Comparison of Adaptive Boosting and Bootstrap Aggregating Performance to Improve the Prediction of Bank Telemarketing

Agus Priyanto
Post Graduate Student
President University
Bekasi, Indonesia
aguspriyanto.st@gmail.com

Rila Mandala
Dean Faculty of Computing
President University
Bekasi, Indonesia
rilamandala@president.ac.id

Abstrak **- Background: Telemarketing is an effective marketing strategy lately, because it allows long-distance interaction making it easier for marketing promotion management to market their products. But sometimes with incessant phone calls to clients that are less potential to cause inconvenience, so we need predictions that produce good probabilities so that it can be the basis for making decisions about how many potential clients can be contacted which results in time and costs can be minimized, telephone calls can be more effective, client stress and intrusion will be reduced. strong.**

Method: This study will compare the classification performance of Bank Marketing datasets from the UCI Machine Learning Repository using data mining with the Adaboost and Bagging ensemble approach, base algorithm using J48 Weka, and Wrapper subset evaluation feature selection techniques and previously data balancing was performed on the dataset, where the expected results can be known the best ensemble method that produces the best performance of both.

Results: In the Bagging experiment, the best performance of Adaboost and J48 with an accuracy rate of 86.6%, Adaboost 83.5% and J48 of 85.9%Conclusion: The conclusion obtained from this study that the use of data balancing and feature selection techniques can help improve classification performance, Bagging is the best ensemble algorithm from this study, while for Adaboost is not productive for this study because the basic algorithm used is a strong learner where Adaboost has Weaknesses to improve strong basic algorithm.

Keyword : Telemarketing, Adaboost, Bagging, Decision Tree, J48

I. INTRODUCTION

In financial institutions marketing is the spearhead for the sustainability of its business. As technology develops, marketing methods also develop, telemarketing is an example of a new method of marketing. Telemarketing is an interactive direct marketing technique by marketing agents where potential customers will be contacted via electronic media such as telephone, fax or other media to offer an item or service.

One of the advantages of telemarketing is to focus long-distance interactions with customers in the contact center, thereby making it easier for marketing promotion management to market their products. According to the "Bank Marketing Survey Report-200" released by the American Banking Association / Bank Marketing Association, banks have sharply increased telemarketing [1].

From previous studies, many researchers used classification algorithms to analyze telemarketing performance improvements with data mining such as decision trees, naïve bayes, K-Nearest Neighbors, Support Vector Machines with good results, but the authors felt there were things that could be further explored for improve prediction performance by applying feature selection to attributes in the dataset and by comparing results when using the Bootstrap Aggregating and Adaptive Boosting algorithm where the algorithm is considered effective to improve the stability and accuracy of machine learning algorithms, especially classification and regression algorithms [2].

This study aims to prove that the use of a Bootstrap Aggregating and Adaptive Boosting algorithm can improve the classification performance of the results obtained by the J48 classifier and after the feature selection process using the Wrapper Subset Evaluation method, where the results of this study are expected to be useful for bank telemarketing world, where from the results of this study will obtain relevant attributes in determining potential clients. The model obtained from the classification results is used to test the testing data from the telemarketing database, so the results obtained from the test produce a prediction that is useful as information for decision making in determining potential clients from their database, so phone calls for telemarketing can be focused on potential clients, telephone calls become more effective, and this will lead to time and cost efficiency.

II. LITERATURE REVIEW

A. Feature Selection

Feature Selection or variable selection consists of reducing available features to optimal or sub-optimal assets and being able to produce the same or better results than the original set. Reducing the feature set will decrease the dimensions of the data which in turn reduces the training time of the selected induction algorithm and computational costs, increases the accuracy of the final result, and makes data mining results easier to understand and more applicable [3].

Dash and Liu (1997) break down the feature selection process into four steps: generation, evaluation, stopping criterion, and validation[4].

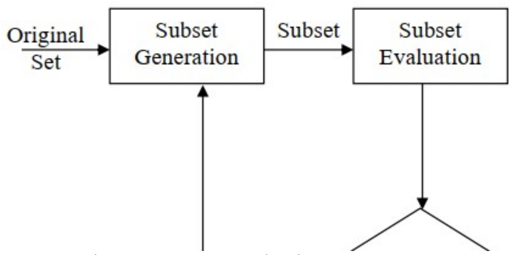


Figure 1. Feature Selection Process

B. Wrapper Subset Evaluation

The "wrapper" subset evaluation of WEKA is the evaluator's implementation of Kohavi (1997). A facility in the WEKA application that is used for feature selection with a wrapper method, where with this application it is possible to carry out a feature selection process using an induction algorithm as needed. This implementation performs a 10-fold cross validation of the training data in evaluating a given subset with respect to the classification algorithm chosen. To minimize bias, this cross validation is carried out in the internal loop of each training fold in the outer cross validation. After the feature set is selected, it is run on the outer loop cross validation [5].

C. Decision Tree C4.5

C4.5 algorithm recursively visits each decision node, selecting the optimal separation, so that no more separation is possible. The C4.5 algorithm step according to [6], to grow a decision tree is as follows:

- Select the attribute for the root node.
- Create a branch for each value of the attribute.
- Cases are divided by branch.
- Repeat the process for each branch until all cases in the branch have the same class

The question is, how can an attribute be selected as the root node? First, calculate the gain ratio of each attribute. The root node is determined by the attribute that has the maximum gain ratio. Gain Ratio is calculated by the following formula [7]:

$$Gain\ Ratio\ (A) = \frac{Gain\ (A)}{SplitInfo\ (A)} \quad \text{.........(1)}$$

Where A is the attribute to which the Gain Ratio will be calculated. Attribute A with maximum Gain Ratio is chosen as the separating attribute. This attribute minimizes the information needed to classify tuples in the resulting partition. Such an approach minimizes the number of tests expected to be needed to classify the tuple given and guarantee that a simple tree if found. To calculate the gain of an attribute, first calculate the entropy of the attribute using the formula:

$$Entropy\ (S) = -\sum_{i=1}^{n} P_i \log_2 P_i \quad \text{.........(2)}$$

Where Pi is the probability that the arbitrary tuple in S belongs to the class Ci and is estimated by ｜Ci; d｜ / ｜D｜. A base 2 log function is used because information is encoded with bits. Entropy (S) is only the average value of information needed to identify the class label of a tuple in S. Now the gain of the Attribute is calculated by the following formula (Beck et al., 2007; Quinlan, 1993):

$$Gain\ (A) = Entropy\ (S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} Entropy\ (S_i) \quad \text{.......(3)}$$

Where, Si = {S1, S2, ... Sn} = Partition of S according to the value of attribute A:
n = Number of attribute A
｜Si｜ = Number of cases in the Si partition
｜S｜ = Total number of cases in S

The Gain Ratio divides the gain with the evaluated separation information, this makes the separation split by many results (Beck et al., 2007; Quinlan, 1993):

$$SplitInfo\ (A) = -\sum_{i=1}^{n} \frac{|S_i|}{|S|} \log_i \frac{|S_i|}{|S|} \quad \text{......... (4)}$$

Split information is a calculation of the weighted average of information using the proportion of cases passed on to each child. when there are cases with unknown results on the split attribute, split information treats this as an additional split direction. this is done to punish the separation made using the case with a lost value. after finding the best separation, the tree continues to grow recursively using the same process.

D. Bagging

To facilitate us in understanding Bagging it will be illustrated as follows. Suppose we are a patient and want to make a diagnosis based on the symptoms experienced. Instead of asking for one doctor, you can choose to ask several doctors. if certain diagnoses occur more than others, you can choose this as the final or best diagnosis. it is the final diagnosis based on the most votes where every doctor gets the same voice. If now replace every doctor with a classifier, then you have the basic idea behind bagging. intuitively, the most votes made by a large group of doctors may be more reliable than the majority votes made by a small group.

Given a set, D, from tuples, Bagging works as follows [6]. For iterations i (I = 1,2,3, ..., k), the training set, the t-tuple is sampled with the replacement of the original tuple set, D. Note that the term bagging is a bootstrap aggregation. Each training set is a bootstrap sample. Because replacement sampling is used, some original D tuples may not be included in Di, while others can occur more than once. The Mi Classifier model is studied for each set of training, Di. To calcify an unknown tuple, X, each classifier, Mi, returns its class prediction, which is counted as one vote and assigns the class with the most votes to X. Bagging can be applied to continuous value predictions by taking the average value of each prediction for the test tuple given.

Bagging Algorithm: Bagging Algorithm [6] creates an ensemble model (Classifier or predictor) for learning schemes where each model is given the same weight prediction.

Input :

- D, a set of training tuples
- K, the number of models in ensemble
- A Learning Scheme (Contoh, decision tree, back propagation, dan lain-lain)

Output : A composite model, M*

Metode:

- For I = 1 to k do // create k models
- Create bootstrap sample, Di by sampling D with replacement
- Use Di to derive a model, Mi
- End for

Untuk menggunakan model komposit pada tupel, X:

- If Classification then
- Let each of the k models classify X and return the majority vote
- If prediction then
- Let each of the k models predict a valur for x and return the average predicted value

Bagging classifiers often have much greater accuracy than a single classifier derived from D, the original training data. It will not be much worse and stronger for noise data effects. Increased accuracy occurs because the composite model reduces the variance of each classifier. For predictions, it is theoretically proven that bagging predictors will considerably increase the accuracy of one predictor derived from D.

E. Adaboost

Boosting is a common method for increasing the accuracy of each learning algorithm that is given. This is an effective method for producing highly accurate prediction rules by combining rough and inaccurate rules of thumb. In this research is the main focus on the AdaBoost algorithm [8], [6].

In AdaBoost, the input includes a D dataset from class labeled d-labels, integers k specifying the number of classifiers in the ensemble and classification learning schemes.

Every tuple in the dataset is weighted. The higher the weight the more influence the theory being studied. Initially, all weights are given the same value 1 / d. The k-repeatedly algorithm. At all times, the Mi model is built on the current dataset obtained by sampling with replacement for the original training dataset D. The framework [6] of this algorithm is as follows:

Algorithm : AdaBoost

Input:

- D, a set of d class-labeled training tuples
- K, the number of rounds
- A classification learning scheme

Output : A composite model

Method :

- Initialize the weight of each tuple in D to 1/d
- For I = 1-k do
- Sample D with replacement according to the tuple weights to obtain Di
- Use training set Di to drive a model, Mi
- Compute the error rate error(Mi) of Mi
- If error(Mi) > 0.5 then
- Reinitialize the weights to 1/d
- Go back to step 3 and try again
- Endif
- Update and normalize the weight of each tuple;
- Endfor

The error rate of Mi, is the sum of the weights of all tuples in Di, that of the tuples in Di which are Mi misclassified:

$$error\,(M_i) = \sum_{j=1}^{d} W_j \times err\,(X_j) \quad \text{……….(5)}$$

Where, err (Xi) = 1, if Xj is misclassified and err (Xj) = 0 otherwise. Then the weights of each tuple are updated so that the weights of the misclassified tuples are increased and the weights of the correctly classified tuples are reduced. This can be done by multiplying the weights of each tuple properly classified by an error (Mi) / (1-error (Mi)). The weights of all tuples are then normalized so that the number is equal to 1. To maintain this limit, the weight of each tuple is divided by the number of new weights. After round K, a composite model will be generated, or classifying ensemble which is then used to classify new data. When a new tuple X comes, it is classified through these steps:

Initialize weight of each class to 0

- For I = 1 - k do
- Get weight wi of classifier Mi
- Get class prediction for X from Mi : c = Mi (Xi)
- Add βi to weight for class c
- Endfor
- Return the class with the largest weight

Bobot Wi untuk masing-masing classifier Mi dihitung dari persamaan berikut :

$$W_i = \log \frac{1 - error\,(M_i)}{error\,(M_i)} \quad \text{……………(6)}$$

III. RESEARCH METHODS

A. Research Framework

In this research, a flow is made that refers to the research framework that aims to make the research gradual and consistent and is an outline of the research steps undertaken in this study.

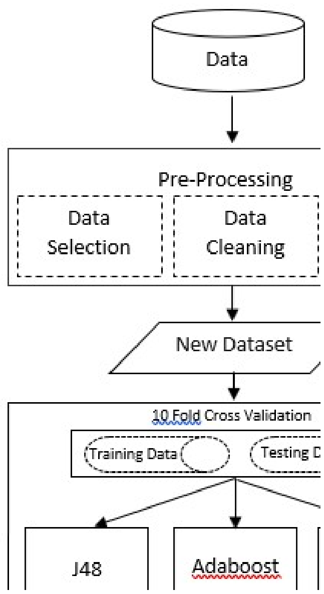


Figure 2. Research Framework

B. Data Preprocessing

- Data Selection

In this study a dataset from the University of California was used. The dataset used is type *. CSV with the name bank.csv. The number of datasets is 4,521 rows with 17 attributes where the last attribute is the data class. This data was downloaded at:

http://archive.ics.uci.edu/ml/datasets/Bank+Marketing

This data is related to the direct marketing campaign of Portuguese banking institutions. Marketing campaigns are based on telephone calls. Often, more than one contact to the same client is needed, to access if the product (bank time deposit) will ('yes') or not ('no') subscribe.

- Data Balancing

The problem of unbalanced data classification is seen when the number of elements in one class is much smaller than the number of elements in another class [9]. If no balance is performed, most machine learning algorithms will predict the dominant class [10]. Can be seen in the dataset used in this study shows that the dataset is not balanced with respect to the class label under study. The majority of the records used in this study, 88%, belong to the "No" class and for the "Yes" class 12%. In this case relating to customers buying deposits or not

- Feature Selection

The feature selection method in this study uses the Wrapper technique. In this method, the classification algorithm is used as part of the selection process in the process. Some relevant attributes will be selected which will hopefully improve the classification results.

C. Performance Measure

1. Performance Classifier
   - Accuracy: Is the ratio of True predictions (positive and negative) to the whole data.
   - Precision: This is a positive positive prediction ratio compared to overall positive predicted results.
   - Recall: Is a true positive prediction ratio compared to overall true positive data.

2. Comparison of ROC curves

The ROC curve summarizes all information provided by the confusion matrix in a visual format. The plot represents the classifier's ability to correctly identify positive labels and negatively identified negative labels. The main advantage of using the ROC curve for the actions mentioned earlier is that the ROC curve provides performance values above all possible thresholds. To better compare the ROC curve generated by the algorithm used, mapping it simultaneously using the WEKA workflow manager.

IV. RESULT AND DISCUSSION

This chapter details the results of the experiment, where the dataset attributes have been selected using the wrapper subset evaluation feature selection method. The first part will explain the relevant features obtained from the feature selection. The second part will examine and compare in detail the results of the comparative use of the Adaboost and Bagging algorithm against the J48 decision tree algorithm. Each method will be described with the performance measures used..

A. Feature Selection

The feature selection method in this study uses the Wrapper technique, in this method, the classification algorithm is used as part of the selection process. The table below shows the results obtained by applying the feature selection method in this experiment.

Because the selection results are sorted by rank, the following arrangement of relevant features is sorted from the best of a total of 16 attributes to 9 attributes.

Table 1. Order of features selected

No	Feature / Attribute
1	Marital
2	Education
3	Housing
4	Loan
5	Contact
6	Day
7	Month
8	Duration
9	Pdays

For the accuracy of the selected attributes using the J48 algorithm there is an increase in accuracy of 0.05% of the results using all the features.

B. Classification Result Performance

After the relevant features are obtained the next process is to classify the three algorithms used by the validation method using 10-cross fold validation. From this test the following results were obtained:

Table 2. Comparative tables of the performance of three classifiers J48, Adaboost and Bagging

Method	Accuracy	AUC	Pre
J48FS	0,859	0,887	0
ADABOOST	0,835	0,904	0

If the results above are illustrated into the graph to facilitate the analysis, you will get a graph like the one below:

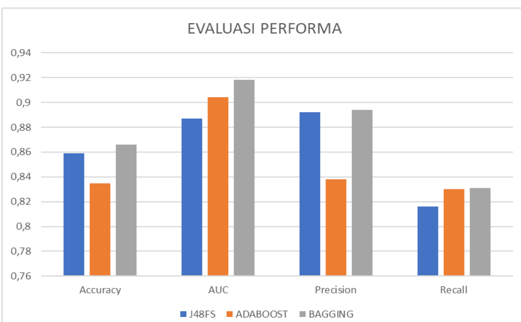


Figure 3. Comparison graph of the performance of three J48, Adaboost and Bagging classifiers

From the graph above it can be seen that the bagging algorithm gets the best results because it excels in all performance evaluations, both Accuracy, AUC, F-Measure, TP Rate, TN Rate, Precision, and Recall, all values show the highest results among the two other algorithms J48 and Adaboost. The Adaboost algorithm which is expected to improve the performance of the J48 base learner does not show good performance and is even worse than the base learner algorithm. It can be seen on the graph of the advantages of Adaboost against J48 only in AUC, TP Rate, and Recall..

C. ROC Curve Evaluation

The ROC curve summarizes all information provided by the confusion matrix in a visual format. The plot represents the classifier's ability to correctly identify positive labels and negatively identified negative labels. The main advantage of using the ROC curve for the actions mentioned earlier is that the ROC curve provides performance values above all possible thresholds. To better compare the ROC curve generated by the algorithm used, mapping it simultaneously using the WEKA workflow manager. For the results of this process can be seen from the picture below that shows the graphical visualization of the three classifiers J48, Adaboost and Bagging.

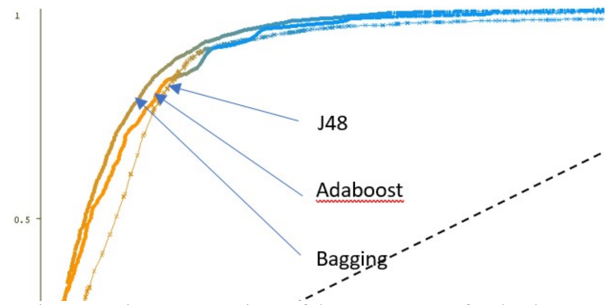


Figure 4. Figure comparison of the ROC curves for the three classifiers

From the graph above it can be concluded that Bagging is the best classifier of this experiment then continued by Adaboost and finally J48. Adaboost on this test gets better results than J48 where in previous tests many testing parameters did not get better results than J48.

From the results of all experiments conducted it can be summarized that the selection of the right features can improve performance which can be seen by using the Wrapper Subset Evaluation feature selection method resulting in nine relevant features where for accuracy values obtain an increase of 0.05% of the dataset using all features. Such an increase is not significant but will at least affect the classification process.

Three data mining classification methods are carried out including J48, Adaboost and Bagging, For Bagging accuracy values get the highest results with 86.6% followed by J48 by 85.9% and the lowest results obtained by Adaboost by 83.5% For AUC values, return Bagging gets the highest score of 91.8% followed by Adaboost of 90.4% while the lowest yield is J48 with a value of 88.7%. For F-Measure, Bagging returned to get the highest result of 0.862 then continued with J48 of 0.852. As for the lowest yield produced by Adaboost of 0.835. ROC graph analysis also shows Bagging is the best classifer among the three methods tested. From the graph it can be seen that the ROC Bagging graphs get the smallest eucledian distance from (0.1) followed by Adaboost and finally J48.

D. Analysis of Experiment Result

1. Data Balancing Effect on Classification Performance

The problem of unbalanced data classification is seen when the number of elements in one class is much smaller than the number of elements in another class [9]. If no balance is performed, most machine learning algorithms will predict the dominant class [10]. Can be seen in the dataset used in this study shows that the dataset is not balanced with respect to the class label under study. The majority of the records used in this study, 88%, belong to the "No" class and to the "Yes" class to 12%. In this case relating to customers buying deposits or not.

To prove this, two tests were conducted, where the first test performed a classification using a unified balanced dataset. Whereas the second test uses a dataset that has been balanced with the Random Under Sampling Technique. For the classification algorithm used Bagging with J48 base classifier, the results of the experiment can be seen from the following table.

Table 3. Comparative table of performance results between balance and un-balance dataset

Un-Balance Dataset Accuracy 90.17%

Cofusion Matrix		Classified As	
		No	Yes
	No	[illegible]	[illegible]

Balance Data[illegible]

Cofusion M[illegible]	

Un-Balance Dataset

Class	FP Rate	Precision	Recall
No	0,516	0,933	0,957
Yes	0.043	0.599	0.484

Balance Dat[illegible]

Class	F[illegible]
No	[illegible]
Yes	[illegible]

It can be seen in the table above that the accuracy value of the unbalanced dataset is higher than the balance dataset of 90.17% compared to 86.65%, but this accuracy does not mean that the classifier shows good performance. We can see in the FP value of the Precission and Recall values, the unbalanced dataset only stands out in one class, the "No" class, while for the "Yes" class, the score is not good.

It is different when compared to a balanced dataset even though the resulting values are not very good, but for the FP Rate, Precision and Recall for each class, the values are almost as good. Therefore based on the results of the analysis of the experiments carried out prove that balancing data can produce good performance for the classification algorithm used..

2. Adaboost Analysis which result in poor performance

According to research conducted by Jeevani Wickramaratna, Sean Holden, and Bernard Buxton in a study entitled "Performance Degradation In Bossting" in 2001, Adaboost was optimally productive when the base classifier used was classified as quite weak (Weak Learner). And usually it will be unproductive or counterproductive to improve the performance of a strong base classifier (Strong Learner) [11]. The J48 algorithm used in this study is included as a strong base classifier (Strong Learner), because the test results show good accuracy results so that Adaboost is not appropriate to use..

To prove that Adaboost provides optimal performance results on weak classifiers, two trials are conducted to test the same dataset (Bank Marketing - UCI) but by using the weak classifier Decision Stump and Naïve Bayes as base learners for Adaboost and Bagging . For the results of both experiments can be seen in the following table:

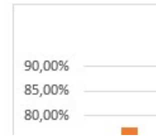
Table 4. Comparative table of performance results between Adaboost and Bagging using the weak classifier Naïve Bayes and Decision Stump

Pengujian 1

Method	Accuracy	Precision	Recall	AUC
Naïve Bayes	77,58%	0,778	0,776	0,866
Adaboost	79,57%	0,797	0,796	0,868
Bagging	77,56%	0,778	0,776	0,866

Pengujian 2

Method	Ac[illegible]
Decision Stump	7[illegible]
Adaboost	7[illegible]
Bagging	7[illegible]



In the test table 1, which uses Naïve Bayes as its basic algorithm, Adaboost can increase the accuracy of its basic algorithm by 2% while Bagging drops by 0.01% from Naïve Bayes. For the value of Precision, Recall, and AUC Adaboost also obtained the best results from the other two algorithms.

Furthermore, for the test table 2, which uses Decision Stump as the basic algorithm, Adaboost can increase the accuracy of the basic algorithm by 5.81%. While Bagging is only able to increase accuracy by 0.09%. For other values such as Precision, Recall, and AUC, Adaboost also gets the best results from all three.

From the results of these two tests, it can be concluded that Adaboost can improve the performance of a base classifier that is classified as weak classifier or weak learner. The weaker the performance produced by a classifier, the better the performance of Adaboost to improve the performance of its base classifier. Returning to previous research where the J48 algorithm gets high performance results and is classified as a strong learner, Adaboost is not appropriate to be applied because the results will not be optimal.

V. CONCLUSION AND SUGGESTION

From the stages of experimentation and testing in this study there are a number of things that can be concluded, including the following:

1. Data balancing at the pre-processing stage needs to be done to avoid the classifier only leaning to one particular class
2. The use of Feature selection can be done to improve classification performance. In this study, it was proven that feature selection can improve accuracy, although not significantly.
3. In this study Bagging algorithm gets the highest performance evaluation results with an accuracy of 86.6% followed by J48 of 85.9% and the lowest results obtained by Adaboost by 83.5%. For the AUC value, Bagging again gets the highest value of 91, 8% followed by Adaboost at 90.4% while the lowest yield was obtained by J48 with a value of 88.7%. ROC graph analysis also shows Bagging is the best classifer among the three methods tested. From the graph it can be seen that the ROC Bagging graph gets the smallest eucledian distance from (0.1) followed by Adaboost and finally J48.

4. The Adaboost method in this study is not appropriate to use because the J48 algorithm used as a base classifier is included in the strong learner. Adaboost is optimally productive when the base classifier used is relatively weak (Weak Learner). And usually it will not be productive or counterproductive to improve the performance of a strong base classifier (Strong Learner). This is evidenced by 2 experiments conducted with Decision Stump and Naïve Bayes as the base classifier, Adaboost can provide optimal results compared to Bagging

Based on the conclusion points of the results of the study described earlier, then there are some recommendations that are recommended for further research studies :

1. Use of different feature selection methods. The feature selection method has been proven to provide improved performance, but for the Wrapper Subset Evaluation feature selection method, the performance obtained does not provide significant results, so it is necessary to try using other feature selection methods such as Correlation-Based Feature Selection.

2. Using a boosting algorithm with J48 as a base classifier does not give good results because J48 is a strong learner and Adaboost does not give good results. Bagging is the right algorithm for this situation. For further research it is necessary to analyze the appropriate classification algorithm so that it can be used with Adaboost.

REFERENCES

[1] Albro, W. and Linsley, C. 2001. "Credit card solicitations: Less for more,". Bank Marketing, 33(2): 11 [Google Scholar]

[2] Kotsiantis, S. (2014). Bagging and boosting variants for handling classifications problems: A survey. The Knowledge Engineering Review, 29(1), 78-100. doi:10.1017/S0269888913000313

[3] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. The Journal of Machine Learning Research, 3, pp. 1157–1182.

[4] Dash, M. & Liu, H. (1997). Feature selection for classification. Intelligent Data Analysis: An International Journal, 1(3), pp. 131–156.

[5] Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In Proc. Twelfth International Conference on Machine Learning. Los Altos, CA: Morgan Kaufmann, pp. 194–202

[6] Han, J. ,Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques, 2rd ed. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers. pp. 336–337. ISBN 1-55860-901-6

[7] Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann.

[8] Yoav Freund, Robert Schapire. (1999). A Short Introduction to Boosting, Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, September, 1999, http://www.yorku.ca/gisweb/eats4400/boost.pdf

[9] Gu, Q., Cai, Z, Zhu, L., & Huang, B. (2008). Data Mining on Imbalanced Data Sets. Proceedings of International Conference on Advanced Computer Theory and Engineering, Phuket, pp. 1020–1024.

[10] Drummond, C., & Holte, R. C. (2005). Severe class imbalance: Why better algorithm aren't the answer. In Machine Learning: ECML 2005, pp. 539–546. Springer

[11] J. Wickramaratna, Sean Holden, & Bernard Buxton.(2001) Performance Degradation in Boosting. MCS 2001 Cambridge (LNCS2096, Springer, 2001)