Modified Balanced Random Forest (MBRF) Algorithm for Classifying Imbalanced Data

Zahra Putri Agusta, Adiwijaya
Surya University, Jln MH Thamrin Km 27
Tangerang, 15143, Indonesia
{zahra.putri@surya.ac.id}

Telkom University, Jl Telekomunikasi no 1
Bandung, 40257, Indonesia
{adiwijaya@telkomuniversity.ac.id}

ABSTRACT: *Customer churn prediction is a method that companies use to anticipate loss in revenue. Some data mining classification techniques can be used to predict customer churn. However, these techniques could become less optimal when faced with imbalanced data conditions. Customer churn data has imbalanced data characteristics, so a process that can handle imbalanced data is required. There are two approaches that can solve these problems, namely sampling method (distribution of training data is modified so that two classes of data can be balanced) and algorithm approach (algorithm process is modified to handle imbalanced data). This paper used the algorithm approach because the consistency of original data distribution will be kept the same as the training data. This will provide more valid data and prediction results that can better represent real conditions. In line with this, we proposed a Modified Balanced Random Forest (MBRF) algorithm as a classification technique to address imbalanced data. The MBRF process changes the process in a Balanced Random Forest by applying an undersampling strategy based on clustering techniques for each data bootstrap decision tree in the Random Forest algorithm. The proposed MBRF method yielded better performance compared to the Balanced Random Forest (BRF) and Random Forest (RF) algorithms, with a sensitivity value or true positive rate (TPR) of 88%, a specificity or true negative rate (TNR) of 94%, and the best AUC accuracy value of 91.65%. Moreover, MBRF also reduced process running time.*

Keywords: Imbalanced Data, Random Forest Algorithm, Balanced Random Forest, Customer Churn, Classification Technique, Machine Learning

Received: 2 September 2019, Revised 10 December 2019, Accepted 23 December 2019

DOI: 10.6025/jic/2020/11/2/41-51

1. Introduction

Customers are a company's main source of revenue [1]. Customer loss caused by strict competition, also known as customer churn, has become an issue in many industries. This has caused business holders to focus more on solving this issue. Customer churn could seriously impact a company and can cause loss of company revenue and profitability. Questions of service quality compared to the competition could also arise from unsatisfied customers [2].

The occurrence of customer churn can be predicted. Companies can use the predicted customer churn rate to take preventive

action [3]. Since the cost of winning a new customer is far greater than the cost of retaining an existing one, many companies have now shifted their focus from customer acquisition to customer retention [4].

Customer churn analysis usually involves the learning and analysis of past customers that have ended relationships with companies. Customer churn data is data with imbalanced class characteristics [5]. This is because, in the real world, there is a higher tendency for customer churn to occur rather than the opposite. Class imbalance happens if one class contains more samples than the other classes [6].

Machine learning classification techniques are known as an efficient way to identify churn, and are therefore often used to predict customer churn. Random Forest is a machine learning classification algorithm that has been proven to perform well in conducting classification compared to other classification algorithms. This is because the algorithm is easy to implement and produces a model with better performance [7]. Random Forest has shown higher performance compared to five other classification algorithms, such as KNN, Naïve-Bayes, C4.5, AdaBoost, and ANN [8].

However, imbalanced data poses a big challenge for machine learning classification techniques. Classifying imbalanced data can decrease the effectiveness of classification techniques, since the classification process always assumes that the data is drawn from the same distribution as the training data and at the same misclassification cost. Therefore, a process for handling imbalanced data for the classification algorithm is required.

Several studies have addressed the issue of imbalanced data. For example, Wu et al. [9] used the Random Forest algorithm in an insurance business problem, where the insurance data had characteristics of class imbalance. The data was analyzed using undersampling with a KKN algorithm approach. The technique reduced the data learning process for the Random Forest.

Khalilia, Chakraborty, and Popescu [10] used Random sub-sampling to handle imbalanced data in predicting disease risk. First, imbalanced medical data was treated, where the training data was divided into multi-sampling data. It was also ensured that each sub-sample data was balanced between the minority and majority. The final result showed that the Random Forest algorithm, which had applied imbalanced data treatment beforehand, produced superior performance compared to the SVM classification algorithm.

One study conducted a data handling technique by combining two sampling data techniques, namely undersampling and SMOTE, to handle the imbalanced data problem in a weighted Random Forest [11]. Another study carried out a combination of RUSBoost and Information Gain as the preprocessing method for churn prediction of imbalanced data [12].

However, some of the above researches handled imbalanced data such that the data handling process would still be in preprocessing before the classification algorithm is executed. Therefore, the direct effect of handling balanced data in the Random Forest algorithm could not be fully observed.

In this research, we discuss methods for handling imbalanced data based on the Random Forest and Balanced Random Forest (BRF) algorithms. The Balanced Random Forest puts imbalanced data handling into an algorithm process. The Balanced Random Forest implements an undersampling technique for every process of decision tree formation in the Random Forest algorithm, and therefore is known as the Balanced Random Forest, as it combines a sampling technique with an ensemble idea. However, BRF has a few weaknesses. For example, its application of a random undersampling process reveals important data that is not used and wasted or some important discriminating instances that may be discarded, all of which could affect the classifier produced.

Therefore, in this paper, we proposed a new approach, namely the Modified Balanced Random Forest (MBRF) algorithm. The method we propose not only improves accuracy but also reduces time complexity. This method changes the process of the Balanced Random Forest algorithm, which takes the data majority. In short, the random undersampling process usually applied in BRF is replaced with an undersampling method based on the clustering technique. The technique of training data distribution is also adjusted to the number of random forest parameters used. We apply the method to real customer churn data in the telecommunication industry i.e. PT Telkom Indonesia and also compare this new algorithm to the Random Forest and Balanced Random Forest algorithms.

In the next section, the materials and methods in this study are explained. The results and discussion are also outlined. At the

end of this paper, we make a conclusion based on the result and discussion and present recommendations for future research.

2. Materials and Methods

In this study, we used the MBRF technique to predict customer churn. This section explains basic algorithm techniques and the flow process of each algorithm in more detail..

2.1. Random Forest

Random Forest is also known as an ensemble learner. Random Forest contains some decision trees, where the final classification result is taken by voting. Random Forest was first introduced by Breiman [7]. It is said to be random because it takes the training data, which is conducted randomly for each tree that will be built, such that the training data given for each tree decision is independent to one another.

The Random Forest algorithm carries out sampling several times—also called the bootstrap method—where for each set of bootstrap sample, a decision tree will be built [13]. Hence, the number of base classifiers formed is equal to the number of samplings done. After all the decision trees are formed, a classification technique is conducted to classify the testing data. The chosen class from the classification prediction result will be the class that has the most votes from the formed decision tree. Random Forest uses a CART algorithm to establish a tree [14].

2.2. Balanced Random Forest

Balanced Random Forest is a modification of the Random Forest, where each iteration in RF takes sample data from minority classes and major classes with the same amount of data as the number of minorities. Therefore, for each formed tree, there are two bootstraps, where the number is equal to the minority class. This technique combines the undersampling majority class process and the ensemble idea [15].

The handling of imbalanced data is carried out at the same time as the classification algorithm execution, so that its direct effect on the performance of the classification algorithm can be observed. Balanced Random Forest is very effective at handling imbalanced data. Classification algorithms, with one of them being the Random Forest algorithm, assumes that the data distribution that will be classified has equal distribution among classes such that any imbalanced data due to the presence of a minority class from another class should be focused on before classification is conducted.

2.3. Modified Balanced Random Forest

In this study, MBRF is proposed to improve the prediction performance of the Random Forest and Balanced Random Forest algorithms. We proposed changes in the process of Balanced Random Forest by taking advantage of other algorithms, namely clustering algorithms.

• Input training data, Ntree

• For i = 1... Ntree

- Split data randomly as Ntree total

Output

- Training Data ($D1$, $D2$, $D3$, Dntree)

• For $i = D1$... Dntree

- Split the data based on class (–) and class (+)

- Calculate total of size data class (+) which is churn class

- Take data to form tree (Boostrap D1)

- Data cluster (–) become as much as the data class size (+), output cluster formed as much as the minority class

- Boostrap (data class (+) is added with centroid data in each cluster)

- Build tree with CART algorithm based on bootstrap data formed

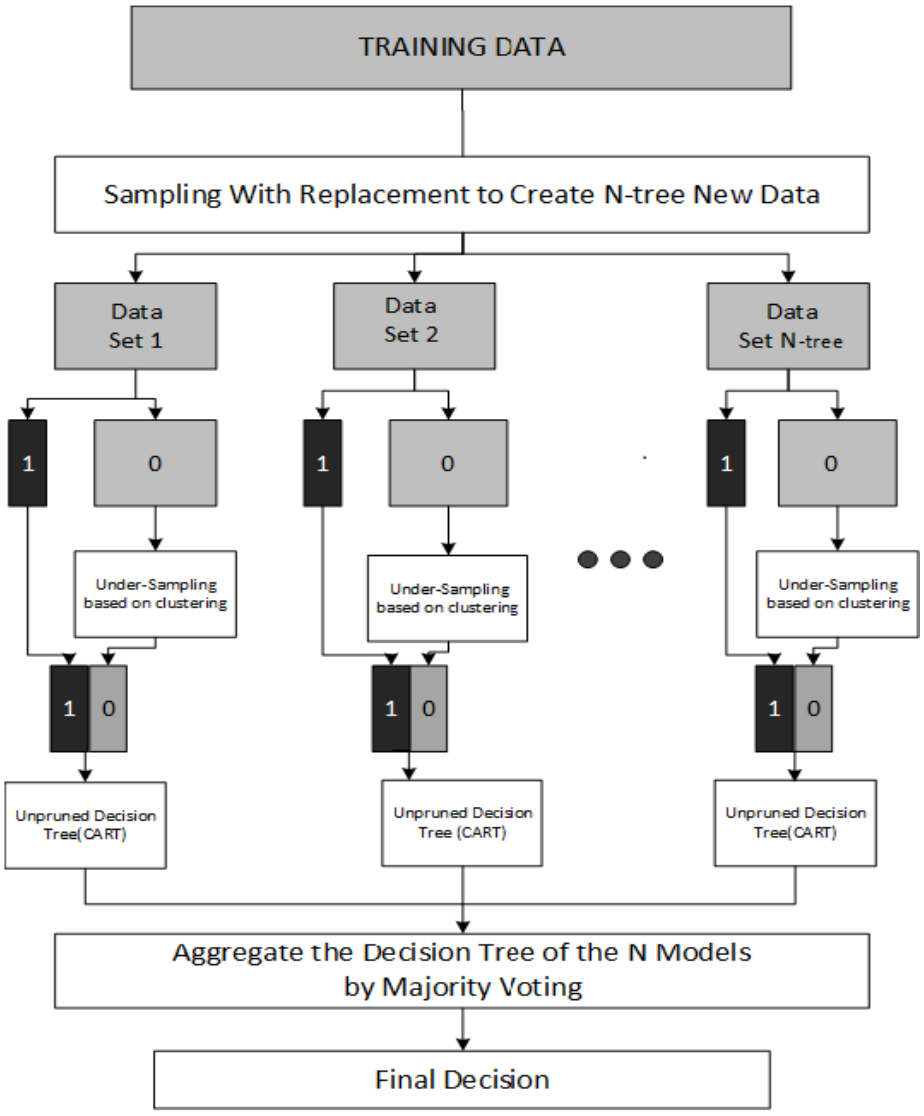

Figure 1. MBRF Flowchart

- Repeat until Dntree

• Output voting result using testing data.

This method begins with taking data from the training data (*D*). The data will be split as many as the input of *N* tree, which becomes *Di* (*D*1, *D*2, *D*3, until *Dntree*). Each *Di* has (*Xi* and *Yi*), where *Xi* is the vector data and *Yi* is the class label. After that, the Balanced Random Forest algorithm is run in which the data is split into two bootstrap samples, *Di*+, *Di*– (the class labeled "+" is churn data and the class labeled "–" is non-churn data).

To form the tree, the two bootstrap samples will be used, in which each bootstrap takes the same total data to that of the data in the minority class. Then, undersampling is conducted on the majority data using a clustering technique, in which the centroids in each cluster will be input into the bootstrap, and the total number of clusters will be the same as the total minority data. In this way, a centroid with the same total as that of the minority class will be produced.

Balanced Random Forest is said to be a process that combines undersampling majority classes and ensemble learning ideas [15].

In the method we propose, under-sampling majority is conducted based on clustering, so undersampling data for building a tree can represent all data, where clustering data will be made into a cluster. Therefore, similar data will be combined such that there will be no data that is not used from down sampling. It also helps to remove the weakness of the undersampling technique, in which there are often various important samples, which are not mentioned in the Random Forest learning process [11]. It is better if every tree formed in the Forest does not have any relationship or high correlation with other trees. Hence, a random process is conducted during classification of bootstrap samples, where the total bootstrap will be equal to the total *N*-tree.

For the clustering process, a *k*-means algorithm was used [16]. *K*-means is a clustering algorithm that is simple and unsupervised. This algorithm can group the given data from a number of clusters *k*. With each data, the cluster has a centroid that will represent the cluster [17]. Each point is inserted into the cluster with the nearest centroid, where the initialization of the number of clusters (*k*) to be formed is the same as class + or class minority.

2.4. Evaluation Measure

In this study, we used sensitivity, specificity, ROC Curve, and AUC to assess the prediction models. Sensitivity and specificity of the effectiveness of an algorithm in one class can be divided into positive and negative, respectively [6]. Based on our case, churn prediction aims to predict the right negative class, so that the higher the sensitivity, the better the prediction of customer churn [18].

Besides sensitivity and specificity, calculation of model accuracy using G-means was also performed for both classes (positive and negative) [6]. This is different from the calculation of general accuracy, which cannot be used for imbalanced data because it has more weight in the majority class (negative) than the minority class (positive). This makes it difficult for the classifier to show in a minority class (negative) [6]. Therefore, if the model wrongly classified the minority class, the accuracy would still be high, whereas G-means would give a more realistic result. G-means measurement avoids inclination towards the majority class (negative class).

Other evaluations used are ROC (Receiver Operating Characteristics) and AUC (Area Under ROC Curve). The benefit of using the ROC curve is that we can easily determine the model with the best performance [19]. Meanwhile, AUC is the area under the ROC curve. AUC summarizes the ROC curve performance into one quantity value. The AUC value is about 0.5–1,where the bigger the AUC value, the better the model [6]. Before performing the model evaluation calculation, we first calculated the total true positive (TP), true negative (TN), false positive (FP) (actual negative but it is predicted positive), and false negative (FN) (actual positive but it is predicted negative) values.

3. Results and Discussion

Before conducting the system examination, the data was first split using the MBRF approach. This is detailed out below:

3.1. Data Set

This study used customer churn data obtained from *PT* Telkom Indonesia. The data amounted to 200387 row data, which consisted of 192863 row non-customer churn data and 7524 row customer churn data. Hence, the churn rate is 3.75%, resulting in imbalanced data and 52 attributes in the data.

3.2. Splitting Training and Testing Data

Data was first divided into training data and testing data. Training data is used for the learning model, while testing data is used to evaluate the model [20]. Two experiments were conducted to get the optimal combination of training data and testing data. First, an experiment to find the optimal proportion of churn and non-churn in the training and testing data was conducted. Then, a second experiment was performed to find the optimal combination for splitting the dataset into training data and testing data.

The first experiment used five combinations of proportioned data: 90%-10%, 80%-20%, 70%-30%, 60%-40%, and 50%-50%, where the data was divided randomly and used in the experiments separately.

Table 1 shows the average result of the five attempts. In this experiment, the AUC calculation was used to observe the changes and differences between the five combinations, where the highest AUC corresponds to the best data proportion. From the combination above, we determined that the best data combination is 50% training data and 50% testing data because while dividing with the same amount of data, the training data and testing data result for the proportion of churn and non-churn

samples represented the proportion of the original dataset.

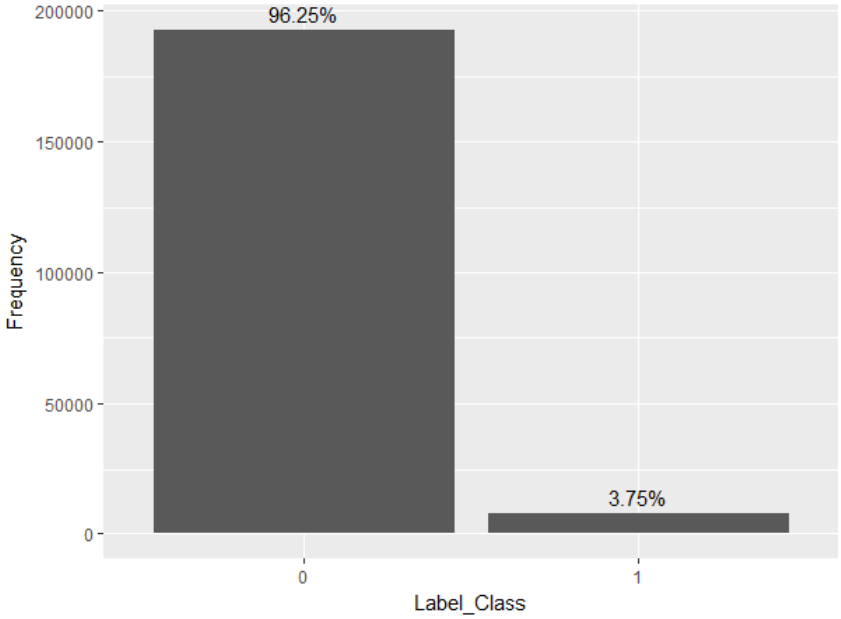

Figure 2. Distribution of non-churn rows (Label_Class: 0) and churn rows (Label_Class: 1) in the Dataset

Exp	Proportion Data	Proportion Training Data		Proportion Testing Data		AUC
		Churn	Not Churn	Churn	Not Churn	
1.1	Train90%, Test10%	3.86%	96.14%	2.75%	97.2%	78.2%
2.1	Train80%, Test20%	0.13%	99.87%	18.2%	81.8%	69%
3.1	Train70%, Test30%	0.15%	99.85%	12.16%	87.8%	82.1%
4.1	Train60%, Test40%	1.76%	98.24%	6.74%	93.2%	80.7%
5.1	Train50%, Test50%	3.75%	96.25%	3.75%	96.2%	88.6%

Table 1. Effect of Proportion Splitting of AUC Rate

Table 1 shows the average result of the five attempts. In this experiment, the AUC calculation was used to observe the changes and differences between the five combinations, where the highest AUC corresponds to the best data proportion. From the combination above, we determined that the best data combination is 50% training data and 50% testing data because while dividing with the same amount of data, the training data and testing data result for the proportion of churn and non-churn samples represented the proportion of the original dataset.

Furthermore, in the second experiment, the divided data was combined based on the optimal proposition obtained in the first experiment. This experiment arranged the splitting of data between training and testing data but the proportion of row data became 3.75% churn and 96.25% non-churn. This proportion will be used in further analysis.

Splitting Combination	Proportion	Row	Churn Row	Not Churn Row
1	Data Train 90%	180348	6764	173584
	Data Test 10%	20039	751	19288
2	Data Train 80%	160310	6012	154298
	Data Test 20%	40077	1502	38575
3	Data Train 70%	140271	5261	135010
	Data Test 30%	60116	2254	57862
4	Data Train 60%	120232	4509	115723
	Data Test 40%	80154	3005	77149
5	Data Train 50%	100194	3757	96437
	Data Test 50%	100193	3757	96437

Table 2. The Training and Testing Data were Resampled using the Optimal Proportion

The result of the examination shows that the best data combination is 90% training data and 10% testing data. This combination resulted in an AUC value of 91.65%. Therefore, this data combination was used for further examination.

Splitting Combination	AUC
1	91.65%
2	89.73%
3	90.01%
4	89.72%
5	88.86%

Table 3. Effect of Proportion Splitting on AUC Rate in the Second Experiment

The optimal proportion was applied so that the churn rate would still depict the actual churn rate proportion and would not result in any bias values in the learning and evaluation process.

3.3. Performance Measurement

This research aims to apply an imbalanced data technique using MBRF (Modified Balanced Random Forest). To examine and determine the improvement from using the proposed model, the proposed model is compared with the Random Forest algorithm and Balanced Random Forest (BRF) algorithm. To obtain optimal parameters, we input 21 parameters for the total trees (10, 15, 20,

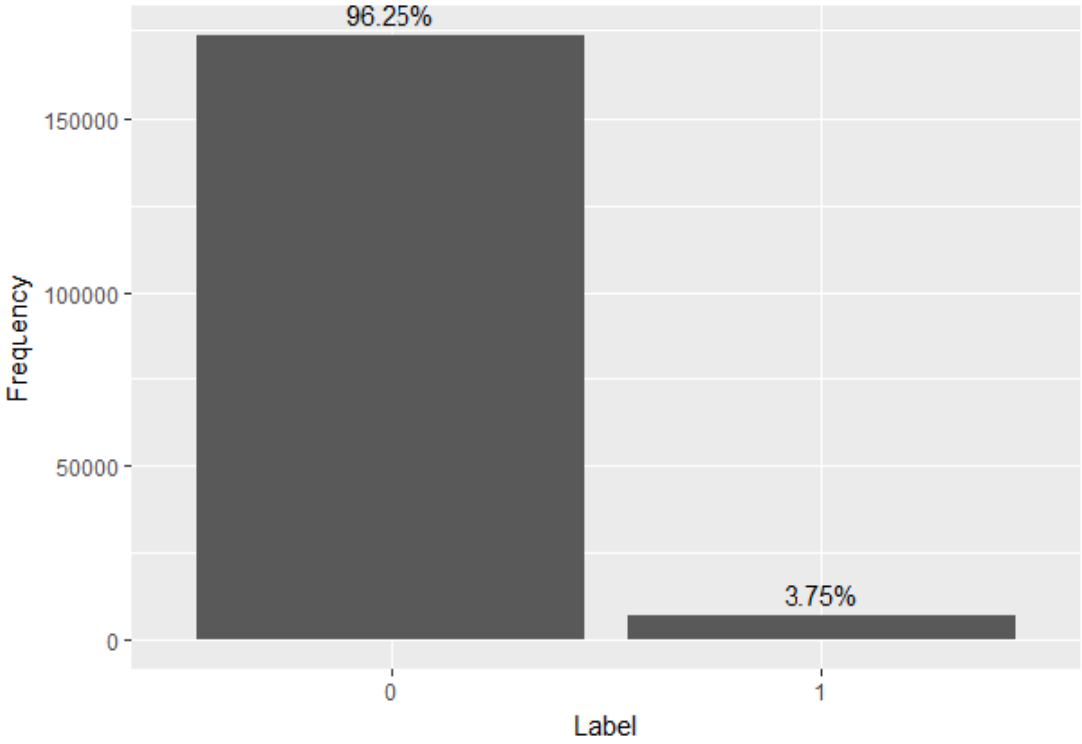

Figure 3. Graph of Training Data Set (Non-Churn Rows: Label 0 and Churn Rows: Label 1) at combination1

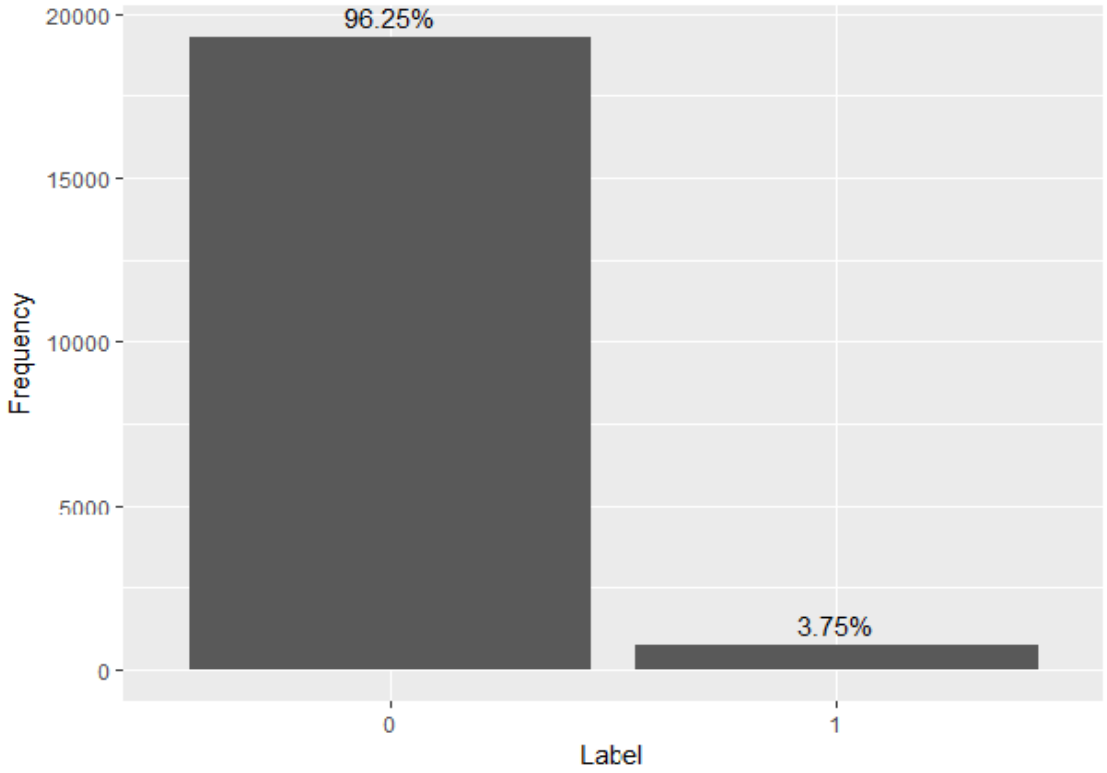

Figure 4. Graph of Testing Data Set (Non-Churn Rows: Label 0 and Churn Rows: Label 1) at combination 1

25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 200, 400, 500, 600, 700, 800, 900). After executing the running process with the 21 parameters, the optimal parameter obtained is a total of 25 trees. However, the parameter of other trees did not give a significant difference on the results. This is because the Random Forest parameter is insensitive; although it did not affect the model accuracy, this parameter did affect time, where the more the parameters of the Random Forest, the longer the time needed for it to run.

Algoritma	**Random Forest (RF)**	**Balanced Random Forest (BRF)**	**Modified Balanced Random Forest (MBRF)**
Sensitifity	14.11%	73.77%	88.64%
Specificity	87.21%	92.98%	94.16%
G-Means	34.99%	82.81%	91.36%
AUC	50.65%	83.37%	91.36%
Running Time	345.45 Sec	42.81 Sec	39.5 Sec

Table 4. Experimental Results for each Method

From Table 4 above, MBRF produced better values than the other two methods—Random Forest and Balanced Random Forest. MBRF yielded a G-means value of 0.9136 or 91.36%, and an AUC value of 0.9136 or 91.36%. In the case of churn prediction, the goal is for the model to attain better recall value. In Table IV above, the MBRF and BRF results for recall value or sensitivity are better than RF. This is because the two methods applied balanced data handling when forming trees. The MBRF results showed better running time than the other algorithms. Therefore, MBRF yielded the most improved churn prediction and the best running time of all the other algorithms.

In Figure 5, the ROC Curve was used to compare the performance of the three methods. The ROC curve consists of x (TPR) and y (1-TNR), in which the point in the curve that forms a line is based on conducting tree tests based on this curve. MBRF, which is marked in yellow, has an AUC value of 1 (bigger under curve area and approach), proving that the proposed MBRF algorithm is better than the BRF and RF algorithms. The ROC curve can be used to determine which method is best [19].

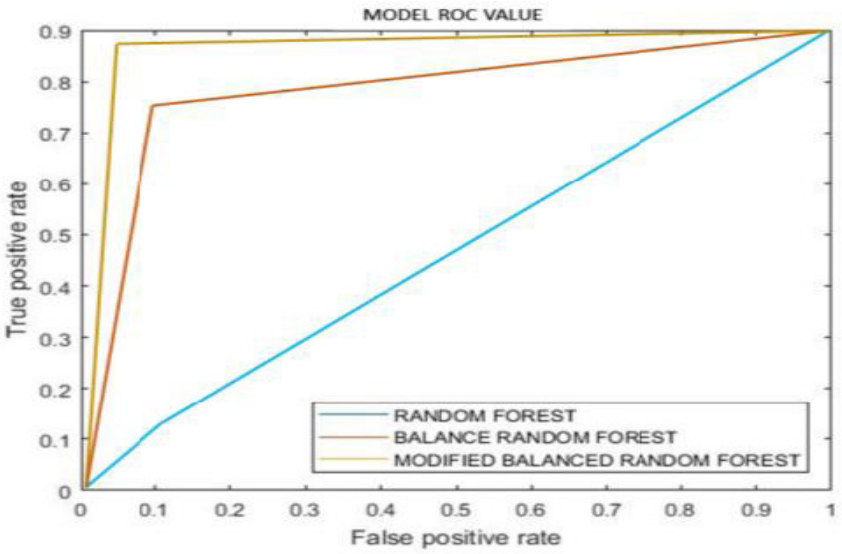

Figure 5. Comparison of ROC Curves between RF, BRF, and MBRF

The tree parameters did not give a significant effect on the result because these could be observed as a point. This means that the AUC value is not too different (almost the same), so the parameter in this method is insensitive towards the result. However, it really affected execution time.

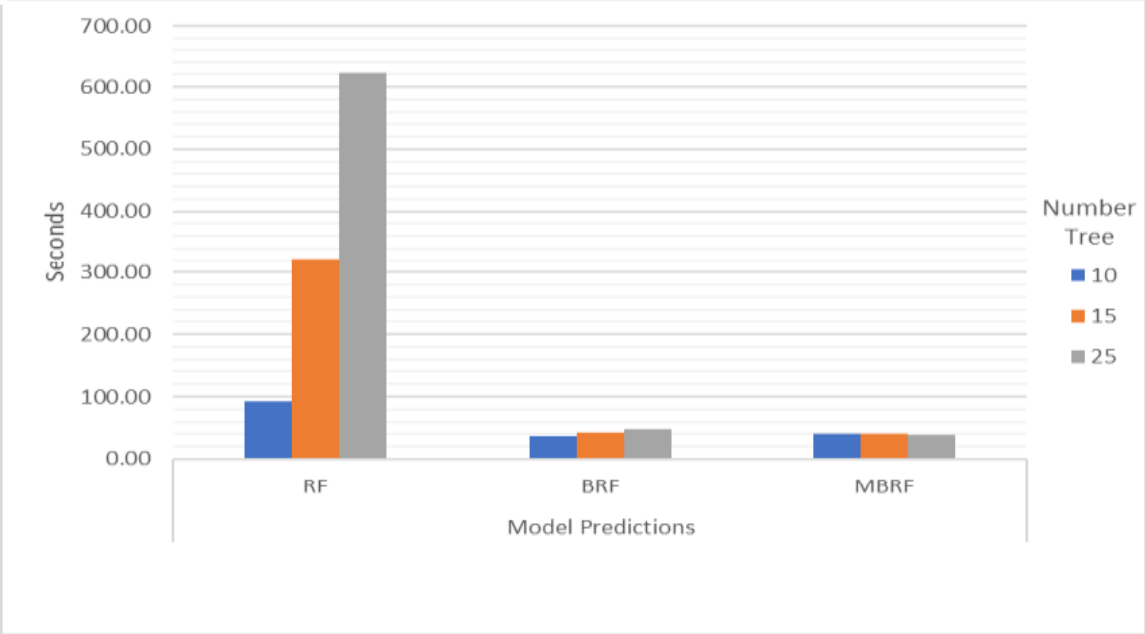

Figure 6. Effect of tree parameter against time

Figure 6 shows that, "the higher the value of the tree in the parameter input, the higher the running time of the Random Forest". Random Forest resulted in the longest running time, with 25 trees resulting in 621 seconds or 10 minutes. Meanwhile, the shortest running time was MBRF with 25 trees in 39.01 seconds.

4. Conclusions

In this study, we proposed a method called the Modified Balanced Random Forest (MBRF) and applied it to imbalanced customer churn data obtained from the telecommunication industry. Based on experiments conducted on the data, the model we proposed yielded better performance compared to the other models (Random Forest and Balanced Random Forest). Moreover, the model also reduced processing time. The Random Forest parameters used did not give different results in each model, because the Random Forest parameter is insensitive. However, this parameter did affect running time, because the more the tree parameters, then the time to form the trees would also take longer.

The optimal proportion for both training data and testing data was a churn sample of 3.25% and non-churn sample of 96.25%. This proportion reflects the actual proportion of the dataset, which has 3.75% churn data and 96.25% non-churn data. Therefore, we can conclude that the combination of training data and testing data must have the same proportion as the real data or data sets to represent the correct population, thus maximizing performance outcome.

Further research should improve upon the effectiveness of the proposed MBRF algorithm in this study because MBRF has limits, where it can only run large data. This is because when the data is split and small data is used, the information contained in the dataset is too insignificant to perform learning and generate models, which means that a small sample data could not be implemented. Future research into this area is also encouraged to explore methods with different domains and different data.

References

[1] Khan, A. A., Jamwal, S., Sepehri, M. M. (2012). Applying Data Mining to Customer Churn Prediction in an Internet Service

Provider, *Int. J. Comput. Appl.,* 2010.

[2] Adebiyi, S. O., Oyatoye, E. O., Amole, B. B. (2016). Relevant Drivers for Customers' Churn and Retention Decision in the Nigerian Mobile Telecommunication Industry, *J. Compet*., 2016.

[3] Umayaparvathi, V., Iyakutti, K. (2016). A Survey on Customer Churn Prediction in Telecom Industry: Datasets, Methods and Metrics, *Int. Res. J. Eng. Technol*., p 2395–56, 2016.

[4] Dalvi, P. K., Khandge, S. K., Deomore, A., Bankar, A., Kanade, V. A. (2016). Analysis of Customer Churn Prediction in Telecom Industry using Decision Trees and Logistic Regression, *Symp. Colossal Data Anal. Netw*., 2016.

[5] Sonak, A., Patankar, R. A. (2015). *A Survey on Methods to Handle Imbalance Dataset*., 4, (11), p 338–343.

[6] Bekkar, M., Djemaa, H. K., Alitouche, T. A. (2013). Evaluation measures for models assessment over imbalanced data sets, *J. Inf. Eng. Appl.,* 2013.

[7] Breiman, L. (2001). Random forests, Mach. Learn., 2001.

[8] Esteves, G., and Mendes-Moreira, J. (2016). Churn perdiction in the telecom business, in 2016 11th International Conference on Digital Information Management, ICDIM 2016, 2016.

[9] Wu, Z., Lin, W., Zhang, Z., Wen, A., Lin, L. (2017). An Ensemble Random Forest Algorithm for Insurance Big Data Analysis, in Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017.

[10] Khalilia, M., Chakraborty, S., Popescu, M. (2011). Predicting disease risks from highly imbalanced data using random forest,BMC Med. Inform. Decis. Mak., 2011.

[11] Effendy, V., Baizal, Z. K. a. (2014). Handling imbalanced data in customer churn prediction using combined sampling and weighted random forest, 2014 2nd *Int. Conf. Inf. Commun. Technol*., 2014.

[12] Dwiyanti, E., Adiwijaya, Ardiyanti, A. (2017). Handling imbalanced data in churn prediction using RUSBoost and feature selection (Case study: PT. Telekomunikasi Indonesia regional 7), *in Advances in Intelligent Systems and Computing*, 2017.

[13] Kobyli Dski, A., Przepiórkowski, A. (2008). Definition extraction with balanced random forests, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008.

[14] Singh, S., Gupta. (2014). Comparative study ID3, cart and C4 . 5 Decision tree algorithm: a survey, *Int. J. Adv. Inf. Sci. Technol.,* 2014.

[15] Chen, C., Liaw, A., Breiman, L. (2004). Using random forest to learn imbalanced data, Univ. California, Berkeley, 2004.

[16] Ghosh, S., Kumar, S. (2013). Comparative Analysis of K-Means and Fuzzy C-Means Algorithms, *Int. J. Adv. Comput. Sci. Appl*..

[17] Oyelade, O. J., Oladipupo, O. O., Obagbuwa, I. C. (2010). Application of k Means Clustering algorithm for prediction of Students Academic Performance, *Int. J. Comput. Sci. Inf. Secur*., 2010.

[18] Weng, C. G., Poon, J. (2008). A new evaluation measure for imbalanced datasets, *Conf. Res. Pract. Inf. Technol. Ser*., 2008.

[19] Fawcett, T. (2006). An introduction to ROC analysis, *Pattern Recognit. Lett*., 2006.

[20] Kotsiantis, S. B., Kanellopoulos, D., Pintelas, P. E. (2006). Data preprocessing for supervised learning, *Int. J. Comput. Sci.,* 2006.