootstrap aggregation or simply called 'bagging'. So the main idea is to run a multiple the random forests model simultaneously in parallel with a different dataset that has been split into several groups based on the number of n, which n is the number set for the models to be run in parallel. A number of 2 to 8 bootstrap with the best performing number of trees and max tree depth will be used to find the best output. After the parameter for random forest model found, a model competitor will be chosen in this experiment and it is the artificial neural network multilayer perceptron model with a backpropagation algorithm, a model that is loosely inspired by the Fig. 5. Bagged Random Forest human brain [13]. The purpose of this model creation is for the comparison with the random forest and will also be built using python, while the main reason it is chosen as a model competitor is because of several reasons, one of

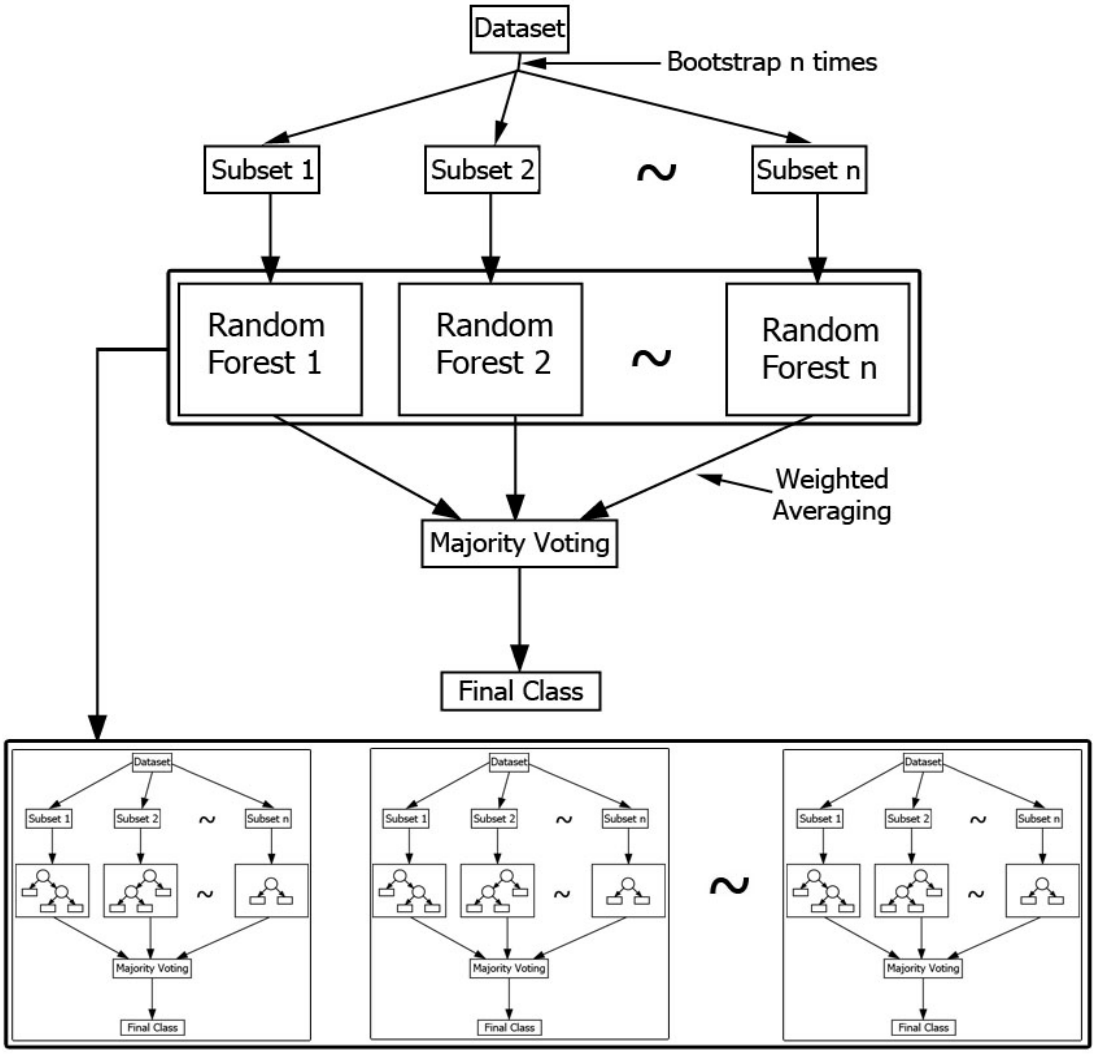

Figure 5. Bagged Random Forest

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

which is due to the increasingly widespread deep learning techniques which can only use neural networks as a model [14]. Second is due to the ability of the model to be flexibly regulated parameters in the input architecture, hidden layer, and output. All of these parameters are called the hyperparameters [15].

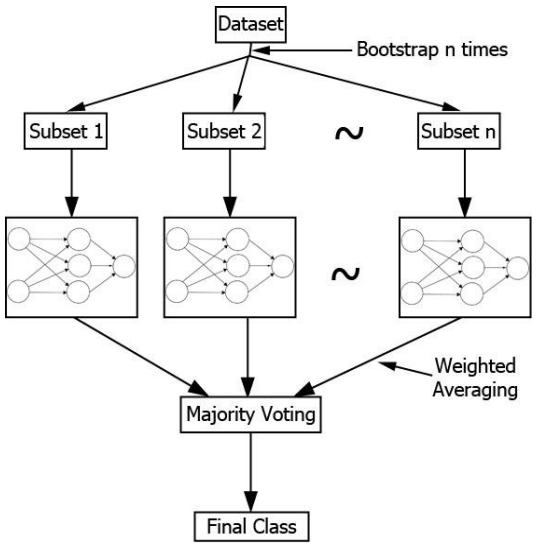

Figure 6. Bagged Neural Network

These hyperparameters include, number of hidden layers, number of nodes each layer, activation function, number of epoch. The best hyperparameters value can also be found the same as the random forest, which is by trials and errors [16]. The first and second trial will start with 23 nodes in the input layer will be used as the variable counts is 23 and a single hidden layer with 10-20 nodes within its layer, while for the epoch will be start using 1000 epoch, then the third and fourth trial will use, 2 hidden layers with 10-20 nodes on each hidden layer. The fifth and sixth trial will use 3 to 4 hidden layer, with also 10-20 nodes on each layer and for the output will still the same with 2 nodes on a range between zero to infinity. After the best parameter for hidden layer and neurons found, using the best parameter, author will try for tuning the epoch parameter for finding the best duration, the trial will try 500 and 1000 epochs. After the best parameter that produced the highest accuracy found, it will be used to create a base neural network that will be applied with the bootstrap aggregation. So the idea here is to run the neural network model simultaneously in parallel with a different dataset that has been split into several groups based on the number of n, which n is the number set for the models to be run in parallel. As in Figure 6, the bagged neural network process shown in detail, this is similar to the architecture of a random forest, but with an artificial neural network inside using the previous architecture with the same best hyperparameters from the last trial. A number of 2 to 5 bootstrap with the best performing number of hidden layer, neurons on each layer, and epoch will be used to find the best output.

In the fifth stage, a prediction test will be carried out to test the models that were made in the previous stage using the splitted dataset, then will be validated using k-fold Cross Validation. The k is the number of splits in the dataset. The idea of using k-fold cross validation is to prevent overfitting [17] . In this study, 5, 10, and 15 folds will be used to test each of every model made. In the sixth stage, a comparison between accuracy produced from each model in the previous stage will take place and a confusion matrix will be generated for each model to take a deep look at each number shown on true positive, true negative, false positive and false negative also the recall and precision output.

4. EXPERIEMENTS AND RESULTS

In this section, the result will be broken down based on the experiment from the stages described in the research method. The first stage in this experiment is to identify which variable will be used in this dataset. The 25 variables consist of attribute and predictor. Attribute will be used to generate new variables to be a predictor. Table 1 will list the detail of each variable that will be used in the dataset.

4.1 Data Preprocessing

Data Preprocessing step will consist of two processes, Data Cleaning and Data Transformation that will be break down in detail on each process.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Data Cleaning. The next step is data preprocessing where the data was cleaned and transformed, first is to fill the null value on every variable, starting from cust_street, cust_house, cust_house_electricity with modus, since it is categorical data. Then filling cust_spouse_income with 0, because it is an integer and may not guess each of every income, so it suits best as these customer's spouses don't have an income.

Data Transformation. The need to transform every string variable into categorical as integer. This string to integer transformation will be applied to objt_obj_type, objt_obj_brand, objt_model, cust_sex, cust_marital_status, cust_education, cust_street, cust_house, cust_occupation, cust_economic_sector, cust_ao_ro_status. Then removing null values on 794 rows in cust_overdue, because this attribute is needed to generate the target variable. After that a new variable will be

Table 1 Variable Details

No	Variable Name	Description	Null Value	Datatype	Type
1	APPL_DATE	Application Date	0	Date	Attribute
2	INST_AMT	Customer Installment	0	Int	Predictor
3	APPL_PERCENTAGE_NET_DP	Customer DP %	0	Int	Predictor
4	TENOR	Instalment Tenor	0	Int	Predictor
5	OBJ_PRICE	Object Price	0	Int	Predictor
6	OBJT_OBJ_TYPE	Object Type	0	String	Predictor
7	OBJT_OBJ_BRAND	Object Brand	0	String	Predictor
8	OBJT_MODEL	Object Model	0	String	Predictor
9	OBJT_MFG_YEAR	Object Creation Date	0	Int	Predictor
10	CUST_SEX	Customer Gender	0	String	Predictor
11	CUST_BIRTH_DATE	Customer Birth Date	0	Date	Attribute
12	CUST_MARITAL_STATUS	Customer Marital Status	0	String	Predictor
13	CUST_NO_OF_DEPENDENTS	Customer Dependency	0	Int	Predictor
14	CUST_EDUCATION	Customer Last Education	0	String	Predictor
15	CUST_STREET	Customer House Street Type	47	String	Predictor
16	CUST_HOUSE	Customer House Status	21	String	Predictor
17	CUST_YEAR_OF_STAY	Customer House Stay Duration	0	Integer	Predictor
18	CUST_HOUSE_ELECTRICITY	Customer House Electrical Power	300	Integer	Predictor
19	CUST_OCCUPATION	Customer Occupation	0	String	Predictor
20	CUST_ECONOMIC_SECTOR	Customer Occupation Sector	0	String	Predictor
21	CUST_YEAR_OF_WORK	Customer Job Duration	0	Integer	Predictor
22	CUST_NET_INCOME	Customer Net Income	0	Integer	Predictor
23	CUST_SPOUSE_INCOME	Customer Spouse Income	7574	Integer	Predictor
24	CUST_AO_RO_STATUS	Customer Loan Type	0	String	Predictor
25	CUST_OVERDUE	Customer Max OD in instalment	794	Integer	Attribute

created as a predictor, "cust_age" from (cust_birth_date-appl_date) and a target variable will also be created for the training process, since this is a supervised learning. The target variable will be named as "credit_classification" where it is created from cust_overdue with value between '0-90 days' means 'good', while overdue '>=90 days' means 'bad'.

4.2 Modelling With Random Forest

When the dataset is ready, the next step is to create he random forest model. With the training dataset and test dataset split 80% and 20% respectively on every model created. As shown in Table 2, it can be seen that a random forest output accuracy is better on every increase in the max tree depth. For every number of trees, author tried with 5,10, 20, 30, 40, and unlimited number of max tree depth. Every increase in number of trees the duration also increase, the same as the increase of max tree depth, the duration is also increase. With 2, 3, 4, and 5 number of trees and 5 and 10 max tree depth, generate the same number 0.71 and 0.73 of accuracy respectively with the difference only on duration of 0.39s and 0.59s for 2 number of trees, 0.52s and 0.76s for 3 number of trees, 0.62s and 0.95s for 4 number of trees, 0.70s and 1.11s for 5 number of trees. But with 20 number of max tree depth the accuracy and duration is all different, for 2 number of trees generate 0.82 accuracy with duration of 0.86s, for 3 number of trees generate 0.83 accuracy with duration of 1.19s, for 4 number of trees generate 0.85 accuracy with duration of 1.54s, for 5 number of trees generate 0.86 accuracy with duration of 1.80s. But with 30, 40, and unlimited max tree depths, generate the same accuracy for 2 and 3 number of trees generate 0.9 accuracy with duration of 0.96s, 0.97s, and 1.03s respectively, but for 4 and 5 number of trees is different from the others, for 4 number of trees generate the same accuracy of 0.93 with duration of 1.75s, 1.83s, and 1.87s respectively. For 5 number of trees generate the same accuracy of 0.92 with duration of 2.06s, 2.10s, 2.16s respectively. Based on this result the accuracy increase to the highest of 0.93 with 4 number of trees and 40 number of max tree depth., but with 5 number of trees and 40 number of max tree depth, the accuracy decrease to 0.92 but with longer duration from 1.83s to 2.10s, this is prove to a theory, a larger forest is not guaranteed to give the best prediction output [12]

While with 30, 40, and unlimited max tree depth give the same accuracy output , this means that the accuracy reach is highest with max tree depth between 20 to 30. As shown in Figure 7 can be seen accuracy over depths for 4 number of trees, can be seen with the increase of number of depths, the accuracy also increase bit by bit and become stable for in between of 20 and 30 until 50 number of max tree depths.

Table 2 Random Forest Model Output

Number of Trees	Max Tree Depth	Accuracy	Duration
2	5	0.71	0.39s
2	10	0.73	0.59s
2	20	0.82	0.86s
2	30	0.9	0.96s
2	40	0.9	0.97s
2	Unlimited	0.9	1.03s
3	5	0.71	0.52s
3	10	0.73	0.76s
3	20	0.83	1.19s
3	30	0.9	1.36s
3	40	0.9	1.36s
3	Unlimited	0.9	1.44s
4	5	0.71	0.62s
4	10	0.73	0.95s
4	20	0.85	1.54s
4	30	0.93	1.75s
4	**40**	**0.93**	**1.83s**
4	Unlimited	0.93	1.87s
5	5	0.71	0.70s
5	10	0.73	1.11s
5	20	0.86	1.80s
5	30	0.92	2.06s
5	40	0.92	2.10s
5	Unlimited	0.92	2.16s

The red line is based on the train dataset, while blue line is based on test dataset. So based on Table 2 and Figure 7, the best number of trees and max tree depths for random forest will be 4 number of trees with 40 number of max tree depth. So it is known that for the next stage, a random forest using 4 number of trees and 40 number of max tree depth as the parameters will be used. Next is the creation of the ensembled model using the classifier with the known best parameters, this is the output from the ensembled model.

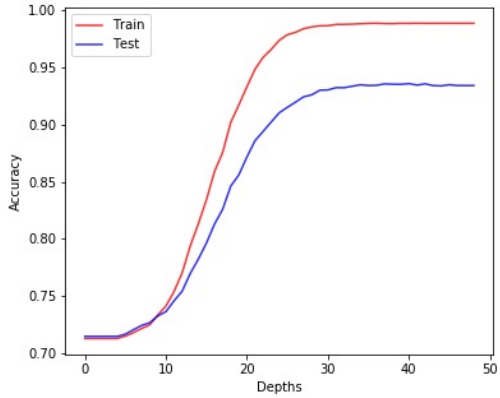

Figure 7. Accuracy over Depths

4.3 Ensembling With Multiple Random Forest

Table 3 Ensembled Random Forest Model Output

Random Forest : Bootstrap Aggregation				
Number of Trees	Max Tree Depth	Bootstrap	Accuracy	Duration
4	40	2	0.91	2.34s
4	40	3	0.92	3.42s
4	40	4	0.93	4.58s
4	40	5	0.93	5.62s
4	**40**	**6**	**0.94**	**6.79s**
4	40	7	0.94	7.83s
4	40	8	0.94	8.92s

As shown in Table 3, it can be seen that the ensembled random forest with 2 bootstrap has an accuracy of 0.91 with duration of 2.34s but with 3 bootstrap the accuracy increased by 0.01 to 0.92 with duration of 3.42s. For 4 and 5 bootstrap the accuracy output is the same, increased also with 0.01 to 0.93 with duration of 4.58s and 5.62s respectively. For 6, 7, and 8 bootstrap the accuracy is the same with also the same increasement as before with 0.01 to 0.94s with duration of 6.79s, 7.83s, 8.92s respectively. Experiment stop at bootstrap 8 because the accuracy did not increase anymore after 3 times increasement in the bootstrap. Based on this result, the best parameter for ensembled random forest is with 6 bootstrap with 4 number of trees and 40 number of max tree depth, that generate 0.94 accuracy with the lowest duration of 6.79s, because in every increase in bootstrap, the number of random forest ran in parallel also increased, making the higher duration because it needs more time to process every dataset on each model, resulting in a voting process for the final classification. So the next step is to compare with the neural network as the comparison model we choose. But first thing first, author will need to find for the best parameter in the neural network.

4.4 Modelling with Neural Network

Using the same dataset used in the random forest, a neural network model will be created. With also the same training dataset and test dataset split to 80% and 20% respectively, as shown in Table 4.4, it can be seen that a neural network with only 1 hidden layer with 10 neurons, have the same accuracy with 2 hidden layers with 10 neurons in each layer, having the same accuracy of 0.71. The difference only on their each duration. Duration in here is the time it took from training the model with 80% of the dataset, until testing the model with 20% of the dataset and generating an accuracy from the model.

While doubling the total number of neurons on each layer from 10 to 20 will result also nearly twice in the duration, can be seen on 1 hidden layer each 10 to 20 neurons will result in 18.38s to 32.55s with accuracy of 0.731 and 0.734, an increase of 0.003, in 2 hidden layer each 10 to 20 neurons will result in 22.81s to 68.32s with accuracy of 0.731 and 0.733, an increase of 0.02, in 3 hidden layer each 10 to 20 neurons will also doubled the duration from 44.45s to 126.01s, this duration depends on the machine used to execute this process, as the machine get faster, the execution process will also be faster which means the smaller duration it can get and vice versa.

Table 4 Neural Network Model Output

Neural Network Multilayer Perceptron				
Hidden layer	Neurons	Epoch	Accuracy	Duration
1	10	1000	0.731	18.37s
1	**20**	**1000**	**0.734**	**32.55s**
2	10	1000	0.731	22.81s
2	20	1000	0.733	68.31s
3	10	1000	0.731	44.45s
3	20	1000	0.733	126.01s
4	10	1000	0.730	79.33s
4	20	1000	0.733	171.83s

But as the hidden layer increased, the accuracy produced the same, as it can seen on Table 4.4, by increasing the number of hidden layers to 3 with the same 10 neurons each, only giving a slight increase in accuracy which is 0.002 from 0.731 to 0.733. But if it combined with the increased number of

Journal of Theoretical and Applied Information Technology
31st July 2022. Vol.100. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

neurons to 20 in each layer, the accuracy output is still the same with 0.733 but only increased the duration. In the last experiment using 4 hidden layer, using 10 neurons on each layer, the accuracy dropped by 0.001 to 0.730, while with 20 neurons on each layer only result with 0.733, the same as before but with the longest duration of 171.83s. So from the experiment in this stage, a neural network using 1 hidden layers with 20 neurons on each of the hidden layers will give a better result with accuracy of 0.734. To find out further, several number of epochs will be tested, to check whether it can also give an increase in accuracy on 1 hidden layers with 20 neurons with 500, 1000, 1500, and 2000 epochs.

Table 5 Neural Network Model Epoch Output

Neural Network Multilayer Perceptron				
Hidden layer	Neurons	Epoch	Accuracy	Duration
1	**20**	**500**	**0.734**	**31.84s**
1	20	1000	0.734	32.55s
1	20	1500	0.734	32.64s
1	20	2000	0.734	32.49s

As shown in Table 5, each epoch tested, 500, 1000, 1500, 2000 all have the same accuracy output with 0.734, the only difference is the duration, but only a few seconds that can be calculated as a network or machine performance issue. This happens because the backpropagation learning algorithm has been sufficiently minimized the error output by doing a number of loops over and over again [18]. In this case, an epoch of 500 is enough for the model to reach the minimum error as possible, which means if the number of epochs is more than 500, it is only for the maximum looping done in the training process.

As shown in Figure 8, it gave insight about accuracy over epochs. The accuracy produced from train and test dataset gets better from each epoch. The red line is for train dataset, while blue line is for test dataset, can be seen from the first epoch, train dataset produced 0.7231 in accuracy, while test dataset produced 0.7261. Through the training process, as the epoch increases, the accuracy also increases all the way until it reaches the minimum error output on epoch 131, producing an accuracy of 0. 7395 for the train dataset and 0.7347 for test dataset respectively. Meaning that beyond epoch 131, the accuracy produced is the same, from the first iteration until it stopped training on epoch 131 because training loss did not improve more than 0.0001 so it stopped the process.

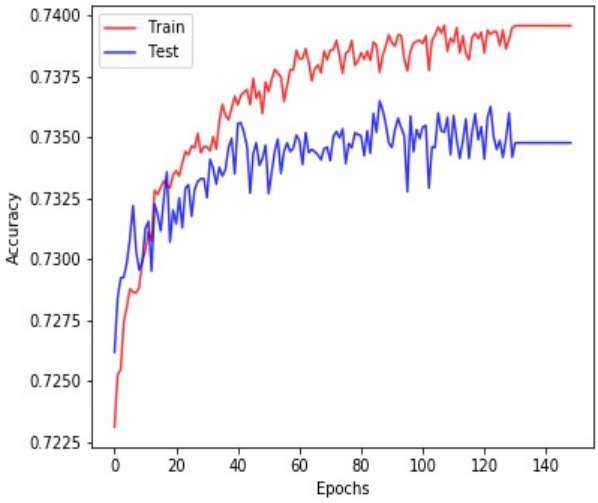

Figure. 8. Accuracy over Epochs

So it is known that for the next stage, a neural network using 1 hidden layers with 20 neurons on each of the hidden layers using 500 of epoch as the parameters will be used. Next is the creation of the ensembled model using the neural network with the known best parameters for comparison with the ensembled random forest as it is a similar architecture.

4.5 Ensembling with Multiple Neural Network

Table 6 shows the output from the ensembled model, it can be seen that the ensembled neural network with 2 bootstrap has an accuracy of 0.73 but with 3 bootstrap has the same accuracy as 4 and 5 bootstrap, the accuracy increased by 0.01 to 0.74. but as the bootstrap increased, the duration gets higher from 56.18 seconds, 87.80 seconds, 118.04 seconds and 138.65 seconds respectively. It is because every increase in bootstrap, the number of neural networks ran in parallel needs more time to process every dataset on each model, resulting in a voting process for the final classification.

Table 6 Ensembled Neural Network Model Output

Neural Network : Bootstrap Aggregation					
Hidden Layer	Neurons	Epoch	Bootstrap	Accuracy	Duration
1	20	500	2	0.73	56.18s
1	**20**	**500**	**3**	**0.74**	**87.80s**
1	20	500	4	0.74	118.04 s
1	20	500	5	0.74	138.65 s

4.6 Validation Test

After these models are created, a validation test will take place in the last stage using K-Fold Cross Validation to test the 2 models that have been created. The ensembled random forest with 4 number of trees and 40 number of max tree depth with 6 bootstrap against ensembled neural network that will use 1 hidden layers with 20 neurons on each layer with 500 of epoch paralleled with 3 bootstrap. The k number of fold will be 5, 10, and 15 fold, while the output prediction is the mean/average from all generated accuracy. And this is the output from the test on each fold.

Table 7 K-Fold Cross Validation Output

K Fold Cross Validation				
	Bootstrap	5 fold	10 fold	15 fold
Random Forest : Bagging	6	0.94	0.95	0.96
Neural Network : Bagging	3	0.74	0.73	0.71

As shown in Table 7, the ensembled random forest with 6 bootstrap test generate an accuracy of an increased of 0.01 to the highest of 0.96 from 0.94 on every fold, 5, 10, and 15 fold, while the ensembled neural network with 3 bootstrap have the output test score of 0.74 on 5 fold, the accuracy decreased by 0.01 to 0.73 on the 10 fold, while decreased by 0.02 to the lowest on 0.71 on 15 fold. Based on Table 7, the highest accuracy output produced by the ensembled random forest with 6 bootstrap in parallel. For the next stage, a confusion matrix from random forest, ensembled random forest, and ensembled neural network models will be generated.

4.7 Comparison & Evaluation

Table 8 shows that the random forest model has an accuracy of 93.52%, which can predict 45.884 True Positive with 16.182 True Negative, while the Precision and Recall generate output of 97% and 94% respectively.

5

Table 8 Random Forest Confusion Matrix

		Actual		
Accuracy	93.52%	Positive	Negative	Precision
Predicted	Positive	45884	1550	97%
	Negative	2748	16182	
	Recall	94%		

Table 9 Ensembled Random Forest Confusion Matrix

		Actual		
Accuracy	94.58%	Positive	Negative	Precision
Predicted	Positive	46354	1080	98%
	Negative	2518	16412	
	Recall	95%		

Table 9 shows that the ensembled random forest model has an accuracy of 94.58%, which have 1.06% difference higher with the single random forest and can predict 46.354 True Positive with 16.412 True Negative, while the Precision and Recall generate output of 98% and 95% respectively, which have and exact of 1% difference higher with the single random forest.

Table 10 Ensembled Neural Network Confusion Matrix

		Actual		
Accuracy	74.01%	Positive	Negative	Precision
Predicted	Positive	43816	3618	92%
	Negative	13630	5300	
	Recall	76%		

Table 10 shows that the ensembled neural network model has an accuracy of 74.01%, which can predict 43.816 True Positive with 5.300 True Negative, while the Precision and Recall generate output of 92% and 76% respectively.

So based on experiments and results that have been done in this chapter, it can be concluded that ensembling a single model using bootstrap aggregation can improve the model performance in terms of accuracy with duration tradeoffs, therefore to improve a credit scoring model, it can be built using the Random Forest using the ensemble method approach especially the bootstrap aggregation method.

5. CONCLUSION

This research shows ensemble method using bootstrap aggregating success to improve the performance classification in a credit scoring problem comparing single method. Two methods are used random forest and neural network to shows the success of this ensemble. It is proven by using 6 bootstrap can boost up to 1% accuracy from a single model . However, in the ensemble, the more bootstrap, the longer duration in training model.

In future, the performance of accuracy can still be improved by doing a hyperparameters automatic tuning to find the most optimal value for the

hyperparameters that can tweak the model performance.

REFERENCES

[1] A. Ghodselahi and A. Amirmadhi, "Application of Artificial Intelligence Techniques for Credit Risk Evaluation," Int. J. Model. Optim., vol. 1, no. 3, p. 243, 2011.

[2] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[3] E. Dumitrescu, S. Hue, C. Hurlin, and S. Tokpavi, "Machine Learning for Credit Scoring: Improving Logistic Regression with Non Linear Decision Tree Effects," Doctoral dissertation, 2018.

[4] J. Abellán and J. G. Castellano, "A comparative study on base classifiers in ensemble methods for credit scoring," Expert Syst. Appl., vol. 73, pp. 1–10, 2017.

[5] S. Ardabili, A. Mosavi, and A. R. Várkonyi-Kóczy, "Advances in Machine Learning Modeling Reviewing Hybrid and Ensemble Methods," in Lecture Notes in Networks and Systems, vol. 101, Springer, 2020, pp. 215–227.

[6] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial," arXiv Prepr. arXiv1905.12787, 2019.

[7] B. Kim, "Ensemble Methods Applied to Classification Problem," Int. J. Internet, Broadcast. Commun., vol. 11, no. 1, pp. 47–53, 2019.

[8] P. Singh, "Comparative study of individual and ensemble methods of classification for credit scoring," in 2017 International Conference on Inventive Computing and Informatics (ICICI), 2017, pp. 968–972.

[9] H. He, W. Zhang, and S. Zhang, "A novel ensemble method for credit scoring: Adaption of different imbalance ratios," Expert Syst. Appl., vol. 98, pp. 105–117, 2018.

[10] C. Zhang and Y. Ma, Ensemble machine learning: methods and applications. Springer, 2012.

[11] J. C. Bezdek, S. Boggavarapu, L. O. Hall, and A. Bensaid, "Genetic algorithm guided clustering," in Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on, 1994, pp. 34–39.

[12] P. Probst and A.-L. Boulesteix, "To Tune or Not to Tune the Number of Trees in Random Forest.," J. Mach. Learn. Res., vol. 18, no. 1, pp. 6673–6690, 2017.

[13] S. S. Haykin, "Neural networks and learning machines/Simon Haykin." New York: Prentice Hall, 2009.

[14] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015.

[15] P. Probst, A.-L. Boulesteix, and B. Bischl, "Tunability: Importance of hyperparameters of machine learning algorithms.," J. Mach. Learn. Res., vol. 20, no. 53, pp. 1–32, 2019.

[16] T.-S. Lee and I.-F. Chen, "A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines," Expert Syst. Appl., vol. 28, no. 4, pp. 743–752, 2005.

[17] T. Tušar, K. Gantar, V. Koblar, B. Ženko, and B. Filipič, "A study of overfitting in optimization of a manufacturing quality control procedure," Appl. Soft Comput., vol. 59, pp. 77–87, 2017.

[18] I. Wahyuni, N. R. Adam, W. F. Mahmudy, and A. Iriany, "Modeling backpropagation neural network for rainfall prediction in tengger east Java," in 2017 International Conference on Sustainable Information Engineering and Technology (SIET), 2017, pp. 170–175.