

The Islamic University – Gaza
Denary of Higher Studies
Faculty of Information Technology



New Method to Improve Mining of Multi-Class Imbalanced Data

By

Marwa Fouad Al-Rouby

Supervisor

Dr. Alaa El-Halees

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master in Information Technology

2012 – 1433H

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Deduction

To my beloved father

To my beloved mother

To sisters and brothers

To my best friends

Acknowledgements

Praises and thanks to Allah, the Almighty for having guided me at every stage of my life.

I would like to thank my parents for their love, pray, support and patience during the years of my study I also extend my thanks to my beloved brothers and sisters. Without my family I would not have been able to achieve anything.

I am heartily thankful to my supervisor, Dr. Alaa El-Halees, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

I would like to thank Mr. Motaz Saad, for his valuable scientific and technical notes also I extend my thanks to Heba El-Lahham for reviewing my thesis.

I also would like to take this opportunity to express my deepest gratitude to the academic staff of information technology program at the Islamic University-Gaza.

I also extend my thanks to Mr. Naseem Tuffaha and Mr. Walid Abu-Hadba for their financial support.

Special thanks to all my friends who have directly or indirectly have contributed to my success in completing this thesis.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the research.

Marwa F. Al-Rouby

April, 2012

New Method to Improve Mining of Multi-Class Imbalanced Data

Marwa Fouad Al-Rouby

Abstract

Class imbalance is one of the challenging problems for data mining and machine learning techniques. The data in real-world applications often has imbalanced class distribution. That is occur when most examples are belong to a majority class and few example belong to a minority class. In this case, standard classifiers tend to classify all examples as a majority class and completely ignore the minority class. For this problem, researchers proposed a lot of solutions at both data and algorithmic levels. Most efforts concentrate on binary class problems. However, binary class is not the only scenario where the class imbalance problem prevails. In the case of multi-class data sets, it is much more difficult to define the majority and minority classes. Hence, multi class classification in imbalanced data sets remains an important topic of research.

In our research, we proposed new approach based on SOMTE (Synthetic Minority Over-sampling TEchnique) and clustering which is able to deal with imbalanced data problem involving multiple classes. We implemented our approach by using open source machine learning tools: Weka, and RapidMiner. The experimental results show our approach is effective to deal with the multi class imbalanced data sets, and can improve the classification performance of minority class and its performance on the whole data set. In the best case, our F-measure improved from 66.91 to 95.18. We compared our approach with other approaches and we find our approach achieved best F-measure results in most cases.

Keywords

Data mining, Classification, Multi class classification, Class imbalanced problem, Sampling methods, SMOTE approach.

طريقة جديدة لتحسين دقة تصنيف البيانات متعددة الفئات والتي تعاني من عدم التوازن.

مروة فؤاد الروبي

المخلص

تعاني كثير من البيانات الموجودة على الأنترنت من مشكلة عدم التوازن في توزيع البيانات على الفئات التي تنتمي إليها البيانات. وتعد هذه المشكلة أحد تحديات مجال تنقيب البيانات وخاصة في مجال تصنيف البيانات. حيث أنه في مجال تصنيف البيانات هناك بيانات تصنيفها يعود لفئتين وبيانات يعود تصنيفها لأكثر من فئتين. وفيما يخص البيانات التي لها فئتان نجد أنه يقصد بمشكلة عدم التوازن هو عندما يعود تصنيف معظم الأمثلة الموجودة لدينا الى الفئة الأولى (الأكثرية) وعدد قليل جدا من الأمثلة يعود الى الفئة الثانية (الأقلية). في هذه الحالة يكون المصنف قد تعلم بشكل كبير على أمثلة الفئة الأولى نظرا لكثرتها ولم يتسنى له فرصة كافية للتعلم على أمثلة الفئة الثانية نظرا لقلة الأمثلة بها. لذلك وعند قدوم بيانات جديدة للمصنف، غالبا يقوم المصنف بتصنيفها الى الفئة الأولى وقد تكون هناك أمثلة في الاصل تصنيفها هو الفئة الثانية ولكن يخطئ في ذلك. ولهذه المشكلة كانت هناك حلول على مستوى التغيير على نفس البيانات وعلى مستوى التغيير على نفس المصنف ليكون قادرا على عدم التأثير بهذه المشكلة. وكانت معظم الأبحاث متركزة على البيانات التي يكون تصنيف بياناتها يرجع الى فئتين فقط نظرا لصعوبة التعامل مع البيانات المتعددة الفئات. لذا عملنا على ايجاد طريقة لحل هذه المشكلة ولكن على البيانات التي يكون تصنيف بياناتها يرجع الى أكثر من فئة. في طريقتنا دمجا ما بين استخدام طريقة ال SMOTE واستخدام ال Clustering . قارنا النتائج قبل استخدام طريقتنا وبعد استخدامها ووجدنا تحسن ملحوظ جدا في عملية تصنيف البيانات (في احدى الحالات حيث كانت دقة البيانات قبل طريقتنا 66.91 وأصبحت 95.18 بعد استخدامنا للطريقة المقترحة). أيضا عرضنا مثالين يظهر فيهما دقة تصنيف الفئات قبل وبعد استخدام الطريقة المقترحة. ثم قارنا عملنا بعمل الأبحاث الأخرى ووجدنا أن طريقتنا قد حققت نتائج أفضل وأكثر دقة في التصنيف في معظم الحالات.

الكلمات المفتاحية

تنقيب البيانات، معالجة البيانات، التصنيف، البيانات متعددة الفئات، مشكلة عدم التوازن.

Table of Contents

Deduction.....	iii
Acknowledgements.....	iv
Abstract.....	v
Table of Contents.....	vii
List of Figures.....	x
List of Tables.....	xi
CHAPTER 1: Introduction.....	1
1.1 Data Mining.....	1
1.2 Imbalance Data Problem.....	1
1.2.1 Imbalance in class distribution.....	2
1.2.2 Lack of data.....	3
1.2.3 Concept complexity.....	4
1.3 Multi-Class Imbalanced Data.....	4
1.4 Evaluation Problem of Imbalance Classification.....	4
1.5 Main Approaches.....	5
1.6 Problem Statement.....	5
1.7 Research Objectives.....	6
1.7.1 Main Objective.....	6
1.7.2 Specific Objectives.....	6
1.8 Research Scope and limitation.....	6
1.9 Significance of the thesis.....	6
1.10 Thesis Structure.....	7
CHAPTER 2: Literature Review.....	8
2.1 Data Mining.....	8
2.2 Classification.....	10
2.2.1 Rule Induction.....	11
2.2.2 Naïve Bayes.....	12
2.2.3 Decision Tree.....	13
2.2.4 Artificial Neural Network.....	14
2.3 Clustering.....	16

2.3.1 K-mean algorithm	16
2.3.2 X-mean algorithm	18
2.4 Major imbalanced class distribution techniques	18
2.4.1 Two class problem	18
2.4.2 Multi-class Problem	29
2.5 Summary	30
CHAPTER 3: Related Works	31
3.1 Data Level Solutions.....	31
3.2 Algorithm Level Solutions.....	34
CHAPTER 4: Research Proposal and Methodology	38
4.1 Approach combines between both Synthetic Minority Over-sampling TEchnique (SOMTE) approach and clustering approach.....	39
4.2 Collection data	40
4.3 Preprocessing Stage	43
4.3.1 Classification experiments without preprocessing.....	43
4.3.2 Under sample experiments.....	43
4.3.3 Over sample experiments.....	44
4.3.4 Under sample based on clustering experiments	44
4.3.5 Over sample based on clustering experiments	45
4.3.6 Apply over sample with use automatic clustering approach.....	45
4.3.7 Apply SOMTE approach	45
4.3.8 Apply SOMTE based on clustering approach.....	46
4.4 Apply the model.....	46
4.4.1 Rule Induction.....	47
4.4.2 Naïve Bayes	47
4.4.3 Decision Tree	48
4.4.4 Artificial Neural Network	48
4.5 Evaluate the model.....	49
4.6 Summary	49
CHAPTER 5: Experimental Results and Analysis	50
5.1 Classification experiments without preprocessing.....	50
5.2 Under sample experiments.....	51

5.3 Over sample experiments.....	52
5.4 Under sample based on clustering	52
5.5 Over sample based on clustering	54
5.6 Choosing optimal number of clusters	55
5.7 SOMTE approach	57
5.8 SOMTE based on clustering	57
5.8 Discussion and summary	60
CHAPTER 6: Conclusion and Future work.....	63
6.1 Conclusion	63
6.2 Future Work.....	64
Reference	64

List of Figures

Figure 1.1: The illustration of class imbalance problems.	2
Figure 1.2: (a) A data set with a between-class imbalance. (b) A data set with both between-class and within-class imbalance.....	3
Figure 1.3: The effect of lack of data on class imbalance problem; the solid line represents the true decision boundary and dashed line represents the estimated decision boundary.....	3
Figure 2.1: Using K-means Algorithm Operation to find three cluster in sample data.	17
Figure 2.2: The distribution of samples before and after apply under sample approach.	19
Figure 2.3: Simple example on under sample based on clustering approach.	21
Figure 2.4: The distribution of samples before and after apply over sample approach.	22
Figure 2.5: Simple example on over sample based on clustering approach.	24
Figure 2.6: (A) Imbalanced data set. (B) Balanced data set.....	26
Figure 4.1: Methodology Steps.....	39
Figure 4.2: General view of our proposed approach.....	40
Figure 4.3: Settings of rule indication.....	47
Figure 4.4: Settings of naïve Bayes.	47
Figure 4.5: Settings of decision tree.....	48
Figure 4.6: Settings of neural network.....	48
Figure 5.1: Average F-measures for each classifier on whole data sets.	53
Figure 5.2: Average F-measures for each classifier on whole data sets.	54
Figure 5.3: Summary for all our experiments.	60

List of Tables

Table 2.1: Basic structure of decision tree algorithm.	14
Table 2.2: Basic structure of backpropagation algorithm.	15
Table 2.3: K-mean clustering algorithm	17
Table 2.4: The structure of under sample based on clustering approach.	21
Table 2.5: The structure of over sample based on clustering approach.	23
Table 2.6: Simple example of generation of synthetic examples (SMOTE).	25
Table 2.7: Cost matrix for two class.	28
Table 4.1: Summary of data sets	41
Table 4.2: The structure of SMOTE based on clustering approach (our approach).	46
Table 5.1: Average accuracy for whole data sets in classification experiments without preprocessing.....	51
Table 5.2: Average F-measure for whole data sets in classification experiments without preprocessing ..	51
Table 5.3: Average F-measure for whole data sets in under sample experiments	52
Table 5.4: Average F-measure for whole data sets in over sample experiments	52
Table 5.5: Average F-measure for whole data sets when determine number of cluster manual and automatic	56
Table 5.6: Average F-measure for whole data sets in SMOTE experiments.....	57
Table 5.7: Average accuracy for whole data sets in SMOTE based on clustering experiments.....	58
Table 5.8: Average F-measure for whole data sets in SMOTE based on clustering experiments	58
Table 5.9: F-measure results of the approaches: over sample based on clustering and SOMTE with clustering for all our data set.....	58
Table 5.10: Average accuracy and F-measure comparison of the approaches: baseline and SOMTE with clustering experiments for all our data set	61
Table 5.11: F-measure results – page blocks	62
Table 5.12: F-measure results – auto-mpg.....	62
Table 6.1: Summary table for compare between some other works	64

CHAPTER 1: Introduction

This chapter introduces data mining, the imbalance data problem, multi-class imbalanced data, evaluation problem of imbalance classification, main approaches, research objective, research scope and limitation and significance of the thesis.

1.1 Data Mining

Data mining is the process of extracting patterns from data. It is the analysis of observational data sets to find unsuspected associations and to sum up the data in new ways that are both clear and useful to the data owner. It is a prevailing technology which has great potential to help companies that focus on the most important information in their data warehouses. Tools of data mining predict future trends and behaviors, allowing businesses to make proactive and the knowledge-driven decisions. One of the core tasks of the data mining is the classification task. Classification is a fundamental task of data mining. The task of the constructed classifier is to predict the class labels for an unseen input objects based on a certain number of observations. An example of classification is the categorization of the bank loan applications as either safe or risky [9] [21] [26].

1.2 Imbalance Data Problem

The classification techniques usually assume a balanced class distribution (i.e. there data in the class is equally distributed). Usually, a classifier performs well when the classification technique is applied to a dataset evenly distributed among different classes. But many real applications face the imbalanced class distribution problem. In this situation, the classification task imposes difficulties when the classes present in the training data are imbalanced.

The imbalanced class distribution problem occurs when one class is represented by a large number of examples (majority class) while the other is represented by only a few (minority class). In this case, a classifier usually tends to predict that samples have the majority class and completely ignore the minority class. This is known as the class imbalance problem. Figure 1.1 illustrates the idea of the class imbalance problem where a minority class is represented by only 1% of the training data and 99% for majority class.

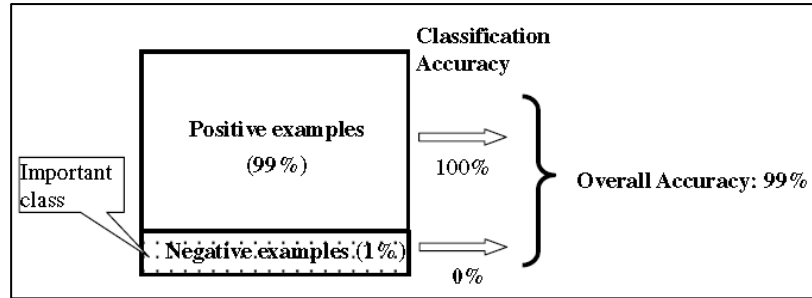


Figure 1.1: The illustration of class imbalance problems [7].

Unfortunately, this problem is very pervasive in many domains. For example, with text classification tasks whose training sets typically contain much fewer documents of interest to the reader than on irrelevant topics. Other domains suffering from class imbalances include target detection, fault detection, or credit card fraud detection problems, disease diagnosis, bioinformatics, oil-spill detection and many other areas, which contain much fewer instances of the event of interest than of irrelevant events [10].

Class imbalanced presents several difficulties in learning, including imbalanced in class distribution, lack of data and concept complexity.

1.2.1 Imbalance in class distribution

The class imbalance problem can appear either from *between classes* (inter class) or *within a single class* (intra class) [23]. *Inter-class* imbalance refers to the case when one class has larger number of example than another class. The degree of imbalance can be represented by the ratio of size of the minority class to size of the majority class. *Intra-class* imbalance occurs when a class consists of several sub-clusters or sub-concepts and these sub-concepts do not have the same number of sample. In a simple example, consider the depicted distributions in Figure 1.2. In this figure, the stars represent the minority and the circles represent the majority classes. Figure 1.2 (a) display a data set with a *between-class* imbalance. The circles class has larger number of instance than stars class and there is no overlapping between the two classes. Figure 1.2 (b) display data set with both *between-class* and *within-class* imbalance and there is overlapping between two classes. As shown in Figure 1.2 (b), for example cluster C represent a sub-concept of the minority class and cluster D represents two sub-concepts of the majority class. In our research we focused mainly on rectifying the *between-class* imbalance.

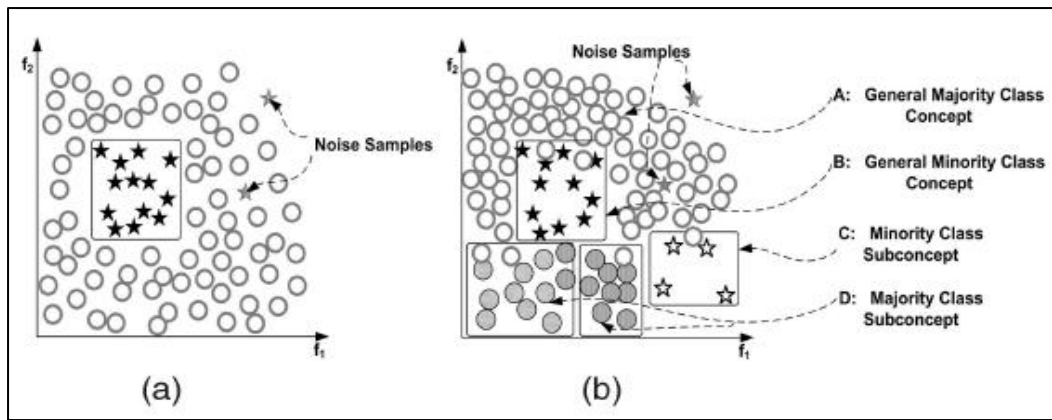


Figure 1.2: (a) A data set with a between-class imbalance. (b) A data set with both between-class and within-class imbalance [18]

1.2.2 Lack of data

One of the primary problems when learning with imbalanced data sets is the associated lack of data where the number of samples is small. In a classification task, the size of data set has an important role in building a good classifier. Lack of example, makes it difficult to uncover regularities within the small classes [23]. Figure 1.3 illustrates an example of the problem that can be caused by lack of data. Figure 1.3 (a) shows the dashed line obtained when using sufficient size from data set for training. Figure 1.3 (b) illustrates the result when using a small size from data set. When there is sufficient data, the estimated decision boundary (dashed line) approximates well the true decision boundary (solid line); whereas, if there is a lack of data, the estimated decision boundary can be very far from the true boundary. It has been shown that as the size of training data increases, the error rate caused by imbalanced training data decreases. However, using the sufficient size from data set for training, the classification system may not be affected by high imbalance ratio.

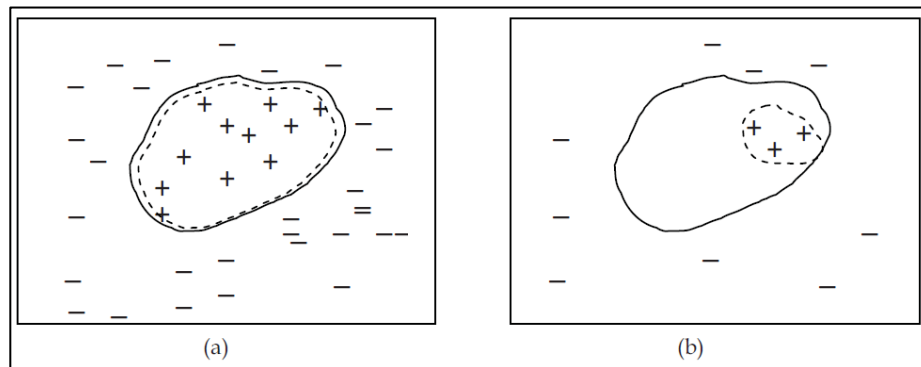


Figure 1.3: The effect of lack of data on class imbalance problem; the solid line represents the true decision boundary and dashed line represents the estimated decision boundary [23].

1.2.3 Concept complexity

Concept complexity is an important factor in a classifier ability to deal with imbalance problem. Concept complexity refers to the separation level between classes within the data. Linear separation between classes means the classifier not liable to any amount of imbalance. On other hand, the high complexity refers to occurs high overlapping between the two classes that means the classifier susceptible to any amount of imbalance [23]. So, for a given data sets that is complex and imbalanced, the challenge is how to train a classifier that correctly recognizes samples of different classes with high accuracy. In a simple example, notice how Figure 1.2 (a) has no overlapping example between its classes and has only one concept pertaining to each class, whereas Figure 2.1 (b) has both multiple concepts and severe overlapping.

1.3 Multi-Class Imbalanced Data

The imbalanced data problem can appear in two different types of data sets: binary problems, where one of the two classes comprises considerably more samples than the other and multi-class problems, where the applications have more than two classes and unbalanced class distribution hinder the classification performance. Most research efforts on imbalanced data sets have traditionally concentrated on two-class problems. However, this is not the only scenario where the class imbalance problem prevails. In the case of multi-class data sets, it is much more difficult to define the majority and minority classes [13] [31]. Hence, multi class classification in imbalanced data sets remains an important topic of research. In our research we focused mainly on multi class imbalance problem which the two-class problem is considered as a special case from multi-class problem.

1.4 Evaluation Problem of Imbalance Classification

Evaluation measures play a crucial role in both assessing the classification performance and guiding the classifier modeling. Traditionally, accuracy is the most commonly used measure for these purposes. The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier [16]. However, for classification of imbalanced data, accuracy no longer a proper measure since the rare class has very little impact on the accuracy as compared to that on the prevalent class. So, it is important to know that when the performances of all classes are interested, classification performance of each class should be

equally represented in the evaluation measure [26] [27]. Therefore, other metrics have been developed to assess classifiers performance for imbalanced data set. One of the most important of these metrics is F-measure. In our research we depend on F-measure as a measure for classification for imbalanced data. For example, we note in car evaluation data set from [38] before preprocessing data, the average accuracy with using naïve Bayes classifier is 86.49 which is consider good while the average F-measure is 69.64 which is consider low. So, the accuracy measure cannot detect the imbalanced problem and cannot give us the actual classification performance especially when the data has imbalanced class distribution problem.

1.5 Main Approaches

In order to overcome the class imbalance problem, some approaches have been introduced at both algorithm and data levels. At algorithm level approaches (also called internal) which try to adapt existing classifier learning algorithm to bias the learning toward the minority class. These methods require special knowledge of both the corresponding classifier and the application domain, comprehending why the classifier fails when the class distribution is uneven. Examples on the algorithm level approaches are recognition based learning, ensemble learning, and cost-sensitive learning. At data level approaches (or external) which are rebalance the class distribution by resampling the data space. This way avoids the modification of the learning algorithm by trying to decrease the effect caused by imbalance with a preprocessing step. Therefore, they are independent of the classifier used, and for this reason, usually more versatile. Examples on the data level approaches are re-sampling techniques and multi classifier committee approach [8][12][23][26][32]. We mainly concern on the methods at the data level in our research. More detail about these approaches are described in chapter two.

1.6 Problem Statement

How to develop a new method able to effectively handle multi-class imbalanced data set to improve the classification performance of minority class.

1.7 Research Objectives

1.7.1 Main Objective

The main objective of this research is to try to increase the classification accuracy of minority class by avoiding the drawbacks of the previous methods.

1.7.2 Specific Objectives

The specific objectives of this research are:

- Investigate the current approaches on handling imbalanced data.
- Propose new more efficient method.
- Apply the model in some data sets.
- Test if the solution works in binary class classifier as well as multi-class classifier.
- Using different classifiers to classify the instance.
- Apply our proposed approach on various real domains and evaluate the results.
- Compare our proposed method with other existing methods.

1.8 Research Scope and limitation

- We concentrate at data level not algorithmic.
- We focus mainly on rectifying the between-class imbalance, and ignore the case where imbalance occurs within each class.
- We focus on multi-class data sets case which the two-class problem is a special case from multi-class problem.
- We use data mining preprocessing methods that can apply the proposed method.
- We assume the data set does not contain missing or noise value.
- We apply change only on rows without change column or use the feature selection method.
- We use only small, medium and large data sets.

1.9 Significance of the thesis

- Unfortunately, many datasets in real applications (such as health examination, inspection, credit fraud detection, spam identification and text mining....etc.) involve imbalanced class distribution problem especially multi-class cases. We

apply our approach on six data sets with different real domain, characteristics and sizes.

- There are a few solutions that have been proposed aiming at multi-class, most current algorithms are discussed and tested by using two-class imbalanced data sets. In this research, our approach able to deal with imbalanced data problem involving multiple classes.
- The current solutions provide little improvement in the imbalance data. In our approach, we achieved good classification accuracy of minority class in imbalanced class distribution problem.
- Re-balance the class distribution to avoid happen imbalanced data distributions considered the important step in preprocessing and preparing data in data mining to use it after that in various fields because this process impact on result accuracy of minority class.
- Our research can be applied to all classifiers since the work in the preprocessing stage while algorithmic solution needs to modify the algorithm of each method.

1.10 Thesis Structure

The rest of the research is organized as follows: Second chapter for literature review. Third chapter presents related work. Fourth chapter include the methodology and proposed model architecture. In fifth chapter, we discuss the experimental results and analysis. Sixth chapter draws the conclusion and summarize the research achievement and future direction.

CHAPTER 2: Literature Review

In this chapter we introduce some important fundamentals and basic terminology that we used in our research. It includes the following topics: section one about data mining. Section two about classification that describes major kinds of classification algorithms which are used in our research: rule induction, naïve Bayes, decision tree and neural network. Section three about kind of clustering. In section four, we give an overview of major existing techniques related to imbalanced class distribution problem which is used in two-class and multi-class imbalanced data set problem.

2.1 Data Mining

Data mining, which is also referred to as *knowledge discovery in databases*, means a process of nontrivial extraction of implicit, previously unknown and potentially useful information (such as knowledge rules, constraints, regularities) from data [16]. There are also many other terms carry a similar or slightly different meaning to data mining, such as *knowledge mining from databases*, *knowledge extraction*, *data/pattern analysis*, *data archaeology*, and *data dredging*. Mining information and knowledge from large databases has been recognized by many researchers as a key research topic in database system and machine learning and by many industrial companies as an important area with an opportunity of major revenues. The discovered knowledge can be applied to information management, query processing, decision making, process control, and many other applications. Researchers in many different fields, including database systems, knowledge-based systems, artificial intelligence, machine learning, knowledge acquisition, statistics, spatial databases, and data visualization have shown great interest in data mining [16].

Data mining is an essential step in the process of knowledge discovery. Knowledge discovery as a process is consist of an iterative sequence of the following steps:

1. *Data cleaning* to remove noise and inconsistent data.
2. *Data integration* where multiple data sources may be combined.
3. *Data selection* where data relevant to the analysis task are retrieved from the database.

4. *Data transformation* where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations.
5. *Data mining* an essential process where intelligent methods are applied in order to extract data patterns.
6. *Pattern evaluation* to identify the truly interesting patterns representing knowledge based on some interestingness measures.
7. *Knowledge presentation* where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. In general, data mining tasks can be classified into two categories: *descriptive mining* and *predictive mining*. *Descriptive mining* tasks characterize the general properties of the data in the database such as *association rule* and *clustering*. *Predictive mining* tasks perform inference on the current data in order to make predictions such as *classification*, *prediction* and *outlier analysis* [16].

Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. It studies the frequency of items occurring together in transactional databases, and based on a threshold called *support*, identifies the frequent item sets. Another threshold, *confidence*, which is the conditional probability than an item appears in a transaction when another item appears, is used to pinpoint association rules. Association analysis is commonly used for market basket analysis [16] [33].

Classification is the organization of data in given classes. Also known as *supervised classification*, the classification uses given class labels to order the objects in the data collection. Classification approaches normally use a training set where all objects are already associated with known class labels. The classification algorithm learns from the training set and builds a model. The model is used to classify new objects [33]. In the next section we will talk about classification and its method which is used during our experiments.

Prediction has attracted considerable attention given the potential implications of successful forecasting in a business context. There are two major types of predictions: one can either try to predict some unavailable data values or pending trends, or predict a class label for some data.

The latter is tied to classification. Once a classification model is built based on a training set, the class label of an object can be foreseen based on the attribute values of the object and the attribute values of the classes. Prediction is however more often referred to the forecast of missing numerical values, or increase/ decrease trends in time related data. The major idea is to use a large number of past values to consider probable future values [33].

Clustering is a division of data into groups of similar objects. It is similar to classification. However, unlike classification, in clustering, class labels are unknown and it is up to the clustering algorithm to discover acceptable classes. Clustering is also called *unsupervised classification*, because the classification is not dictated by given class labels. There are many clustering approaches all based on the principle of maximizing the similarity between objects in a same class (intra-class similarity) and minimizing the similarity between objects of different classes (inter-class similarity) [16] [33]. In the section 2.3 we will talk about clustering and its method which is used during our experiments.

Outlier analysis: Outliers are data elements that cannot be grouped in a given class or cluster. Also known as *exceptions* or *surprises*, they are often very important to identify. While outliers can be considered noise and discarded in some applications, they can reveal important knowledge in other domains, and thus can be very significant and their analysis valuable [33].

2.2 Classification

Classification is a main task in data mining. The classification is a supervised learning task that estimates the correct classes of objects [16]. In general, there are two steps for data classification. In the first step, a classifier is built describing a predetermined set of data classes or concepts which are known as “model”. This is the training phase, where a classification algorithm builds the classifier by analyzing or “learning from” a training set made up of database tuples and their associated class labels. Each tuple is assumed to belong to a predefined class called class label attribute. In the second step, the model is used for classification. First, the predictive accuracy of the classifier is estimated. Therefore, a test set is used, made up of test tuples and their associated class labels. These tuples are randomly selected from the general data set. The accuracy of a classifier on a given test set is the percentage of test set tuples that are

correctly classified by the classifier. The associated class label of each test tuple is compared with the learned classifier's class prediction for that tuples.

In this next sub sections we describe major kinds of classification algorithms which are used in our research: rule induction, naïve Bayes, decision tree and neural network.

2.2.1 Rule Induction

IF condition THEN conclusion.....2.1

An example is rule R1,

R1: IF age = youth AND student = yes THEN buys_computer = yes.

The “IF” part (or left-hand side) of rule is known as the rule antecedent or precondition. The “THEN” part (or right-hand side) is the rule consequent. In rule antecedent, the conduction consists of one or more attribute tests (such as *age = youth* and *student = yes*) that are logically AND. The rule's consequent contains a class prediction (in this case, it predicting a customer will buy a computer). If the condition (that is, all of the attribute tests) in rule antecedent holds true for a given tuple, we say that the rule antecedent is satisfied (or simply, that the rule is satisfied) and that the rule covers the tuple.

A rule **R** can be assessed by its coverage and accuracy. Given a tuple, **X**, from a class labeled data set, **D**, let n_{covers} is the number of tuples covered by **R**; $n_{correct}$ is the number of tuples correctly classified by **R**; and $|D|$ be the number of tuples in **D**. The coverage and accuracy of **R** can be defined as:

$$Coverage(R) = \frac{n_{covers}}{|D|} \dots\dots\dots 2.2$$

$$Accuracy (R) = \frac{n_{correct}}{n_{covers}} \dots\dots\dots 2.3$$

That is, a rule's coverage is the percentage of tuples that are covered by the rule (i.e, whose attribute values hold true for the rule's antecedent). For a rule's accuracy, it looks at the tuples that it covers and see what percentage of them the rule can correctly classify [16].

2.2.2 Naïve Bayes

Bayesian classifiers are statistical classifiers [16]. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on Bayes' theorem. Studies comparing classification algorithms have found a simple Bayesian known as naïve Bayesian classifiers. A naïve Bayesian classifier assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. The naïve Bayesian classifier work as following steps [16]:

Step1: let D be training set of tuples and their associated class labels. Each tuple is represented by an n-dimensional attribute vector, $X = (x_1, x_2, \dots, x_n)$, n measurements made on the tuple from n attribute, respectively, A_1, A_2, \dots, A_n .

Step2: assume that there are m classes, C_1, C_2, \dots, C_m . given a tuple, X, the classifier will predict that X belongs to the class having the highest probability, conditioned on X. That is, the naïve Bayesian classifier predicts that tuple X belongs to the class C_i if and only if

$$P(C_j|X) > P(C_i|X) \text{ for } 1 \leq j \leq m, j \neq i \dots\dots\dots 2.4$$

The class C_i for which $P(C_i|X)$ is the maximized is called the maximum posteriori hypothesis. By Bayes' theorem (Equation 2.5),

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{p(X)} \dots\dots\dots 2.5$$

Step3: as $P(X)$ is constant for all classes, only $P(X|C_i) P(C_i)$ need maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally.

Step4: based on the assumption is that attributes are conditionally independent (no dependence relation between attributes), $P(X|C_i)$ using Equation 2.6.

$$P(X|C_i) = \prod_{k=1}^n P(x_k|C_i) \dots\dots\dots 2.6$$

Equation 2.6 reduces the computation cost, only counts the class distribution. If A_k is categorical, $P(X_k|C_i)$ is the number of tuples in C_i having value x_k for A_k divided by $|C_i, D|$ (number of tuples

of C_i in D). And if A_k is continuous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ and $P(X_k|C_i)$ is

$$P(X|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) \dots \dots \dots 2.7$$

$$g(x_k, \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \dots \dots \dots 2.8$$

Where μ is the mean and σ is the variance. If an attribute value doesn't occur with every class value, the probability will be zero, and a posteriori probability will also be zero.

2.2.3 Decision Tree

A decision tree is a flowchart like tree structure, where each internal node (nonleaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label [16]. The topmost node in tree is the root node. Instances are classified starting at the root node and sorted based on their feature values. Decision tree can easily be converted to classification rules. The most popular algorithm in the literature for building decision tree is the C4.5. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision tree generated from a set of training data by C4.5, using the concept of information entropy. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits. The criterion is the normalized information gain which is result from choosing an attribute for splitting the data. So, the attribute which has the highest normalized information gain is chosen to make the decision tree. Decision tree algorithm was constructed in a top-down recursive divide-and-conquer manner. Table 2.1 presents decision tree algorithm [16].

Table 2.1: Basic structure of decision tree algorithm [16].

Input:

- Data partition, D , which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *attribute_selection_method*, a procedure to determine the splitting criterion that “best” partitions the data tuples in to individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A Decision tree.

Method:

1. create a node N ;
2. **if** tuples in D are all of the same class, C **then**
3. return N as a leaf node labeled with the class C ;
4. **if** *attribute_list* is empty **then**
5. return N as a leaf node labeled with the majority class in D ; // majority voting
6. apply **Attribute_selection_method** (D , *attribute_list*) to **find** the “best” *splitting_criterion*;
7. label node N with *splitting_criterion*;
8. **if** *splitting_attribute* is discrete-valued **and** multiway splits allowed **then** // not restricted to binary trees
9. *attribute_list* \leftarrow *attribute_list* - *splitting_attribute*; // remove *splitting_attribute*
10. **for each** outcome j of *splitting_criterion* // partition the tuples and grow subtrees for each partition
11. let D_j be the set of data tuples in D satisfying outcome j ; // a partition
12. **if** D_j is empty **then**
13. attach a leaf labeled with the majority class in D to node N ;
14. **else** attach the node returned by **Generate_decision_tree** (D_j , *attribute_list*) to node N ;
15. **end for**
15. Return N ;

2.2.4 Artificial Neural Network

A neural network is a set of connected input/output units in which each connection has a weight associated with it [16]. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input tuples. Neural network learning is also referred to as connectionist learning due to the connections between units. Neural networks involve long training times and are therefore more suitable for applications where this is feasible. They require a number of parameters that are typically best determined empirically, such as the network topology or “structure”. Neural networks have been criticized for their poor interpretability. For example, it is difficult for humans to interpret the symbolic meaning behind the learned weights and of “hidden units” in the network. These features initially made neural networks less desirable for data mining.

There are many different kinds of neural networks and neural network algorithms. The most popular neural network algorithm is backpropagation, which gained reputation in the 1980s. The backpropagation algorithm performs learning on a multilayer feed-forward neural network. It iteratively learns a set of weights for prediction of the class label of tuples. A multilayer feed-forward neural network consists of an input layer, one or more hidden layers, and an output layer.

The inputs to the network correspond to the attributes measured for each training tuple. These inputs pass through the input layer and are then weighted and fed simultaneously to a second layer, known as a hidden layer. The outputs of the hidden layer units can be input to another hidden layer, and so on. The number of hidden layers is arbitrary, although in practice, usually only one is used. The weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction for given tuples. The backpropagation algorithm is summarized in Table 2.2 [16].

Table 2.2: Basic structure of backpropagation algorithm [16].

<p>Input:</p> <ul style="list-style-type: none"> • D, a data set consisting of the training tuples and their associated target value; • l, the learning rate; • $network$, a multilayer feed-forward network. <p>Output: A trained neural network.</p> <p>Method:</p> <ol style="list-style-type: none"> 1. Initialize all weights and biases in network; 2. While terminating condition is not satisfied { 3. for each training tuple X in D { 4. // propagate the inputs forward: 5. for each input layer unit j { 6. $O_j = I_j$; // output of an input unit is its actual input value 7. for each hidden or output layer unit j { 8. $I_j = \sum_i w_{ij}O_i + \theta_j$; // compute the net input of unit j with respect to the previous layer, i 9. $O_j = \frac{1}{1 + e^{-I_j}}$; } // compute the output of each unit j 10. // Backpropagate the errors: 11. for each unit j in the output layer 12. $Err_j = O_j(1 - O_j)(T_j - O_j)$; //compute the error 13. for each unit j in the hidden layers, from the last to the first hidden layer 14. $Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}$; // compute the error with respect to the next higher layer, k 15. for each weight w_{ij} in $network$ { 16. $\Delta w_{ij} = (l)Err_j O_i$; //weight increment 17. $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update 18. for each bias θ_j in network {
--

```

19.       $\Delta \theta_j = (1)Err_j$ ; // bias increment
20.       $\theta_j = \theta_j + \Delta \theta_j$ ; } // bias update
21.    } }

```

2.3 Clustering

Clustering is one of data mining technique, it is an unsupervised learning problem widely studied in many research areas such as statistics, machine learning, data mining, and pattern recognition [3]. The objective of clustering process is the organization of data in clusters through grouping of similar objects and a clustering of a set is a partition of its elements that is chosen to minimize some measure of dissimilarity. However, it is unlike classification, in clustering, class labels are unknown. Clustering algorithms are often useful in applications in various fields such as visualization, pattern recognition, learning theory, computer graphics, neural networks, AI, and statistics.

In the next sub sections we describe the major kinds of clustering algorithms which are used in our research: k-mean algorithm, X-mean algorithm.

2.3.1 K-mean algorithm

One of the most popular used algorithms for clustering is called k-mean cluster. K-mean clustering is a method of cluster analysis which aims to partition n instances into k clusters in which each observation belongs to the cluster with the nearest mean [3]. The basic K-means algorithm requires time proportionate to number of patterns and number of cluster per iteration. This is computationally expensive especially for large datasets which sizes ranging from hundreds of thousands to millions. The worst case complexity of k-means is $O(nkt)$, where n is the number of data points or objects, k is the number of desired clusters, and t is the number of iterations the algorithm takes for converging to a stable state. It operates on numerical and binary data and it is cannot handle missing data and outliers. The pseudo code of k-means algorithm is shown in Table 2.3. The algorithm take two inputs, X that is the dataset examples to be clustered and the K that is the number of clusters. It starts by place K points into the space represented by the objects that are being clustered. These points represent initial group centroids. Then the algorithm assigns each object to the group that has the closest centroid using Euclidean distance Equation 2.10. The distance between two points $P = (x_1(P), x_2(P), \dots)$ and $Q = (x_1(Q), x_2(Q), \dots)$ is given by:

$$D(P, Q) = \sqrt{(x_1(P) - x_1(Q))^2 + (x_2(P) - x_2(Q))^2 + \dots} \dots \dots \dots 2.9$$

$$= \sqrt{\sum_{j=1}^p (x_j(P) - x_j(Q))^2} \dots \dots \dots 2.10$$

When all objects have been assigned, recalculate the positions of the K centroids. This produces a separation of the objects into groups from which the metric to be minimized can be calculated. The operation of k-means algorithm is illustrated in Figure 2.1

Table 2.3: K-mean clustering algorithm [16].

<p>Input:</p> <ul style="list-style-type: none"> • <i>K</i>: the number of clusters, • <i>D</i>: a data set containing n objects. <p>Output: A set of <i>k</i> clusters.</p> <p>Method:</p> <ol style="list-style-type: none"> 1. Arbitrary choose <i>k</i> objects from <i>D</i> as the initial cluster centers; 2. Repeat 3. (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster; 4. update the cluster means, i.e., calculate the mean value of the objects for each cluster; 5. until no change;

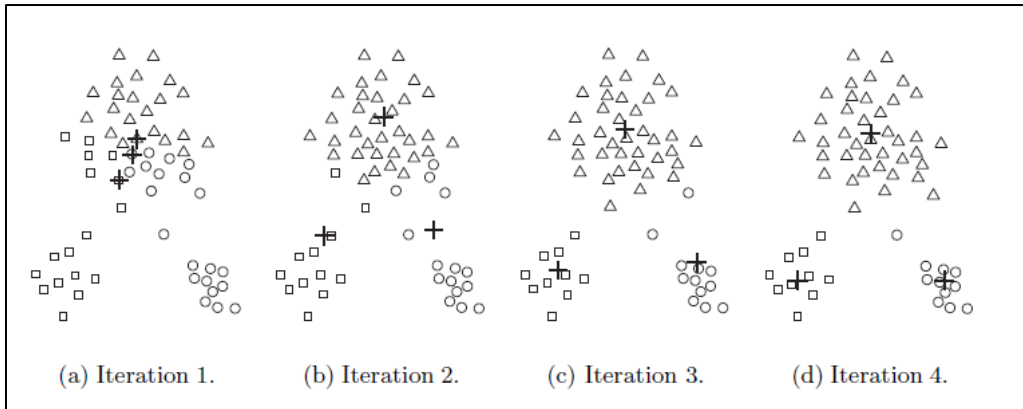


Figure 2.1: Using K-means Algorithm Operation to find three cluster in sample data [28].

2.3.2 X-mean algorithm

X-mean is K-mean extended by an improve structure part through efficient estimation of the number of cluster automatically [24]. That means we do not need to enter the number of clusters by ourselves. The x-mean algorithm starts with K (k : number of cluster) equal to the lower bound of the given range and continues to add centroids where they are needed until the upper bound is reached. During this process, the centroid set that achieves the best score is recorded, and this is the one that is finally output.

2.4 Major imbalanced class distribution techniques

The imbalanced data problem in classification can appear in two different types of data sets: binary problems, where one of the two classes comprises considerably more samples than the other and multi-class problems, where the applications have more than two classes and unbalanced class distribution hinder the classification performance. In order to overcome the class imbalance problem, many approaches have been introduced. Most research efforts on imbalanced data sets have traditionally concentrated on two-class problems. In this section, we review various techniques, which have been proposed in two-class and multi-class imbalanced data set problem.

2.4.1 Two class problem

It occurs when there are significantly fewer training instances of the first class compared to other one [23]. For example, in credit card usage data there are very few cases of fraud transactions as compared to the number of normal transaction. So, the instances of this data set belong to either fraud class or normal class only. For two-class problem, researcher proposed many solutions to the class imbalance problem at both data level and algorithm level. In data level different re-sampling techniques are applied to balance class distribution, such as re-sampling techniques and multi classifier committee approach. In algorithm level solutions try to adapt existing classifier learning algorithms to strengthen learning with regards to the small class, such as recognition based learning, ensemble learning, and cost-sensitive learning. In the next sub section we describe these approaches.

2.4.1.1 Re-sampling Techniques

One of the common approaches to tackle class imbalance problem is sampling. Sampling methods modify the distributions of the majority and minority class in the training data set to obtain a more balanced number of instances in each class [23]. To minimize class imbalance in training data, there are two basic methods, under sampling and over sampling.

➤ Under-sampling

It removes data from the original data set by randomly select a set of majority class examples and then remove this sample [18]. Hence, an under sample approach is aim to decrease the skewed distribution of majority class and minority class by lowering the size of majority class [32]. Under-sampling is suitable for large application where the number of majority samples is very large and lessening the training instances reduces the training time and storage [23]. Figure 2.2 illustrate the distribution of samples in a dataset before and after apply under sample approach. For example, from the Figure 2.2 we find the red circle is represent minority class which has two instances. So, for this reason we take randomly only two instances from other circles: black, blue and green which are represent majority classes in this case. The drawback of this technique is that there does not exist a control to remove patterns of the majority class, thus it can discard data potentially important for the classification process [13], which degrade classifier performance.

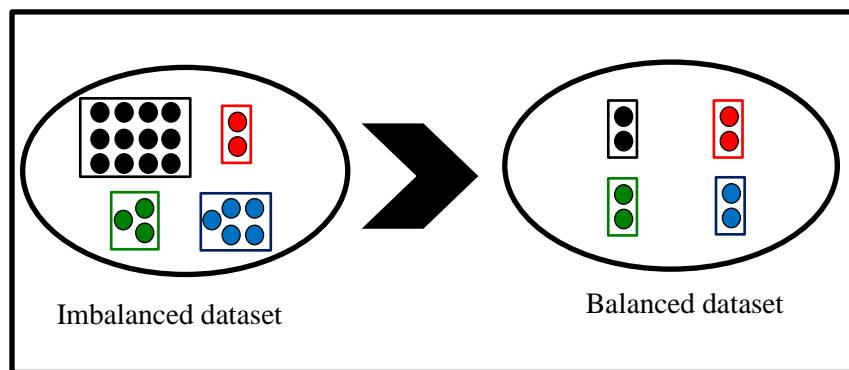


Figure 2.2: The distribution of samples before and after apply under sample approach.

One of research related to under sampling approach is in [32], where the authors proposed approach that is used under sampling approach with clustering algorithm which is named, **under sample based on clustering approach**.

➤ **Under sample based on clustering approach**

It is combining between under sample approach and clustering technique. The authors first cluster all the training samples in to some clusters. The main idea is that there are different clusters in a dataset, and each cluster seems to have distinct characteristics. They define the number of majority class samples and minority class samples in the each cluster as $size_{MA}$ and $size_{MI}$, respectively. Therefore, the ratio of the number of majority class samples to the number of minority class samples in the each cluster is $size_{MA} / size_{MI}$. They suppose the ratio of $size_{MA}$ to $size_{MI}$ in the training dataset is set to be $m: 1$ ($m \geq 1$). They select a suitable number of majority class samples from each cluster by considering the ratio of the number of majority class samples to the number minority class samples in the each cluster. The number of selected majority class sample in the cluster is shown in Equation 2.11:

$$SSize_{MA} = (m \times Size_{MI}) \frac{Size_{MA}/Size_{MI}}{\sum Size_{MA}/Size_{MI}} \dots\dots\dots 2.11$$

In Equation 2.11, m is the ratio of the number of majority class samples to the number minority class samples in the cluster. $Size_{MA}$ is number of majority class samples and $Size_{MI}$ is number minority class samples. $M \times Size_{MI}$ is the total number of selected minority class samples that is supposed to have in the final training dataset. $\sum Size_{MA} / Size_{MI}$ is the total ratio of the number of majority class samples to the number of minority class samples in all clusters. After determining the number of majority class samples which are selected in the each cluster by using Equation 2.9, choose randomly majority class samples from each cluster. Now, the total number of selected majority class samples is equals the total number of existing minority class samples. Finally, they combine the whole minority class samples with the selected majority class samples to construct a new balance training data set. Table 2.4 shows the steps for under sample based on clustering approach. Also Figure 2.3 illustrate simple example on under sample based on clustering approach.

Table 2.4: The structure of under sample based on clustering approach.

Step 1	Determine the ratio of the number of majority class samples to the number minority class samples in the cluster is set to be 1:1
Step 2	Cluster all the samples in the data set in to some clusters.
Step 3	Determine the number of selected majority class samples in each cluster by using Equation 3.1, and then randomly select the majority class samples in each cluster.
Step 4	Combine the selected majority class samples and all the minority class samples to obtain the balance training dataset.

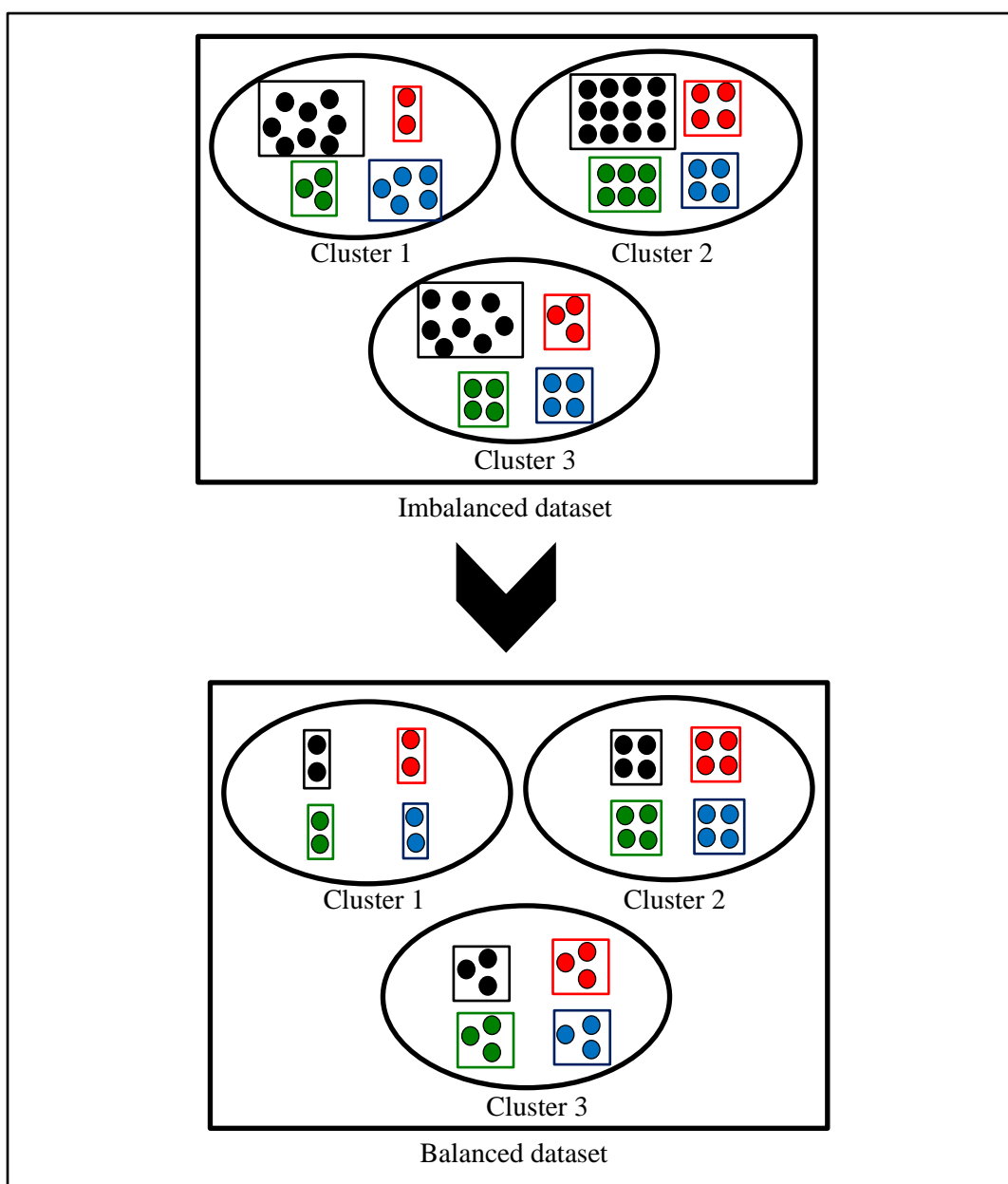


Figure 2.3: Simple example on under sample based on clustering approach.

In Figure 2.3 we assume there are four classes: red circle represents minority class samples and black, blue and green circle represent majority class samples. We cluster data in to three clusters and we note the red circle class has the smallest number of instances in each cluster and we note the other classes have greater number of instances than the red circle class in each cluster. Therefore, we look to class which has smallest number of instances to apply under sample approach on other classes based on smallest class. After that we find each class has the same or close number of instances to each other classes in each cluster. Finally, we combine whole classes' instances to create new balance training data set as illustrate in box two in Figure 2.3.

➤ **Over sampling**

It is a method to adding a set of sampled from minority class by randomly select minority class examples and then replicating the selected examples and adding them to data set [18]. The advantage is that no information is lost, all instances are employed. However, the major problem of this technique is leads to a higher computational cost. 2.4 illustrate the distribution of samples in a dataset before and after apply over sample approach. For example, from the Figure 2.4 we find the black circle is represent majority class which has twelve instances. So, for this reason we replicate instances from other circles: black, blue and green which are represent minority classes until they reach to twelve instances approximately in this case.

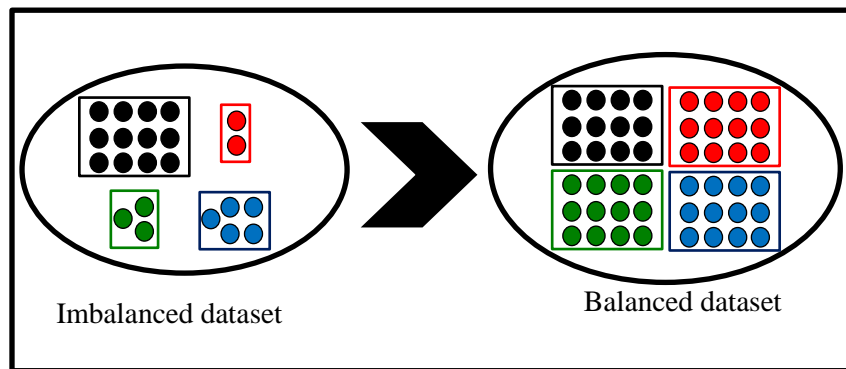


Figure 2.4: The distribution of samples before and after apply over sample approach.

The drawback of this technique is if some of the small class samples contain labeling error, adding them will actually deteriorate the classification performance on the small class [23]. There are many of research related to over sampling approach as in [8], the authors proposed

approach that is used over sampling approach with clustering algorithm which is named, **over sample based on clustering approach**, and as in [5], the author present **Synthetic Minority Over-sampling TEchnique (SMOTE) approach** which is consider as a one of the famous over-sampling approach.

➤ **Over sample based on clustering approach**

It is combining between over sample approach and clustering technique. The authors in [8] use k-mean clustering technique. This procedure takes a random set of K examples form each cluster for majority and minority classes and computes the mean feature vector of these examples, which is designated as the cluster center. Next, the remaining training examples are presented one at a time and for each example, the Euclidean distance vector between it and each cluster center is computed. Each training example is then assigned to the cluster that exhibits the smallest distance vector magnitude. Lastly, all cluster means are updated and the process is repeated until all examples are exhausted (i.e., only one cluster mean is essentially updated for each example). Now, the minority class samples are randomly oversampled until reach the same number of majority class samples in each cluster. Finally, they obtain new data set which contains the same number of samples from majority and minority classes. Table 2.5 shows the steps for over sample based on clustering approach. Figure 2.5 illustrate simple example on over sample based on clustering approach.

Table 2.5: The structure of over sample based on clustering approach.

Step 1	Determine the ratio of the number of majority class samples to the number minority class samples in the cluster is set to be 1:1
Step 2	Cluster all the samples in the data set in to some clusters.
Step 3	Compute number of majority class samples and the number of minority class samples in each cluster.
Step 4	Adding a set of sampled by randomly select of minority class samples and then replicating the selected examples until reach same or close to number of majority class samples in each cluster.
Step 5	Combine the majority class samples and all the minority class samples to obtain the balance training dataset.

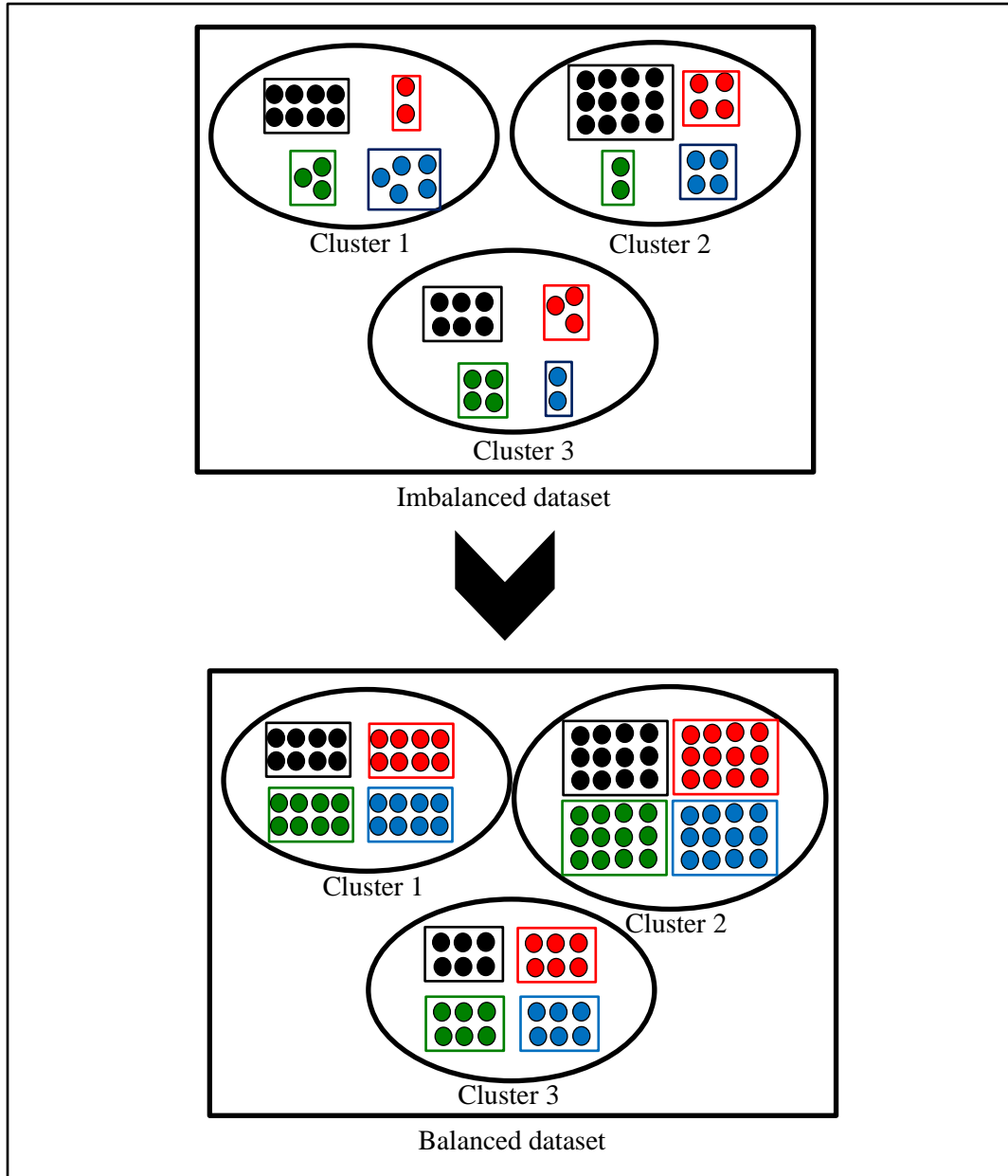


Figure 2.5: Simple example on over sample based on clustering approach.

In Figure 2.5 we assume there are four classes: black circle represents majority class samples and red, blue, green circle represent minority class samples. We cluster data in to three clusters and we note that the black circle class has the greatest number of instances in each cluster and we note the other classes have smaller number of instances than the black circle class in each cluster. Therefore, we look to class which has largest number of instances to apply over sample approach on other classes. After that we find each class has the same or close number of

instances to each other classes in each cluster. Finally, we combine whole classes' instances to create new balance training data set as illustrate in box two in Figure 2.5.

➤ **Synthetic sampling with data generation**

One of the famous over-sampling approaches is SOMTE (Synthetic Minority Over-sampling TEchnique). SMOTE was introduced by Cieslak and Chawla [5], who suggested a local implementation of sampling based on create “synthetic” instances from existing minority class samples. The SOMTE is a powerful method that has shown a great deal of success in various applications [18]. This approach is proposed in [5]. They created extra training data by performing certain operation on real data. The SMOTE algorithm creates artificial data based on the feature space similarities between existing minority examples. Specifically, for subset of minority class samples, consider k-nearest neighbors for each instance from minority class samples, for some specified integer K; the k-nearest neighbors are defined as the K elements of set of minority class samples whose Euclidian distance between itself and instance under consideration exhibits the smallest magnitude along the n-dimension of feature space. To create a synthetic sample, randomly select one of the k-nearest neighbors, and then multiply the corresponding feature vector difference with random number between [0,1], and finally add this vector to instance as shown in Equation 2.12.

$$x_{new} = x_i + (\hat{x}_i - x_i) \times \delta \dots\dots\dots 2.12$$

Where x_{new} represents new instance, x_i is the minority under consideration, \hat{x}_i is one of the k-nearest neighbors for x_i and δ is random value between 0 and 1. Table 2.6 shows simple example of calculation of random synthetic samples.

Table 2.6: Simple example of generation of synthetic examples (SMOTE).

<p>Consider a sample (6, 4), let (4, 3) be its nearest neighbor and assume (x, y) is a new synthetic example. (6, 4) is the sample for which k-nearest neighbors are being identified. (4, 3) is one of its K-nearest neighbors.</p> <p>Let:</p> <p>$X = 4 - 6 = -2$</p> <p>$Y = 3 - 4 = -1$</p> <p>$(x, y)_{new} = (6,4) + \text{rand}(0-1) \times (-2, -1)$</p> <p>rand(0-1) generates a random number between 0 and 1</p>

2.4.1.2 Multi Classifier Committee Approach

Multi classifier committee approach uses all information of a training dataset. Multi classifier committee approach divides the samples with majority class randomly into several subsets, and then takes every subset and all the samples with minority class as training dataset, respectively. The number of the subsets depends on the ratio of majority size to minority size [32]. For example, suppose in a dataset, the size of majority is 10 samples and the size of minority is 2 samples. If we think the best ratio of majority size to minority size is 1:1 in a training dataset, then the number of training subsets will be $10/2 = 5$. Each of these 5 subsets contains minority and a subset of majority that both sizes are 2, and the ratio of them is exactly 1:1 as illustrate in Figure 2.6.

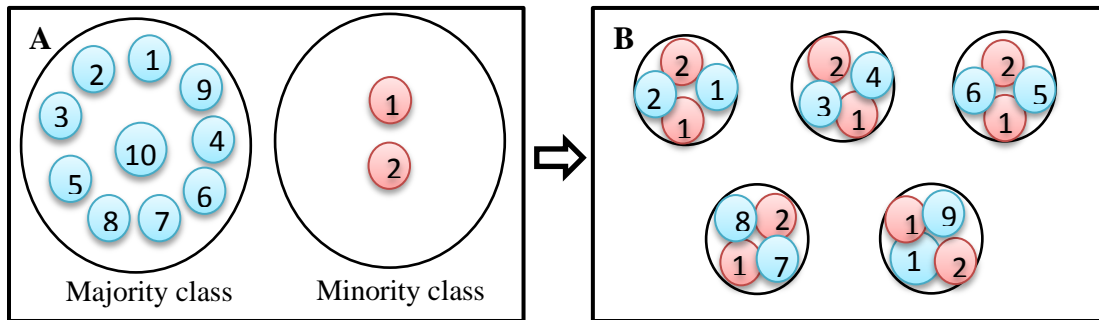


Figure 2.6: (A) Imbalanced data set. (B) Balanced data set.

2.4.1.3 Recognition Based Learning

A recognition-based or one-class approach is another solution where the classifier is modeled on the example of the target class (minority class) in the absence of examples of the non-target class. However, it focuses on the separation between the minority and the majority classes. There are two main strategies for one-class learning. The first one is tries to recognize instances of the target class rather than discriminate between instances of all classes. In this case, the minority class can be viewed as the target class, whereas the majority class will be the outlier class. As a result, the goal of this strategy is to define a boundary around the target class such that as many objects as possible of the target classes are recognized, while a minimal amount of outliers are accepted. The second approach to one-class learning uses instances of both classes to make predictions about the target class, and uses internal bias strategies during the learning process to

achieve this goal [9]. Finally, one-class learning is particularly useful on extremely unbalanced data sets with a high dimensional noisy feature space [13].

2.4.1.4 Ensemble Learning

The idea of classifier ensemble learning is to construct multiple classifiers from the original data and then combine their predictions when classifying new objects (unknown class). There are a number of ensemble models proposed to solve real-world applications, such as Bagging, Random forests and Boosting.

Bagging also called bootstrap aggregation, is based on constructing different specialized classifiers. It does by providing each classifier with a different training bag, which is sampled uniformly and with replacement from the original training set. Usually, minority training instances are sampled with a different ratio than majority instances, such that over-/under-sampling is performed in each training set. This allows each classifier to focus more (specialize) on specific aspects of the minority data. After a set of different classifiers is trained, their predictions are combined by voting. As a result, the ensemble will have a better grasp of the relevant concepts than a single classifier, since mistakes made by each classifier are neglected by the voting scheme. Bagging proves especially successful when applied to classifier learning methods that are sensitive (instable) to modifications of the input data [9].

Random forest is an ensemble classifier that consists of many decision trees, where each tree is generated based on an independent set of random vectors of a data set. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest) [35].

Boosting is a method for providing the performance of a weak learning algorithm. Boosting algorithm, such as AdaBoost algorithm (Adaptive Boosting) generates a set of classifiers by re-sampling like bagging, but the two algorithms differ substantially. The AdaBoost algorithm generates the classifiers sequentially, while Bagging can generate them in parallel. Each training example keeps a weight in AdaBoost and is updated after each time of iteration of constructing a classifier. The examples which are misclassified currently will be assigned larger weight, in order to be more likely to be chosen as a member of training subset

during re-sampling at next round. However, consecutive classifiers tend to focus on "hard" examples. A final classifier is formed using a weighted voting scheme-the weight of each classifier depends on its performance on the training set used to build it [30].

2.4.1.5 Cost-sensitive Learning

Cost-Sensitive Learning is a type of learning in data mining that takes the misclassification cost into consideration. The concept of the cost matrix is fundamental to the cost-sensitive learning methodology. The cost matrix can be considered as a numerical representation of the penalty of classifying examples from one class to another [13]. Table 2.7 defined cost matrix for two classes.

Table2.7: Cost matrix for two class [29].

	Predicted positive	Predicted negative
Actual positive	TP (number of True Positives) or C(+,+)	FN (number of False Negative) or C(+,-)
Actual negative	FP (number of False Positives) or C(-,+)	TN (number of True Negative) or C(-,-)

True Positives (TP) denote the number of positive examples correctly recognized as being positive, and False Negatives (FN) represent the number of positives incorrectly recognized as being negative. TN and FP represent the number of negative examples correctly identified as being negative, and incorrectly identified as being positive, respectively.

Most classifiers assume that the misclassification costs (false negative and false positive cost) are the same. In most real-world applications, this assumption is not true. For example, in customer relationship management, the cost of mailing to non-buyers is less than the cost of not mailing to the buyers [29].

For two-class problem, if we assume $C(\text{Min},\text{Maj})$ as the cost of misclassifying a majority class example as minority example and $C(\text{Maj},\text{Min})$ as the cost of misclassifying a minority class example as majority example. The cost of misclassifying minority examples is higher than the cost of misclassifying majority examples ($C(\text{Maj},\text{Min}) > C(\text{Min},\text{Maj})$) and there is no cost for correct classification. In class imbalance problem, the objective of cost sensitive learning is to develop a hypothesis that seek to minimize the high cost errors (misclassifying a minority class) and the total misclassification cost [27].

A cost-sensitive classification technique takes the cost matrix into consideration during model building and generates a classifier that has the lowest cost. There are many different ways of implementing cost sensitive learning, but in general existing cost sensitive learning for dealing with imbalanced data sets can be divided in to three different categories: 1) weighting the data space, 2) making a specific classifier learning algorithm cost-sensitive and 3) using Bayes risk theory to assign each sample to its lowest risk class as in [13]. The drawback is this method assumes the misclassification costs are known but in practice the specific cost information is often depend on a number of factors that are not easily compared. Also Weiss found the cost sensitive learning may lead to over fitting during training [15]. Also there are many of researches which are combine between this approaches as in [10][20][27][29].

2.4.2 Multi-class Problem

The two class imbalanced data is not the only scenario where the class imbalance problem prevails. The multi class imbalance problem is an extension of the traditional two class imbalanced data where a data set consists of k classes instead of two. While imbalance is said to exist in the binary class imbalance problem when one class severely outnumber the other class, extended to multiple classes the effects of imbalance are even more problematic [19]. For example, in car evaluation data set from [38], all instances are grouped in to four classes which is determine the car acceptability. Class “unacc” is considered as a majority class which has 70.03% from all data. Other classes “acc”, “good” and “vgood” are considered as a minority class with only 22.22%, 3.99% and 3.76 % of the entire samples, respectively. In the case of multi-class data sets, it is much more difficult to define the majority and minority classes. For example, one class A can be majority with respect to class B, but minority with respect to another class C. or there are two or more minority classes with respect to one majority class.

The multi class imbalance problem is therefore interesting for two important reasons. First, most learning algorithms do not deal with the wide variety of challenges multi class imbalance presents. Second, a number of classifiers do not easily extend to the multi class domain. So, there are few works addressing the imbalance multi-class problem [13]. Most methods designed to solve this problem are based on splitting the K-class classification problem into a number of smaller two-class subproblems. For each subproblem, an independent binary classifier is built. Then, the results of the binary classifiers are combined to get the classification

result. Several techniques were proposed for decomposing the multi-class problem, such as in [14][22].

2.5 Summary

This chapter gave an overview for basic theoretical foundation about data mining, classification and its method. Then, it introduced an overview of major existing techniques related to imbalanced class distribution problem which is used in two-class and multi-class imbalanced data set problem. The next chapter will review the related work that was done for imbalanced class distribution problem domain.

CHAPTER 3: Related Works

Some of researches have been done for imbalanced class distribution problem domain. These approaches have been introduced at both algorithm and data levels. In this chapter we review the most important ones.

3.1 Data Level Solutions

At the data level, the objective is to re-balance the class distribution by re-sampling the data space including over sampling instances of the positive class and under sampling instances of the negative class [8][12][23][26][32]. The following are some well known works on imbalanced data mining implemented at data level:

First, Yen and Lee in [32] proposed cluster-based under-sampling approaches for selecting the representative data as training data. The first one is called Under-sampling based on clustering (SBC) and the other five methods are called Under-sampling based on clustering and distances between samples (SBCNM-1, SBCNM-2, SBCNM-3, SBCMD and SBCMF). The difference between the SBC method and the five proposed under-sampling methods is the way to select the majority class samples from each cluster. In SBC, majority class samples are selected randomly. For the five proposed methods, the majority class samples are selected according to the distances between the majority class samples and the minority class samples in each cluster. In the experiments they use neural network for classify instances and k-mean clustering algorithm for their approaches. To evaluate the performance of a classifier, the authors used two criteria: F-measure and time consuming. They compare their approaches with the other under-sampling approaches on synthetic data sets and on two real datasets which is represented the two classes case. Finally, they found SBC has better prediction accuracy and stability than other methods also it has fast execution time. But when the datasets contain more exceptional samples and disordered samples, SBCMD approach has better prediction accuracy and stability. Other approaches (SBCNM-1, SBCNM-2, SBCNM-3, SBCMD and SBCMF) do not have stable performances in their experiments. The five methods take more time than SBC on selecting majority class sample as well. The disadvantage for their approaches that is not able to deal with multi-class problem. Also we note they define the number of cluster manually in their experiments and they used one classifier (neural network) for classification. They applied most

experiments on synthetic data sets and only two real datasets used during their experiments so we believe that is not enough. In general the F-measure term does not exceed 79% which is considering low. Also, as stated before under sampling approach may lose useful information about the majority class.

Second, Chawla et al. in [5] presented the Synthetic Minority Over-sampling TEchnique (SMOTE) approach, which is generate synthetic minority samples by interpolating between two minority samples that lie together at an over sampling rate. They focused on two class problem. For every minority sample, find its k (set to 5) nearest neighbors of the same class, then select the samples randomly among them according to the over sampling rate. The new synthetic samples are generated along the line between the minority sample and selected nearest neighbors. Experiments are performed using C4.5 decision tree, Ripper and Bayes classifier. The method is evaluated using the area under the Receiver Operating Characteristic curve (AUC) and ROC convex hull strategy. This method has been validated to be effective. However, they apply only for binary class.

Third, Chen et al. in [8] proposed a novel over sampling strategy to handle imbalanced data based on ensembles, named Cluster Ensembles Based SMOTE (CE-SMOTE), which first used cluster ensemble to generate multiple partitions. Next, the clustering consistency index lower than the given threshold value are identified. Finally, they over sample these boundary minority samples to balance the original data set, and then classifier can be trained on the over sampled data set applying a lot of traditional classification methods, such as the C4.5 decision tree. In order to test the performance of CE-SMOTE, they applied their experiments on ten imbalanced data sets from the UCI machine learning repository [35]. All data sets consider as a two class problem. They use the minority class as the positive class, and the majority class as the negative class. Also, they use F-measure and G-mean to evaluate the performances of the compared algorithms for the minority class and the whole data set respectively. Finally, we note they define the number of cluster manually in their experiments. They reach to average F-measure is 73.44 which is consider low. Also, they not applied their approach on multi class learning problem.

Fourth, Zhang and Mani in [34] presented the compared results within four informed under-sampling approaches and random under sampling approach. The first method “NearMiss-

1” selects the majority class samples which are close to some minority class samples. In this method, majority class samples are selected while their average distances to three closest minority class samples are the smallest. The second method “NearMiss-2” selects the majority class samples while their average distances to three farthest minority class samples are the smallest. The third method “NearMiss-3” take out a given number of the closest majority class samples for each minority class sample. The fourth method “Most distant” selects the majority class samples whose average distances to the three closest minority class samples are the largest. Finally, they find the “NearMiss-2” method and random under sampling method perform the best. They applied their experiments on one dataset so we believe that is not enough. Also we must do not forget under sampling approach may lose useful information about the majority class.

Fifth, Nguyen et al. in [23] introduced a new approach to deal with the class imbalance problem by combining both unsupervised clustering and supervised learning to handle imbalanced data set and applied this learning approach for training feed-forward neural networks. They first proposed a new under sampling method based on clustering. A clustering technique is employed to partition the training instances of each class independently into smaller set of training prototype patterns. Then a weight is assigned to each training prototype to address the class imbalance problem. The weighting strategy is introduced in the cost function such that the class distributions become roughly even. In the extreme imbalance cases, where the number of minority instances is small, they applied unsupervised learning to resample only the majority instances, and select cluster centers as prototype samples, and keep all the small class samples. The experimental results showed that the proposed approach can effectively improve the classification accuracy of the minority classes. The disadvantage for their approach that is not able to deal with multi-class problem. Also we note they define the number of cluster manually in their experiments. In general the F-measure term does not exceed 79% which is considering low. Also we must do not forget under sampling approach may lose useful information about the majority class.

3.2 Algorithm Level Solutions

At algorithm level, solutions try to adapt the existing classifier learning algorithm to bias towards the positive class [8][12][23][26][32]. The following are some well known works on imbalanced data mining implemented at algorithm level:

First, Ghanem et al. in [14] proposed a new approach, named Multi-IM, to handle the imbalanced situation between multiple pattern classes. Multi-IM derives its fundamentals from the probabilistic relational technique (PRMs- IM), designed for learning from imbalanced relational data for the two-class problem. Multi-IM is based on extending PRMs-IM to the multi-class problem by embedding the balancing concept of PRMs-IM in All-and-One (A&O) approach. Multi-IM firstly follows the A&O approach by training One-Against-All (OAA) approach and One-Against-One (OAO) approach. Consider a three class problem ($C_1;C_2;C_3$), with imbalanced data distribution. For the OAA, they construct classifiers ($OAA_1;OAA_2;OAA_3$), one classifier for each class. The training data of OAA_i includes all the samples of C_i as positives and all the other samples of the other classes as negatives. For the OAO, they build three classifiers ($OAO(1;2);OAO(1;3);OAO(2;3)$) for each pair of classes. The training data of $OAO(i;j)$, includes the samples of C_i and C_j as positives and negatives, respectively. To address the imbalanced problem, the balancing concept of PRMs-IM is used in building the classifiers of the OAO and OAA. Thus, the training data of each classifier is used to obtain balanced subsets that include all the minority samples and a similar number of random samples of the majority class. Then, an independent classifier is trained on each balanced subset. For classifying new samples, the OAA system is used to find the top two candidates ($C_i;C_j$). Then, the corresponding binary OAO classifier $OAO(i;j)$ is used to find the final answer. Although the result of Multi-IM was generally better than other strategies, but it can be better. Also, they applied an algorithmic approach, which dislike data approach, work only with one classifier which is namely Multi-IM. Finally, they use AUC as evaluation measure which is not applicable for multi class evaluation as stated by [16]

Second, Chen et al. in [7] proposed a new methodology which combines “Information Granulation” and LSI (Latent semantic indexing) to solve class imbalance problems. The method contains two major parts: Part one; they construct IGs (clusters) by using K-means. Then they set

the “granularity selection criteria” (i.e., the threshold of H-index and U-ratio) to determine the suitable level of granularity.

$$H - index = \sum_m \frac{i}{n} / m \dots\dots\dots 3.1$$

Where i = amount of objects possessing the majority class, n= number of all objects in one granule, m= number of all IGs.

$$U - ratio = \frac{u}{m} \dots\dots\dots 3.2$$

Where u=number of undistinguishable granules, m= number of all IGs.

Next, they checked the data type. If the data is continuous, it will be discretized. After that they describe the constructed IGs by concept of sub-attributes. Part two, they reduce number of dimensions of the sub-attributes through determine the optimal number of features (LSI) and finally build feed-forward neural network as classifier for calculate the classification accuracy. They evaluate the performance of classifiers using three measures: accuracy (based on the confusion matrix), execution time and storage space. They applied experiments on four data sets (two balanced and two imbalanced data). Finally, they found proposed GrC method has good potential in handling imbalanced data. In addition, not only improves the classification performance, but it also saves much execution time and storage space compering with two other methods: 1- using NN to discover knowledge from IGs (without implementing LSI) and 2- using NN to extract knowledge from numerical data. The disadvantage for their approach is complex and contains many steps. Also the average overall accuracy was 72.68% which is consider low and they used one classifier (neural network) for classification.

Third, Murphey et al. in [22] proposed a new pattern classification algorithm, One Against Higher Order (OAHO), that effectively learn multi-class patterns from the imbalanced data. The idea is building a hierarchy of classifiers based on the data distribution. OAHO constructs K - 1 classifiers for K classes in a list of {C1;C2; :::;CK}. The first classifier is trained using the samples of the first class in the list C1 against all the samples of all the other classes. Then, the second classifier is trained using the samples of the second class in the list C2 against

the samples of the higher ordered classes $\{C_3; \dots; C_K\}$, and so on until the last classifier is trained for C_{K-1} against C_K . To classify new samples, a hierarchical approach is used. Thus, the sample is first classified by the first classifier. If the sample is classified as C_1 , then the process terminates and the sample is assigned to class C_1 . Otherwise, the second classifier is used to classify the sample, and so on till the last classifier. To resolve the imbalanced class problem in this approach, the classes are ordered in descending order based on the size of the samples in each class. This order is chosen to reduce the imbalanced situation, in which the small classes are grouped together against the majority class.

They implemented OAHO algorithm using a base architecture of one-hidden layer neural networks trained with feed-forward back propagation (BP) learning algorithm. They applied their experiments on all minority classes in Glass and Shuttle, the two extremely imbalanced data collection from the UCI machine learning data base and they found a system modeled by OAHO is effective and performed extremely well in machine learning from imbalanced training data. Although OAHO has been proposed to handle the imbalanced problem for multi-class classification, its performance is sensitive by the classifier order, as misclassification made by the top classifiers cannot be corrected by the lower classifier.

Fourth, Han and Mao in [15] presented an approach, namely fuzzy –rough k-nearest neighbor algorithm for imbalanced data sets learning to improve the classification performance of minority class. The approach defines the new fuzzy membership function that can reduce the disturbance of majority class to minority class. Considering the fuzziness and roughness existed in the data set, and constructs fuzzy equivalent relation between the unlabeled instances and its k nearest neighbors. They used four data sets to validate the performance of their approach. Two are imbalanced data sets of two classes. Two are multi class data sets. As multi class problem can be transformed into two class problem, for multi-class datasets, they respectively select class “1”, “2” as the minority class, and combine the remainder classes into the majority class. They compare their approach with k-nearest neighbor algorithm and fuzzy k-nearest neighbor. The performance of fuzzy –rough k-nearest neighbor is better as it takes not only the fuzziness but also the roughness of the nearest neighbor of an instance into consideration. So, they find comparing with other two algorithms their approach is best and effectively improves the classification performance of minority class and its performance on the whole data set. The

average f-measure for whole data sets was 61.84 which is consider low. Also we note they solved imbalanced problem at algorithm level with modify k-nearest neighbor, but we think solved imbalanced problem at data level is better because after that we can use this data with different classifier. For this reason we work at data level to handle imbalanced class distribution.

Fifth, Adam et al. in [1] solved imbalanced data set problem through introduced feed forward ANN that is used particle swarm optimization (PSO). PSO is an advanced optimization intelligent technique that easy to implement in optimization problems and it has been successfully applied in various fields. The experimental results show that the proposed ANN model can achieve better performance to ANN classifier without using any sampling technique. It is able to solve imbalanced data set problems with better performance compared to the standard ANN. We note they solved imbalanced problem at algorithm level with modify ANN, but we think solved imbalanced problem at data level is better because after that we can use this data with different classifier. For this reason we work at data level to handle imbalanced class distribution.

From the previous works we can conclude that we note few of works proposed for multi class problem because it is much more difficult to define the majority and minority classes. We preferred to work at the data level than work at the algorithm level because at data level after preprocessing data we can use this data with different classifier but at algorithm level we need to modify each classifier that is used with imbalanced datasets. Also we note in the researches which is used clustering technique, they determine the number of cluster manually. However, in our research we try to test the results with clustering data automatically. Also we think the performance of minority class can be improved.

Summary

In this chapter we gave an overview about some of researches done in imbalanced class distribution problem. We draw a conclusion that we can improve multiclass imbalance problem which will be proposed in the next chapter. We will also give the steps of our methodology.

CHAPTER 4: Research Proposal and Methodology

This chapter explains our proposed approach and methodology which we followed in this research. Section one, presents general view of our proposed approach. Section two, will give description of the collecting various data sets for design experimental data. Section three, perform pre-processing to convert data set from imbalanced data to balanced data. Section four, applies the model by using data mining method. Section five, evaluates the model using accuracy and F-measure to evaluate classification performance.

To implement and evaluate this approach we use the following methodology steps as presented in Figure 4.1:

1. **Collection data:** we collect various real domain, characteristics and sizes from UCI machine learning repository.
2. **Preprocessing data:** through apply our proposed approach which is combining between both SOMTE approach and clustering approach.
3. **Apply the model:** through implement our model by using one of the classification algorithms such as: Rule induction, Naïve Bayes, decision tree and neural network.
4. **Evaluate the model:** to evaluate the classification performance of our model, we use accuracy and F-measure.
5. **Comparing phase:** we applied two comparison:
 - A. Compare performance before using our proposed approach and after using it.
 - B. Compare performance between proposed approach and other works which can be used for imbalanced problem.

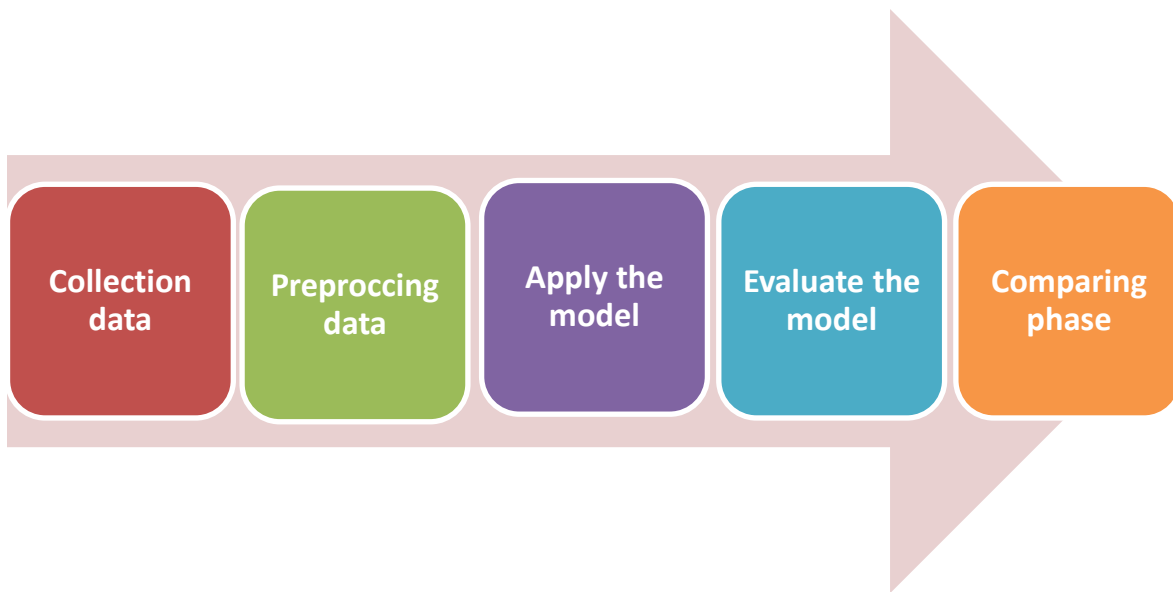


Figure 4.1: Methodology Steps

4.1 Approach combines between both Synthetic Minority Over-sampling TEchnique (SOMTE) approach and clustering approach

Our main objective in this research is to try to increase the classification accuracy of minority class by avoiding the drawbacks of the existing methods. For that, we propose an efficient approach combine between both Synthetic Minority Over-sampling TEchnique (SOMTE) approach and clustering approach which is able to deal with multi class imbalanced data problem. To do that we propose the following steps in the preprocessing stage which are:

1. Clustering the data into clusters using random clustering algorithm provided by RapidMiner environment to obtain clusters with equal number of instance approximately.
2. Also in other experiments we used X-mean algorithm to test the effect of determine the number of clusters automatically.
3. Use over sampling which duplicates the sample of the minority class and adding them to data set.
4. Use SMOTE approach which generates new synthetic minority instances by interpolating between several minority examples that lie close together.

Figure 4.2 presents general view of our proposed approach.

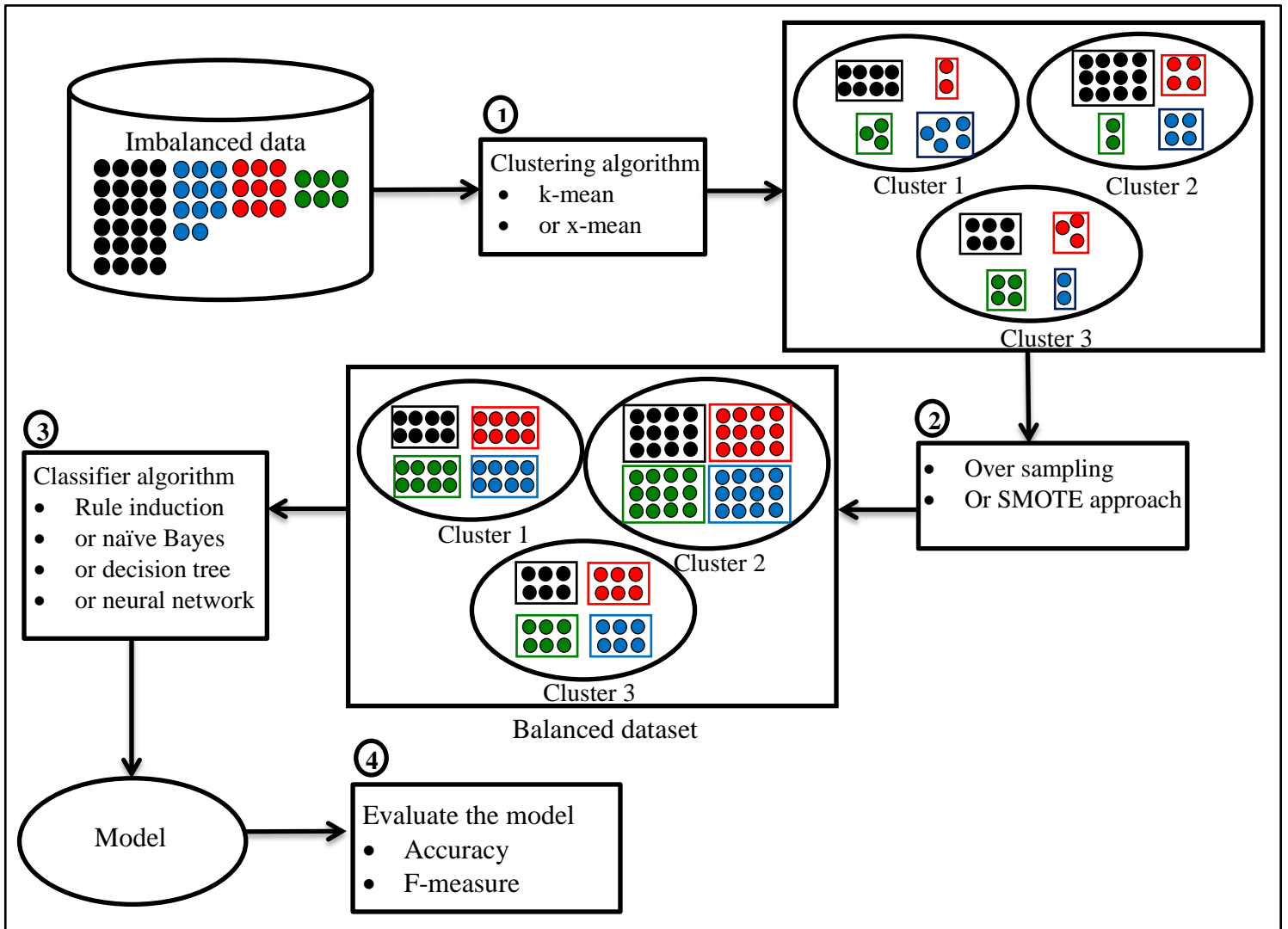


Figure 4.2: General view of our proposed approach

4.2 Collection data

Many real applications face the imbalanced class distribution problem especially in UCI Machine Learning Repository [11]. The UCI is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. For our experiments, six data sets are chosen from different real domain, characteristics and sizes. Five from data sets (page blocks, cardiocography, car evaluation, auto MPG and glass identification) represent multi class problem case and the other

one data set (breast cancer-w) represent two class problem case. General information about these eight data sets is tabulated in Table 4.1.

Table 4.1: Summary of data sets

Data set	Data type	# instance	# Attribute	# class	Class distribution	% Class distribution	Reference
Page Blocks	Real	5473	10	5	<ul style="list-style-type: none"> Text: 4913 Horiz-line: 329 Picture: 115 Vrt-line: 88 Graphic: 28 	Text:89.8% Horiz-line:6% Picture:2% Vrt-line:1.6% Graphic:0.6%	[36]
Cardiotocography	Real	2126	23	3	<ul style="list-style-type: none"> Normal: 1655 Suspect: 295 Pathology: 176 	Normal:77.8% Suspect:13.9% Pathology:8.3%	[37]
Car Evaluation	Categorical	1728	6	4	<ul style="list-style-type: none"> Unacc: 1210 Acc: 384 Good: 69 Vgood: 65 	Unacc:70% Acc:22% Good:4% Vgood:4%	[38]
Auto MPG	Real	398	8	5	<ul style="list-style-type: none"> Class 4: 204 Class 8: 103 Class 6: 84 Class 3: 4 Class 5: 3 	Class 4: 51.3% Class 8: 25.7% Class 6: 21% Class 3: 1% Class 5: 1%	[39]
Glass Identification	Real	214	10	6	<ul style="list-style-type: none"> Class 2: 76 Class 1: 70 Class 7: 29 Class 3: 17 Class 5: 13 Class 6: 9 	Class 2: 35.5% Class 1: 32.7% Class 7:13.55% Class 3: 7.9% Class 5: 6% Class 6: 4%	[40]
Breast Cancer - w	Real	699	10	2	<ul style="list-style-type: none"> Benign: 458 Malignant: 241 	Benign: 65.5% Malignant:34.5%	[41]

➤ **Page Block dataset:**

Page block dataset from [36], the problem consists in classifying all the blocks of the page layout of a document that has been detected by a segmentation process. This is an essential step in document analysis in order to separate text from graphic areas. This data set contain of 5473 examples come from distinct documents. Each observation concerns one block. The instances are described by 10 attributes, of which all are numeric data type. All data are classified in to 5 classes: text (1) represents 89.8% from all data, horizontal line (2) represents 6.0% from all data, picture (3) represents 2.1% from all data, vertical line (4) represents 1.6% from all data and

graphic (5) represents 0.5% from all data as illustrate in Table 4.1. So, we note in this data set the imbalance between classes is very high especially between text class and other classes.

➤ **Cardiotocography:**

Cardiotocography dataset from [37], this data set contains of 2126 instances. It uses to classify of fetal heart rate (FHR) signals and unterine contraction (UC), the important feature of Cardiotocograms classified by expert obstetricians. Each observation concerns one measurement. The instances are described by 23 attributes, of which all are numeric data type. The classification is with respect to a fetal heart rate class code (N-Normal, S-Suspect and P-Pathologic).

➤ **Car Evaluation Dataset:**

Car evaluation dataset from [38], this dataset was derived for car evaluation. There are 1728 instances with each described by 6 nominal ordered attributes. All data are grouped in to 4 classes. Table 4.1 describes the class distribution. Class acc for car acceptability, good and vgood are small classes with only 22.22%, 3.99% and 3.76 of the entire samples, respectively.

➤ **Auto MPG dataset:**

The Auto MPG data set [39] used to classifying on the number of cylinders. The data set contains of 398 instances. The instances are described by 8 attributes, of which all are numeric data type. All data are grouped in to 5 classes. Table 4.1 describes the class distribution.

➤ **Glass Identification:**

Glass Identification from [40], this data set used in classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence. If it is give correctly identified. The data set contains of 214 instances. The instances are described by 10 attributes, of which all are numeric data type. All data are grouped in to 6 classes. Table 4.1 describes the class distribution.

➤ **Breast Cancer – w Data set:**

This breast cancer database [41] was obtained from the University of Wisconsin Hospital, Madison from Dr. William H. Wolberg. The data set includes 699 instances. The instances are described by 10 attributes, of which all are numeric data type. Each instance has one of two possible classes: benign which represents 65.5% from all data or malignant which represents 34.5% from all data. This data represents two class problem case.

4.3 Preprocessing Stage

In this section, we present our strategy which we followed to achieve our goals. We implemented the following steps:

- A. Classification experiments without preprocessing (Base line experiment).
- B. Apply under sample approach.
- C. Apply over sample approach.
- D. Apply under sample based on clustering approach.
- E. Apply over sample based on clustering approach.
- F. Apply over sample with use automatic cluster.
- G. Apply SMOTE approach.
- H. Apply SOMTE based in clustering approach (our proposed approach).

4.3.1 Classification experiments without preprocessing

In our experiments, we start with classify instances before done any change on data sets to test the classification accuracy of minority class. We used this experiment as a baseline for the whole experiments.

4.3.2 Under sample experiments

We apply under sample approach which is supposed to reduce the number of samples with the majority class. Hence, an under sample approach is aim to decrease the skewed distribution of majority class and minority class by lowering the size of majority class [32]. In this approach, first we look at minority class which has smaller number of instances, and then take the same number of instances from other majority classes. In order to do this we use sample

operator provided by RapidMiner environment [42]. This operator performs a random sampling from each majority classes. Finally, we obtain new data set with balance number of instances in each class.

4.3.3 Over sample experiments

We apply over sample approach which is duplicate the sample of the minority class and adding them to data set [18]. It is different than under sample approach so there is no information is lost, all instances are employed. In this approach, first we look at the majority class which has greater number of instance and then we replicate sample from other minority classes until reach to the same or close number of instance in majority class. Finally, we obtain new data set with balance number of instances in each class.

4.3.4 Under sample based on clustering experiments

We apply under sample based on clustering approach which was discussed in chapter 2. First, we cluster all the training samples in to some clusters. The main idea is that there are different clusters in a dataset, and each cluster seems to have distinct characteristics. So to obtain clusters with equal number of instance approximately, we use random clustering algorithm provided by RapidMiner environment. Then we compute the number instances of each class in all clusters. If a cluster has more majority class samples and less minority class samples, it will behave like the majority class samples. Therefore, the approach selects a suitable number of majority class samples from each cluster by considering the ratio of the number of majority class samples to the number minority class samples in the cluster which is equal 1 in our experiments. After determining the number of majority class samples which are selected in the each cluster by using Equation 2.5 from chapter 2, choose randomly majority class samples from each cluster by using sample operator provided by RapidMiner environment. Now, the total number of selected majority class samples is equals the total number of existing minority class samples. Finally, we combine the whole minority class samples with the selected majority class samples to construct a new balance training data set.

4.3.5 Over sample based on clustering experiments

We apply over sample based on clustering approach which was discussed in chapter 2. First, we cluster all the training samples in to some clusters. The main idea is that there are different clusters in a dataset, and each cluster seems to have distinct characteristics. So to obtain clusters with equal number of instance approximately, we use random clustering algorithm provided by RapidMiner environment. Then we compute the number instances of each class in all clusters. If a cluster has more majority class samples and less minority class samples, it will behave like the majority class samples. Therefore, we look at the majority class which has greatest number of instance and the replicate sample from other minority classes until reach to the same or close number of instance in majority class in each cluster. After that adding replicated sample to data set. Finally, combine between whole classes to produce new balance training data set.

4.3.6 Apply over sample with use automatic clustering approach

On the other hand, we think that is choosing the most appropriate number of clusters plays an important role. So, in our experiments we try to apply over sample approach after determining the optimal number of clusters for each data set. To find the optimal number of clusters we use the automatic method, which in our case x-means clustering algorithm provided by RapidMiner program. X-mean is K-mean extended by an improve structure part through efficient estimation of the number of cluster automatically [24]. That means we do not need to enter the number of clusters by ourselves. The x-mean algorithm starts with K (k: number of cluster) equal to the lower bound of the given range and continues to add centroids where they are needed until the upper bound is reached. During this process, the centroid set that achieves the best score is recorded, and this is the one that is finally output. Then, we apply the same steps of over sample based on clustering approach but with using X-mean clustering algorithm.

4.3.7 Apply SOMTE approach

We apply SOMTE (Synthetic Minority Over-sampling TEchnique) approach which is provided WEKA environment [43]. WEKA is a popular suite of machine learning software written in Java, developed at the University of Waikato. It is free software available under the GNU General Public License. WEKA provides a large collection of machine learning algorithms

for data pre-processing, classification, clustering, association rules, and visualization, which can be invoked through a common Graphical User Interface. The SMOTE approach is different than over sample approach. Instead of merely replicating cases belonging to the minority class samples, it generates new synthetic minority instances by interpolating between several minority examples that lie close together.

4.3.8 Apply SOMTE based on clustering approach

We apply SOMTE based on clustering approach. In our approach, we use the SMOTE method provided by WEKA environment. So, we apply the same steps of over sample based on clustering approach with using SMOTE approach instead of use normal over sample approach. Table 4.2 shows the steps for SMOTE based on clustering approach.

Table 4.2: The structure of SMOTE based on clustering approach (our approach).

Input	Imbalanced data sets.
Step 1	Determine the ratio of the number of majority class samples to the number minority class samples in the cluster is set to be 1:1
Step 2	Cluster all the samples in the data set in to some clusters.
Step 3	<p>Compute number of majority class samples and the number of minority class samples in each cluster to determine the amount of minority class samples which is needed in each cluster.</p> <ul style="list-style-type: none"> - Given C as number of instances in the majority class. - For each minority class, - If d is number of instances in the class. - Duplicate d to d' such that $d' = C$.
Step 4	Adding a set of sampled by randomly select of minority class samples and then generate synthetic minority samples until reach same or close to number of majority class samples in each cluster.
Step 5	Combine the majority class samples and all the minority class samples to obtain the balance training dataset.
Output	Balanced data sets.

4.4 Apply the model

This section describes the major kinds of classification algorithms which are used in our research: Rule induction, Naïve Bayes, decision tree and neural network which are provided by RapidMiner environment. In the following sub-sections we present these classification algorithms and their settings which are used during experiments results.

4.4.1 Rule Induction

We used rule induction in our research which is consider as one of the most important techniques of machine learning that is extraction of useful if-then rules from data based on statistical significance. Figure 4.3 illustrate the settings of rule indication. We chose the information gain for the criterion term. The sample ratio and pureness was 0.9.

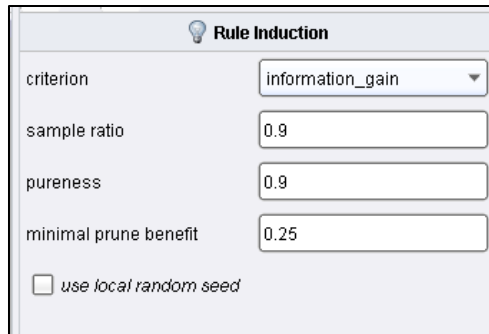


Figure 4.3: Settings of rule indication.

4.4.2 Naïve Bayes

We use naïve Bayes in our research which is consider as one of the most widely used classifiers. Figure 4.4 illustrate the settings of naïve Bayes. We use Laplace correction to prevent high influence of zero probabilities.

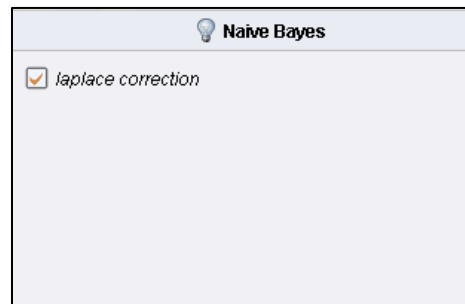


Figure 4.4: Settings of naïve Bayes.

4.4.3 Decision Tree

Tree-shaped structures that represent set of decisions. These decisions generate rules for the classification of a dataset. Figure 4.5 illustrate the settings of decision tree. We chose the gain ratio for the criterion term.

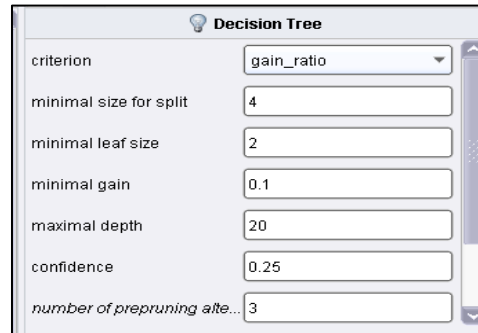


Figure 4.5: Settings of decision tree.

4.4.4 Artificial Neural Network

Neural network is a simulation of the human brain acquires knowledge through learning. Also it is nonlinear predictive models and resembles biological neural networks in structure. A multilayered perceptron network (MLPN) from the RapidMiner environment were trained on dataset using the feed forward back propagation (FFBP) algorithm with one hidden layer and the number of training cycles is 500. The learning rate was 0.3 and the momentum value was 0.2. Figure 4.6 illustrate the settings of neural network in our research.

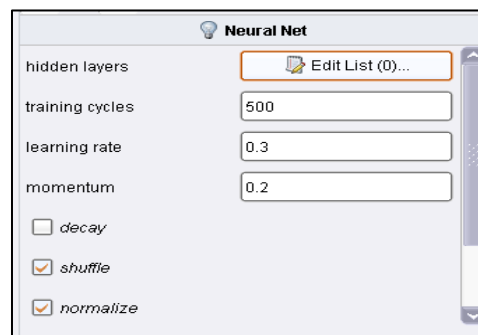


Figure 4.6: Settings of neural network.

4.5 Evaluate the model

Evaluating metrics play an important role to evaluate classification performance. Accuracy measure is the most commonly for these purposes. The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier [15]. However, for classification of imbalanced data, accuracy is no suitable metric since the minority class has very little impact on the accuracy as compared to that of the majority class [24]. For example, in a problem where a minority class is represented by only 1% of the training data and 99% for majority class, a simple strategy can be one that predicts the majority class label for every example. It can achieve a high accuracy of 99% may mean nothing to some application where the learning concern is the identification of the minority cases. Therefore, other metrics have been proposed to evaluate classifiers performance for imbalanced data sets. F-measure is one of measures that most relevant to imbalanced data. It is defined as the harmonic mean of recall and precision [23]. The recall is the ratio of the number of positive example correctly recognized and the number of all positive examples. The precision is the ratio of the number of positive examples correctly recognized and total number of examples (both positive and negative) recognized [34]. A high F-measure value signifies a high value for both recall and precision. It is evaluated when the learning objective is to achieve a balanced performance between the identification rate (recall) and the identification accuracy (precision) of a specific class [25]. F-measure which is shown in Equation 3.9

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{precision}}{\text{Recall} + \text{precision}} \dots\dots\dots 3.9$$

In our experiments, we use F-measure and compare it with accuracy to evaluate the performances of the compared classifier for the imbalanced data set. Also, for evaluation purpose, we use cross-validation method provided by RapidMiner environment.

4.6 Summary

This chapter describes the methodology used in our research. It presents our preprocessing strategy which we followed to achieve our goals with more detail. Also, we explain the classification algorithms which are used during experiments results. The next chapter will be discussing the results of our experiments using our approach and the described methodology.

CHAPTER 5: Experimental Results and Analysis

In this chapter we present and analyze experimental results. We used different machine learning classifier for our experiments named, rule induction, naïve Bayes, decision tree and neural network on the selected datasets to classify the instances. All classifier were run on machine environment has 64-bit with 4GB RAM. For evaluation purpose, we use cross-validation method provided by RapidMiner environment. Also we assume that the ratio of the number of majority class samples to the number of minority class samples in the training data is set to be 1:1. In other word, there are the whole 100 majority class samples and there are must existing 100 minority class samples in this training data set.

We apply set of experiments, in the first section we classified instances without doing any preprocessing in the data sets. In the second section we classified instances after apply under sample approach on the data sets. In the third section we classified instances after apply over sample approach on the data sets. In the fourth section we classified instances after apply under sample approach based on clustering. In the fifth section we classified instances after apply over sample approach based on clustering. Section the six we shall discuss the importance of choosing the most appropriate number of clusters. Section the seven we applied SMOTE (Synthetic Minority Over-sampling TEchnique) approach. Section the eight we applied SMOTE based on clustering. Finally, we discussed and summarized the results of all our experiments.

5.1 Classification experiments without preprocessing

In this section we classify instances before done any change on data sets to compare the classification performance of classifier before using our proposed method and after using it. We used this experiment as a baseline for the whole experiments.

Table 5.1 illustrates the average accuracy for all the data. We note that the average accuracy range from 82 to 87 which is considered as a good result. However; we cannot depend on accuracy metric as a measure for classification for imbalanced data as mentioned in chapter 3. Therefore, we compute average F-measure of whole classes to evaluate classification performance. Table 5.2 shows the average F-measure for all the data. We note that in general the average of F-measure is less than the average of accuracy that is means the data sets have

imbalanced problem. For instance, the average accuracy of neural network classifier is 87.04 while the average F-measure is 65.74. So, the accuracy measure cannot detect the imbalanced problem and cannot give us the actual classification performance especially when the data has imbalanced class distribution problem.

Table 5.1: Average accuracy for whole data sets in classification experiments without preprocessing

Data set / Classifier	Rule Induction	Naïve Bayes	Decision Tree	Neural Network
Page-Blocks	95.86	94.64	93.12	95.49
CTG	94.67	86.68	98.12	97.49
Car evaluation	88.03	86.49	64.86	70.66
Auto-mpg	95.80	89.08	95.80	96.64
Glass	54.69	42.19	65.62	67.19
Breast - w	91.43	95.71	91.90	94.76
Average	86.75	82.46	84.9	87.04

Table 5.2: Average F-measure for whole data sets in classification experiments without preprocessing

Data set / Classifier	Rule Induction	Naïve Bayes	Decision Tree	Neural Network
Page-Blocks	75.16	75.6	57.28	59.5
CTG	90.95	79.57	97.25	94.88
Car evaluation	68.77	69.64	34.09	20.7
Auto-mpg	58.13	68.49	72.33	58.97
Glass	49.66	48.3	49.8	64.7
Breast - w	90.21	95.3	90.71	94.04
Average	72.15	72.82	66.91	65.47

5.2 Under sample experiments

In this experiment, we apply under sample approach which is supposed to reduce the number of samples with the majority class. Hence, an under sample approach is aim to decrease the skewed distribution of majority class and minority class by lowering the size of majority class. Finally, we obtain new data set with balance number of instances from each class. Table 5.3 present the average F-measure for all the data. Under sample approach made improvement on an average F-measure when we use decision tree and neural network classifiers. But there is no improvement with rule induction and naïve Bayes classifier. The reason is that under sampling may remove some instances which are important for the classification process.

Table 5.3: Average F-measure for whole data sets in under sample experiments

Data set / Classifier	Rule Induction	Naïve Bayes	Decision Tree	Neural Network
Page-Blocks	89.62	86.63	77.72	97.74
CTG	97.24	81.01	97.15	98.73
Car evaluation	76.7	85.67	60.74	39.01
Auto-mpg	36.69	13.3	34.28	40
Glass	19.83	72.89	83.3	75.9
Breast - w	93.13	95.86	92.39	97.92
Average	68.87	72.56	74.26	74.88

5.3 Over sample experiments

In this experiment, we apply over sample approach which is duplicate the sample of the minority class and adding them to data set. It is difference than under sample approach so there is no information is lost, all instances are employed. Finally, we obtain new data set with balance number of instances in each class. Table 5.4 present the average F-measure for all the data. We find over sample approach create significant improvement on an average F-measure comparing with the results obtain from normal classification and under sample experiments. For example, the average F-measure of decision tree is 66.91 in the normal classification experiments, the average F-measure of decision tree is 74.26 in the under sample experiments and the average F-measure of decision tree is 88.2 in the over sample experiments.

Table 5.4: Average F-measure for whole data sets in over sample experiments

Data set / Classifier	Rule Induction	Naïve Bayes	Decision Tree	Neural Network
Page-Blocks	90.61	74.5	98.8	96.03
CTG	95.55	82.1	90.84	99.5
Car evaluation	80.93	87.66	90.07	38.61
Auto-mpg	94.58	93.42	97.92	96.2
Glass	45.88	60.77	54.82	85.1
Breast - w	94.22	97.09	96.73	98.56
Average	83.63	82.59	88.2	85.67

5.4 Under sample based on clustering

In this experiment we apply cluster-based under-sampling approach on our data sets to produce new dataset has balance number of instances from each class. We evaluate the

performance for under-sampling method using different number of clusters two, three, four and five. Figure 5.1 shows the classifiers curve of the average F-measure for whole data sets when apply under sample approach with different number of clusters.

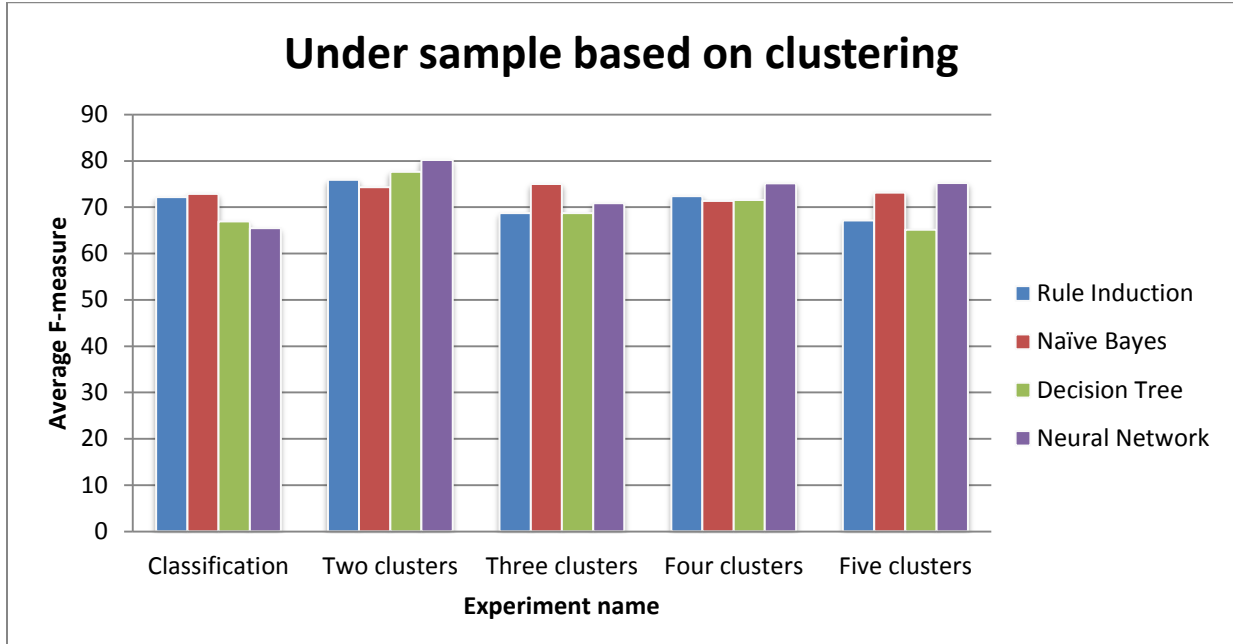


Figure 5.1: Average F-measures for each classifier on whole data sets.

In rule induction, the average F-measure improved from 67 to 75. The highest F-measure result (75.88) was in under sample approach with two clusters. We note the under sample based on clustering achieved small improvement only with two and four of clusters comparing with the F-measure result in normal classification experiments (72.15), and there is no any improvement with three and five clusters.

In naïve Bayes, the average F-measure improved from 71 to 74. The highest F-measure result (74.97) was in under sample approach with three clusters. We note the under sample based on clustering achieved small improvement only with two, three and five of clusters comparing with the F-measure result in normal classification experiments (72.82), and there is no any improvement with three and four clusters.

In decision tree, the average F-measure improved from 65 to 77. The highest F-measure result (77.62) was in under sample approach with two clusters. We note the under sample based

on clustering achieved improvement with two, three and four clusters comparing with the F-measure result in normal classification experiments (66.91), and there is no any improvement with five clusters.

In neural network, the average F-measure improved from 70 to 80. The highest F-measure result (80.18) was in under sample approach with two clusters. We note the under sample based on clustering achieved improvement in different number of clusters comparing with the F-measure result in normal classification experiments (65.47).

In general, from the cluster-based under-sampling experiments if there is no any improvement in some cases the reason return to discard some samples will be effect on classification performance. Also we can find the average F-measure for cluster-based under-sampling approach is better than under sample approach in most cases.

5.5 Over sample based on clustering

In this experiment we apply cluster-based over-sampling approach on our data sets to produce new dataset that has balance number of instances for each class. We evaluate the performance for over-sampling method using different number of clusters two, three, four and five. Figure 5.2 shows the classifiers curve of the average F-measure for whole data sets when apply over sample approach with different number of clusters.

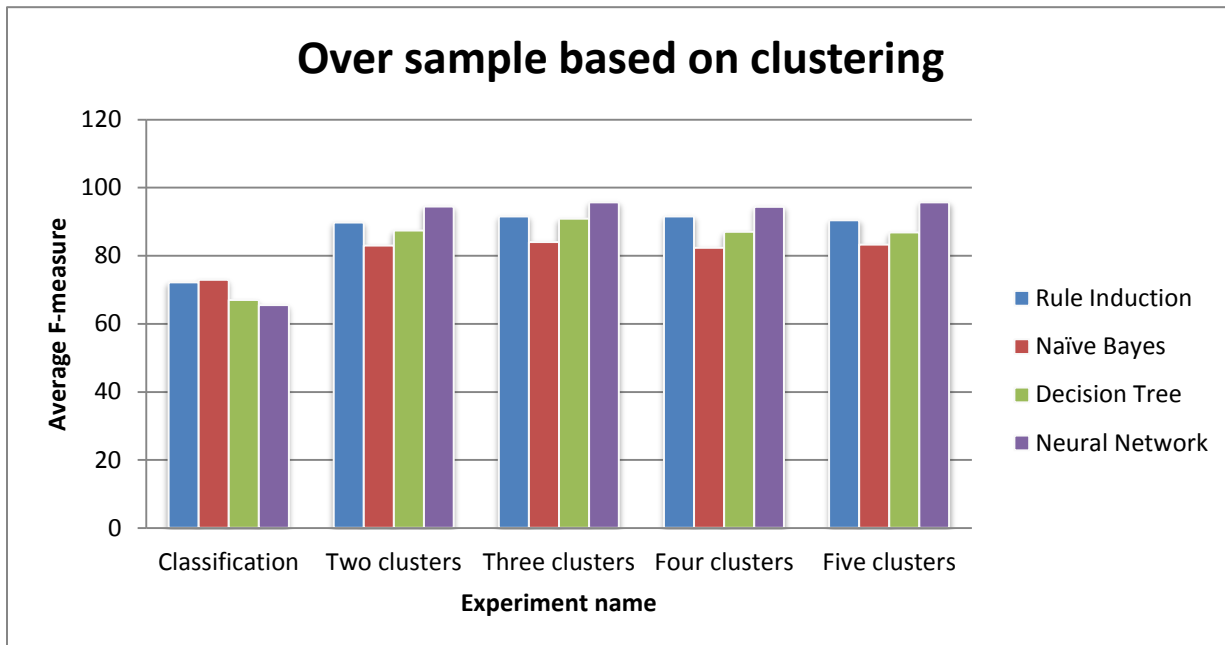


Figure 5.2: Average F-measures for each classifier on whole data sets.

In rule induction, the average F-measure improved from 89 to 91. The highest F-measure result (91.54) was in over sample approach with four clusters. We note the over sample based on clustering achieved significant improvement in different number of clusters comparing with the F-measure result in normal classification experiments (72.15).

In naïve Bayes, the average F-measure improved from 82 to 83. The highest F-measure result (83.92) was in over sample approach with three clusters. We note the over sample based on clustering achieved significant improvement in different number of clusters comparing with the F-measure result in normal classification experiments (72.82).

In decision tree, the average F-measure improved from 86 to 90. The highest F-measure result (90.78) was in over sample approach with three clusters. We note the over sample based on clustering achieved significant improvement in different number of clusters comparing with the F-measure result in normal classification experiments (66.91).

In neural network, the average F-measure improved from 94 to 95. The highest F-measure result (95.64) was in over sample approach with three clusters. We note the over sample based on clustering significant achieved improvement in different number of clusters comparing with the F-measure result in normal classification experiments (65.47).

In general, we can notice that the average F-measure for cluster-based over-sampling approach is better than over sample approach in most cases. From all the previous experiments, cluster-based over-sampling approach makes the best average F-measure.

5.6 Choosing optimal number of clusters

In the previous example, the number of cluster is chosen by trial and error. In this section we try to find optimal the number of clusters by using automatic method, which in our case x-means. Choose the most appropriate number of clusters play an important role on F-measure results. For this reason, we try cluster data by using x-mean algorithm which is determining the number of cluster automatically. From over sample with clustering by x-mean algorithm experiments, we find that in most cases these experiments achieved good F-measure results. Table 5.5 illustrate the average F-measure results when determine number of cluster manual and automatic.

Table 5.5: Average F-measure for whole data sets when determine number of cluster manual and automatic

	Experiment name	F-measure	X-mean	Number of clusters the x-means gave
Rule Induction	2cluster	89.69	91.63	3cluster
	3cluster	91.51		
	4cluster	91.5		
	5cluster	90.39		
Naïve Bayes	2cluster	82.92	81.23	3cluster
	3cluster	83.92		
	4cluster	82.23		
	5cluster	83.16		
Decision Tree	2cluster	87.37	93.19	4cluster
	3cluster	90.78		
	4cluster	86.94		
	5cluster	86.77		
Neural Network	2cluster	94.38	94.58	3cluster
	3cluster	95.64		
	4cluster	94.34		
	5cluster	95.63		

In rule induction, determining the number of cluster automatically made close result (91.63) to (91.51) when determining the number of cluster manually that is in over sample approach with three clusters experiment. So, we note the highest average F-measure (91.51) of rule induction in manual experiments also was in three clusters.

In naïve Bayes, determining the number of cluster automatically made close result (81.23) to (82.23) when determining the number of cluster manually that is in over sample approach with four clusters experiment. But we note the highest average F-measure (83.92) of naïve Bayes in manual experiments was in three clusters not in four clusters.

In decision tree, determining the number of cluster automatically made close result (93.19) to (90.78) when determining the number of cluster manually that is in over sample approach with three clusters experiment. So, we note the highest average F-measure (90.78) of rule induction in manual experiments also was in three clusters.

In neural network, determining the number of cluster automatically made close result (94.58) to (95.64) when determining the number of cluster manually that is in over sample

approach with three clusters experiment. So, we note the highest average F-measure (95.64) of rule induction in manual experiments also was in three clusters.

In most cases, we can say the best average F-measure from results which is coming from over sample approach with using the optimal number clusters.

5.7 SOMTE approach

In this experiment, the minority class is over-sampled by creating “synthetic” examples rather than by over-sampling with replicate the same existing examples. Then, we obtain new data set with balance number of instances in each class. Table 5.6 present the average F-measure for all the data. We find SMOTE approach create significant improvement on an average F-measure comparing with the results obtain from normal over sample experiments. For example, the average F-measure of neural network is 85.67 in the over sample experiments and the average F-measure of neural network is 93.97 in the SOMTE approach experiments

Table 5.6: Average F-measure for whole data sets in SMOTE experiments

Data set / Classifier	Rule Induction	Naïve Bayes	Decision Tree	Neural Network
Page-Blocks	94.51	78.37	96.51	96.79
CTG	90.01	76.44	92.21	93.17
Car evaluation	79.78	86.82	96.33	98.15
Auto-mpg	93.16	93.74	98.4	96.46
Glass	80.37	60.02	64.86	81.45
Breast - w	95.97	96.74	95.63	97.83
Average	88.96	82.02	90.66	93.97

5.8 SOMTE based on clustering

In this experiment, we applied SMOTE based on clustering (our approach) on the best experiment of each data set to know if there is any improvement happens. In general SMOTE from [5] based on clustering create significant improvement on the classification accuracy of minority class in most cases comparing with the results for all other experiments. We compute the average accuracy for all the data set of this experiment as illustrated in Table 5.7. Also, Table 5.8 presents the average F-measure for all the data.

Table 5.7: Average accuracy for whole data sets in SMOTE based on clustering experiments

Data set / Classifier	Rule Induction	Naïve Bayes	Decision Tree	Neural Network
Page-Blocks	96.54	81.27	97.20	96.13
CTG	96.96	82.64	98.24	98.24
Car evaluation	86.34	88.10	95.87	95.10
Auto-mpg	97.34	96.68	99.00	98.01
Glass	78.46	66.15	84.62	83.85
Breast - w	96.74	98.19	97.10	96.74
Average	92.06	85.51	95.34	94.68

Table 5.8: Average F-measure for whole data sets in SMOTE based on clustering experiments

Data set / Classifier	Rule Induction	Naïve Bayes	Decision Tree	Neural Network
Page-Blocks	96.55	82.69	97.23	96.18
CTG	96.9	83.73	98.25	98.25
Car evaluation	86.83	88.16	95.93	94.96
Auto-mpg	96.26	96.5	98.89	98.88
Glass	79.15	66.83	83.66	82.69
Breast - w	96.74	98.2	97.09	96.74
Average	92.07	86.02	95.18	94.62

We compare our approach results with the best previous experiment results of each data set to make sure. These experiments will be discussed separately for each data set in this section.

Table 5.9 summarized all F-measure results for each data set.

Table 5.9: F-measure results of the approaches: over sample based on clustering and SOMTE with clustering for all our data set

Data set name	Experiment name	Rule Induction	Naive Bayes	Decision Tree	Neural Network
<i>Page Blocks</i>	Over sample + 4clusters	95.78	75.71	71.71	96.53
	SOMTE + 4clusters	96.55	82.69	97.23	96.18
<i>Cardiotocography</i>	Over sample + 4clusters	97.47	81.49	97.79	99.2
	SOMTE + 4clusters	96.9	83.73	98.25	98.25
<i>Car Evaluation</i>	Over sample + 3clusters	79.14	87.7	92.46	97.9
	SOMTE + 3clusters	86.83	88.16	95.93	94.96
<i>Auto-MPG</i>	Over sample + 3clusters	97.4	95.01	99.68	98.35
	SOMTE + 3clusters	96.26	96.5	98.89	98.88
<i>Glass identification</i>	Over sample + 3clusters	89.2	67.24	85.72	83.35
	SOMTE + 3clusters	79.15	66.83	83.66	82.69

Breast Cancer - W	Over sample + 4clusters	94.51	97.08	97.46	97.47
	SOMTE + 4clusters	96.74	98.2	97.09	96.74

In **page blocks data set**, the best F-measure results were in experiment of over sample approach with four clusters. So, we apply the SMOTE approach with four clusters. We find this approach performs significant improvement on F-measure with rule induction, naïve Bayes, and decision tree and there is no improvement with neural network as presented in Table 5.9.

In **cardiotocography data set**, the best F-measure results were in experiment of over sample approach with four clusters. So, we apply the SMOTE approach with four clusters. We find this approach performs significant improvement on F-measure with naïve Bayes and decision tree and there is no improvement with rule induction and neural network as presented in Table 5.9.

In **car evaluation data set**, the best F-measure results were in experiment of over sample approach with three clusters. So, we apply the SMOTE approach with three clusters. We find this approach performs significant improvement on F-measure with rule induction, naïve Bayes, and decision tree and there is no improvement with neural network as presented in Table 5.9.

In **auto-MPG data set**, the best F-measure results were in experiment of over sample approach with three clusters. So, we apply the SMOTE approach with three clusters. We find this approach performs significant improvement on F-measure with naïve Bayes and neural network and there is no improvement with rule induction and decision tree as presented in Table 5.9.

In **breast cancer-w data set**, the best F-measure results were in experiment of over sample approach with four clusters. So, we apply the SMOTE approach with four clusters. We find this approach performs significant improvement on F-measure with rule induction, naïve Bayes, and decision tree and there is no improvement with and neural network as presented in Table 5.9.

Finally, in **glass identification data set** the best F-measure results were in experiment of over sample approach with three clusters. But there is no any improvement with all learning algorithm so the reason is may due to the nature of these data set where the smaller two classes have 3 and 4 instances which makes a problem in the clustering distribution. That means we can

find some clusters do not have the instances from this smaller classes. Table 5.9 illustrates the F-measure results for this experiment.

5.8 Discussion and summary

The following Figure 5.3 shows an overview of the all experiment results.

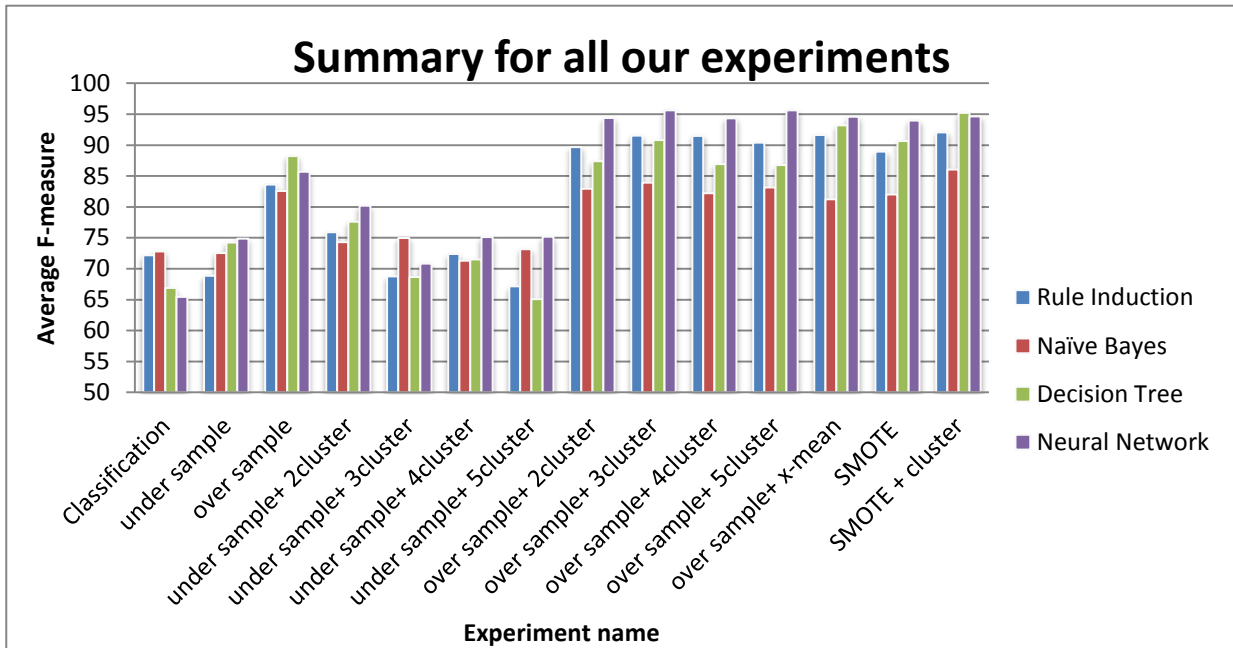


Figure 5.3: Summary for all our experiments.

We can summarize our experiments results as is in rule induction, the highest F-measure result (92.07) was in our approach (SOMTE based on clustering). In naïve Bayes, the highest F-measure result (86.02) was in our approach. In decision tree, the highest F-measure result (95.18) was in our approach. In neural network, the highest F-measure result (95.64) was in over sample approach with three clusters.

We find under sample approach is good solution for imbalanced data distribution but the over sample approach is better than under sample approach because it is difference than under sample approach so there is no information is lost, all instances are employed. Also we preferred use clustering with both two samples approach: under and over sample because we find some of kind of distribution between the data inside the cluster that is helping us in covers all the characteristics of all the existing data. And that when select the majority class samples from each cluster in under sample approach and adding them to new data set, or when select minority class

examples from each cluster in over sample approach and then replicating the selected examples and adding them to new data set.

From all our experiments we can say the over sample with optimal number of clusters achieved the good classification accuracy of minority class. Also, we note using SMOTE based on clustering perform significant improvement on F-measure results and better than over sample based on clustering approach in most cases. Table 5.10 illustrates accuracy and F-measure results for the baseline and our approach experiments with all classifier.

Table 5.10: Average accuracy and F-measure comparison of the approaches: baseline and SOMTE with clustering experiments for all our data set

Classifier	Accuracy of baseline	Accuracy of our approach	F-measure of baseline	F-measure of our approach
Rule Induction	86.75	92.06	72.15	92.07
Naïve Bayes	82.55	85.51	72.82	86.02
Decision Tree	84.46	95.34	66.91	95.18
Neural Network	87.04	94.68	65.47	94.62

We can note the great difference in improvement before preprocessing and after apply our approach in accuracy and F-measure. For example, in the decision tree the accuracy is 84.46 and the F-measure is 66.91 in the baseline experiment, and after we apply our approach we obtain 95.34 for accuracy and 95.18 for F-measure.

Also we can show the great difference in improvement with different classes in page block data set as in Table 5.11 and Auto-MPG data set as in Table 5.12. In page blocks data set all classes except class 1 are consider as a minority class and note that this data set has high imbalances, because class 1 represents 89.8% from all data and other classes represent the remained. In Auto-MPG data set all classes except class 4 are consider as a minority class especially class 3 and 5. Class 3 and 5 has only 4 and 3 instances respectively.

Table 5.11: F-measure results – page blocks

Classifier	Experiment name	Class 1	Class 2	Class 4	Class 5	Class 3	F-measure
Rule Induction	Baseline	99.39	73.79	71.43	30	50	75.16
	Our approach	94.28	97.82	97.77	93.13	99.86	96.55
Naive Bayes	Baseline	97.50	69.90	90.48	50	70	75.6
	Our approach	83.51	86.58	94.85	50.77	91.47	82.69
Decision Tree	Baseline	99.93	33.01	76.19	0	20	57.28
	Our approach	94.41	97.68	97.64	96.36	100	97.23
Neural Network	Baseline	98.99	73.79	66.67	50	0	59.5
	Our approach	92.59	96.90	97.04	94.77	99.65	96.18

Table 5.12: F-measure results – auto-mpg

Classifier	Experiment name	Class 8	Class 4	Class 6	Class 3	Class 5	F-measure
Rule Induction	Baseline	100	100	91.67	0	0	58.13
	Our approach	100	89.29	96.61	100	100	96.26
Naive Bayes	Baseline	100	93.65	70.83	50	0	68.49
	Our approach	100	93.55	88	100	100	96.5
Decision Tree	Baseline	100	100	87.50	50	0	72.33
	Our approach	100	96.77	98.00	100	100	98.89
Neural Network	Baseline	100	98.41	100	0	0	58.97
	Our approach	98.46	95.16	96.00	100	100	98.88

From all the above, experimental results confirm our findings which are saying the SMOTE based on clustering achieved best classification accuracy of minority class in imbalanced class distribution problem with both two and multi classes cases. Because the two class problem is a special case from multi class problem. The disadvantage for our approach is the size of data set will be increasing when adding new instances with amount close to majority size to create balance data set.

CHAPTER 6: Conclusion and Future work

6.1 Conclusion

Many of real-world applications are encountered the class imbalanced problem. It is occur when there are many more instances of some classes than others. In such cases, standard classifiers tend to be overwhelmed by the large classes and ignore the small ones. Our research proposes a new approach combine between both Synthetic Minority Over-sampling TEchnique (SOMTE) approach and clustering approach which is able to deal with multi class imbalanced data problem. First, we cluster all the training samples in to some clusters. Then we compute the number instances of each class in all clusters. If a cluster has more majority class samples and less minority class samples, it will behave like the majority class samples. After that in each cluster we apply the SMOTE approach which is generate new synthetic minority instances by interpolating between several minority examples that lie close together. Finally, combine between whole classes to produce new balance training data set.

For our experiments, six data sets are chosen from different real domain, characteristics and sizes. Five from data sets (page blocks, cardiotocography, car evaluation, auto MPG and glass identification) represent multi class problem case and the other one data set (breast cancer-w) represent two class problem case. For evaluation purpose, we use cross-validation method provided by RapidMiner environment. Also we assume that the ratio of the number of majority class samples to the number of minority class samples in the training data is set to be 1:1. Experimental results show the SMOTE based on clustering approach perform significant improvement on F-measure results and better than normal over sample based on clustering approach in most cases. In some case F-measure improved from 66.91 to 95.18.

To confirm our conclusion, Table 6.1, compares our work with some other published work in the field of imbalanced class distribution problem domain.

Table 6.1: Summary table for compare between some other works

Paper	Approaches	Average F-measure	with automatic selection of number of cluster	Handle multi class	Level
Our research	SMOTE based on clustering	95.18%	√	√	Data
Yen and Lee in [32]	Cluster based under sampling	79%	×	×	Data
Chen et al. in [7]	Information Granulation Based Data Mining	72.68%	-	√	Algorithm
Chen et al. in [8]	Cluster Ensembles Based SMOTE	73.44%	×	×	Data
Han and Mao in [15]	Fuzzy –rough k-nearest neighbor algorithm	61.84%	-	×	Algorithm
Nguyen et al. in [23]	Under sampling method based on clustering	79%	×	×	Data

6.2 Future Work

In future work, we will need to find solution of the size of data set that will be increasing when adding new instances with amount close to majority size to create balance data set. This is considering problem especially when dealing with very large data sets. Also we can extend our method to deal with within class imbalance problem. Also, we need to consider the problem of imbalance data with noisy dataset especially if the noise in class attribute. Another direction could be working with data types other than numbers and categories such as multimedia data.

Reference

- [1] Adam, A., Shapiai, I., Ibrahim, Z., Khalid, M., Chun Chew, L. , Wen Jau, L. and Watada, J.: *A Modified Artificial Neural Network Learning Algorithm for Imbalanced Data Set Problem*, cicsyn, pp.44-48, 2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks, 2010.

- [2] Barandela, R., Sánchez, J., García, V., and Rangel, E.: *Strategies for learning in class imbalance problems*. Pattern Recognition 36(3), 849-851, 2003.
- [3] Berkhin, P.: *Survey of clustering data mining techniques*, Accrue Software, San Jose, CA, Tech. Rep., 2002.
- [4] Chawla, N.: *Data Mining for Imbalanced Datasets: An Overview*. In: Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, pp. 853--867. Springer, 2005.
- [5] Chawla, N., Bowyer, K., Hall, L., and Kegelmeyer, W. P.: *SMOTE: synthetic minority over-sampling technique*. In International Conference on Knowledge Based Computer Systems, 2002.
- [6] Chen, C., Liaw, A., and Breiman, L.: *Using random forest to learn imbalanced data*. Technical Report 666, University of California, Berkeley, <http://www.stat.berkeley.edu/tech-reports/666.pdf>, 2004.
- [7] Chen, M., Chen, L., Hsu, C., and Zeng, W.: *An information granulation based data mining approach for classifying imbalanced data*, Information Sciences, vol. 78, no. 16, pp. 3214-3227, 2008.
- [8] Chen, S., Guo, G. and Chen, L.: *A New Over-Sampling Method Based on Cluster Ensembles*, AINA Workshops 2010: 599-604
- [9] Debray, T.: *Classification of Imbalanced Data Sets*, Master's Thesis in Artificial Intelligence Faculty of Humanities and Sciences, Maastricht University, 2009
- [10] Estabrooks, T. Japkowicz, Jo and N.: *A multiple resampling method for learning from imbalanced data set*. Computational Intelligence, 20(1):18-36, 2004.
- [11] Frank, A. And Asuncion, A.: UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science, 2010.
- [12] Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera. F.: *A Survey on Ensembles for Class Imbalance Problem: Bagging, Boosting and Hybrid Based Approaches*, IEEE Transactions on System, Man and Cybernetics - Part C: Applications and Reviews, doi: 10.1109/TSMCC.2011.2161285, 2012
- [13] García, V., Sánchez, J.S., Mollineda, Alejo, R., R., and Sotoca, M.: *The class imbalance problem in pattern classification and learning*, Tamida, Saragossa, Spain, pp. 283-291, 2007.

- [14] Ghanem, A., Venkatesh, S., and West, G.: *Multi-Class Pattern Classification in Imbalanced Data*, In International Conference on Pattern Recognition, pp. 2881-2884. Istanbul, Turkey: IEEE, 2010
- [15] Han, H. and Mao, B.: *Fuzzy-rough k-nearest neighbor algorithm for imbalanced data sets learning*, FSKD 2010: 1286-1290
- [16] Han J., and Kamber M.: *Data Mining: Concepts and Techniques*, (2nd Ed), the Morgan Kaufmann Series in Data Management Systems, 2006.
- [17] Hand, D. and Tell, R.: *A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems*, Machine Learning, 45, 171-186, 2001
- [18] He, H., and Garcia, E.: *Learning from imbalanced data*. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263 –1284, 2009.
- [19] Hoens, T., Qian, Q., Chawla, N., and Zhou, Z.: *Building decision trees for the multiclass imbalance problem*, In: Proceedings of the 16th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'12), LNAI xxxx, Kuala Lumpur, Malaysia, 2012.
- [20] Hu, S., Liang, Y., Ma, L., and He, Y.: *MSMOTE: Improving Classification Performance When Training Data is Imbalanced*, iwccse, vol. 2, pp.13-17, Second International Workshop on Computer Science and Engineering, 2009.
- [21] Kaur, G. and Singh, L.: *Data Mining: An overview*, IJCST, vol. 2, Issue 2, pp. 336-339, June 2011.
- [22] Murphey, Y., Wang, H., Ou, G., and Feldkamp, L.: *OAHO: an effective algorithm for multi-class learning from imbalanced data*, in International Joint Conference on Neural Networks (IJCNN), pp. 406–411, 2007
- [23] Nguyen, G. Hoang., Bouzerdoum, A. and Phung, S.: *Learning pattern classification tasks with imbalanced data sets*. In P. Yin (Eds.), Pattern recognition (pp. 193-208). Vukovar, Croatia: In-The, 2009.
- [24] Pelleg, D. and Moore, A.: *X-means: Extending K-means with Efficient Estimation of the Number of Clusters*, ICML 2000: 727-734
- [25] Sun, Y., Kamel, M.S., Wong, A.K.C., and Wang, Y.: *Cost-Sensitive Boosting for Classification of Imbalanced Data*, Pattern Recognition, vol. 40, no. 12, pp. 3358-3378, 2007.

- [26] Sun, Y.: *Cost-Sensitive Boosting for Classification of Imbalanced Data*, Thesis requirement for the degree of Doctor of Philosophy In Electrical and Computer Engineering, Waterloo University, 2007
- [27] Sun, Y., Kamel, M., Wang, Y.: *Boosting for Learning Multiple Classes with Imbalanced Class Distribution*, IEEE Conference on Data Mining (ICDM), Hong Kong, China, December 18-22, 2006
- [28] Tan, P., Steinbach, M., Kumar, V.: *Introduction to Data Mining*, Addison-Wesley, Reading, MA, 2006.
- [29] Thai-Nghe, N., Gantner, Z., and Schmidt-Thieme, L.: *Cost-sensitive learning methods for imbalanced data*. In Proceeding of IEEE International Joint Conference on Neural Networks (IJCNN'10), 2010.
- [30] Wang, S.: *Class Imbalance Learning*, Thesis Proposal, 2008
- [31] Wasikowski, M., and Chen, X.: *Combating the small sample class imbalance problem using feature selection*, IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 13881400, 2010.
- [32] Yen, S.-J., and Lee, Y.-S.: *Cluster-based Under-sampling Approaches for Imbalanced Data Distributions*. Expert Systems with Applications, 36, 5718-5727, 2009.
- [33] Zaiane, O.: *Introduction Survey to Data Mining*, CMPUT690 Principles of Knowledge Discovery in Databases, University of Alberta, 1999
- [34] Zhang, J., and Mani, I.: *kNN approach to unbalanced data distributions: A case study involving information extraction*. In Proceedings of the ICML'2003 workshop on learning from imbalanced datasets
- [35] http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm, (2012, March), [Online].
- [36] UCI Machine Learning Repository: Page Blocks Data set, Available: <http://archive.ics.uci.edu/ml/datasets/Page+Blocks+Classification>, (2012, March), [Online].
- [37] UCI Machine Learning Repository: Cardiotocography Data set, Available: <http://archive.ics.uci.edu/ml/datasets/Cardiotocography>, (2012, March), [Online].
- [38] UCI Machine Learning Repository: Car Evaluation Data set, Available: <http://archive.ics.uci.edu/ml/datasets/Car+Evaluation>, (2012, March), [Online].

- [39] UCI Machine Learning Repository: Auto MPG Data set, Available: <http://archive.ics.uci.edu/ml/datasets/Auto+MPG>, (2012, March), [Online].
- [40] UCI Machine Learning Repository: Glass Identification Data set, Available: <http://archive.ics.uci.edu/ml/datasets/Glass+Identification>, (2012, March), [Online].
- [41] UCI Machine Learning Repository: Breast Cancer Wisconsin Data set, Available: <http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29>, (2012, March), [Online].
- [42] <http://rapid-i.com/content/view/181/190/>, (2012, March), [Online].
- [43] <http://www.cs.waikato.ac.nz/ml/weka/>, (2012, March), [Online].