ISSN 2347 - 3983
Volume 8. No. 5, May 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter19852020.pdf
https://doi.org/10.30534/ijeter/2020/19852020

Classification Performance for Credit Scoring using Neural Network

Christopher Edmond[1], Abba Suganda Girsang[2]
[1] Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480, christopher.edmond@binus.ac.id
[2] Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480, agirsang@binus.edu

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 will use a real world dataset, where data is obtained directly from a financial company and will be used to feed an Artificial Neural Network to differentiate between good and bad potential new customers. In this research, the Artificial Neural Network will be created, and then will be applied with the ensemble method. The idea of applying the parallel is to test whether it can increase accuracy or not. Finally, the output accuracy from the final class of the ensemble methods that resulted from the voting will be compared with the original unmodified single model and it shows that the modified multiple model surpasses the unmodified single model in terms of accuracy with a tradeoff on duration in the process.

Key words: Bootstrap Aggregation, Credit Scoring, Ensemble Methods, Neural Network

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, a large number of studies have been made on this topic, proposed using many different models, such as K-nearest neighbor [2], Support Vector Machine [3], Logistic Regression [4], Neural Network [5], Naive Bayes [6], as well as Decision Tree [7] and several hybrid models such as Fuzzy C-Means [8], The Lasso Group [9], Markov Chain [10], Ensemble Methods [11] were also proposed for further analysis.

This research study is all focused on the ability of each model to make predictions with the highest accuracy values that can be achieved. Although a lot of research study has been done in this field, it is still possible to continue doing research study to improve more [12]. All of this research is carried out with the same goal, which is the need for a reliable model for making accurate predictions to prevent the high risk of lending, but the weak learner or single model learner that is often used by researchers can be said that the performance is not so good when compared to strong learner or multiple model learner [13], which is a combination resulting from more than one of the same (homogeneous) or different model (heterogeneous) that is run in parallel will enhance the performance of the model [14]. With all of the information that has been obtained from many of these studies, a research by combining several weak learners model with similar models and run them in parallel will be conducted, by using a method called the bootstrap aggregation which is one of the hybrid types of ensemble methods technique that is used in the random forest architecture to make a strong learner [11].

The weak learner or single model that is chosen in this research study is the neural network, a model that is loosely inspired by the human brain [15]. The main reason it is chosen as a model to be studied is because of several reasons, one of which is due to the increasingly widespread deep learning techniques which can only use neural networks as a model [16]. Second is due to the ability of the model to be flexibly regulated in the input architecture, hidden layer, and output [17]. Third is because neural networks are too flexible in their parameters, it makes all of the studies on these neural networks not the same because with different architectures, the results are different; each research has different parameters to adjust to the conditions that want to be completed [18].

Next in chapter 2, a quick explanation on the theory of every model and methods that will be used in chapter 3. Then in chapter 3, will be further explaining the research method on how the model is made, starting from variable identification to how to make the model in detail using the ensemble method

referred to earlier, until the comparison between models. Later in the chapter, the result from each model will be discussed and compared. For the last chapter, conclusions will be drawn from the results of this study.

2. MODELS

In this section, each model used in this study to make credit scoring will be discussed.

2.1 Artificial Neural Network Multilayer Perceptron

Artificial Neural Network (ANN), also known as Multilayer Perceptron is a class of feed forward neural network and it is the most commonly used method. It is a method of Artificial Intelligence which is a concept that is made like a network designed to resemble the human brain. Where are built nodes that are interconnected with one another. These nodes are connected through a link which is commonly referred to as weight. The basic idea is to adopt the workings of the human brain that have the characteristics of parallel processing, processing elements in large numbers and fault tolerance.

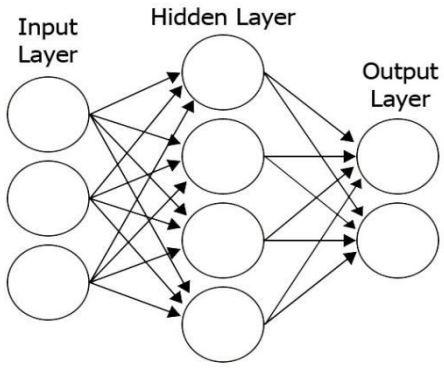

Figure 1: Basic Multilayer Perceptron with 1 Hidden Layer

Artificial Neural Network is a large number of processors that are distributed in parallel and consist of simple processing units, where each unit has a tendency to store knowledge that is experienced and can be reused [15]. In general, neural networks are divided by layers as explained in Figure 1, the input layer, hidden layer and output layer. Each node in each layer has an error rate, which will be used for the training process. This network is usually implemented using electronic components or simulated on a computer application. In a basic multilayer perceptron shown in Figure 1, can be used to explain the formula shown on Equation 1 from of multilayer perceptron itself, where n is the number of neurons in the input layer, then will be multiplied by weight on every neurons to get the output value of y. In every output neuron in the next layer, there is an offset value that is called bias. This value of bias b will be added to each weighted sum from every neuron from the previous layer. After this summing takes place, whether the value can be passed to the next neuron in the next layer will be decided by passing the output value to the activation function σ.

$$y = \sigma\left(\sum_{i=0}^{n} x_i w_i \right) = \sigma\left(w^T x \right) \quad (1)$$

or if the formula shown in Equation 1 transformed into a one big picture, it can be viewed as shown in Figure 2.

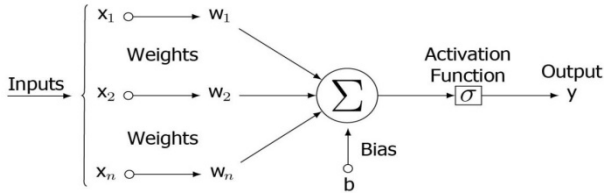

Figure 2: Neural Network Calculation in a Big Picture

2.1.1 Backpropagation Algorithm

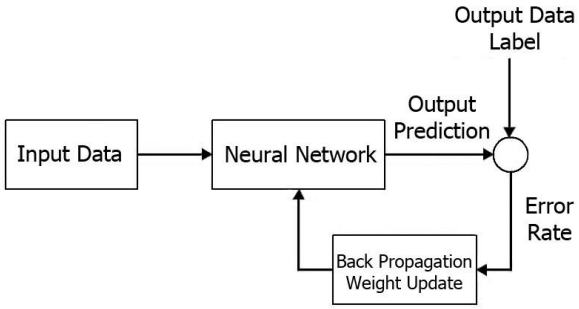

Figure 3: Model of Backpropagation Algorithm

Backpropagation is a training algorithm that is applied to the multilayer perceptron and it is a supervised learning algorithm [15]. The function of this algorithm as shown in Figure 3, is to train the network to obtain the desired weight. Every time a training process is done, the neural network will produce an output prediction. The difference between the output prediction produced by the neural network and the output data label or the desired target is called the error output. As described above in Figure 3, Backpropagation will use the error output to modify the weight values in the backward direction. The training algorithm in backpropagation has three phases, namely feedforward propagation phase, backward propagation phase and weight modification phase. In the feedforward phase, the input pattern is calculated forward from the input layer to the output layer. In the backpropagation phase, error output will be calculated from every output from the neural network, then the error output will be propagated backwards. In the backpropagation phase the weight is modified to reduce the error until the desired condition is met.

2.3 Ensemble Method: Bootstrap Aggregation

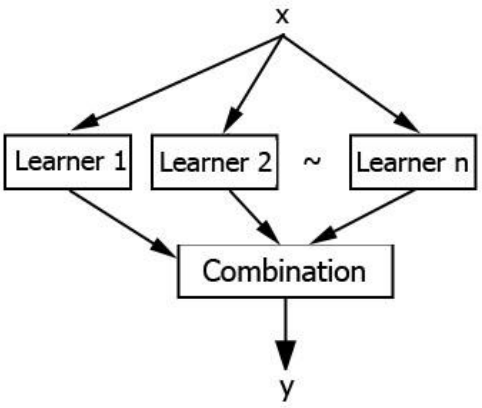

Figure 4: Common Ensemble Architecture

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 [19]. 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 4 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 [20]. The most commonly used ensemble methods are bagging, boosting, and stacking.

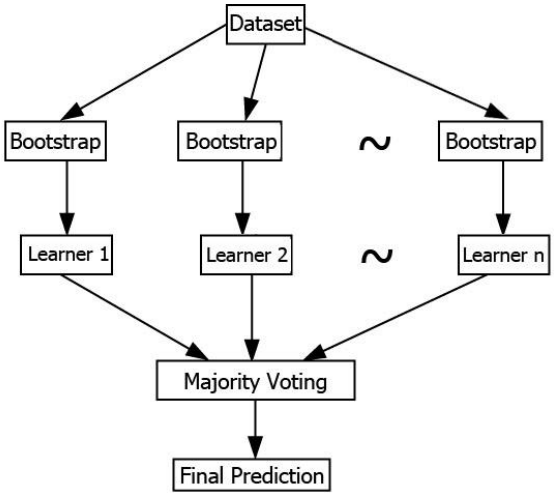

Figure 5: Bootstrap Aggregation Architecture

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 5, modeling 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 method has better accuracy and is quite significant compared to the single (base) classification model. The increase in accuracy occurs because the combination of models can reduce the variance of a single grouping. As shown on Equation 2, a bagging has B which separates the bootstrapped samples from the training set and using those separated samples to feed the model [21], then it will produce a new model with a new formula as shown in Equation 2.

$$\tilde{f}_{avg}(x) = \frac{1}{B}\sum_{b-1}^{B} \hat{f}^{b}(x) \qquad (2)$$

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 6, there will be 6 stages to be carried out. In this study, a real-world dataset directly from a financing company is used.

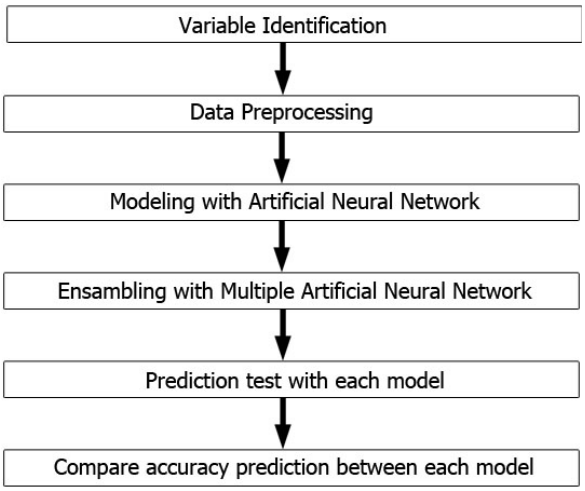

Figure 6: Research Method Stage

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 Preprocessing" 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 preprocess 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. 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 labeling purpose is for the 'supervised learning' process in the training stage.

In the third stage, the artificial neural network multilayer perceptron model with a backpropagation algorithm is built using python. An artificial neural network has many parameters that need to be set for it to run, all of these parameters are called the hyperparameters [22]. These hyperparameters include, number of hidden layers, number of nodes each layer, activation function, number of epoch. The best hyperparameters value can be found by trials and errors [23], but for the first time, a 22 nodes in the input layer will be used as the variable counts is 22, using 1-3 hidden layers with 10-20 nodes on each hidden layer, and 2 nodes on output layer as to classify a good or bad customer, using rectified linear unit as the activation function, which have a range between zero to infinity.

In the fourth stage, an ensembled of multiple artificial neural networks will be made using a parameter that produced the highest accuracy from the previous stage in parallel using one of the ensemble methods called the bootstrap aggregation or simply called 'bagging'. 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. A number of 2 to 4 bootstrap will be used to find the best output. Figure 7 shows the bagged neural network process in detail, this is similar to the architecture of a random forest.

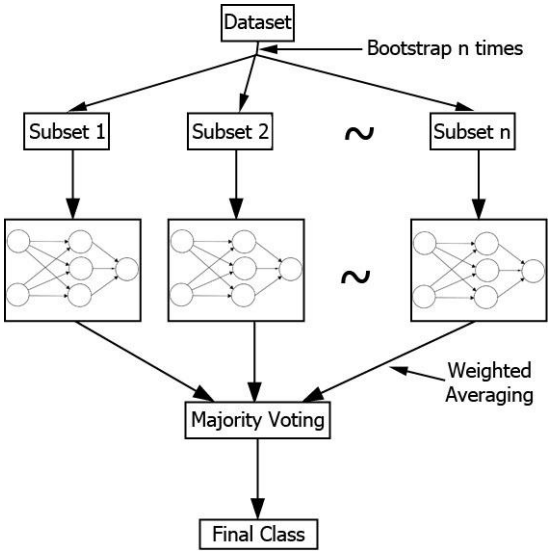

Figure 7: Bagged Neural Network

In the fifth stage, a prediction test will be carried out to test the models that were made in the previous stage by 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 [24]. 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 the model with the highest accuracy to take a deep look at each number shown on true positive, true negative, false positive and false negative.

4. EXPERIMENT & RESULT

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:

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	Integer	Predictor
3	APPL_PERCENTAGE_NET_DP	Customer DP %	0	Integer	Predictor
4	TENOR	Installment Tenor	0	Integer	Predictor
5	OBJ_PRICE	Object Price	0	Integer	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	Integer	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	Integer	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 installment	794	Integer	Attribute

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.

Now 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 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’.

After the dataset is ready, the next step is to create the neural network model. With the training data set and test dataset split 80% and 20% respectively on every model created. As shown in Table 2, it can be seen that a neural network with only 1 hidden layer with 10 and 20 neurons, have the same accuracy with 2 hidden layers with 10 neurons in each layer, having the same accuracy of 0.67. 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.

Table 2: Neural Network Model Output

Neural Network Multilayer Perceptron				
Hidden layer	Neurons	Epoch	Accuracy	Duration
1	10	1000	0.67	17s
1	20	1000	0.67	31s
2	10	1000	0.67	26s
2	20	1000	0.71	57s
3	10	1000	0.68	67s
3	20	1000	0.73	127s

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 17 to 31 seconds, in 2 hidden layer each 10 to 20 neurons will result in 26 to 57 seconds, in 3 hidden layer each 10 to 20 neurons will also doubled the duration from 67 to 127 seconds, 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. But as the hidden layer increased, the accuracy produced better results, as it can seen on Table 2, by

increasing the number of hidden layers to 3 with the same 10 neurons each, only giving a slight increase in accuracy which is 0,01 from 0.67 to 0.68. But if it combined with the increased number of neurons to 20 in each layer, will a better output with 0.71 and 0.73 respectively. So from the experiment in this stage, a neural network using 3 hidden layers with 20 neurons on each of the hidden layers will give a better result. To find out further, several number of epochs will be tested, to check whether it can also give an increase in accuracy on 3 hidden layers with 20 neurons with 500, 1000, 1500, and 2000 epochs.

Table 3: Neural Network Model Epoch Output

Neural Network Multilayer Perceptron				
Hidden layer	Neurons	Epoch	Accuracy	Duration
3	20	500	0.73	126s
3	20	1000	0.73	127s
3	20	1500	0.73	130s
3	20	2000	0.73	129s

As shown in Table 3, each epoch tested, 500, 1000, 1500, 2000 all have the same accuracy output with 0.73, 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 [25]. 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.

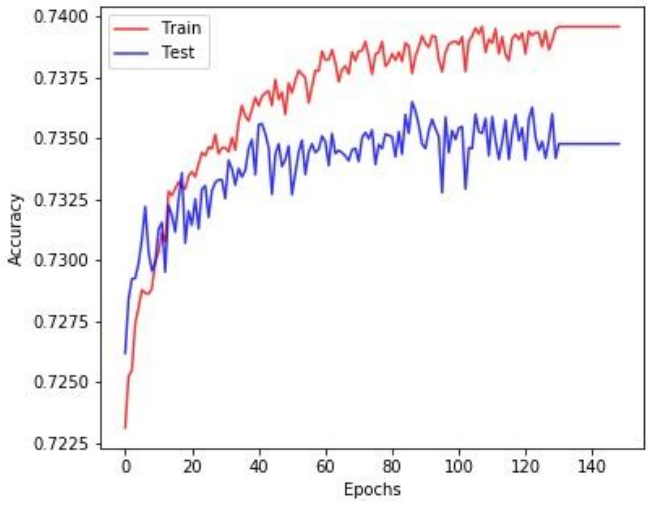

Figure 8: Accuracy over Epochs

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. Beyond epoch 131, the accuracy produced is the same, can be seen on Figure 9 about the training loss curve, 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. These 2 diagrams, both showing the whole process log while training the model.

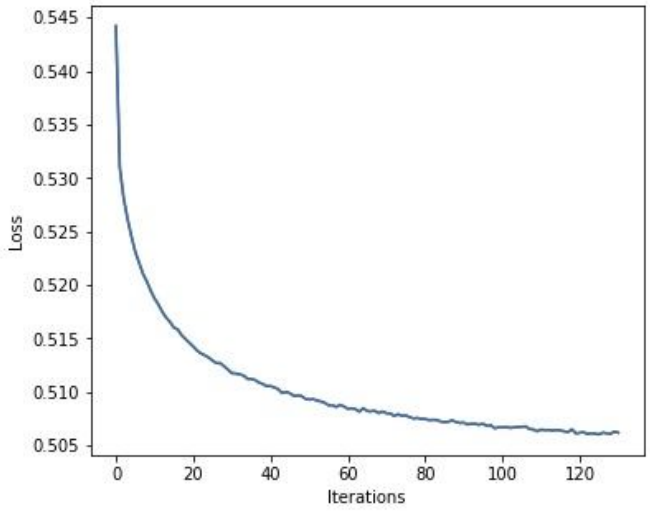

Figure 9: Loss over Epochs

So it is known that for the next stage, a neural network using 3 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 classifier with the known best parameters, this is the output from the ensembled model.

Table 4: Ensembled Neural Network Model Output

Multilayer Perceptron : Bootstrap Aggregation					
Hidden Layer	Neurons	Epoch	Bootstrap	Accuracy	Duration
3	20	500	2	0.73	234s
3	20	500	3	0.74	393s
3	20	500	4	0.74	532s

As shown in Table 4, 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 bootstrap, the accuracy increased by 0.01 to 0.74. but as the bootstrap increased, the duration gets higher from 234 seconds, 393 seconds, and 532 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.

After these models are created, a validation test will take place in the fifth stage using K-Fold Cross Validation to test the 2 models that have been created. The neural network will use 3 hidden layers with 20 neurons on each layer with 500 of epoch against the ensembled neural network with the same parameters of 3 hidden layers with 20 neurons on each layer with 500 of epoch but paralleled with 2 bootstrap. The k number of fold will be 5, 10, and 15 fold, while the output prediction is the mean from all generated accuracy. And this is the output from the test on each fold.

Table 5: K-Fold Cross Validation Output

	Estimator	5 fold	10 fold	15 fold
MLP	1	0.71	0.71	0.71
MLP : Bagging	3	0.73	0.74	0.74

As shown in Table 5, the Neural Network test generate the accuracy a decreased of 0.02 to 0.71 from 0.73 on every fold, 5, 10, and 15 fold, while the Ensembled Neural Network with 3 bootstrap have the output test score of 0.73 on 5 fold, the accuracy decreased by 0.01 from 0.74, but for the 10 fold and 15 fold, the accuracy is the same as before which is 0.74. Based on table 5, the highest accuracy output produced by Ensembled Neural Network with 3 bootstrap in parallel and a confusion matrix from these 2 models will be generated.

Table 6: Neural Network Confusion Matrix

		Actual		
Accuracy	71%	Positive	Negative	Precision
Predicted	Positive	41.510	6.324	87%
	Negative	12.278	5.152	
	Recall	77%		

Table 6 shows that the neural network model has an accuracy of 71%, which can predict 41.510 True Positive with 5.152 True Negative, while Precision and Recall of 87% and 77% respectively.

Table 7: Ensembled Neural Network Confusion Matrix

		Actual		
Accuracy	74%	Positive	Negative	Precision
Predicted	Positive	44.084	3.350	93%
	Negative	13.776	5.154	
	Recall	76%		

Table 7 shows that the ensembled neural network model has an accuracy of 74%, which can predict 44.084 True Positive with 5.154 True Negative, while Precision and Recall of 93% and 76% respectively.

5. CONCLUSION

The creation of both Artificial Neural Network and Artificial Neural Network with the Ensemble Method are both can help to a credit scoring problem. A real-world data pulled directly from a financial company is used to compare both of the model performance in solving customer classification problems. The test results show that the Ensembled Neural Network with 3 numbers of bootstrap can boost the accuracy up to 3% from the single Neural Network classifier. The application of ensembled models can be used to help in credit scoring classification problems to improve the accuracy of the model. As for further research, a deep learning model with hyperparameters automatic tuning that can also tweak the model performance can be deeply investigated.

REFERENCES

1. A. Ghodselahi, **A Hybrid Support Vector Machine Ensemble Model for Credit Scoring,** International Journal of Computer Applications, vol. 17, no. 5, pp. 1–5, 2011. https://doi.org/10.5120/2220-2829
2. M. A. Mukid, T. Widiharih, A. Rusgiyono, and A. Prahutama, **Credit scoring analysis using weighted k nearest neighbor**, Journal of Physics: Conference Series, vol. 1025, p. 012114, 2018.
3. S. Maldonado, J. Pérez, and C. Bravo, **Cost-based feature selection for Support Vector Machines: An application in credit scoring**, European Journal of Operational Research, vol. 261, no. 2, pp. 656–665, 2017. https://doi.org/10.1016/j.ejor.2017.02.037
4. D. A. V. D. Paula, R. Artes, F. Ayres, and A. M. A. F. Minardi, **Estimating credit and profit scoring of a Brazilian credit union with logistic regression and machine-learning techniques**, RAUSP Management Journal, vol. 54, no. 3, pp. 321–336, Aug. 2019.
5. C. C. Escolar-Jimenez, K. Matsuzaki, and R. Gustilo, **A Neural-Fuzzy Network Approach to Employee Performance Evaluation**, International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, no. 3, pp. 573–581, 2019. https://doi.org/10.30534/ijatcse/2019/37832019
6. 6 M. Alaraj and M. F. Abbod, **Classifiers consensus system approach for credit scoring**, Knowledge-Based Systems, vol. 104, pp. 89–105, 2016.
7. A. S. Girsang and Abner, **Optimization of Debtor Credit Quality Determining Prediction using Decision Tree**, International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, no. 2, pp. 1076–1081, 2020.
8. S. Sitohang, A. S. Girsang, and S. Suharjito, **Prediction of the Number of Airport Passengers Using Fuzzy C-Means and Adaptive Neuro Fuzzy Inference**

System, International Review of Automatic Control (IREACO), vol. 10, no. 3, p. 280, 2017. https://doi.org/10.15866/ireaco.v10i3.12003

9. H. Chen and Y. Xiang, **The Study of Credit Scoring Model Based on Group Lasso**, Procedia Computer Science, vol. 122, pp. 677–684, 2017. https://doi.org/10.1016/j.procs.2017.11.423
10. X. Feng, Z. Xiao, B. Zhong, Y. Dong, and J. Qiu, **Dynamic weighted ensemble classification for credit scoring using Markov Chain**, Applied Intelligence, vol. 49, no. 2, pp. 555–568, May 2018.
11. 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. https://doi.org/10.1016/j.eswa.2016.12.020
12. S. Walusala W, D. R. Rimiru, and D. C. Otieno, **A Hybrid Machine Learning Approach for Credit Scoring Using PCA and Logistic Regression**, IJC, vol. 27, no. 1, pp. 84-102, Sep. 2017.
13. C. R. D. Devi and R. M. Chezian, **A relative evaluation of the performance of ensemble learning in credit scoring**, 2016 IEEE International Conference on Advances in Computer Applications (ICACA), 2016.
14. T. G. Dietterich, **Ensemble Methods in Machine Learning**, Multiple Classifier Systems Lecture Notes in Computer Science, pp. 1–15, 2000. https://doi.org/10.1007/3-540-45014-9_1
15. S. S. Haykin, **Neural networks and learning machines,** Delhi: Pearson, 2016.
16. J. Schmidhuber, **Deep learning in neural networks: An overview**, Neural Networks, vol. 61, pp. 85–117, 2015.
17. K. K. Lai, L. Yu, S. Wang, and L. Zhou, **Credit Risk Analysis Using a Reliability-Based Neural Network Ensemble Model**, Artificial Neural Networks – ICANN 2006 Lecture Notes in Computer Science, pp. 682–690, 2006. https://doi.org/10.1007/11840930_71
18. S. Yang and A. Browne, **Neural network ensembles: combining multiple models for enhanced performance using a multistage approach**, Expert Systems, vol. 21, no. 5, pp. 279–288, 2004.
19. C. Zhang and Y. Ma, **Ensemble machine learning: methods and applications**, Springer Science & Business Media: Springer, 2014.
20. B. Ghojogh and M. Crowley, **The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial, Machine Learning**, 2019.
21. G. James, D. Witten, T. Hastie, and R. Tibshirani, **An introduction to statistical learning: with applications in R**, New York: Springer, 2017.
22. P. Probst, A. Boulesteix and B. Bischl, **Tunability: Importance of Hyperparameters of Machine Learning Algorithms.**, Journal of Machine Learning Research, vol. 20, no. 53, pp. 1-32, 2019.
23. T. Lee and I. Chen, **A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines**, Expert Systems with Applications, vol. 28, no. 4, pp. 743–752, 2005. https://doi.org/10.1016/j.eswa.2004.12.031
24. T. Tušar, K. Gantar, V. Koblar, B. Ženko, and B. Filipič, **A study of overfitting in optimization of a manufacturing quality control procedure**, Applied Soft Computing, vol. 59, pp. 77–87, 2017. https://doi.org/10.1016/j.asoc.2017.05.027
25. I. Wahyuni, N. R. Adam, W. F. Mahmudy, and A. Iriany, **Modeling backpropagation neural network for rainfall prediction in tengger east Java**, 2017 International Conference on Sustainable Information Engineering and Technology (SIET), 2017. https://doi.org/10.1109/SIET.2017.8304130