Spring 2017

A Proposed Frequency-Based Feature Selection Method for Cancer Classification

Yi Pan
Western Kentucky University, yi.pan169@topper.wku.edu

A PROPOSED FREQUENCY-BASED
FEATURE SELECTION METHOD
FOR CANCER CLASSIFICATION

A Thesis
Presented to
The Faculty of the Department of Computer Science
Western Kentucky University
Bowling Green, Kentucky

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

By
Yi Pan

May 2017

A PROPOSED FREQUENCY-BASED
FEATURE SELECTION METHOD
FOR CANCER CLASSIFICATION

Date Recommended 04/21/2017

Huanjing Wang, Director of Thesis

Guangming Xing

Zhonghang Xia

Dean, Graduate School   Date 4/24/17

I dedicate this thesis to my son, Leo, and my wife, Yunxin Kuang, who are a great inspiration to me.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

A PROPOSED FREQUENCY-BASED FEATURE SELECTION METHOD FOR CANCER CLASSIFICATION

Yi Pan May 2017 33 Pages

Directed by: Huanjing Wang, Guangming Xing and Zhonghang Xia

Department of Computer Science Western Kentucky University

Feature selection method is becoming an essential procedure in data pre-processing step. The feature selection problem can affect the efficiency and accuracy of classification models. Therefore, it also relates to whether a classification model can have a reliable performance. In this study, we compared an original feature selection method and a proposed frequency-based feature selection method with four classification models and three filter-based ranking techniques using a cancer dataset. The proposed method was implemented in WEKA which is an open source software. The performance is evaluated by two evaluation methods: Recall and Receiver Operating Characteristic (ROC). Finally, we found the frequency-based feature selection method performed better than the original ranking method.

1. INTRODUCTION

In our daily life, we are always surrounded by a variety of big data. How to efficiently find useful information in the collected data is becoming more and more important, such as the advertising and financial industry. Therefore, data mining technology becomes more and more necessary. Data mining technology includes many aspects, like classification, clustering, association rule discovery, sequential pattern discovery, regression and deviation detection [4]. Among these techniques, data pre-processing plays an essential role, it involves several tasks: data cleaning, data integration, data reduction, data transformation and data discretization [4]. The original feature selection is to rank all the attributes through the particular ranking method to select the valuable attributes and delete those attributes with lower scores. Through this approach, feature selection technique can definitely filter out the irrelevant attributes so that it can improve the accuracy and efficiency of the model.

In this study, we proposed a new frequency-based feature selection method. We have chosen several different classification models and ranking methods to examine whether this new method can effectively improve the accuracy rate. We studied three filter-based ranking methods including Gain Ratio (GR), OneR, and Symmetric Uncertainty (SU). Four classification models were presented in this research: Naive Bayes (NB), J48 decision tree (J48), Sequential Minimal Optimization (SMO), and IBK. Also we used two model evaluation metrics: Recall and Receiver Operating Characteristic (ROC). We applied the ranking methods for each classification models on the cancer inhibitor dataset. In order to compare the performance, we implemented the proposed method in an open source software, WEKA [3]. Therefore, based on the experimental results the frequency-based feature selection method performed a better result with NB, J48 and SMO classification models.

This paper including the following part: Section 2 represents a review of related work; Section 3 introduces some detailed information about filter-based feature ranking methods, classifiers, model evaluation and k-fold cross-validation; Section 4 offers the experimental design, the frequency-based feature seletion method, and the results analysis; Section 5 gives a summary of this study and the discussion of future work.

2. RELATED WORK

In the real dataset, there are many unfavorable factors that affect the accuracy of the model, like the missing value, noisy data, redundant features and irrelevant features [4]. So this requires us to reduce these interference items as much as possible in the data pre-processing. Feature selection is a technique that is applied to data mining and machine learning, and it is used to optimize the dataset. Its goal is to shrink the dataset and find the optimal subset that has the closest result like all attributes are included [4].

Das [16] claims that the good feature subset is either a good predictor of the class by itself or a good predictor of the class when taken together with some other subset of the feature in the set. Usually, feature selection algorithm has two ways: forward selection and backward elimination [16]. Guyon et al. [23] conclude that the feature selection algorithm includes the variable ranking as its simplicity and excellent empirical success. A single attribute may be useless by itself, but it may provide a significant power when combining with other attributes [23]. Saeys [15] also points out that the general issue of the traditional feature selection method is each feature considered independently.

Wang et al. [1] provide an empirical study by using six filter-based feature ranking methods and five different classifiers. They offer an assessment and comparison between six ranking methods with five different classifiers. Ding et al. [13] also used two classifiers to compare the performance and perform a shrink process to the microarray gene data.

3. MOTHODOLOGY

3.1 Filter-based Feature Ranking Methods

Feature ranking method can provide a list that contains all the scores for each attribute. Based on this list, we can easily find out the attributes with high scores. Generally, we prefer to select the attributes with a higher score rather than the one with a lower score. In this study, we use three different filter-based feature ranking methods include Gain Ratio (GR), OneR, and Symmetric Uncertainty (SU). Most of them are commonly used techniques [1].

Novakovic [2] mentioned that entropy is a measure which is widely used in the information theory. It describes the purity of an arbitrary collection of examples. The entropy of Y is

$$H(Y) = -\sum_{y \in Y} p_i \log_2(p_i) \tag{1}$$

where p_i is the probability of a random tuple in Y is part of the class C_i. Information Gain (IG), Gain Ratio (GR), and Symmetrical Uncertainty (SU) are based on the idea of entropy [1].

Information Gain is a measuring method that compares the amount of change of two situations: one is the data including the particular attribute, the other is the data excluding that attribute. If the difference is huge, it means that attribute plays a significant role in the dataset. Witten et al. [3] notes that Information Gain discretizes the attributes by using the MDL-based discretization method. Wang et al. [1] claim that the disadvantage of IG has a tendency to choose the attribute with multiple possibilities. However, the Gain Ratio (GR) improved the Information Gain by using "split information" value which normalizes the Information Gain [4]. The Gain Ratio is defined as

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}. \qquad (2)$$

OneR applies a simple measuring method from OneR classifier, and it has two parameters, one is the number of folds as it has internal cross-validation, the other parameter is the minimum bucket size [3].

The Symmetrical Uncertainty (SU) [6] defined as

$$SU(X,Y) = 2\times\frac{Gain}{H(X)+H(Y)}. \qquad (3)$$

In this formula, $H(X)$ is the entropy of X. Symmetrical Uncertainty improves the Information Gain's backward by using more values and normalizes the value to the range [0,1] [7]. SU value has two primary purposes: (1) remove the features that SU is less than the threshold and (2) it can calculate the weight for every feature [6].

3.2 Classifiers

In this study, classification models are created by four major classifier algorithms: Naive Bayes (NB), J48 decision tree (J48), Sequential Minimal Optimization (SMO), and IBK. The reason why these algorithms were chosen is that most of them are commonly used in data mining and machine learning. In this case, default setting and parameters are used if there is no significant performance change.

The Naive Bayes (NB) classifier is a statistical classifier based on Bayes' theorem [8]. It's the simplest probability model in the Bayesian network and one the most efficient learning algorithm in machine learning [9]. Leung (2007) concludes that (1) NB classifier has the similar performance with the decision tree and selected neural network classifiers and (2) NB classifier can show an excellent accuracy and speed when the dataset is large [8].

The J48 decision tree classifier is one of the most useful ways to solve the classification problems, and it will build a binary tree model during the classification process [10]. The J48 algorithm is an expansion of the ID3 algorithm, however there are more features of J48 like accounting for missing values, decision trees pruning, continuous attribute value ranges, derivation of rules, etc [11].

Support Vector Machine (SVM) also called Sequential Minimal Optimization (SMO) in WEKA [3]. It can provide a secure and accurate way among all famous algorithms. It is a straightforward and efficient algorithm for solving the quadratic programming problem in support vector machines [12]. The central approach of SVM is to find the maximum marginal hyper-plane [3].

The IBK classifier also called K-Nearest Neighbors (KNN) in WEKA which is widely used in the area of pattern recognition and described in the early 1950s [4]. The IBK classifier provides several options to speed up the task to find nearest neighbors [3]. It has two steps: (1) determine the nearest neighbors and (2) determine the primary class that those neighbors used [14]. When given an unknown tuple, the IBK classifier will be looking for k training tuples that are close to the unknown tuple, and then these k training tuples are the k “nearest neighbors” of the unknown tuple [4]. The default parameter k is one, in this study we use five instead.

3.3 Model Evaluation

Model evaluation is always used to estimate how accurately the classifier can predict [4]. Basically, the classification model is a mapping from instances to predicted classes [17]. In this case, we considering the classification problems using two classes, therefore, each instance is mapped to two collections which are {positive, negative} and {Y, N}. The first collection indicates the actual value and the second one means the prediction, thus, in the two-class classification problem there are four possible outcomes: the TP (true positive) means the actual label is "positive" and the prediction result is correct. By contrast, if the actual label is "positive" and the prediction result is wrong then we call it FN (false negative). If the actual label is "negative" and the prediction is correct then we name it TN (true negative); if the actual label is "negative" and the prediction is wrong that is FP (false positive).

		Actual class	
		Positive	**Negative**
Predicted class	Y	TP	FP
	N	FN	TN

Figure 1. Two-class classification matrix

Figure 1. shows a two-by-two confusion matrix, this matrix is the basic of many model evaluations. The term true positive rate (TPR) is generally understood to mean sensitivity which is the proportion of true positive and all the positive tuples, defined as $\frac{TP}{TP+FN}$ [4]. The term false positive rate (FPR) refers to specificity which means the rate of negative tuples that are correctly classified, defined as $\frac{FP}{FP+TN}$ [4]. In this study, we choose several famous and useful measure methods for the evaluation.

Recall or Sensitivity is used to describe the ratio that the real positive tuples are correctly classified [5]. It is the same as true positive rate (TPR).

Receiver operating characteristic (ROC) has long been used, and it is a good way to measuring the performance of a classifier [18]. It is defined as the relationship between the TPR and the FPR. The value of ROC always from 0 to 1 and the ideal result of a perfect classifier is 1, therefore, the goal for ROC is to be in the upper-left-hand corner [19].

3.4 K -Fold Cross-Validation

In the model selection problems, one valuable method is cross-validation (CV) [20]. Basically, the process of CV is to split the dataset into two parts once or several times, for each time one part of the dataset is used as training data to build a particular model, the remaining part is used as test data to determine the performance of the model. The cv method circumvents the overfitting problem as the training set is independent from the test set [20]. Random sampling should be used in CV to make sure each class can properly appear in both training set and test set [3]. 10-fold cross-validation is always used to separate the data into ten parts, nine of them as training data, the other one as test data. 10-fold CV has been chosen as the best way to estimate the error based on the comprehensive examines on the different datasets and learning algorithms [3]. Witten et al. (2016) also suggested that only one 10-fold CV is not reliable enough to estimate the result [3]. In this study, a 10-round 10-fold CV is applied to compute the average results. That means the 10-fold CV runs ten times. Note, every time before the dataset is split, the data need to be randomized.

4. EXPERIMENTS

4.1 Dataset

A cancer inhibitor dataset is used in this study, the original owner, Kelvin Xiao provided this dataset on the Kaggle which is a website for the data science [21]. This dataset collected more than 7,000 fingerprints of small molecules which are collected from chEMBL Database, the molecules with IC50 lower than 10 uM always are the cancer inhibitors, otherwise non-inhibitors [21]. Here the owner used RDKit which is an open source toolkit for cheminformatics written in Python. The RDKit can generate a bit vector or count vector for a molecule, typically it will extract the features from a molecule and hash it [22]. Figure 2 shows how RDKit generates fingerprints for a molecular. There are eight different kinds of protein kinases in this dataset, in this study, we focus on the Cyclin-dependent kinase 2 (cdk2). The cdk2 dataset has 8192 attributes and 1 class attribute with 1558 instances. Table 1 shows the details of the cdk2 dataset.

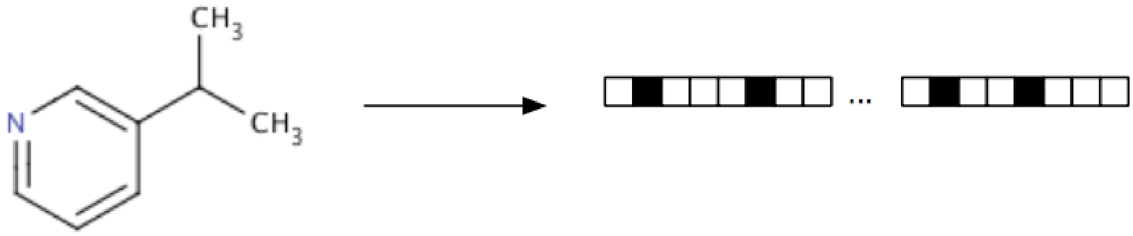


Figure 2. RDKit working process [22]

Dataset Name	Number of Instances	Number of Attributes	Number of Classes	Class Distribution
Cdk2	1558	8193	2	Inhibitor: 459 Non-inhibitor: 1099

Table 1. Cdk2 dataset

4.2 WEKA

In this study, we have used WEKA which is a software that contains many algorithms for data mining tasks. It contains functions for data pre-processing, classification, regression, clustering, association rules and visualization. WEKA is developed by the machine learning group at the University of Waikato. Also, WEKA is an open source software, therefore, in this study, we have used several functions of WEKA in the Java program. We also implemented our proposed method in WEKA. Here, some key methods are listed:

(1) Evaluation.evaluateModel, (2) AttributeSelection.setEvaluator, and (3) Filter.useFilter.

4.3 Experimental Design

In this research, our goal is to find the more accurate and efficient feature selection method by comparing two different approaches: the original feature selection method and the proposed frequency-based feature selection method. The experiment is designed as follows. First of all, we select four particular classification models: NB, J48, SMO and IBK. Secondly, we pick three ranking methods: GR, SU and OneR. After that, we decide to use Recall and Receiver Operating Characteristic (ROC) as the evaluation metrics. Also, the k-fold cross-validation was applied in the experiment.

In this study, to compute the performance based on the original feature selection method including 5 steps: (1) remove the useless attributes by using the WEKA built-in method, (2) apply the ranking method, (3) choose top 50, 80, 200, 500, 800, 1000, 1200, 2000 attributes (threshold) separately based on the score from the previous step, (4) build the classification models, (5) export the result for each fold and each threshold.

Similarly, the frequency-based feature selection method also has 4 steps: (1) remove the useless attributes by using the WEKA built-in method, (2) choose top 50, 80, 200, 500, 800, 1000, 1200, 2000 attributes (threshold) separately based on the frequency from the original method as we mentioned before, (3) build the classification model, (4) export the result for each fold and each threshold. Table 2 and Table 3 show the detail steps for each method.

The main difference between these two methods is the original method pick the attributes based on the ranking list from a particular ranking method. However, the frequency-based method selects the attributes based on the frequency from the frequency list which is used to calculate the accumulative amount for each attributes. That means the original method may choose different attributes for each fold and the frequency-based method will pick top n attributes based on the number of appearances according to the original method.

In order to apply these two approaches, we implemented the proposed feature selection method in WEKA, we also modified WEKA to get the result for each fold and threshold. The original feature selection methods are build-in functions in WEKA. Basically, the core part of this program is the two loops: the outside loop is used to apply the different thresholds, and the inner loop is used to perform the 10-fold CV. We set up the classification model and the ranking method at the beginning of the program. Finally, this program will generate an Excel file that contains the result.

The schema of original feature selection method

```
FOR each classification model
    FOR each ranking method
        FOR each threshold x in (50, 80, …2000)
            FOR each fold 1 to 10
                Perform ranking method, get ranking list
                Select top x features
                Build classification model
                Get performance measure
            ENDFOR
        Average performance measure
        ENDFOR
    ENDFOR
ENDFOR
```

Table 2. The schema of original feature selection method

The schema of frequency-based feature selection method

```
FOR each classification model
    FOR each ranking method
        FOR each threshold x in (50, 80, …2000)
            Create frequency list
            FOR each fold 1 to 10
                Perform ranking method, get ranking list
                Select top x features, save into frequency list
            ENDFOR
            FOR each fold 1 to 10
                Select top x features from frequency list
                Build classification model
                Get performance measure
            ENDFOR
        Average performance measure
        ENDFOR
    ENDFOR
ENDFOR
```

Table 3. The schema of frequency-based feature selection method

4.4 Experiments and Result Analysis

For the purpose of analysising the results, we build classification model using four classifiers with three filter-based ranking methods on the cancer dataset. In order to determine the performance, we built 240 classification models (2 × 4 classifers × 3 ranking methods × 10 folds). Experimental results are presented in the tables (Table 4-9) and figures (Figure 3-26). In each figure, there are two lines indicate the result by using two different methods. Also the table includes all the details from the test. Note that the data in the table represent the average result based on the 10-fold cross-validation.

4.4.1 Case Study I

First of all, we used Naïve Bayes classification models and three ranking methods: GR, SU and OneR. In this case, we chose 50, 80, 200, 500, 800, 1000, 1200, 2000 (threshold) as the number of attributes. As we can see, Figure 3 shows the Recall of NB-SU. When 50 attributes are selected, both methods are effective, especially the frequency-based method. After 50 attributes, both of them are descreasing until 200 attributes are selected. Then the increase is very slow. But the frequency-based way still better than the original way. Figure 4 presents the result of ROC. There are no big differences between two methods, the frequency-based method just moderately enriches the original method, both of them are gradually growing up.

Secondly, the Figure 5 and Figure 6 compared the Recall and ROC based on NB-GR. In Figure 5, they all reached the bottom when 80 attributes are selected. Later, both methods are increasing significantly. We can see no matter how many attributes are chosen, the frequency-based method still got a higher score than the original way.

Similarly, Figure 6, Figure 7 and Figure 8 also represent two lines that have the same trend, the frequency-based method still got a higher score.

The result of NB is shown in Table 4. Interestingly, for those ranking methods, both ways have a very similar shape, but the frequency-based way got a better result in the most situations. In the most cases, the performance can increase 3% to 10% by using the frequency-based method.

Classifier	**NaiveBayes**											
Method **Feature Subset**	**Recall**						**ROC**					
	GR		**SU**		**OneR**		**GR**		**SU**		**OneR**	
	Orignal	**Frequency**	**Orignal**	**Frequency**	**Orignal**	**Frequency**	**Orignal**	**Frequency**	**Orignal**	**Frequency**	**Orignal**	**Frequency**
Top 50 attributes	0.52	0.51	0.79	0.89	0.45	0.50	0.73	0.77	0.82	0.87	0.80	0.84
Top 80 attributes	0.44	0.50	0.82	0.87	0.48	0.54	0.76	0.83	0.85	0.87	0.84	0.86
Top 200 attributes	0.54	0.59	0.77	0.82	0.53	0.57	0.83	0.91	0.86	0.89	0.84	0.88
Top 500 attributes	0.61	0.65	0.79	0.83	0.57	0.65	0.87	0.94	0.90	0.91	0.89	0.91
Top 800 attributes	0.67	0.70	0.82	0.85	0.67	0.72	0.91	0.95	0.91	0.92	0.91	0.93
Top 1000 attributes	0.72	0.76	0.81	0.85	0.66	0.73	0.92	0.95	0.92	0.93	0.91	0.93
Top 1200 attributes	0.74	0.82	0.82	0.85	0.67	0.74	0.91	0.95	0.91	0.93	0.90	0.93
Top 2000 attributes	0.78	0.86	0.81	0.86	0.69	0.75	0.92	0.96	0.92	0.94	0.92	0.94

Table 4. The performance of NB

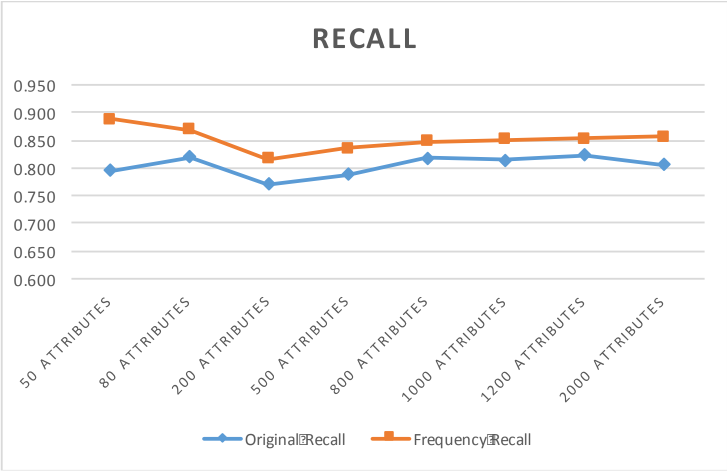


Figure 3. NB-SU Recall

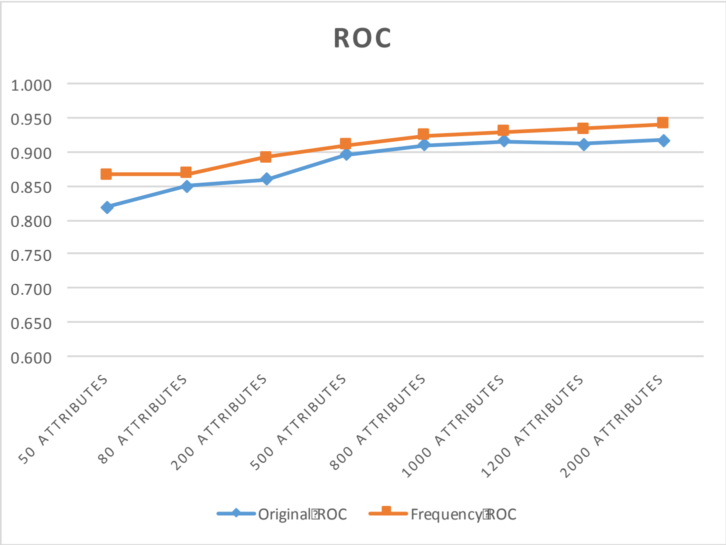


Figure 4. NB-SU ROC

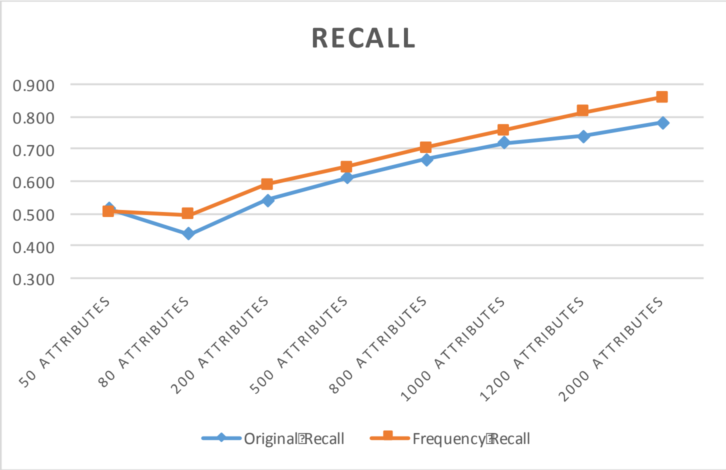


Figure 5. NB-GR Recall

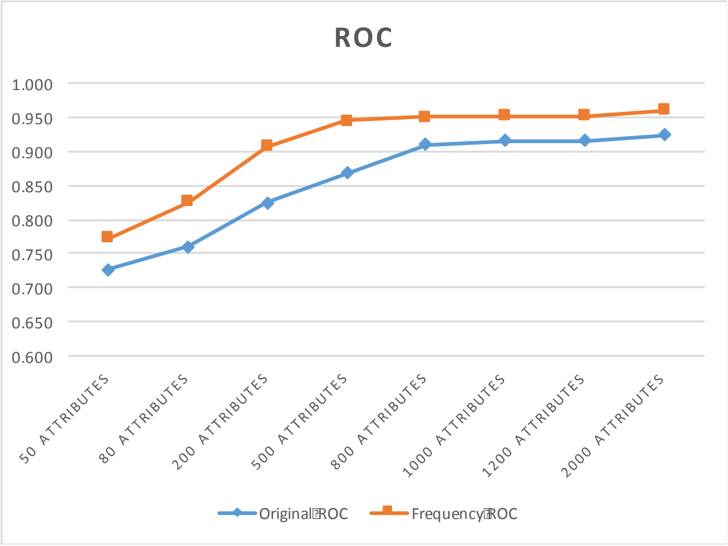


Figure 6. NB-GR ROC

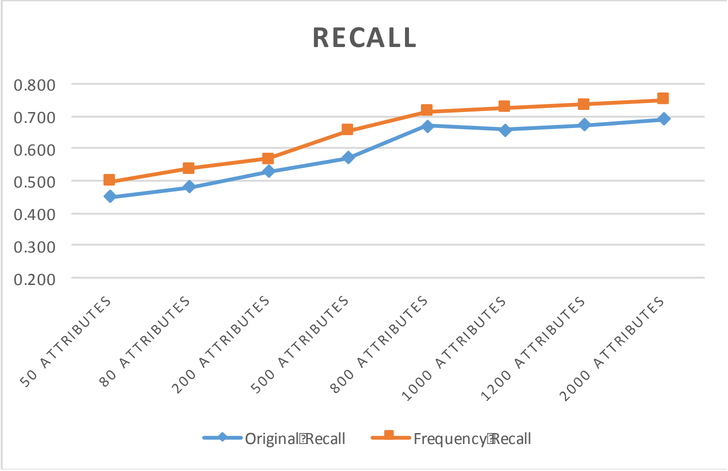


Figure 7. NB-OneR Recall

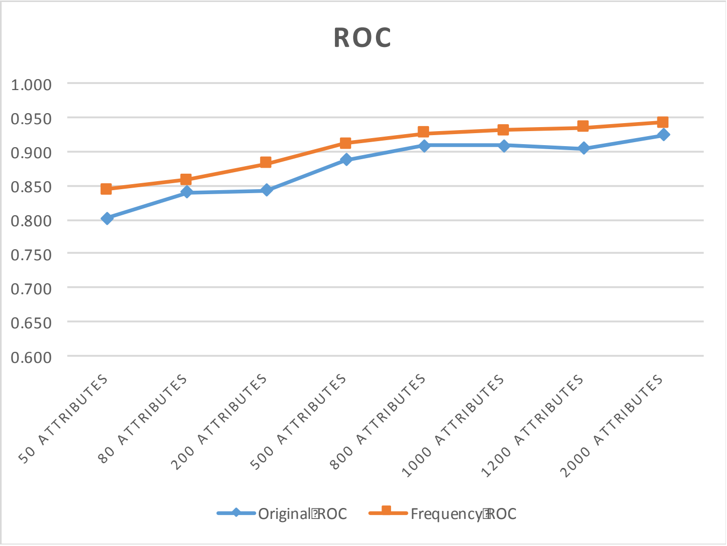


Figure 8. NB-OneR ROC

4.4.2 Case Study II

In this case, we built the classification models using the J48 decision tree model. We also chose the threshold from 50 to 2000. Figure 9 (J48-GR Recall) shows that the frequency-based method gets a better performance than the original method in most conditions. But they are very close to each other. However, in Figure 10 (J48-GR ROC), we can see the frequency-based method significantly upgraded the original method. For Figure 11 (J48-SU Recall) and Figure 12 (J48-SU Recall) the frequency-based method also achieved the goal, but the performance is very close. Figure 13 (J48-OneR Recall) and 14 (J48-OneR ROC) are also supporting the frequency-based method. In Table 5, comparing two model evaluation methods, the average increase is 6%.

Classifier	J48											
Method Feature Subset	Recall						ROC					
	GR		SU		OneR		GR		SU		OneR	
	Orignal	Frequency	Orignal	Frequency	Orignal	Frequency	Orignal	Frequency	Orignal	Frequency	Orignal	Frequency
Top 50 attributes	0.45	0.49	0.56	0.60	0.47	0.53	0.69	0.74	0.81	0.85	0.72	0.76
Top 80 attributes	0.48	0.54	0.58	0.60	0.48	0.56	0.71	0.77	0.85	0.86	0.74	0.77
Top 200 attributes	0.52	0.54	0.60	0.64	0.51	0.59	0.72	0.77	0.86	0.87	0.75	0.79
Top 500 attributes	0.59	0.63	0.58	0.62	0.57	0.60	0.74	0.81	0.80	0.82	0.76	0.78
Top 800 attributes	0.60	0.62	0.61	0.63	0.63	0.66	0.76	0.80	0.77	0.80	0.78	0.81
Top 1000 attributes	0.63	0.68	0.60	0.68	0.63	0.68	0.76	0.80	0.76	0.82	0.77	0.80
Top 1200 attributes	0.67	0.68	0.67	0.70	0.66	0.68	0.78	0.80	0.78	0.83	0.78	0.81
Top 2000 attributes	0.66	0.69	0.65	0.68	0.65	0.69	0.77	0.81	0.78	0.81	0.77	0.80

Table 5. The performance of J48

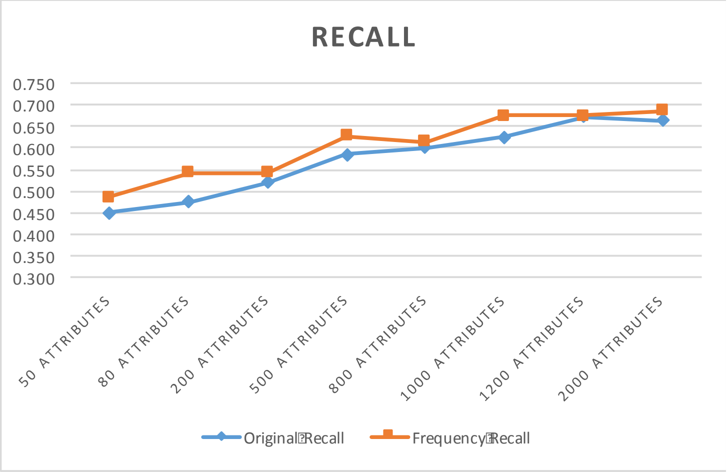


Figure 9. J48-GR Recall

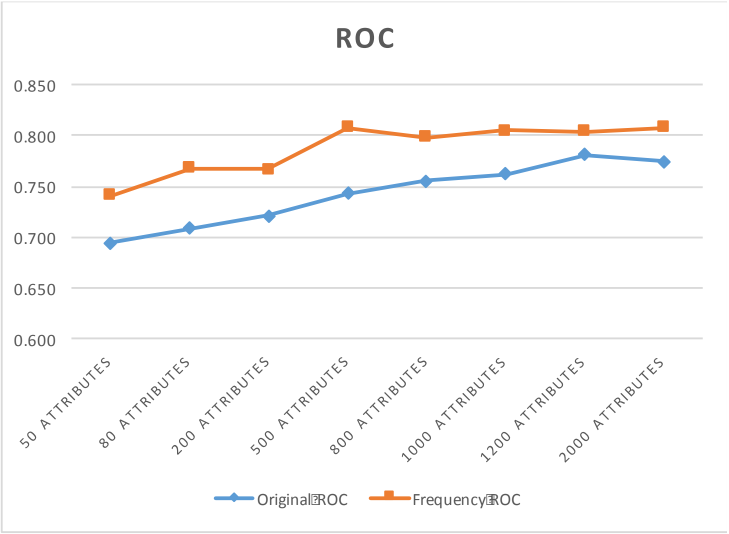


Figure 10. J48-GR ROC

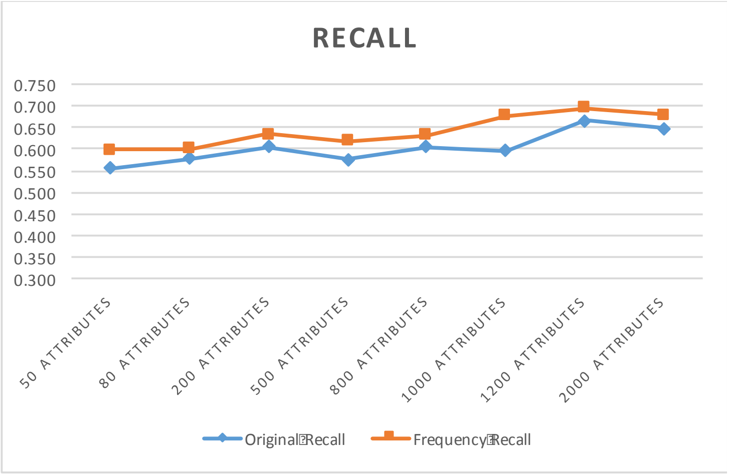


Figure 11. J48-SU Recall

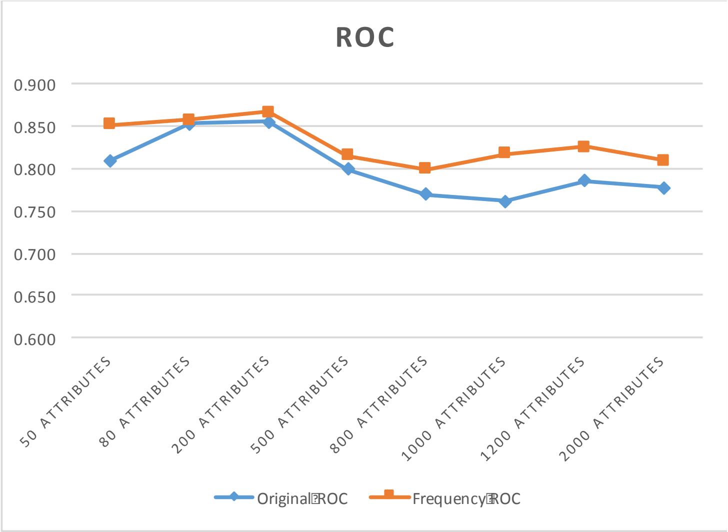


Figure 12. J48-SU ROC

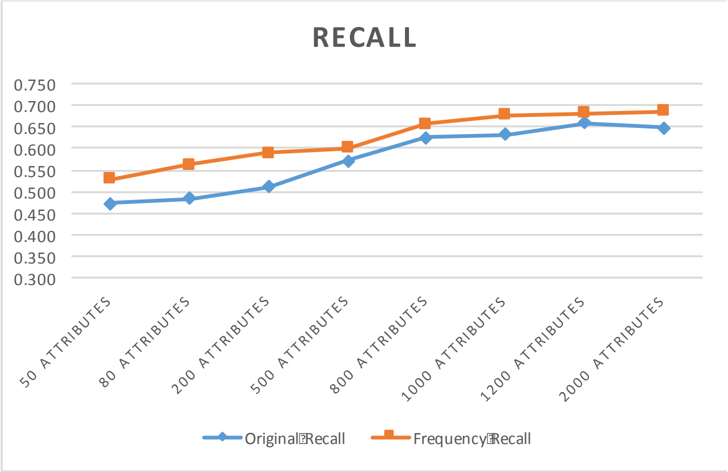


Figure 13. J48-OneR Recall

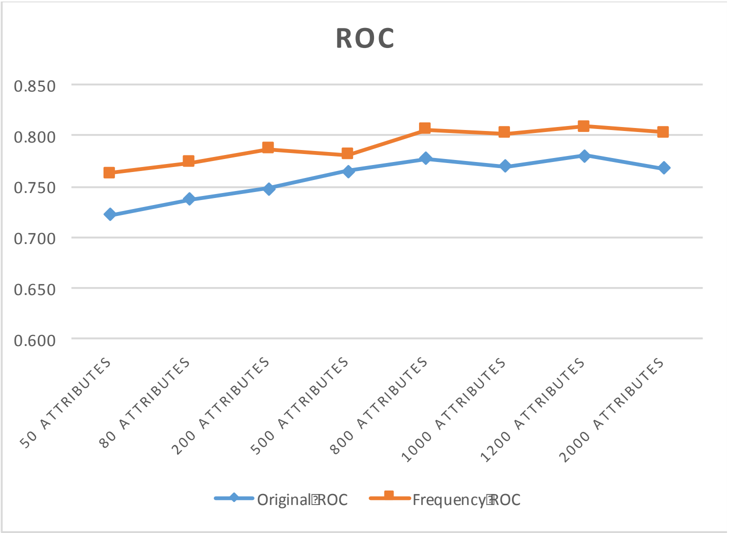
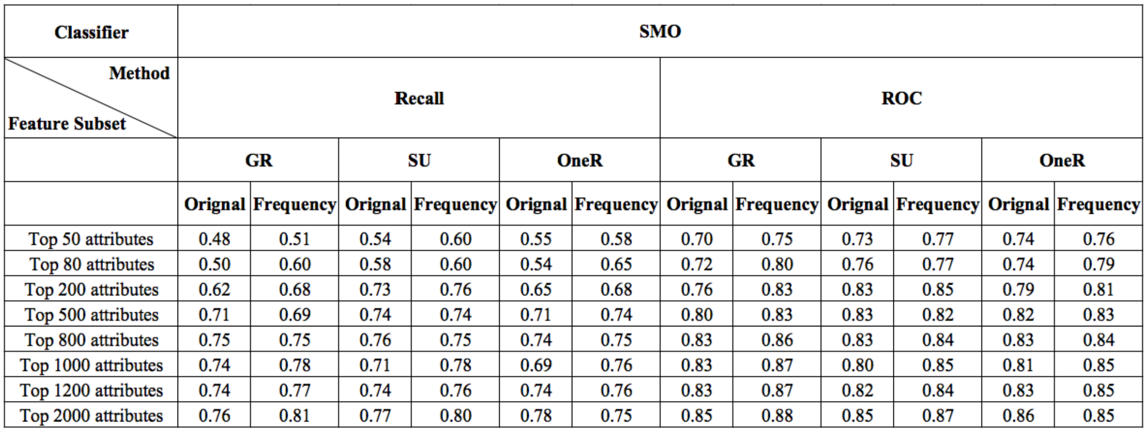


Figure 14. J48-OneR ROC

4.4.3 Case Study III

In this case, the SMO classification model is chosen and three ranking methods: GR, SU and OneR. We also picked top 50, 80, 200, 500, 800, 1000, 1200 and 2000 attributes as threshold. In Figure 15 (SMO-GR Recall), there is no big difference between two lines. In Figure 16 (SMO-GR ROC), the frequency-based method is better than the original method from the beginning until the end. No significant changes through Figure 17 (SMO-SU Recall) to Figure 18 (SMO-SU ROC). Based on Figure 19 (SMO-OneR Recall) and Figure 20 (SMO-OneR ROC), the same thing happened again, no significant change between two lines but when 2000 attributes selected, the original method performed a better performance than the frequency-basedmethod. Table 6 shows that the frequency-based method only slightly upgraded the original method, the growth rate is from 2% to 6%.



Classifier	SMO											
Method / Feature Subset	Recall						ROC					
	GR		SU		OneR		GR		SU		OneR	
	Orignal	Frequency	Orignal	Frequency	Orignal	Frequency	Orignal	Frequency	Orignal	Frequency	Orignal	Frequency
Top 50 attributes	0.48	0.51	0.54	0.60	0.55	0.58	0.70	0.75	0.73	0.77	0.74	0.76
Top 80 attributes	0.50	0.60	0.58	0.60	0.54	0.65	0.72	0.80	0.76	0.77	0.74	0.79
Top 200 attributes	0.62	0.68	0.73	0.76	0.65	0.68	0.76	0.83	0.83	0.85	0.79	0.81
Top 500 attributes	0.71	0.69	0.74	0.74	0.71	0.74	0.80	0.83	0.83	0.82	0.82	0.83
Top 800 attributes	0.75	0.75	0.76	0.75	0.74	0.75	0.83	0.86	0.83	0.84	0.83	0.84
Top 1000 attributes	0.74	0.78	0.71	0.78	0.69	0.76	0.83	0.87	0.80	0.85	0.81	0.85
Top 1200 attributes	0.74	0.77	0.74	0.76	0.74	0.76	0.83	0.87	0.82	0.84	0.83	0.85
Top 2000 attributes	0.76	0.81	0.77	0.80	0.78	0.75	0.85	0.88	0.85	0.87	0.86	0.85

Table 6. The performance of SMO

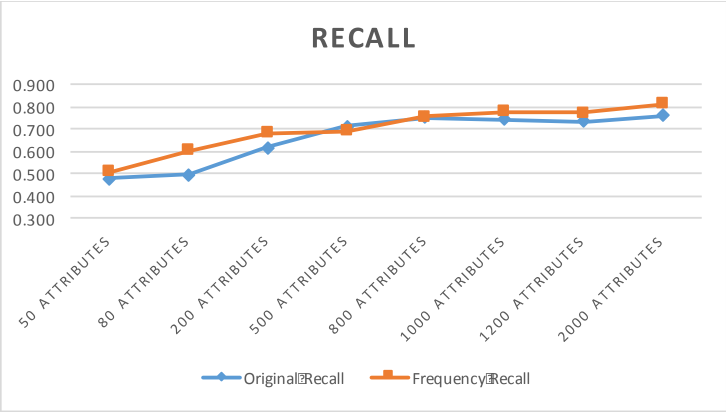


Figure 15. SMO-GR Recall

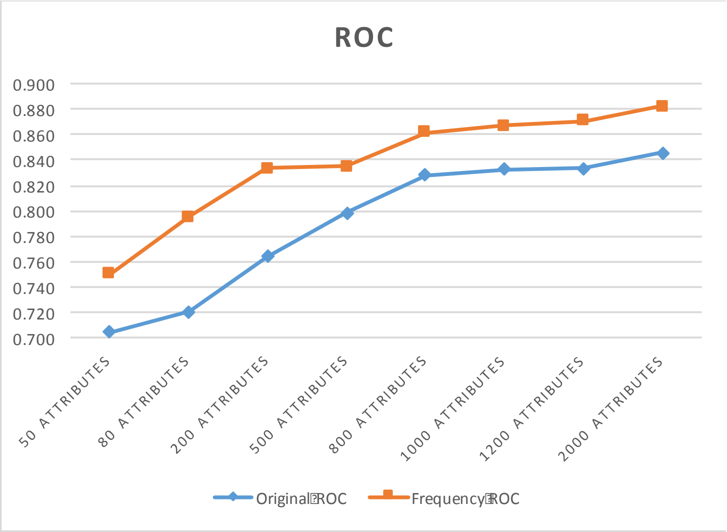


Figure 16. SMO-GR ROC

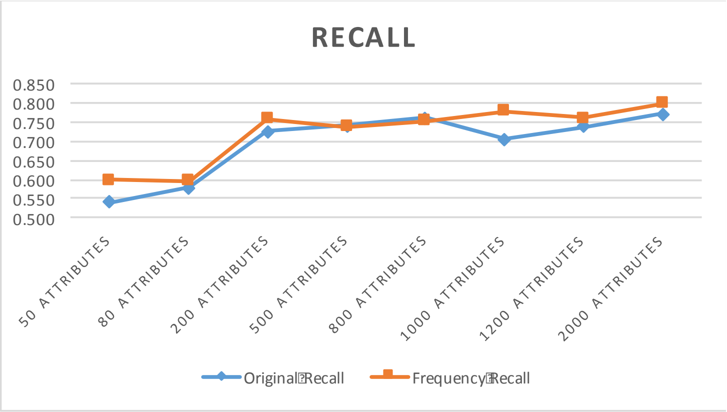


Figure 17. SMO-SU Recall

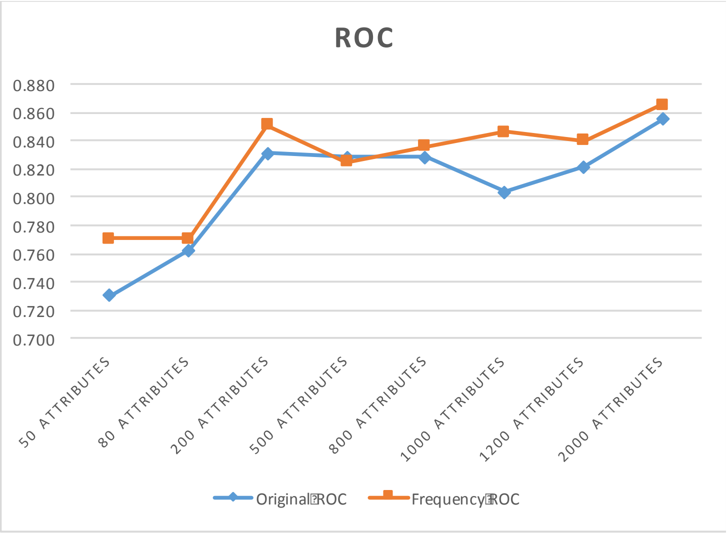


Figure 18. SMO-SU ROC

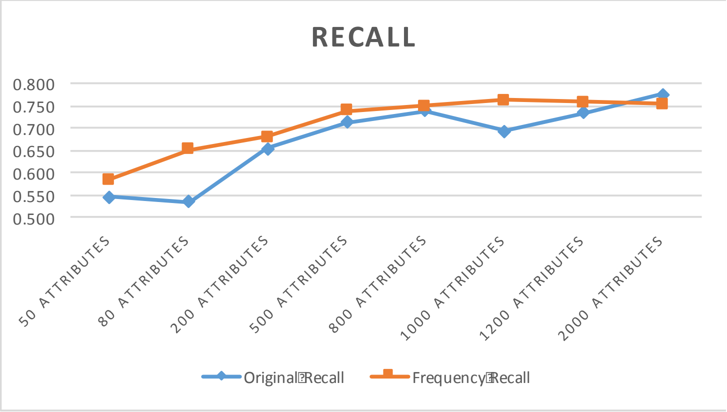


Figure 19. SMO-OneR Recall

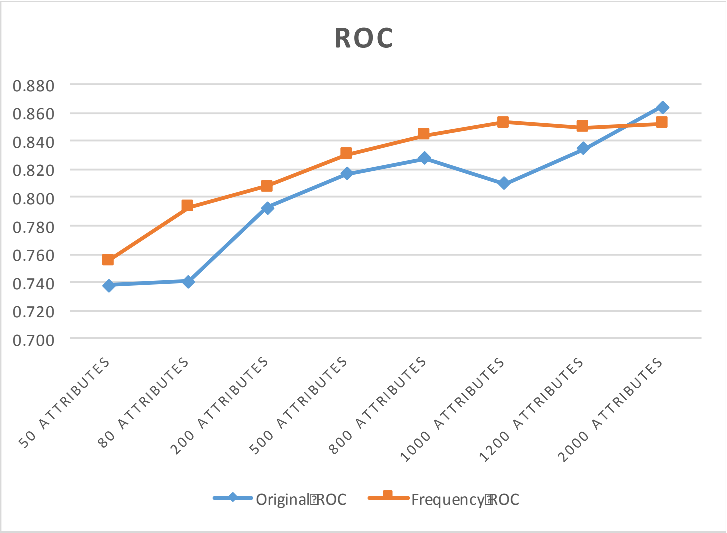


Figure 20. SMO-OneR ROC

4.4.4 Case Study IV

In the last case, we represent the performance of IBK. Through Figure 21 (IBK-GR Recall) to Figure 26 (IBK-OneR ROC), we can see, in most cases, the performance of two methods are the same, even in some cases, the original method has a better performance than the frequency-based method. In addition, based on Table 7, the performance is not reliable and stable.

Classifier	**IBK**											
Method / **Feature Subset**	**Recall**						**ROC**					
	GR		**SU**		**OneR**		**GR**		**SU**		**OneR**	
	Orignal	**Frequency**	**Orignal**	**Frequency**	**Orignal**	**Frequency**	**Orignal**	**Frequency**	**Orignal**	**Frequency**	**Orignal**	**Frequency**
Top 50 attributes	0.39	0.40	0.55	0.55	0.42	0.48	0.72	0.76	0.86	0.88	0.83	0.86
Top 80 attributes	0.38	0.45	0.54	0.59	0.49	0.51	0.75	0.81	0.89	0.89	0.85	0.87
Top 200 attributes	0.38	0.42	0.63	0.62	0.51	0.56	0.81	0.86	0.91	0.91	0.86	0.89
Top 500 attributes	0.39	0.42	0.67	0.68	0.56	0.55	0.82	0.82	0.92	0.91	0.87	0.88
Top 800 attributes	0.49	0.47	0.73	0.70	0.59	0.55	0.86	0.87	0.91	0.91	0.87	0.88
Top 1000 attributes	0.42	0.46	0.69	0.72	0.50	0.56	0.88	0.89	0.91	0.92	0.85	0.88
Top 1200 attributes	0.45	0.46	0.75	0.70	0.54	0.57	0.88	0.89	0.92	0.92	0.88	0.87
Top 2000 attributes	0.66	0.71	0.71	0.71	0.58	0.57	0.91	0.93	0.91	0.92	0.87	0.87

Table 7. The performance of IBK

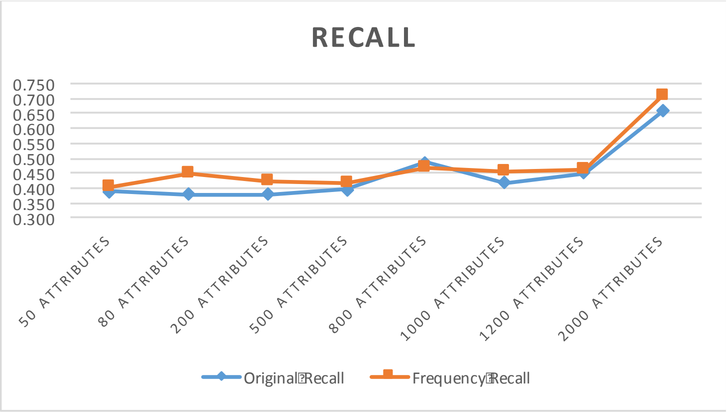


Figure 21. IBK-GR Recall

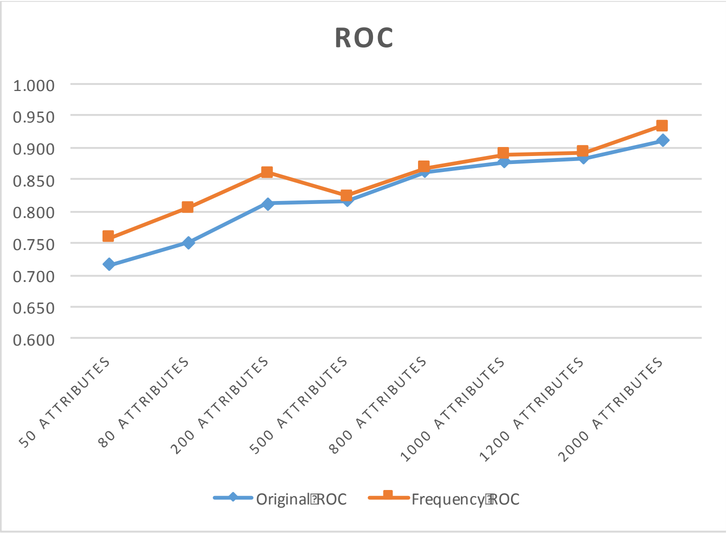


Figure 22. IBK-GR ROC

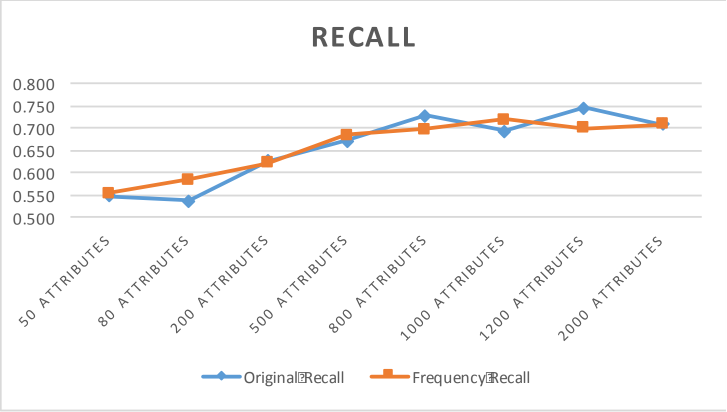


Figure 23. IBK-SU Recall

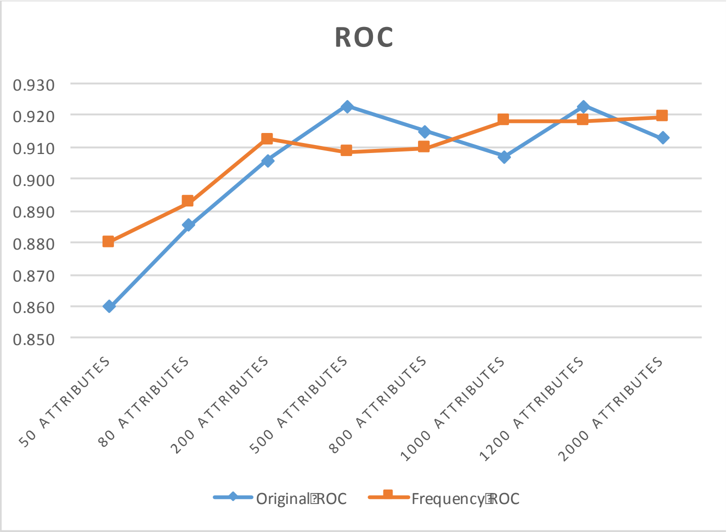


Figure 24. IBK-SU ROC

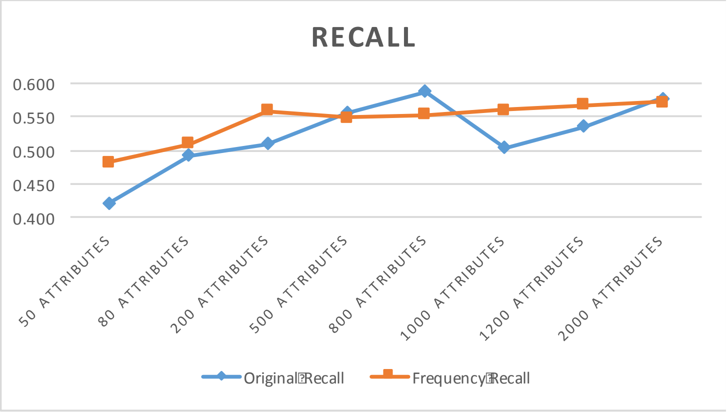


Figure 25. IBK-OneR Recall

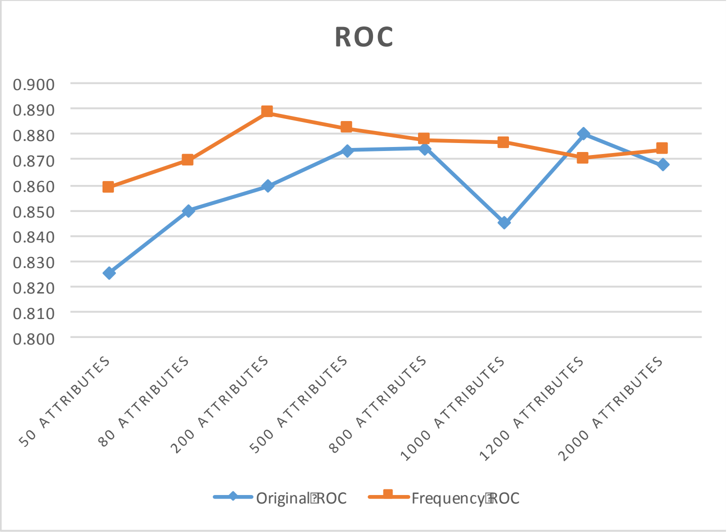
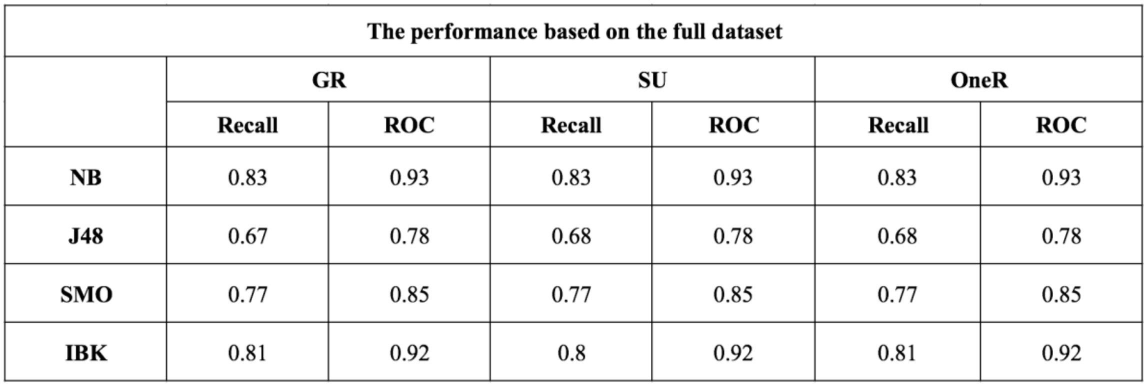


Figure 26. IBK-OneR ROC

4.4.5 Case Study V

In this case, we will study the performance of each classification models based on the full dataset. In Table 8, the NB classifier get the highest score on all three ranking methods. Then, the performance of IBK is very similar to the NB, the difference is very small. The performance of SMO is fair, the average ROC is 0.85 which means the classification model can provide a reasonable accuracy prediction. By contrast, the performance of the J48 based on the full dataset got the lowest score.



The performance based on the full dataset

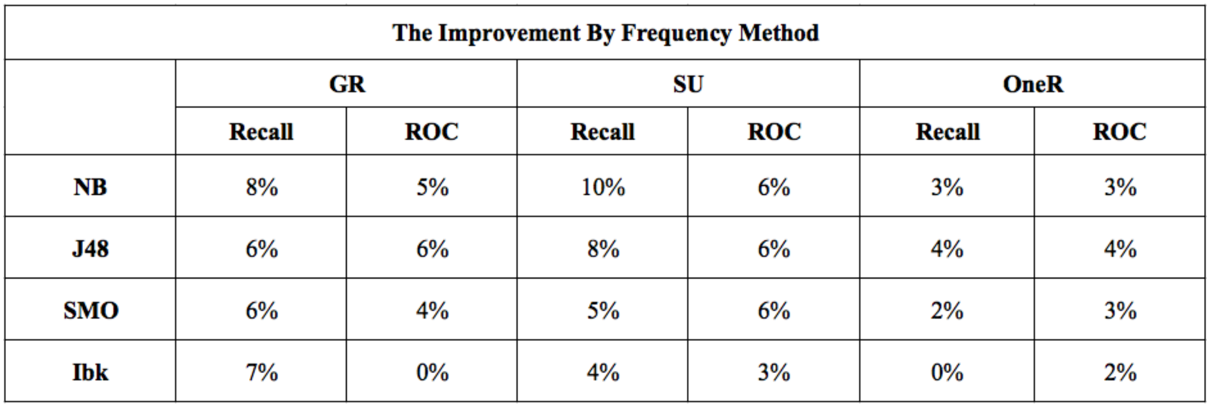
	GR		**SU**		**OneR**	
	Recall	**ROC**	**Recall**	**ROC**	**Recall**	**ROC**
NB	0.83	0.93	0.83	0.93	0.83	0.93
J48	0.67	0.78	0.68	0.78	0.68	0.78
SMO	0.77	0.85	0.77	0.85	0.77	0.85
IBK	0.81	0.92	0.8	0.92	0.81	0.92

Table 8. The performance based on the full dataset

Compare to Table 4 to Table 7, for most cases, the performance of the dataset that applied the frequency-based feature selection is better than the performance based on the full dataset and in some cases the performances are very similar. That means, the frequency-based feature selection method reduced the size of the dataset, but the performance is not getting worse or even better. So, we can say the performance of frequency-based feature selection method is better than the performance based on the full dataset.

4.4.6 Summary

Table 9 shows the improvement of the performance by using frequency-based feature selection method compare to the original feature selection method. In summary, the frequency-based method can work very well with NaiveBayes, J48 decision tree, and SMO when using GR, OneR and SU ranking method. In addition, the frequency-based method increased the original method up to 10%, and in most cases the improved methods steadily rising up. There is no sharp decrease. However, as we can see, when IBK classification model is applied, the frequency-based method cannot effectively improve the performance.



The Improvement By Frequency Method						
	GR		**SU**		**OneR**	
	Recall	**ROC**	**Recall**	**ROC**	**Recall**	**ROC**
NB	8%	5%	10%	6%	3%	3%
J48	6%	6%	8%	6%	4%	4%
SMO	6%	4%	5%	6%	2%	3%
Ibk	7%	0%	4%	3%	0%	2%

Table 9. The improvement by frequency method

5. CONCLUSION AND FUTURE WORK

Either feature selection or classification plays a significant role in data mining. By choosing the best subset, feature selection can remove most irrelevant attributes in a big dataset. In this case, the frequency-based feature selection method can work like most original feature methods did. In this study, we tested the proposed feature selection method using four classification models, and three ranking methods. In order to compare the original method and the proposed frequency-based method, we implemented a java program in WEKA to apply the frequency-based feature selection method and output the result. The chosen classification models are NB, J48, SMO and IBK. The models are evaluated by Recall and ROC. Overall, these results indicate that in most conditions especially when the NB, J48 and SMO classification model are applied, the frequency-based feature selection method can improve the performance, that is to say, the classification model can be constructed more efficiently and accurately.

Future work will continue the investigation, include more classification models and ranking methods. In addition, we may use other datasets to discuss the performance of this improved feature selection method.

REFERENCES

[1] Wang, H., Khoshgoftaar, T. M., & Gao, K. (2010, August). A comparative study of filter-based feature ranking techniques. In *Information Reuse and Integration (IRI), 2010 IEEE International Conference on* (pp. 43-48).

[2] Novakovic, J. (2009, November). Using information gain attribute evaluation to classify sonar targets. In *17th Telecommunications forum TELFOR* (pp. 1351-1354).

[3] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[4] Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

[5] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

[6] Kannan, S. S., & Ramaraj, N. (2010). A novel hybrid feature selection via Symmetrical Uncertainty ranking based local memetic search algorithm. *Knowledge-Based Systems*, *23*(6), 580-585.

[7] Hall, M. A. (1999). *Correlation-based feature selection for machine learning* (Doctoral dissertation, The University of Waikato).

[8] Leung, K. M. (2007). Naive bayesian classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*.

[9] Cheng, Iunniang. (2013). Hybrid Methods for Feature Selection. *Masters Theses & Specialist Projects*, Western Kentucky University.

[10] Patil, T. R., & Sherekar, S. S. (2013). Performance analysis of Naive Bayes and J48 classification algorithm for data classification. *International Journal of Computer Science and Applications*, *6*(2), 256-261.

[11] Kaur, G., & Chhabra, A. (2014). Improved J48 classification algorithm for the prediction of diabetes. *International Journal of Computer Applications, 98*(22).

[12] Keerthi, S. S., & Gilbert, E. G. (2002). Convergence of a generalized SMO algorithm for SVM classifier design. *Machine Learning*, *46*(1-3), 351-360.

[13] Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, *3*(02), 185-20.

[14] Cunningham, P., & Delany, S. J. (2007). k-Nearest neighbour classifiers. *Multiple Classifier Systems*, *34*, 1-17.

[15] Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *bioinformatics*, *23*(19), 2507-2517.

[16] Das, S. (2001, June). Filters, wrappers and a boosting-based hybrid for feature selection. In *ICML* (Vol. 1, pp. 74-81).

[17] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861-874.

[18] Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, *30*(7), 1145-1159.

[19] Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning* (pp. 233-240).

[20] Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, *4*, 40-79.

[21] Kelvin, X. (2016). Cancer Inhibitors. Retrieved from https://www.kaggle.com/xiaotawkaggle/inhibitors

[22] Landrum, Gregory. "Fingerprints in the RDKit." *Fingerprints in the RDKit* (2012): n. pag. *Fingerprints in the RDKit*. Gregory Landrum, 2012. Web. 10 Apr. 2017.

[23] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157-1182.