# DATA MINING TECHNIQUES TO IDENTIFY FINANCIAL RESTATEMENTS

By

**Ila Dutta**

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of
**Master of Computer Science**

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

uOttawa

University of Ottawa

# Abstract

Data mining is a multi-disciplinary field of science and technology widely used in developing predictive models and data visualization in various domains. Although there are numerous data mining algorithms and techniques across multiple fields, it appears that there is no consensus on the suitability of a particular model, or the ways to address data preprocessing issues. Moreover, the effectiveness of data mining techniques depends on the evolving nature of data. In this study, we focus on the suitability and robustness of various data mining models for analyzing real financial data to identify financial restatements. From data mining perspective, it is quite interesting to study financial restatements for the following reasons: (i) the restatement data is highly imbalanced that requires adequate attention in model building, (ii) there are many financial and non-financial attributes that may affect financial restatement predictive models. This requires careful implementation of data mining techniques to develop parsimonious models, and (iii) the class imbalance issue becomes more complex in a dataset that includes both intentional and unintentional restatement instances. Most of the previous studies focus on fraudulent (or intentional) restatements and the literature has largely ignored unintentional restatements. Intentional (i.e. fraudulent) restatements instances are rare and likely to have more distinct features compared to non-restatement cases. However, unintentional cases are comparatively more prevalent and likely to have fewer distinct features that separate them from non-restatement cases. A dataset containing unintentional restatement cases is likely to have more class overlapping issues that may impact the effectiveness of predictive models.

In this study, we developed predictive models based on all restatement cases (both intentional and unintentional restatements) using a real, comprehensive and novel dataset which includes 116 attributes and approximately 1,000 restatement and 19,517 non-restatement instances over a period of 2009 to 2014. To the best of our knowledge, no other study has developed predictive models for financial restatements using post-financial crisis events. In order to avoid redundant attributes, we use three feature selection techniques: Correlation based feature subset selection (CfsSubsetEval), Information gain attribute evaluation (InfoGainEval), Stepwise forward selection (FwSelect) and generate three datasets with reduced attributes.

Our restatement dataset is highly skewed and highly biased towards non-restatement (majority) class. We applied various algorithms (e.g. random undersampling (RUS), Cluster based undersampling (CUS) (Sobhani et al., 2014), random oversampling (ROS), Synthetic minority oversampling technique (SMOTE) (Chawla et al., 2002), Adaptive synthetic sampling (ADASYN) (He et al., 2008), and Tomek links with SMOTE) to address class imbalance in the financial restatement dataset.

We perform classification employing six different choices of classifiers, Decision three (DT), Artificial neural network (ANN), Naïve Bayes (NB), Random forest (RF), Bayesian belief network (BBN) and Support vector machine (SVM) using 10-fold cross validation and test the efficiency of various predictive models using minority class recall value, minority class F-measure and G-mean. We also experiment different ensemble methods (bagging and boosting) with the base classifiers and employ other meta-learning algorithms (stacking and cost-sensitive learning) to improve model performance. While applying cluster-

based undersampling technique, we find that various classifiers (e.g. SVM, BBN) show a high success rate in terms of minority class recall value. For example, SVM classifier shows a minority recall value of 96% which is quite encouraging. However, the ability of these classifiers to detect majority class instances is dismal.

We find that some variations of synthetic oversampling such as 'Tomek Link + SMOTE' and 'ADASYN' show promising results in terms of both minority recall value and G-mean. Using InfoGainEval feature selection method, RF classifier shows minority recall values of 92.6% for 'Tomek Link + SMOTE' and 88.9% for 'ADASYN' techniques, respectively. The corresponding G-mean values are 95.2% and 94.2% for these two oversampling techniques, which show that RF classifier is quite effective in predicting both minority and majority classes. We find further improvement in results for RF classifier with cost-sensitive learning algorithm using 'Tomek Link + SMOTE' oversampling technique. Subsequently, we develop some decision rules to detect restatement firms based on a subset of important attributes.

To the best of our knowledge, only Kim et al. (2016) perform a data mining study using only pre-financial crisis restatement data. Kim et al. (2016) employed a matching sample based undersampling technique and used logistic regression, SVM and BBN classifiers to develop financial restatement predictive models. The study's highest reported G-mean is 70%. Our results with clustering based undersampling are similar to the performance measures reported by Kim et al. (2016). However, our synthetic oversampling based results show a better predictive ability. The RF classifier shows a very high degree of predictive capability for minority class instances (97.4%) and a very high G-mean value (95.3%) with cost-sensitive learning. Yet, we recognize that Kim et al. (2016) use a different restatement dataset (with pre-crisis restatement cases) and hence a direct comparison of results may not be fully justified.

Our study makes contributions to the data mining literature by (i) presenting predictive models for financial restatements with a comprehensive dataset, (ii) focussing on various datamining techniques and presenting a comparative analysis, and (iii) addressing class imbalance issue by identifying most effective technique. To the best of our knowledge, we used the most comprehensive dataset to develop our predictive models for identifying financial restatement.

To my parents, for their endless love and support

# Acknowledgement

First and foremost I sincerely thank my supervisor, Professor Bijan Raahemi of the University of Ottawa, for his invaluable guidance and supports. This dissertation wouldn't have been completed successfully without his continuous support and guidance throughout my thesis work. I would like to express my deepest gratitude to the committee members, Professor Herna Viktor and Professor Wei Shi for their invaluable comments and suggestions which have improved my thesis work substantially. I thank Professor Shantanu Dutta for his valuable comments and sharing his insights throughout the data collection and data processing stages. I am thankful to Professor Natalie Japkowicz for teaching the 'Concept learning systems' course at University of Ottawa, where I have learned important concepts on data mining techniques.

I thank all my colleagues at Knowledge Discovery and Data mining Laboratory of University of Ottawa for their continual support and encouragement.

I am also truly indebted to the University of Ottawa for the generous financial support in the form of Admission scholarships for my master's studies. Thank you to the staffs at graduate office (School of Engineering and Computer Science) for their excellent administrative support.

Finally, I thank my daughters and my husband for their support, patience and encouragement throughout my research years.

# Table of Contents

# List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| ADASYN | Adaptive Synthetic Sampling |
| ANN | Artificial Neural Network |
| AUC | Area Under the Curve |
| Bagging | Bootstrap Aggregating |
| BBN | Bayesian Belief Network |
| CART | Classification and Regression Trees |
| CfsSubset | Correlation Based Feature Subset Selection |
| CfsSubsetEval | Correlation Based Feature Subset Selection |
| CUS | Cluster Based Undersampling |
| DR | Decision Rule |
| DT | Decision Tree |
| FN | False Negative |
| FP | False Positive |
| ID3 | Iterative Dichotomiser 3 |
| InfoGain | Information Gain Attribute Feature Selection |
| InfoGainEval | Information Gain Attribute Feature Selection |
| KDD | Knowledge Discovery in Databases |
| LOGIT | Logistic Regression |
| ML | Machine Learning |
| MLogit | Multinomial Logistic Regression |

| | |
|---|---|
| NRS | Non-restatement |
| PCA | Principle Component Analysis |
| RF | Random Forest |
| ROC | Receiver Operating Curve |
| RS | Restatement |
| SVM | Support Vector Machine |
| TN | True Negative |
| TP | True Positive |
| VI | Variable Importance |
| WEKA | Waikato Environment for Knowledge Analysis |

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivations

Data mining is a multi-disciplinary field of science and technology. Relevant data mining techniques and algorithms are widely used in developing predictive models and data visualization in various domains. Although there are numerous data mining studies across multiple fields, it appears that there is no consensus on the suitability of a particular model or the ways to address data preprocessing issues (e.g. class imbalance). Moreover, the effectiveness of data mining techniques depends on the evolving nature of data. Most of the studies address the aforementioned issues in part, which may not provide us a holistic perspective on the challenges associated with data mining approaches and the effectiveness of various techniques. This has motivated us to undertake a comprehensive empirical study that addresses a wide spectrum of different data mining issues including the class imbalance problem, suitability of different classification techniques, and model validity in the context of evolving nature of data. Since the practical significance of the results is as important as the technical validity of the models, we are also motivated to use a real and comprehensive dataset that captures the interests of academics and practitioners alike.

In particular, we focus on financial restatement data to address these issues pertaining to data mining techniques. Financial restatements occur when management makes unintentional errors while compiling financial statements (termed as 'unintentional restatement') or intentionally alter financial data to mislead inventors and regulators (termed

as 'intentional restatement'). In the context of data mining literature, it is quite motivating to study financial restatements for the following reasons:

***Lack of financial restatement predictive models:*** most of the data mining predictive models concentrate on fraudulent or intentional restatements for model building (e.g. Cecchini et al., 2010; Kim et al. 2016). Although Bowen et al. (2017) show that both types of restatements are of concern to the regulators, investors and market participants because of their potential impact on investors' wealth, the unintentional restatements have received very little attention in the literature. Therefore, it will be quite useful to develop predictive models for all financial restatement instances.

***Inadequate attention to class imbalance issue:*** Typically, the ratio of restatement to non-restatement instances is quite low. If a dataset is highly imbalanced, it requires adequate attention in model building. Existing literature suggests that class imbalance generally causes a classifier to be biased towards the majority class (Longadge, Dongre and Malik, 2013). Further, the class imbalance issue becomes more complex in a dataset that includes both intentional and unintentional restatement instances. Most of the previous studies focus on fraudulent (or intentional) restatements and the literature has largely ignored unintentional restatements. Intentional (i.e. fraudulent) restatements instances are rare and likely to have more distinct features compared to non-restatement cases. However, unintentional cases are comparatively more prevalent and likely to have fewer distinct features that separate them from non-restatement cases. Therefore, a dataset containing unintentional restatement cases is likely to have more class overlapping issues that may impact the effectiveness of predictive models.

***Limited exploration of classifiers:*** there is no consensus in the data mining literature on the most suitable technique to detect financial restatements. Therefore, it is important to explore various classifiers while developing predictive models. To the best of our knowledge there is only one large scale study by Kim et al. (2016) that presents predictive models for both types (i.e. unintentional and intentional) of financial restatements. This study has used three classifiers, namely Logit, SVM and Bayesian Network to detect financial restatements and performance of these models are not encouraging (G-means are less than 0.70). It reinforces the view that we need to explore other classifiers and hybrid models to develop more efficient predictive models. Besides, various undersampling or oversampling techniques are not explored in the context of financial restatement predictive models.

The goal of this study is to address the above mentioned issues and to develop effective predictive models for financial restatements.

## 1.2 Objectives

The primary objective of this study is to address the class imbalance issue in the financial restatement dataset and develop an effective predictive model by using data mining techniques. In order to achieve the goal of this study, we:

- construct a comprehensive real dataset that includes 1,000 restatement and 19,517 non-restatement cases over a period of 2009 to 2014 and 116 attributes.
- implement both undesampling and oversampling techniques to address class imbalance issue at the data level.
- employ various classification techniques to identify financial restatements and use a number of hybrid models to further improve the model performance.

## 1.3 Contributions

The intended contributions of this study are as follows:

### 1.3.1 Implementation of a detail data mining framework

One of the main contributions of this study is the implementation of a detail data mining framework and the development of a highly effective financial restatement predictive model with high minority (restatement) recall and G-mean values. The salient features of the implemented framework are presented below (chapter 5 discusses the framework in detail):

- Use of different feature selection techniques (namely, CfsSubset, InfoGain, and FwSelect) for attribute reduction, which will lead to more parsimonious predictive models.
- Use of various undersampling techniques (Random undersampling (RUS), Cluster-based undersampling (CUS)) and oversampling techniques (Random oversampling (ROS), SMOTE, ADASYN) to address the class imbalance issue. Earlier data mining studies on financial restatement detection primarily relies on matching sample technique to obtain a balanced sample and generally do not employ other cluster based undersampling or synthetic oversampling techniques.
- Use of various classifiers (including Decision Tree (DT), Artificial Neural Networks (ANN), Naïve Bayes (NB), Support Vector Machine (SVM), and Bayesian Belief Network (BBN) classifiers) to predict financial restatements
- Use of ensemble models (employing Random Forest (RF), and other base classifiers with bagging, boosting, and stacking learning algorithms) and hybrid models

(combining various sampling methods and classification algorithm with cost-sensitive learning) to further improve model effectiveness.

### 1.3.2 Creating a new financial restatement dataset

We assemble a new financial restatement dataset in this study. Quite importantly, this dataset includes all financial restatements in the post financial crisis period (2009-2014) which have not been examined in the data mining literature previously. The dataset also includes a comprehensive list of firm specific attributes (116) which increases the likelihood of developing a more effective model. We further contribute to the data mining literature by showing that it is important to put emphasis on data quality to have better and dependable predictive models (Banarescu, 2015). We have taken a number of steps to ensure the integrity of data quality.

- First, we have considered a comprehensive list of restatement cases in developing our predictive models. Only a few studies have focussed on unintentional restatement cases while developing predictive models. However, it appears that number of restatement cases included in those studies are quite limited.[1] Therefore, by employing a more comprehensive dataset on financial restatement in this study, we are able to present more reliable predictive models.

[1] For example, Kim et al. (2016) include 355 unintentional cases in their study that focusses on predictive model developments; other studies that not necessarily focus on predictive model developments, also use fewer unintentional restatement cases. For example, Hennes et al. (2008) include only 83 unintentional cases, and Hennes et al. (2014) uses approximately 1,500 restatement cases. Both studies focusses on the consequences of restatements and do not employ any data mining technique to develop a predictive model.

- Second, it is also quite important to recognize that the choice of restatement data sources may significantly affect the validity of predictive models. Majority of the studies use U.S. GAO database. However, since this database mainly focusses on restatement announcement year and does not include the information on restatement periods, it is quite challenging to use this dataset to develop predictive models (Dechow et al. 2011). The other commonly used database, Accounting and Auditing Enforcement Releases (AAER) database, mainly includes intentional restatement cases. While this database is quite useful for developing predictive models for fraudulent cases, it is less effective for developing a more holistic model that focusses a broader range of restatement cases. In order to overcome these challenges, we use Audit Analytics database that includes a comprehensive set of both intentional and unintentional restatement cases.

Overall, we believe the results of this study can help improving expert and intelligent systems by providing more insights on both intentional and unintentional financial restatements. This study will also benefit academics, regulators, policymakers and investors. In particular, regulators and policymakers can pay a close attention to the suspected firms and investors can take actions in advance to reduce their investment risks.

## 1.4. Publication Resulted from This Research

Dutta, I., Dutta, S.; and Raahemi, B. (2017). Detecting Financial Restatements Using Data Mining Techniques, *Expert Systems with Applications*, 90, 374-393.

## 1.5 Thesis Organization

The remainder of the thesis is organized into seven chapters. Chapter 2 and 3 consists of the background study. In Chapter 2, we review the fundamentals of the classification and ensemble techniques. In Chapter 3, we discuss the class imbalance problem focusing on the classifiers used in this thesis. In Chapter 4, we review the existing literature on class imbalance and fraud and restatement detection. Chapter 5 presents research design and methodology, where we discuss CRISP-DM methodology and a detail experimental framework in the context of this study. In Chapter 6, we present a detail description of data collection and data preprocessing steps and discuss related attributes. In this chapter, we also discuss different attribute selection procedures and corresponding selected attributes. Chapter 7, we present experimental results and analysis of the results and discuss in depth research findings. Finally, in Chapter 8, we conclude and summarize the research findings and answer all the research questions and present other ideas for future work.

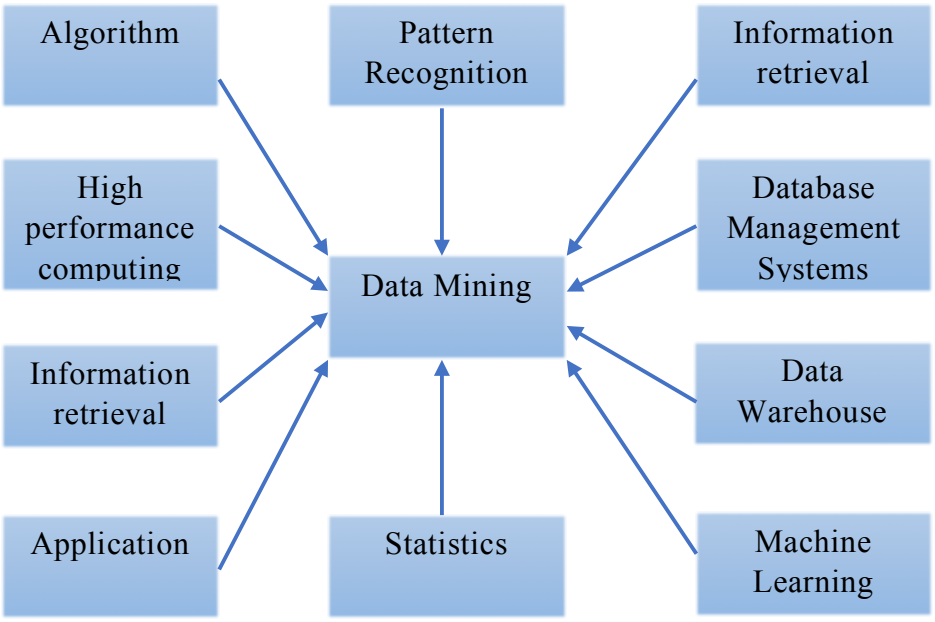
# CHAPTER 2

# BACKGROUND STUDY

Data mining focuses on extraction of knowledge or valuable patterns from large amounts of data from various data repositories, such as, the databases, data warehouses, and web or data streams. Data mining or Knowledge Discovery in Databases (KDD) identifies interesting patterns through several steps of data cleaning, data integration, data selection, data transformation and finally evaluates these discovered patterns to knowledge representations (Han, Kamber and Pei, 2012). Many researchers have recognized data mining as a key research topic in database systems, machine learning, artificial intelligence, statistics and data visualization (Chen, Han and Yu, 1996). Since the databases in various fields and organizations are constantly increasing in size, it is becoming an imperative strategy to use new tools and techniques for data mining and to automate data processing (Chen, Han and Yu, 1996).

Data mining is a multidisciplinary field. Data mining research and application development have incorporated many techniques from various disciplines, such as, machine learning, artificial intelligence, statistics, data visualization, pattern recognition and database systems. Figure 1 represents the multidisciplinary nature of data mining (Han et al., 2012). Among these disciplines, statistics and machine learning are widely used by many researchers in data mining studies.

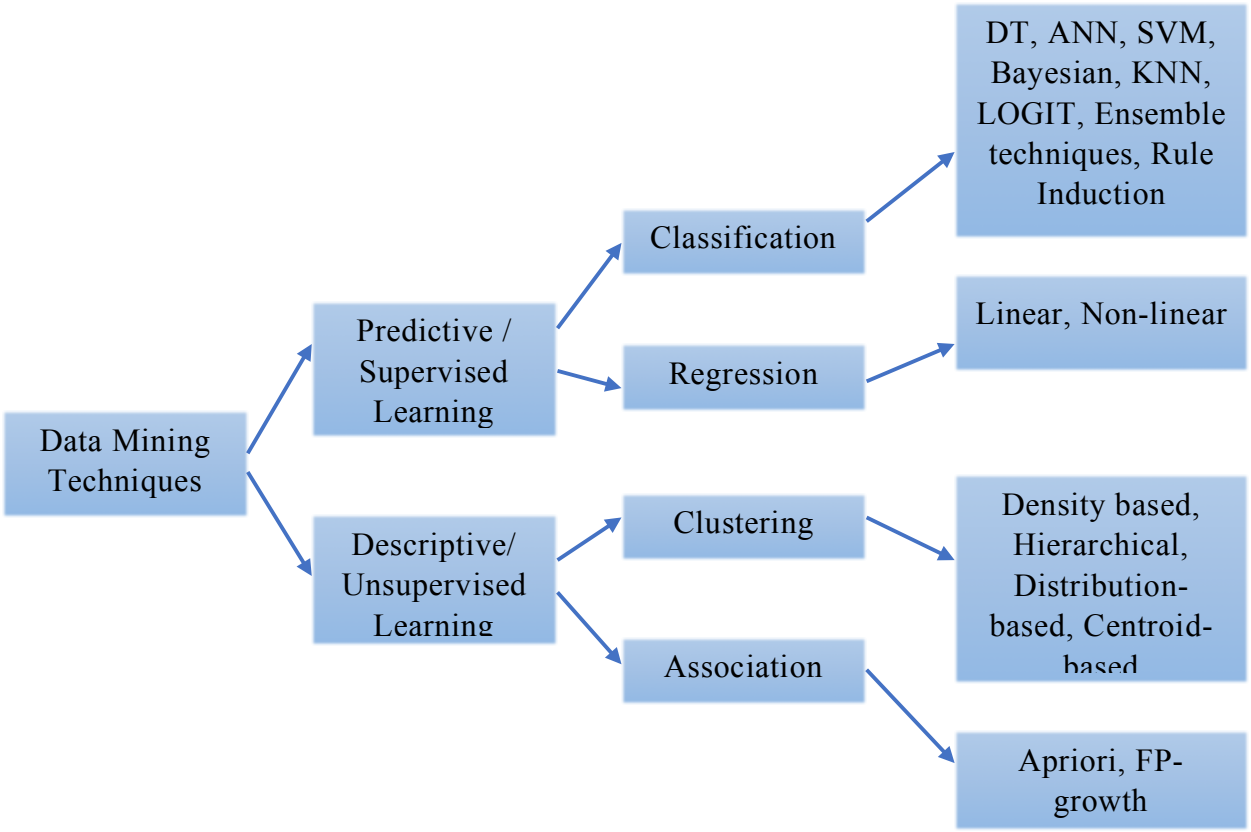
Figure 1: Multidisciplinary nature of Data mining




Machine Learning is a sub-field of data science that focuses on designing algorithms that can learn from the behaviour of the data and can make predictions based on the data. Like Machine learning, data mining problems can also be categorized as Supervised (or Predictive) and Unsupervised (or Descriptive) or Semi-Supervised Learning based on the learning techniques.

Unsupervised or descriptive learning focuses on finding hidden patterns within various attributes of an unlabeled dataset. In an unsupervised learning technique, the target attribute is not specified. The hidden patterns are established based on the relationship among the data instances. Experts then interpret the revealed patterns to discover the knowledge or information in that data. An unsupervised learning technique, clustering, is used to discover natural groupings or classes within an unlabelled dataset.

Figure 2: Datamining technique categorizations




Supervised or Predictive learning involves classification and regression modeling which uses statistical techniques to predict and validate future behaviour of relevant data. In this learning technique, a model is built using a training dataset for which the value of response variable is already known. A predictive model can be built and used using simple validation or cross validation technique. In simple validation procedure, after building the model using training dataset, it is used to predict the classes for the test dataset. On the other hand, in cross validation, all the data is randomly divided into equal sets. In this technique, a predictive model is built on the first set and being validated on the second set. Among researchers the most popular cross validation technique is 10-fold cross validation. Using the

above mentioned validation methods, classification and regression techniques identifies the relationship between training attributes and target attributes of the labeled data. In regression, the target variable is numeric whereas in classification technique, it predicts the categorical variable. Figure 2 represents various categorizations of data mining techniques (Kotu and Deshpande, 2015).

The third learning technique, Semi-Supervised learning, uses both supervised and unsupervised learning technique in order to build a model. In this approach, labeled examples are used to learn class models and unlabeled examples are used to refine the boundaries between classes.

## 2.1 Classification Techniques

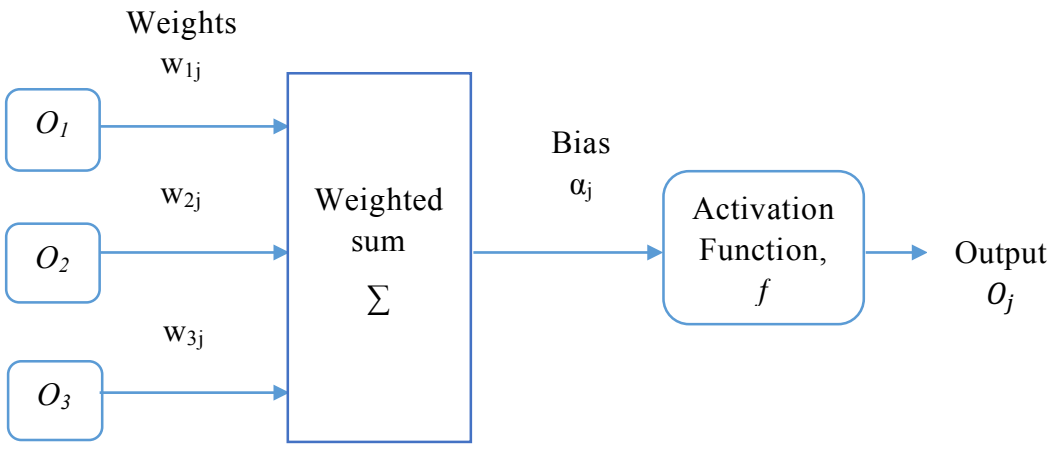
Classification is the most widely used data mining techniques in various studies (Kotu and Deshpande, 2015). A classification model is built to predict a target variable given a set of input variables of a labeled dataset. There are various data mining classification algorithms that can be used to predict the outcome. We present a brief review of some of the techniques that applies to this research work below.

### 2.1.1 Artificial neural network

The neural networks is a supervised machine learning algorithm that is inspired by the neurons in human brain system. A network of node are arranged in three layers; mainly, input layer, hidden layer and output layer. Though artificial neural network (ANN) is criticized for its poor interoperability, researchers find that neural networks give more accurate result, is highly adaptive and very flexible in noise tolerance. Hence, it can generate

a robust model (Ngai et al., 2011). The non-linear statistical data modelling feature of neural network can be used as supervised learning, unsupervised learning and reinforced learning.

Figure 3: A schematic diagram of ANN classifier




The output unit of artificial neural network takes a weighted sum of the outputs from the units in the previous layer. This classification model can be easily modified by changing the weights in the hidden layers. A multi-layer feed forward artificial neural network uses the backpropagation algorithm to modify the weights to minimize the mean- squared error between the network's prediction and the actual target value (Figure 4).

Figure 4 shows a model of ANN backpropagation (Han et al. 2012 p. 402). The inputs of a given unit, $O_1$, $O_2$, $O_3$ …. $O_n$, are either the outputs from the previous hidden layer or the input tuples of the dataset. These inputs are multiplied by their corresponding weights to form a weighted sum.

$$I_j = \sum_{i=0}^{n} w_{ij} O_i + \alpha_j \qquad (i)$$

The weighted sum is then added to the bias associated to the unit. An activation function is applied to calculate the final output of the unit, $O_j$. The pseudocode of the algorithm is presented below (Han et al. 2012 p. 401).

$$O_j = \frac{1}{1 + e^{-I_j}} \qquad (ii)$$

Many researchers (Lin et al., 2015, Ravisankar et al., 2011, Ata et al., 2009, Kirkos et al., 2007, Kotsiantis et al., 2006) have used artificial neural network in their study. This techniques is frequently used in credit card fraud detection, economic and financial forecasting.

### 2.1.2 Decision tree

Decision tree is a supervised learning algorithm that uses a tree-like structure to represent the attributes and possible outcomes. The outcomes or predictions are represented by leaves and attributes by branches. One of the advantages of decision tree (DT) classifier(s) is that it does not require any prior domain knowledge to develop the branches and leafs or to develop the predictive model. Based on top-down attribute selection approach, DT algorithm first identifies the initial node. Subsequently, it uses a series of 'if then' steps along with the attribute selection method to complete the predictive model. DT algorithms use various attribute selection measures, such as, information gain, gain ratio or Gini index. It is also commonly used to predict corporate, credit card and other financial fraud (Sharma and Panigrahi, 2012).

The machine learning algorithms ID3, CART, C4.5 are commonly used as decision tree classification models. ID3 uses information gain as its attribute selection measure. C4.5

uses gain ratio whereas CART uses the Gini index. All these algorithms use simple IF-THEN classification rules to represent the learned knowledge. In our study, we use decision tree C4.5 algorithm.

### 2.1.3 Naïve Bayes

A Naïve Bayes (NB) classifier is based on Bayes' theorem. "This approach calculates the probability for each value of the class variable for given values of input variables. With the help of conditional probabilities, for a given unknown record, the model calculates the outcome of all values of target classes and comes up with a predicted winner" (Kotu and Deshpande, 2015, p. 13). One advantage of using Naïve Bayes classifier is that in order to estimate the parameters it only needs small portion of training data. This method is quite useful when input dimensionality is high (Han et al. 2012). This classification technique is widely used in banking and financial fraud detection (Sharma et al., 2012). Phua et al. (2005) find that in the context of financial fraud detection model, NB classifier outperforms other classifiers. It results in lower false negative cases and almost no false positive cases.

This probabilistic model assumes strong independence among the predictor variables. In other words, it assumes that for any given class (y), the value of a predictor (x) does not depend on other predictors. According to the Bayes theorem, the conditional probability $P(y|x)$, can be determined in terms of $P(y)$, $P(x)$, and $P(x|y)$.

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \qquad (iii)$$

$$P(y|x) = P(x_1|y) * P(x_2|y) * P(x_3|y) * \ldots * P(x_m|y) * P(y) \qquad (iv)$$

Where, $P(y|x)$ is the conditional probability of a class for a given attribute (predictor); P(y) is the prior probability of a class; $P(x|y)$ is the likelihood which is the probability of predictor given a class; and P(x) is the prior probability of predictor.

Since the Naïve Bayes classifier makes the class independence assumption, if the assumption is valid for a given dataset, the Naïve Bayes classifier can predict the best outcome. In such cases, it can have the best accuracy compare to other classifiers.

### 2.1.4 Bayesian Belief Network

Bayesian Belief networks (BBN) follows Bayes theorem. But unlike the Naïve Bayes classifier, BBN does not make the assumption of conditional independence. In real life problems, dependencies can exist among the attributes. In such cases, BBN is preferable over naïve Bayes classifier.

A BBN is a directed acyclic graph (DAG) where the nodes have one-to one correspondence with the attributes. Any node in the network can be selected as a 'class attribute' node. An arrow drawn to node B from node A represents node A as a parent of node B (Han et al. 2012). Based on Bayes theorem, each attribute or node is conditionally independent of its non-descendants given its parents in the network graph. For each data tuple, $X = (x_1, x_2, x_3, \ldots\ldots x_n)$, there is a conditional probability table that represents the conditional probability of the data tuple $X$ for each combination of the values of its parents. Let $P(X)$ be the probability of a data tuple having $n$ attributes.

$$P(X) = P(x_1, x_2, x_3, \ldots\ldots x_n) \qquad (v)$$

$$P(X) = P(x_1|Parents(y_1)) * P(x_2|Parents(y_2)) * \ldots * P(x_n|Parents(y_n)) \qquad (vi)$$

Where $P(x_i|Parents(y_i))$ is the conditional probability of each combination of tuple values of a parent and $i = 1, 2, \ldots . n$. This is one of the widely used classification techniques in fraud detection studies (Kim et al., 2016; Albashrawi, 2016; Sharma et al., 2012; Ngai et al., 2011; Kotsiantis et al, 2006; Kirkos et al., 2007).

2.1.5 Support Vector Machine

Support vector machines (SVM) is one of the most widely used data mining classification techniques (Vapnik, 1995) and may lead to better predictive models in financial domains (Ceccchini et al., 2010). However, some of the existing fraud detection studies (Kotsiantis, et al., 2006; Ravisankar et al., 2011) find that other data mining techniques outperform SVM. This classifier works with feature space instead of the data space. In a binary classification problem, given labeled training dataset, SVM finds an optimal hyperplane to classify the examples based on the largest minimum distance to the training tuples. This hyperplane is expressed as a 'margin'. In a predictive mode, a trained SVM classifier then classifies the target class based on which side of the margin the test sample falls in.

Let $X = (x_{i,}\, y_{i,})$ be the dataset where $x_{i,}$ is the instances, $y_{i,}$ is the class label and $i = 1, 2, \ldots . n$.

For a linearly separable dataset, a SVM with linear kernel can be denoted as

$$f(x) = W \,.\, (x_{i,}\, y_{i,}) + b \qquad (vii)$$

Where, $W$ is the weight vector or the co-efficient and $b$ is a scalar value called bias (Han et al., 2012). However, most of the real world problems falls into the category where

the dataset is not linearly separable. Therefore, a linear margin does not exists that will clearly separate the positive and negative tuples of the dataset. In such cases, a non-linear mapping is used to transform the dataset into a higher dimensional feature space. Thus the non-linear SVM method are kernelized to define a separating hyperplane. Among the SVM's commonly used kernels, radial basis function (RBF) kernel is the most widely used kernel in the extant literature. A RBF kernel can be denoted as

$$K(x_{i,}\, y_{i,}) = e^{-\frac{||x_i - y_i||^2}{2\sigma^2}} \qquad (viii)$$

Where σ is a kernel parameter. Han et al. (2012) pointed out that there is no consensus on administering a specific kernel that will result in the most accurate SVM. According to the authors "the kernel chosen does not generally make a large difference in resulting accuracy. SVM training always finds a global solution" (p. 415). In a recent study on fraud detection, Li et al. (2015) state that "It seems that either financial kernel or expansion is not necessary, as they provide only marginal improvement" (p. 182). Based on the suggestions in existing literatures, we employ SVM RBF kernel in predicting the financial restatements in this study.

### 2.1.6 Random Forest

A Random forest (Breiman, 2011) is a supervised learning technique based on tree-based algorithm. In particular, a random forest (RF) model is an ensemble of classification or regression trees where the trees vote for the most popular class (Han et al., 2012). In this algorithm, a number of decision trees are constructed during training time. Each decision tree classifier is generated using random attribute selection at each of the nodes to determine the tree split.

Let $X = x_1, x_2, x_3, \ldots\ldots, x_n$ be the training dataset with response (i.e. class) variable, $Y = y_1, y_2, y_3, \ldots\ldots, y_n$ with $Z$ attributes. For each iteration, $i = 1, 2, \ldots\ldots k$ the training set is sampled with replacement from $X$ and generates a bootstrapped sample, $X_i$. At each node, a subset of attributes,$B < Z$, is randomly selected to determine the best split at each node using the set of reduced attribute set $B$. Thus, the trees are encouraged to grow to its maximum size without pruning. Later, for classification, majority of voting is used to determine the class prediction. Random forest algorithm uses a subset of attributes for each split and repeat the process for the whole dataset, which make the process quite efficient (Han et al., 2012).

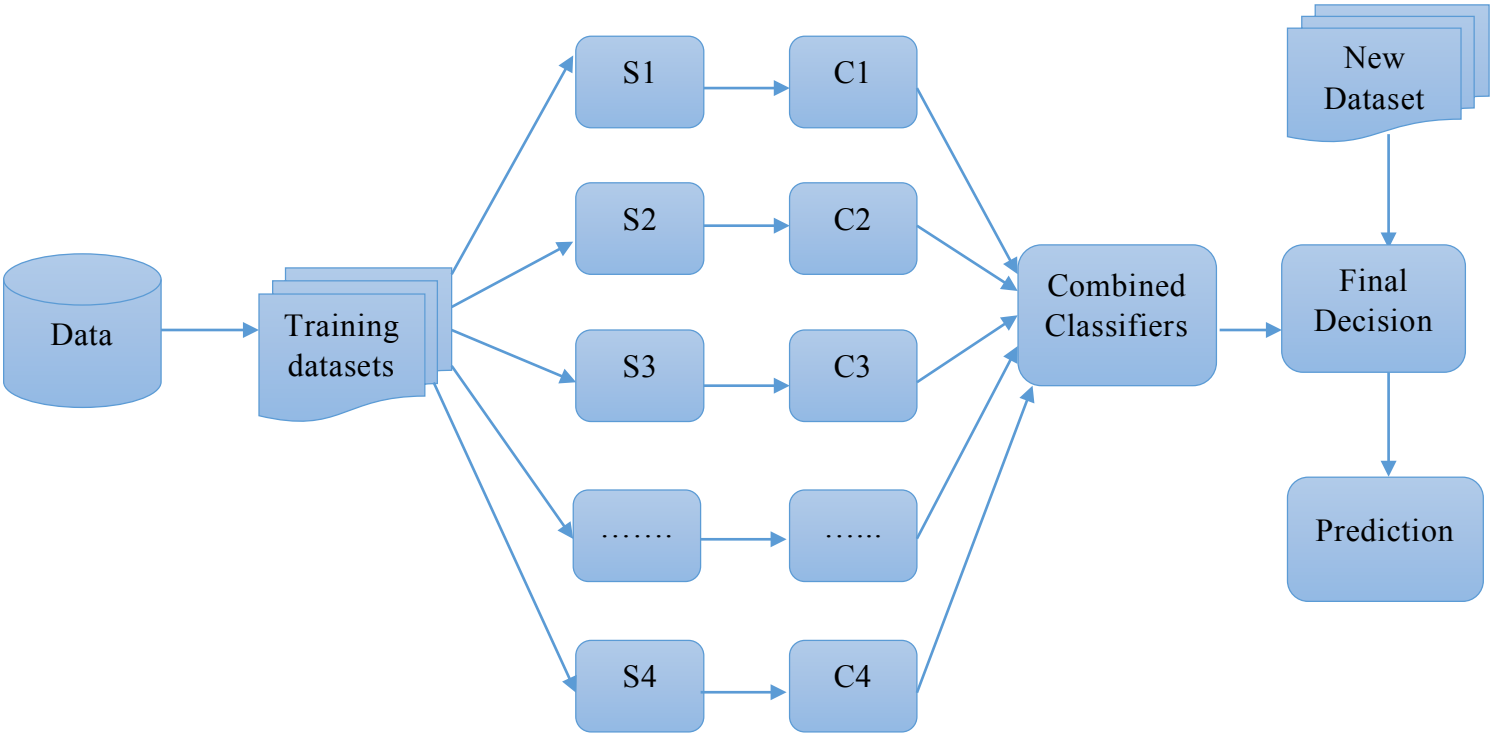
## 2.2 Ensemble Learning

To improve classification accuracy and to make decisions more reliable many researchers have been using ensemble techniques in data mining. Bagging and boosting are two very popular ensemble techniques among researchers. There are also known as 'meta-learning algorithms'. The bagging and boosting technique has been applied to several different classifiers, such as, decision tree (Quinlan, 1997), Naive Bayesian classification (Elkan, 1997).

The ensemble techniques combine various outputs of different models into a single prediction. In ensemble technique procedure a weighted vote or weighted average is calculated. Figure 5 shows an ensemble of classification technique (Kotsiantis et al., 2006). Both bagging and boosting follow this approach to derive the individual model but uses somewhat different approach in terms of weighting schemes. Bagging uses equal model weighting, whereas boosting technique uses a differential weighting scheme, by emphasizing

more on the successful models. Figure 4 presents the schematic diagram for ensemble techniques where S, C represents the samples and classifiers respectively.

Figure 4: Ensemble in classification technique




### 2.2.1 Bagging

Bagging or Bootstrap Aggregating (Breiman, 1996) creates an ensemble of classification models where each model gives an equally weighted prediction.

The pseudocode of bagging algorithm is presented below.

**Algorithm: Bagging**

Input:
*D*, training dataset;
*T*, iteration or the number of models to be created;
A, a classification technique (DT, naïve Bayes, etc.).

Output: The ensemble model, *M*

Method:
Model generation:
for $i$ = 1 to $T$ do
generate bootstrap sample, $Di$ , from dataset D by sampling with replacement;
let model, $M_i$ be the result of training A on $Di$;
endfor

Prediction:
Use of the ensemble to classify a test instance, $X$:
for $i$ = 1 to $T$ do
let $C_i$ be the output of $M_i$ on $X$;
endfor
*return* $C_i$ *that is predicted most often among* $C_1$ *to* $C_T$
let each of the $k$ models classify $X$ and return the majority vote;

### 2.2.2 Boosting

Unlike bagging, boosting algorithm (Freund and Schapire, 1997) focuses on misclassified tuples in an iterative way. It takes into account the performance of built models in previous iteration. In this iterative process, the instances that are not handled correctly in previous iteration are assigned greater weight to boost new models. It is performed through a base learning algorithm. This algorithm trains a series of base learners, and pays more attention to the training tuple wrongly predicted by a base learner. We present AdaBoost or Adaptive Boosting (Freund and Schapire, 1997; Han, Kamber and Pei, 2012), a well-known boosting algorithm, below.

**Algorithm: Adaboost**

Input: *D* = a dataset of labeled training tuples;
*T* = the number of iterations (one classifier is generated per iteration);
A = a classification learning scheme (For example, DT, NB).
Output: An ensemble model.

Let 1/d be the initial weight of each tuple in *D*;
For *i* = 1 to *T* do:
generate *Di* by sampling *D* with replacement according to the tuple weights;
let model, $M_i$ be the result of training A on *Di;*
compute $error(Mi) = \sum_{i=1}^{d} w_i \ err(Di)$
if *error*(*Mi*) > 0.5 then
go back to next iteration and try again;
endif
for each tuple in *Di* that was correctly classified do
multiply the weight of the tuple by *error*(*Mi*) / (1- *error*(*Mi*));
normalize the weight of each tuple;

End For

Prediction:
//Use of the ensemble to classify a test instance, *X*
Let 0 be the initial weight of each class.

For *i* = 1 to *T* do
$w_i = log \frac{1-error(Mi)}{error(Mi)}$ // weight of the classifier's vote
c= $M_i(X)$ // class prediction for test instance X from model M(i)
weight of class c = weight of class c + $w_i$

End For
*return $C_i$ with the largest weight*

## 2.3 Other Meta-learning Techniques

### 2.3.1 Stacking

Stacking or stacked generalization is a meta-learning technique that combines multiple classifiers. Stacking comprises two phases: base learner phase (phase 1) and stacking model learner phase (phase 2). In the first phase, base level learner uses multiple

models and learn from the dataset. The output of each model are synthesized thereafter to create a new dataset. In this dataset, each instance is linked to the real category – which gives a clear indication as to whether the instance is properly predicted. In the second phase, this dataset is used by another learning algorithm, termed as meta-learner, to provide a final output. The intended benefit of stacking algorithm is: if a particular base classifier makes wrong predictions, the second phase classifier might be able to correct this behaviour.

The stacking algorithm is presented below:

**Algorithm: Stacking**

**Input:** $D = \{x_i , y_i\}$ , $i = 1, \dots\dots, m$
, where $x_i$ is an instance in the m dimensional feature space
$C_1, \dots\ \dots\ , C_T$ = Base-level learning algorithm
$M$ = Meta-level learning algorithm

For $i = 1, \dots\dots\dots, T$
$H_i = C_i(D)$ // Train base level individual learner $H_i$
// by applying each learning algorithm $C_T$
End For
$D_{new} = 0$ // Initialize new dataset

For $i = 1, \dots\dots\dots, m$
For $t = 1, \dots\dots\dots, T$
$Z_{it} = H_t(x_i)$ // $H_i$ applied to classify $x_i$
End For
$D_{new} = D_{new} \cup \{ ( (Z_{i1}, \dots.\ , Z_{iT}),\ y_i )\}$
End For
$H_{new} = M(D_{new})$ // Train meta-level learner $H_{new}$ applying meta-learning algorithm $M$

**Output:** $H(x) = H_{new}\ ( H_1(x), \dots\ \dots\ \dots\ , H_T(x))$

## 2.3.2 Cost-Sensitive Learning

Cost sensitive learning associates a penalty with the misclassification of a minority instance in order to highlight the importance of a minority instance detection. In case of a

binary classification, (i.e. minority and majority), where the minority class is the class of interest, the detection model leads to four potential outcomes: (i) true positive, a minority instance is correctly classified as a minority firm; (ii) false negative, a minority instance is incorrectly classified as a majority class; (iii) true negative, a majority instance is correctly classified as a majority class; and (iv) false positive, a majority instance is incorrectly classified as a minority class. False negative and false positive cases are associated with misclassification costs that would reduce the efficiency of a predictive model. Further, cost associated with false negative and false positive classifications are likely to be different. For example, in the context of this study, it would be more costly to misclassify a true restatement instance (minority) as a non-restatement (majority) case. "The classifiers, therefore, have to be both trained using appropriate cost ratios in the training sample and evaluated using relevant cost ratio assumptions" (Perols, 2011, p. 26). In order to implement cost sensitive learning, we assign different cost ratios in the confusion matrix. An example of confusion matrix with different cost ratios is presented in Table 1.

Table 1: A confusion matrix and a cost matrix

| | Actual Positive | Actual Negative |
|---|---|---|
| Predicted Positive | True Positive (TP) | False Positive (FP) |
| Predicted Negative | False Negative (FN) | True Negative (TN) |
| Total | P = TP + FN | N = TN + FP |

| Classified as --> | Cost matrix | |
|---|---|---|
| | R | NR |
| Restatement (R) | 0 | 1 |
| Non-restatement (NR) | 10 | 0 |

In the above cost matrix, a cost ratio 10:1 is assigned for the false negative cases (relative to false positive cases) to penalize the misclassification of minority instances. The intended outcome of this algorithm is to minimize the total misclassification cost.

## 2.4 Chapter Summary

In this chapter, we review the data mining concepts, its categorization and its multidisciplinary nature. We analyse various data mining classification techniques, especially, those are considered in this study. We discuss other meta-level algorithms, such as, bagging, boosting, stacking and cost-sensitive learning that can used to improve the performance of prediction models. In chapter 3, we discuss class imbalance issue and some of the techniques that are commonly used to mitigate class imbalance problem as the dataset used in this study is highly imbalanced.

# CHAPTER 3

# CLASS IMBALANCE

The machine learning (ML) classification algorithms work well when the number of instances of each response variable are roughly equal. In real dataset, it is almost rare to get a balanced dataset. In fraud detection studies, it is very common to have an imbalanced dataset that may have the ratio of 1:100 (Chawla, Bowyer, Hall and Kegelmeyer, 2002). In many other applications the imbalance may be even more; the ratio rise up to 1:10000 (Provost and Fawcett, 2001). Such imbalance generally causes a classifier to be biased towards the majority class (Longadge, Dongre and Malik, 2013). In machine learning, the problem refers to the class imbalance problem where the number of instances of one class significantly exceeds the other classes. Class imbalance problem is well recognized by many researchers in various domains, such as, fraud detection, medical diagnosis, credit scoring, telecommunication as well as text classification. While the class imbalance problem is primarily attributed to the highly skewed distribution of instances in different class, it may have other additional characteristics. Various characterises of an imbalanced dataset are presented below:

- **Between class imbalance:** If the number of instances of one class is significantly less than the instances of other classes, it is termed as the *between class imbalance* in the dataset. Sobhani et al. (2013) posit that regardless of the extent of between class imbalance, if the classes are distinctly separated, standard classification techniques should be able to classify different instances correctly (Japkowicz and Stephen, 2002). However, once the between class imbalance is associated with other types of

complexity (such as within class imbalance and class overlapping), it can lead to misclassification of instances and this problem is more acute with minority class instances.

- **Within class imbalance:** If a class has a number of smaller sub-classes that represent separate subconcepts, it may lead to within class imbalance (Jo and Japkowicz, 2004). "Subconcepts with limited representatives are called "small disjuncts". Classification algorithms are often not able to learn small disjuncts. This problem is more severe in the case of undersampling techniques. This is due to the fact that probability of randomly selecting an instance from small disjuncts within the majority class is very low. These regions may thus remain unlearned" (Sobhani et al., 2014, p. 2).
- **Class overlapping:** In an imbalanced dataset if the instances belonging to different classes overlap, it increases the possibility of misclassification by various standard learning algorithms. The difficulties associated with separating overlapping instances makes a dataset more complex and often classify instances belonging to minority class as the one in majority class (Lin et al., 2017, Batista et al., 2004).

Researchers have developed different methods to mitigate the class imbalance problem to improve classification results. As Krawczyk (2016) mentioned, there are three different techniques to address this issue. These are (i) data level approach, (ii) algorithm level approach, and (iii) hybrid methods. We present a brief description of these three approaches/methods below.

- **Data level approach:** Data level approach deals with sampling techniques to find a suitable distribution of both majority and minority class. Sampling techniques are divided into two categories: undersampling and oversampling. Undersampling techniques reduce the instances of majority class while oversampling techniques increase the instances of minority class with an objective to have a more balanced sample. We discuss these techniques in more detail in the subsequent section.
- **Algorithm level approach:** in the algorithm level approach, the existing classification algorithms are modified to overcome the bias of the majority class for an imbalanced dataset. This requires domain specific knowledge and a good insight into the learning algorithms. One of the widely used techniques in this category is the cost sensitive learning. In this algorithm a penalty is associated with the misclassification of a minority instance in order to highlight the importance of a minority instance detection. Another method to address the class imbalance problem using an algorithm level approach is the deployment of one-class learning technique. This technique, however, requires more specialized knowledge about the dataset and associated complexity. In this study, we primarily focus on cost sensitive learning methodology to address class imbalance problem.
- **Hybrid method:** Hybrid method uses a combination of data level and algorithm level approach to eliminate the majority class bias. There are many ways to do the hybridization. As Krawczyk (2016) states (p. 225), "Hybridization of Bagging, Boosting and random Forests with sampling or cost-sensitive methods prove to be highly competitive and robust to difficult data". Other hybrid techniques include the combination of under or oversampling and stacking. Stacking relies on meta-learning

by combing various learning algorithms and improving the predicting capability of meta-learner based on the feedback from various base learners. In this study, we examine the usefulness of hybrid method by combining under or oversampling, cost sensitive learning and stacking.

In the remainder of this chapter, we highlight various data level approaches that alleviate the concern of class imbalance. Various algorithm level methods such as bagging, boosting, cost sensitive learning and stacking are discussed in chapter 2.

## 3.1 Undersampling

### 3.1.1 Random Undersampling

One of the simplest methodology to create a balanced dataset is random undersampling. To create a balanced distribution of majority and minority class, a subset of the majority class is randomly removed from the tuple (Han, Kamber and Pei, 2012). This technique reduces the data size, hence the run-time of the classification task reduces significantly. On the other hand, random undersampling is subject to high variance (Wallace et al., 2011). The majority class may include some sub-classes. Random undersampling may completely exclude instances from some sub-classes or may select only a few cases from a certain sub-class. This induces risk of losing valuable information and may lead to poor predictive models. Random undersampling algorithm is presented below:

**Algorithm: Random Undersampling (RUS)**

**Input:** $D = \{x_i , y_i\}$ , $i = 1, \dots\dots, n$

Split $D$ into $D_{min}$ and $D_{maj}$
For each classifier $C_j$ , $j = 1, \dots\dots, m$ where $m$ be the total number of folds in cross validation
    $|E|$ be the number of samples to be decreased in $D_{maj}$
    Randomly select $|E|$ samples, $S_{maj}$ from $D_{maj}$
    $T_r = D_{maj} + D_{min} - S_{maj}$
    Train $C_j$ using $T_r$
    $E_j =$ Error rate of $C_j$ on $D$
    $W_j = \log(1/E_j)$
End For
Output: $C_{final}(x) = argmax\ (c)\ \sum_{i=1}^{m} W_j | C_j(x) == c$

### 3.1.2 Cluster based undersampling (CUS)

As we have discussed above, random undersampling has the risk of overfitting the minority class and often valuable information from the majority class gets discarded (Han et al., 2012). When random undersampling removes a large number of instances from a sample, it can create small disjuncts. As Holte et al. (1989) argues this can make classification algorithms error prone. Akbani et al. (2004) also made a similar argument. In light of Sobhani et al. (2014), we present the CUS algorithm.

**Algorithm: CUS**

**Input:** $D = \{x_i , y_i\}$ , $i = 1, \dots\dots, n$

Split $D$ into $D_{min}$ and $D_{maj}$
Cluster $D_{maj}$ into $k$ partition $P_i$, where $i = 1, \dots\dots, k$
For each classifier $C_j$ , $j = 1, \dots\dots, m$
    For each cluster $P_i$:
        $E_{maj}$ += Randomly selected $|D_{min}| / k$ instances of $P_i$
    End For
    $T_r = E_{maj} + D_{min}$

Train $C_j$ using $T_r$
$E_j =$ Error rate of $C_j$ on $D$
$W_j = \log(1/E_j)$
End For

Output: $C_{final}(x) = argmax\ (c)\ \sum_{i=1}^{m} W_j |\ C_j\ (x) == c$

In the CUS algorithm, the final data structure depends on how we partition the majority class instances (i.e. the value of 'k'). The CUS algorithm does not provide any specific guidelines on the determination of optimal number of clusters. Generally, three different methods are employed to find optimal number of clusters: elbow method, gap statistic method and the average silhouette method. However, we recognize that despite having different methodologies, the optimal clustering is somehow subjective. It depends on the parameters used for partitioning with different methodologies.

In this study, we use the average *silhouette* method to find the optimal number of clusters (Kaufman and Rousseeuw, 1990). In this methodology, each cluster is represented by a silhouette score which is calculated based on the cohesiveness and separation of a cluster's objects. Average silhouette method computes the average silhouette of observations for different values of k. The optimal number of clusters k is the one that maximize the average silhouette over a range of possible values for k. We use this optimal k-value in CUS algorithm. Figure 6 to Figure 8 present the optimal k-values obtained for three financial restatement datasets (with different feature selection techniques) used in this study. We find that optimal cluster size for our dataset is 2. Subsequently, we vary k value and check the validity of the optimal cluster solution provided by the average silhouette method.

Figure 5: Optimal number of clusters for CfsSubsetEval dataset

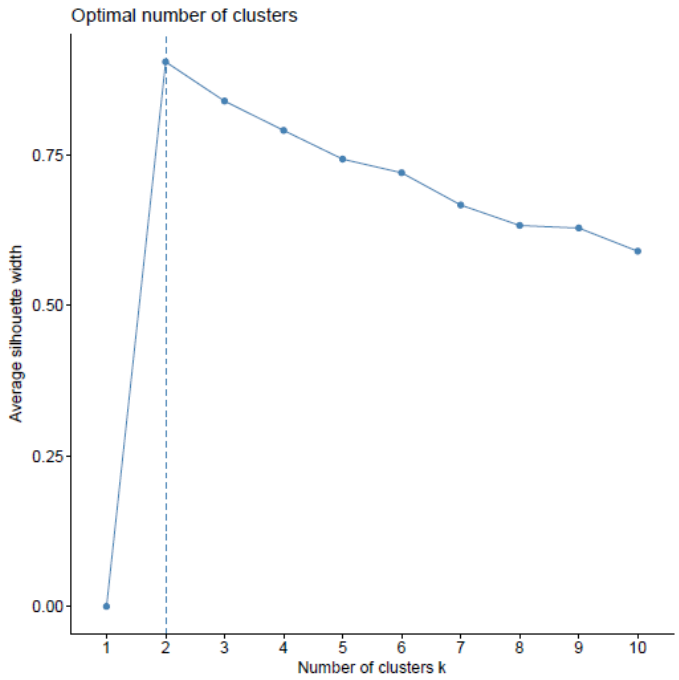



Figure 6: Optimal number of clusters for InfoGainEval dataset

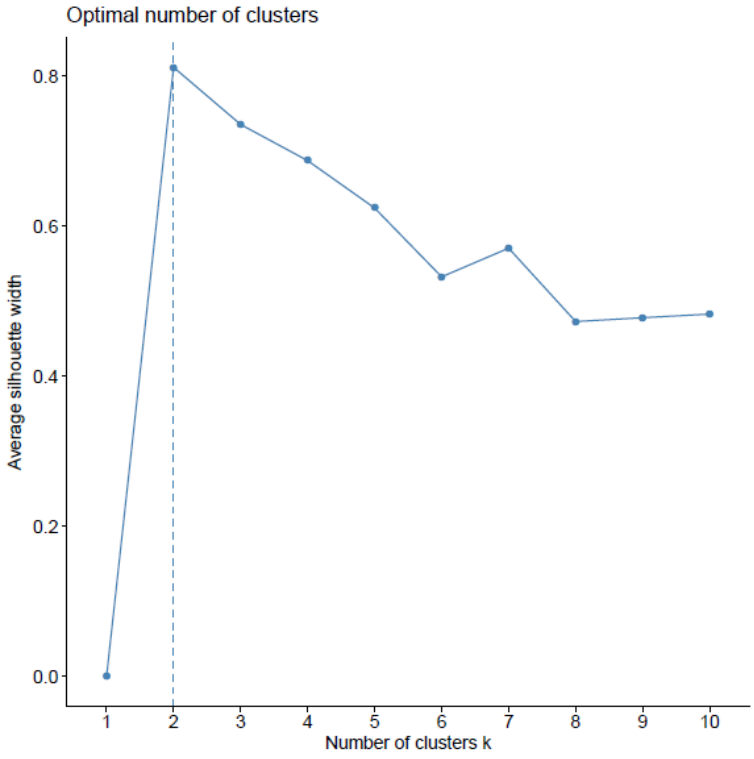
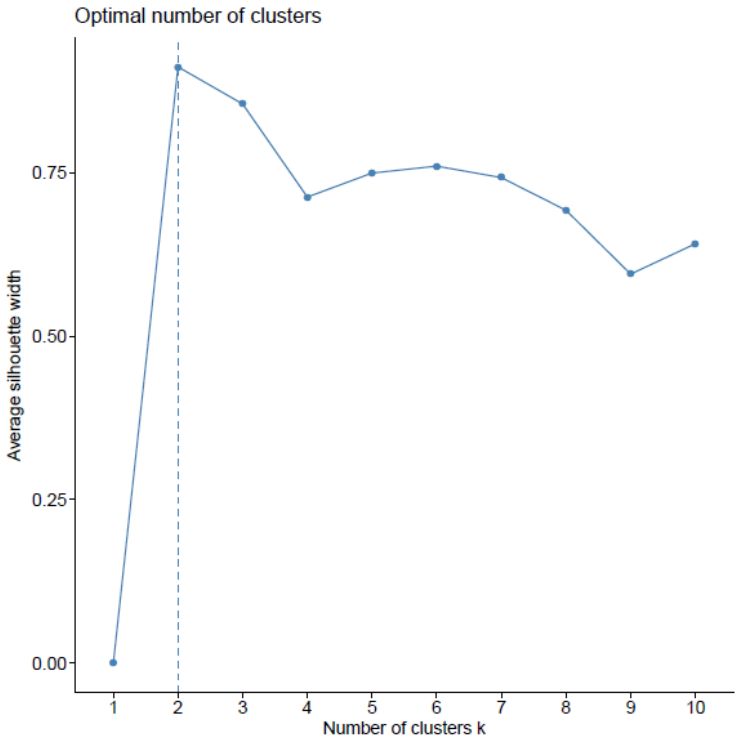


Figure 7: Optimal number of clusters for FwSelect dataset




## 3.2 Oversampling

### 3.2.1 Random Oversampling (ROS)

Oversampling in data mining techniques adjust the training class distribution so that there is equal distribution of minority and majority class. Both oversampling and undersampling works with similar concept but uses opposite technique. In Oversampling, the minority tuples are resampled so that the resulting training set contains an equal number of positive and negative tuples (Han, Kamber and Pei, 2012). Random oversampling produces or replicates minority tuples by sampling with replacement. While it alleviates the class imbalance problem, it increases the risk of overfitting (Batista et al., 2004). To address this

issue we need to add new instances to the minority class. This is challenging and deem less practical due the limited size of minority class sample.

**Algorithm: ROS**

**Input:** $= \{x_i, y_i\}$ , $i = 1, \ldots\ldots, n$

Split $D$ into $D_{min}$ and $D_{maj}$
For each classifier $C_j$ , $j = 1, \ldots\ldots, m$ where $m$ be the total number of folds in cross validation
    $|E|$ be the number of samples to be increased in $D_{min}$
    Randomly select $|E|$ samples, $S_{min}$ from $D_{min}$
    $T_r = D_{maj} + D_{min} + S_{min}$
    Train $C_j$ using $T_r$
    $E_j =$ Error rate of $C_j$ on $D$
    $W_j = \log(1/E_j)$
End For

Output: $C_{final}(x) = argmax\ (c)\ \sum_{i=1}^{m} W_j |\ C_j\ (x) == c$

### 3.2.2 Synthetic Minority Oversampling (SMOTE)

Chawla, Bowyer, Hall and Kegelmeyer (2002) proposed synthetic minority oversampling technique (SMOTE) which generates random synthetic instances of the minority class on the feature space rather than replicating the existing instances of the minority sample. Thus it improves the bias. SMOTE algorithm uses k-nearest neighbour technique to create these synthetic minority examples. For each minority class sample $X_i$ , SMOTE finds the k-nearest neighbours, $N_i$ for that instance. Then, it calculates the difference between the feature sample and its nearest neighbours, $\Delta = X_i - N_i$ . Next, after multiplying this difference by random number 0 or 1, it adds it to the feature sample; thus creating a set of random points along the line segments between two specific features (Chawla, Bowyer, Hall and Kegelmeyer, 2002).

The random points are generated as

$$F_{new} = X_i + (X_i - N_i) \times rand\ (0,1) \qquad (ix)$$

Where $F_{new}$ is the generated random point and $rand\ (0,1)$ is the random variable generated a random number either 0 or 1. In order to create a balanced dataset, different amount of synthetic oversampling may be needed for different datasets.

### 3.2.3 Adaptive Synthetic Sampling (ADASYN)

To overcome the limitations of SMOTE, He et al. (2008) proposed a novel adaptive learning algorithm, ADASYN, to mitigate class imbalance problem in a dataset. Like SMOTE, this algorithm also increases the size of minority instances based on synthetic instance creation. However, in ADASYN, synthetic instance creation for minority class sample is not performed uniformly. More instances are created for the minority class samples that are harder to learn. The difficulty level is determined based on the k-nearest neighbours of the minority samples that belong to the majority class. If a minority class instance is surrounded by more majority class instances, it is considered harder to learn. The ADASYN algorithm (He et al., 2008) is presented below:

**Algorithm: ADASYN**

**Input:** $D = \{x_i, y_i\}$, $i = 1, \ldots\ldots, m$
, where $x_i$ is an instance in the m dimensional feature space $Y$ and
$y_i \in Y = \{1, -1\}$ is the class identity label associated with $x_i$.
$m_j$ = the number of majority class
$m_{in}$ = the number of minority class, where, $m_{in} \leq m_j$ and $m_{in} + m_j = m$

$d_c = m_{in} / m_j$ , the degree of class imbalance, where $d_c \in [0,1]$

G = $(m_j - m_{in}) \times \beta$ , the total number of synthetic minority sample need to be generated, where $\beta \in [0,1]$

For each minority sample $x_i$ , $i = 1, \ldots\ldots, m_{in}$

$k_i$ = the k nearest neighbors of $x_i$ calculated using HEOM distance

$\Delta_i$ = the number of majority class sample in $k_i$

$R_i = \Delta_i / k$ where $R_i \in [0,1]$

Normalize $R_i$: $R_{i-norm} = \frac{R_i}{\sum_{i=1}^{m_{in}} R_i}$

,where $R_{i-norm}$ density distribution $\sum_i (R_{i-norm}) = 1$

$G_i = R_{i-norm} \times G$ , the number of synthetic minority sample need to be generated

For $i = 1, \ldots\ldots, G_i$

$x_{min}$ = Randomly selected minority data sample from $k_i$

$S_i = x_i + (x_{min} - x_i) \times \gamma$

where $S_i$ synthetic data sample, $\gamma \in [0,1]$ , a random number

End For

End For

### 3.2.4 Tomek Links +SMOTE

Tomek link techniques usually can be viewed as data cleaning technique. In a real dataset, there is a possibility that majority and minority class samples are overlapped. These 'overlaps' can be defined by Tomek links (Tomek, 1976; He and Garcia, 2009). For example, dataset = $\{x_i, y_i\}$ , $i = 1, \ldots\ldots, m$ consists of $D_{maj}$ and $D_{min}$ as majority and minority class. Let $E_l = \{x_l, y_l\}$ is an instance of class $D_{maj}$ and $E_k = \{x_k, y_k\}$ is an instance of $D_{min}$ such that $y_l \neq y_k$ where the distance between $E_l$ and $E_k$ is $S(E_l, E_k)$. A Tomek link is represented by the pair $(E_l, E_k)$, if there is no sample $E_X$ exist such that $S(E_l, E_X) < S(E_l, E_k)$ or $S(E_k, E_X) < S(E_l, E_k)$ (Batista et al, 2004). In an imbalance dataset, for a tomek link pair, any one of these samples can be a noise in the dataset. Sometimes it appears that both of these samples exists near the boundary of the minority and majority classes. In such cases, Tomek links can be used to remove the overlap between the classes either by removing both samples forming a Tomek link or by only removing the majority class sample.

The utility of Tomek links depends on the nature of data. If there are no many borderline cases, the Tomek link approach may not give the best result.

## 3.3 Chapter Summary

In this chapter we discuss various undersampling and oversampling techniques to address class imbalance issue in a dataset. Financial restatement datasets are highly imbalanced. Therefore, it is important to address this issue in the financial restatement datasets to mitigate prediction model bias towards majority class. As we discuss in this chapter, random undersampling and random oversampling may not yield the best solution to address class imbalance problem. Existing literature has suggested cluster based undersampling and synthetic instance creation for the minority class (i.e. synthetic oversampling) to improve performance of the predictive models.

# CHAPTER 4

# RELEVANT EXPERIMENTAL EVIDENCE

In this chapter we present an overview of experimental studies in the areas of class imbalance and financial restatement. Since the financial restatement datasets are highly imbalanced, it is important to review the relevant studies on class imbalance that help us in developing better predictive models. Accordingly, in the first part of this chapter, we discuss relevant studies that focus on addressing class imbalance issue using undersampling and oversampling techniques.

In the second part, we focus on the literature on financial restatements. First we summarize the studies on the impact of financial restatements on firm performance. Second, we examine the actions taken by firms and regulators following financial restatements. Finally, we summarize the studies on financial fraud (i.e. intentional restatements) detection models. As we stated in Chapter 1, most of the data mining studies on financial restatements focus on intentional restatements and studies on unintentional restatements are very limited.

## 4.1 Literatures on Class Imbalance

### 4.1.1 Undersampling techniques

Yen and Lee (2009) propose cluster based undersampling technique in selecting the samples from the dataset to improve the classification accuracy for the minority class. They cluster all samples in the dataset into k clusters. The majority class samples in each cluster is then undersampled into the size of m times the size of minority sample in that cluster,

where m is the ratio of majority and minority class in the whole sample. The authors study the undersampling performances on different number of clusters. In the paper, the authors also proposed other methods for undersampling based on clustering and distances between samples and compare the results with previous literatures. Their experiment shows that proposed cluster based undersampling approaches outperform the other undersampling techniques.

In a recent study, Sobhani et al. (2014), discuss the importance of class imbalance problem in various domain, such as, fraud detection, telecommunication management and text classification. As the literature suggest that the traditional classifier may not lead to good model because of within-class and between-class imbalance. The authors proposed a cluster based undersampling algorithm, termed as ClustFirstClass algorithm, which uses cluster analysis to remove the bias of the majority class. The algorithm uses a balanced sample of majority and minority instances where the majority instances are selected from each cluster group. The authors suggest that the proposed algorithm shows encouraging results compared to the other recent methods in terms of G-mean and F-measure.

Agrawal et al. (2015) proposed a hybrid sampling method SCUT, which considers within class imbalance in a multi-class dataset. The authors mention that the proposed algorithm handles within-class and between class imbalance in the dataset. It oversamples minority class using SMOTE and employs cluster based approach to undersample the majority class. The authors have done several experiment and comes with the conclusion that the SCUT approach obtains highly accurate models compared to that of the state-of-the-art models.

In a recent study, Lin et al. (2017) have introduced two undersampling techniques where the clustering technique is used in the data preprocessing step. In their methodology, the number of clusters in the majority class is set to be equal to the number of instances in the minority class. The two techniques include: (i) using cluster centers to represent the majority class, and (ii) using k-nearest neighbours of the cluster center. They have used multi-layer perceptron, NB, SVM and C4.5 algorithm with various datasets from UCI machine learning repository for their experiment. Their experimental results show that using k nearest neighbour approach in cluster based undersampling outperforms the other state-of-the-art techniques.

### 4.1.2 Oversampling techniques

Chawla, Bowyer, Hall and Kegelmeyer (2002) proposed a novel oversampling algorithm, synthetic minority oversampling technique (SMOTE), which generates random synthetic instances of the minority class on the feature space rather than replicating the existing instances of the minority sample. Thus it improves the bias. SMOTE algorithm uses k-nearest neighbour technique to create these synthetic minority examples. A number of subsequent studies have reported improved capability of predictive models using modified SMOTE methodology. Akbani et al. (2004) proposed a variant of SMOTE algorithm - SMOTE with different costs (SDC) that combines different error cost in the SMOTE algorithm. The authors use different error costs with SMOTE for each classes to push the boundary away from the minority instances. They compare the performance of the proposed algorithms with SVM, SMOTE, RUS, different error cost (DEC). They conclude that the proposed algorithm outperforms all the other algorithms experimented in the study.

He et al. (2008) proposed adaptive synthetic sampling approach (ADASYN), a novel adaptive synthetic sampling approach to mitigate class imbalance problem. This algorithm uses the weighted density distribution on the dataset to decide the number of synthetic sample generation for the minority samples according to their level of difficulty in learning. More synthetic instances are created for the minority class samples that are harder to learn compared to those that are easier to learn. Hence, it removes the re-distributes the bias moves the decision boundary towards the difficult learners. The results in this study suggest that ADASYN improves the accuracy for both minority and majority classes in a class imbalance scenario.

He and Garcia (2009) discuss the issue of class imbalanced in knowledge discovery and data engineering domain. In this study, the authors discuss some state-of-the-art algorithms (random undersampling, random oversampling, informed undersampling, SMOTE, cluster based sampling, integration of sampling and boosting) which are used to overcome the class imbalance problem as well as the other existing techniques (e.g. cost-sensitive learning, active learning). He and Garcia (2009) also suggest the effective and accurate performance measures for imbalanced learning.

### 4.1.3 Class Imbalance in financial restatement studies

Financial restatement instances are relatively rare. Many studies use balanced samples to create financial fraud detection model (Deng, 2010; Cecchini et al., 2010; Green and Choi, 1997; Ata and Seyrek, 2009; Chen, 2016). A few studies such as Kim et al. (2016) use a matching sample technique (generally using industry and size matching) to make the data structure more balanced. The authors use cost sensitive learning to mitigate the class

imbalance problem in a multi class framework. Their result shows that the G-mean values achieved by multinomial logistic regression (MLogit), SVM and Bayesian networks (BayesNet) are 69.8%, 65.6% and 58.4% respectively. The authors mention that the resulting G-mean values in the study are not very satisfactory because it is difficult to classify unintentional misstatement instances correctly (Kim et al., 2016).

Moepya et al. (2014) use cost sensitive classification for financial fraud detection. They authors shows the effectiveness of cost sensitive classifier using South Asian market data. They use PCA and factor analysis in the study. The result show that SVM outperforms cost sensitive NB and K-nearest neighbour (KNN) classifier.

In another recent study (Lin et al., 2015), the authors use 'matched-pair sampling' technique by taking one fraud firm to match with four non-fraud firms. They match the control firms based on year, assets size, industry and trade market in the year preceding the fraud event year. The dataset thus contains 129 fraud companies and 447 non-fraud companies. The authors employs DT, ANN, Logistic regression. The study reports overall classification accuracy of 91.2% achieved by ANN model.

Dechow et al. (2011) question the appropriateness of such matching technique in developing a restatement detection model. They argue that matching techniques are useful only if we want to find whether a variable is significantly different relative to a control firm. In the process, however, we ignore majority of non-restatement firms in model building, which reduces the inclusiveness/ generality of a model. Dechow et al. (2011) further stress that "it is more difficult when matching to determine Type I and Type II error rates that users will face in an unconditional setting" (p. 23). Another detail study by Burns and Kedia (2006)

also argues that a matched control firm approach would overstate the likelihood of restatement.

Other options to mitigate the data imbalance issue include undersampling and oversampling. In undersampling, like a matched sample approach, the instances of majority class are reduced. While a number of studies have used this technique (Perols, 2011), there are many challenges with this strategy. First, as we eliminate number of instances from the control group (i.e. majority class), we may lose valuable information on non-restatement firm characteristics (Han et al., 2012; Batista et al., 2004). Second, the resulting predictive model will be less general as it would exclude many firms in the population. Both reasons may also artificially lead to a reduction in the Type I and Type II error rates and improve model performance. Finally, random undersampling may also lead to high variance in the dataset (Wallace et al., 2011).

The other method to address data imbalance issue is oversampling. However, random oversampling that relies on producing minority tuples by sampling with replacement, does not prevent the risk of overfitting (Batista et al., 2004). Chawla, Bowyer, Hall and Kegelmeyer (2002) proposed a modified oversampling method, namely, Synthetic Minority Oversampling Technique (SMOTE), which does not rely on random oversampling. This process generates new instances of restatements using nearest neighbour algorithm and hence does not induce any systematic bias in the data structure (Chawla, Bowyer, Hall and Kegelmeyer, 2002). This algorithm does not rely on any specific attribute(s) of restatement or control groups while generating more instances. Perols (2011) suggest that data mining techniques that focuses on class imbalance problem, such as SMOTE, could be explored to improve classification performance.

## 4.2 Literatures on Financial Restatement

There is a large body of literature on financial restatements. These studies can be broadly categorized into three groups: (i) market reactions to financial restatements, (ii) actions following financial restatements; (iii) data mining based financial restatements/fraud detection models. While our primary interest is to review the usage and effectiveness of various detection models and data mining techniques, we also pay close attention to two other categories of literature to address important methodological issues: data sources and data imbalance. Earlier studies show that these two factors can impact the model outcome considerably. We highlight these two issues more elaborately towards the end of this sub-section.

### 4.2.1 Market reactions to financial restatements

One of the earlier studies by Turner, Dietrich, Anderson, and Bailey (2001) examined the market reactions to 173 restatement cases from 1997 to 1999 period. They report significant negative abnormal market returns, ranging from -12.3% to -5% for different types of restatement over an eight day event window. Palmrose, Richardson, and Scholz (2004) identified 403 cases by searching the Lexis-Nexis News Library using key-word searches for restatements over the period of 1995 and 1999. They find an average abnormal return of about −9 percent over a 2-day announcement window for these restatement cases. While these results are significant - both economically and statistically - market reacts even more negatively for the restatement cases involving frauds (-20%). A later study by Files, Swanson and Tse (2009) used a restatement sample from GAO database and analyzed 381 cases over

a period of 1997 to 2002. They report an abnormal returns of -5.5% over a 3-day (-1 to +1) event window. They further report that market reactions vary according to the prominence of announcements. Overall, these studies report that financial restatements destroy shareholder value quite significantly.

### 4.2.2 Actions following financial restatements

Given the significant impact of financial restatements on shareholders' wealth and firm reputation, a number of studies have focused on the actions taken by firms and their boards following such instances. As discussed in Hennes, Leone and Miller (2014) firms take both short-term and long-term actions following a restatement. In the short-term, relevant studies report that restating firms generally reshape their governance structure and involved committees and replace responsible executives. For example, Farber (2005) finds that following a fraudulent restatement, a firm improves its board structure by inducting more independent directors within next three years. Further, fraud firms hold more audit committee meeting compared to their matched firms following the damaging restatements. Desai, Hogan, and Wilkins (2006), Hennes et al. (2008), and Burks (2010) among others find that restating firms are more likely to replace their top executives following significant restatements.

In order to restore the accounting and auditing quality and to prevent future restatement instances, many firms also replace their auditors. Srinivasan (2005) reports that auditor turnover is significantly higher for restating firms than for non-restating firms. Hennes, Leone and Miller (2014) report that probability of auditor turnover is related to the severity of financial restatements. However, the results are more pronounced for non-Big 4

auditors rather than Big 4 auditors, which can be attributable to the higher switching costs in replacing a Big 4 auditor. Auditor turnover could also be attributed to auditor resignation, as restatements are damaging for auditor reputation. Huang and Scholz (2012) find that due to increased client risk, reputed auditing firms are more likely to resign following restating instances.

### 4.2.3 Studies on financial fraud detection models

As discussed above, financial restatements can have a significant negative effect on shareholders' wealth. In response, firms and auditors take many actions to mitigate reputational damage, restore investor confidence and lower the probability of future restatement incidences. However, it is important to note that there are two types of financial restatements: intentional (i.e. frauds) and unintentional (i.e. material errors in financial statements). Most of the data mining predictive models concentrate on fraudulent or intentional restatements for model building (e.g. Cecchini et al., 2010; Kim et al. 2016) and unintentional restatements have received very little attention in the literature. Further, we find that there is no consensus on the best financial restatement model. It appears that efficiency of detection models depend on data structure and time period. Given the focus of this study, we present a brief overview of data mining studies involving both types of financial restatements. We also summarize the results in Table 35 (in Appendix A).

Existing studies on financial fraud detection mainly explore different classification techniques. In an empirical study, Lin et al. (2015) employ logistic regression, decision trees (CART) and artificial neural networks (ANN) to detect financial statement fraud. The authors concluded that ANN and CART provide better accuracy than the logistic model. In another

comparative study, Ravisankar et al. (2011) apply logistic regression, Support Vector Machine (SVM), Genetic Programing (GP), Probabilistic Neural Network (PNN), multilayer feed forward Neural Network (MLFF) and group method of data handling (GMDH) to predict the financial fraud. The authors used the t-statistics to achieve feature selection on the dataset. They conducted the experiment with or without the feature selection on different classifiers. Their study results showed that PNN has outperformed all the other techniques in both cases.

Ata and Seyrek (2009) use decision tree (DT) and neural networks for the detection of fraudulent financial statements. The authors selected the variables that are related to liquidity, financial situation and the profitability of the firms in the selected dataset. The authors concluded that neural network did better prediction of the fraudulent activities than decision tree method. Kirkos et al. (2007) explore the performance of Neural Networks, Decision Tree and Bayesian Belief Network to detect financial statement fraud. Among these three models, Bayesian Belief Network showed the best performance in terms of class accuracy. Kotsiantis et al. (2006) examine the performance of Neural Networks, Decision Tree, Bayesian Belief Network, support vector machine and K-Nearest Neighbour to detect financial fraud. They proposed and implemented a hybrid decision support system to identify the factors which would be used to determine the probability of fraudulent financial statements. According to the authors, the proposed stacking variant method achieves better performance than other described methods, such as, BestCV, Grading, and Voting. Both studies (Kirkos et al., 2007 and Kotsiantis et al., 2006) used accuracy as performance measure, and ignores the embedded issues of class and cost imbalance in the dataset (Perols 2011).

In a recent paper, Zhou and Kapoor (2011) has discussed the challenges faced by the traditional data mining techniques and proposed some improvements. They posit that while data mining methodologies and fraud detection models have evolved over the years, yet they face an increase level of difficulty in detecting financial frauds. Because "managements have adapted certain ways to avoid being identified by automated detection systems" (2011). In order to improve the efficiency of the existing data mining framework, Zhou and Kapoor (2011) have proposed an adaptive learning framework. According to Zhou and Kapoor, one way to improve the efficiency of data mining technique is to use adaptive learning framework (ALF). This framework proposes the inclusion of many exogenous variables, governance variables and management choice variables that will contribute towards the adaptive nature of financial fraud detection techniques.

Sharma and Panigrahi (2012) suggest that a fraud detection model based on only financial statement data may not give the best prediction. The authors presented a comprehensive review of the exiting literatures on financial fraud detection using the data mining techniques. Their finding shows that regression analysis, decision tree, Bayesian network, Neural network are the most commonly used data mining techniques in fraud detection. Gupta and Gill (2012) present a data mining framework for financial statement fraud risk reduction. In the framework, classification technique is proposed to successfully identify the fraudulent financial statement reporting. In another survey paper, Phua et al. (2005) discuss technical nature of fraud detection methods and their limitations. In a similar study, Ngai et al. (2011) present a comprehensive literature review on the data mining techniques. Their review is based on 49 articles ranging from 1997 to 2008. They show that Logistic Regression model, Neural Networks, the Bayesian belief Network and Decision

Tree are the main data mining techniques that are commonly used in financial fraud detection.

In light of Ngai et al. (2011), Albashrawi (2016) present a recent comprehensive literature survey based on 40 articles on data mining applications in corporate fraud, such as, financial statement fraud, automobile insurance fraud, accounting fraud, health insurance fraud and credit card fraud. Their detail study suggests that logistic regression, DT, SVM, NN and BBN have been widely used to detect financial fraud. However, these techniques may not always yield best classification results (Albashrawi, 2016).

Bolton and Hand (2002) review statistical and data mining techniques used in fraud detection with several subgroups, such as, telecommunications fraud, credit card fraud, medical and scientific fraud, and as well as in other domains such as money laundering and intrusion detection. The authors describe the available tools and techniques available for statistical fraud detection. Weatherford (2002) focusses on recurrent neural networks, backpropagation neural networks and artificial immune systems for fraud detection. In a similar study, Thiruvadi and Patel (2011) classify the frauds into management fraud, customer fraud, network fraud and computer-based fraud and discuss data mining techniques in light of fraud detection and prevention.

Green and Choi (1997) developed a neural network fraud classification model for endogenous financial data. Following Green and Calderon's (1995) study, the sample dataset initially consisted of 113 fraudulent company information. The authors used SEC filed financial statements which clearly indicates fraudulent account balances. Further, they used 10-K company annual reports for 1982 to 1990 from SSEC's Accounting and Auditing

Enforcement Releases (SEC/AAER). Their study revealed that the neural network model has significant fraud detection capabilities as a fraud investigation and detection tool.

Hannes et al. (2008) propose a classification procedure to distinguish unintentional restatement (errors) and intentional restatement (irregularities) in financial restatements. In this paper, the authors used 8-K annual reports for 2002 to 2006 with a clear indication of error, irregularity or any investigation on accounting restatements to distinguish between intentional and non-intentional restatement. Many researchers followed their methodology in classifying errors and irregularities or fraud related to financial restatement.

Abbasi et al. (2012) proposed a meta-learning framework to enhance financial model fraud detection. This framework uses organizational and industry contextual information, quarterly and annual data and more robust classification methods using stacked generalization and adaptive learning. Cecchini et al. (2010) provide a methodology based on support vector machine and domain specific kernel. Their methodology correctly detect 80% of fraud cases using publicly available data and also correctly label 90.6% of the non-fraud cases with the same model. In their paper, the authors classify misstatement vs non-misstatement firms using a graph kernel. Their methodology suggests that using exhaustive combinations of fraud variables has high fraud prediction probability. Further, their experiment shows that their model outperforms F-score model by Dechow et al. (2011) and the neural network model by Green and Choi (1997).

Another recent study by Perols (2011) attempts to evaluate the efficiency of different statistical and machine learning models in detecting financial frauds. Based on prior data mining research, the study uses various classification algorithms available in Weka (a popular data mining software) and perform a comparative analysis. In particular, Perols

(2011) evaluates following six algorithms from Weka: (1) J48, (2) SMO, (3) MultilayerPerceptron, (4) Logistics, (5) stacking, and (6) bagging.[2] The study finds that logistic regression and support vector machine outperforms other four data mining models.

Aris, Arif, Othman and Zain (2015) evaluates the possibility of fraudulent financial statements using statistical analysis, such as, financial ratio, Altman Z-score and Beneish model. They analyzed the financial statement data in a small medium automotive company in Malaysia and suggest further investigation by the management.

Li, Yu, Zhang, and Ke (2015) evaluates machine learning techniques in detecting accounting fraud. Their dataset initially consists of the fraud samples over the period between 1982 and 2010 from SEC's Accounting and Auditing Enforcement Releases (AAERs). The authors excluded the fraud occurrences prior to 1991 since there was a significant shift in U.S. firm behaviour. For further preprocessing, they also excluded the data samples for year 2006 to 2008 because of low fraud percentages. Thus their final dataset consists of fraud sample ranging from 1997 to 2005. In this study, the authors employ linear and non-linear SVM and ensemble methods and use raw accounting variables. They show that raw accounting variables have a better discriminant ability and help in developing improved predictive models.

As stated earlier, most of the earlier data mining models (in this area) focus on intentional or fraudulent restatements. Models involving both intentional and unintentional

[2] "J48 is a decision tree learner and Weka's implementation of C4.5 version 8. SMO is a support vector machine (SVM) and Logistics is Weka's logistic regression implementation. Both these classification algorithms are linear functions. MultilayerPerceptron is Weka's backpropagation ANN implementation, and stacking and bagging are two ensemble-based methods" (Perols, 2011, p. 25).

restatements are very limited. In a recent paper, Kim, Baik and Cho (2016) developed a three-class financial misstatement detection model. Inspired by Hannes, Leone and Miller (2008), they investigate three groups; 1) intentional restatement or fraud, 2) unintentional restatement or error, and 3) no restatement. They have used a matched sample approach to identify non-restatement cases to develop their final dataset. The authors then developed three multi-class classifier model using multinomial logistic regression, support vector machine and Bayesian networks to predict fraudulent intention in financial restatements. However, the study has used only a limited set of unintentional restatement cases which can affect the generality of their predictive models. Further, the use of a matched sample methodology could overstate the likelihood of restatement (Burns and Kedia, 2006) and impact the generality of the model (Dechow et al. 2011).

## 4.3 Chapter Summary

In this chapter, we discuss the existing literatures on class imbalance and financial fraudulent activities using data mining techniques. In the context of class imbalance issue, the existing literature suggests two techniques: undersampling and oversampling - that may mitigate the adverse effect of class imbalance while developing predictive models.

The literature on financial restatements suggests that data mining techniques have been widely used to detect financial fraudulent activities by the practitioners and regulators. These techniques are also well researched by the academics. However, studies involving all types of financial restatements are quite limited. Further, it appears that effectiveness of datamining techniques vary with respect to data sources, data structure and time period. In

order to respond to such dynamic environment, it is important to continue with testing various data mining techniques with newer data sets and time period.

# CHAPTER 5

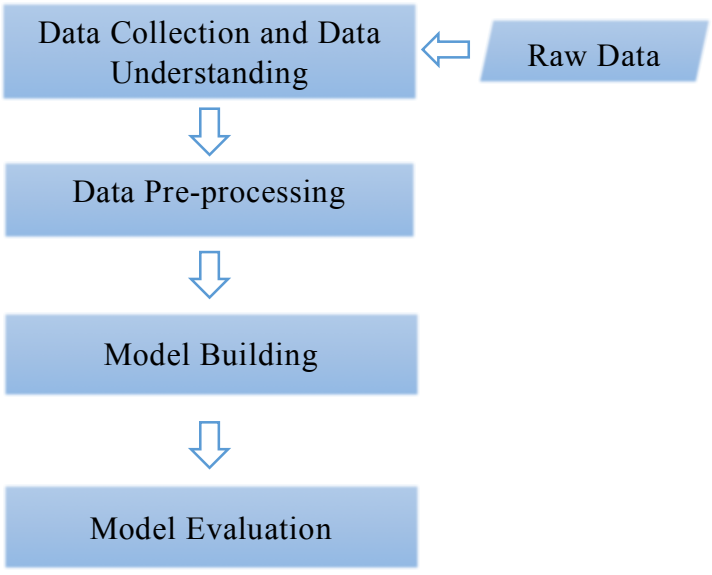
# REARCH DESIGN AND METHODOLOGY

## 5.1 CRISP- DM methodology

Our research methodology follows one of the most popular and widely adopted framework, the Cross Industry Standard Process for Data Mining (CRISP - DM). A consortium of many companies who were involved in data mining developed this frame-work (Chapman et al., 2000).

It consists of four phases. A schematic diagram of the research method that is implemented in this study is shown in Figure 9. In the first phase (data collection phase) we collect data from various sources, such as Compustat, and Audit analytics. This first phase also includes gathering domain knowledge about the data.

Figure 8: A schematic diagram of the research methodology: CRISP-DM Approach



In the second phase or the data pre-processing phase, we first address the missing and duplicate data issue by deleting these instances. Thereafter, we select appropriate attributes by using domain knowledge and following existing research (Dechow et al, 2011; Perols, 2011; Ravishankar et al., 2011). In order to mitigate the outlier effect, we further winsorize the data at 1% and 99% percentiles.

The Feature selection is a process in which attributes in a data set is reduced to a most important attributes. We perform feature subset selection using correlation based feature selection (CfsSubset), information gain feature selection (InfoGain) and stepwise forward selection (FwSelect) to remove less significant or redundant attributes. This step reduces total attributes or features from 116 to 26, 37 and 15 attributes using CfsSubset, InfoGain and FwSelect respectively. This process creates three datasets for further analysis, which help building a more parsimonious model. These datasets are used to examine the efficiency of different classifiers as employed in the study.

We find that financial restatement dataset is highly imbalanced (1:20). In order to address the class imbalance issue, we use various undersampling techniques (random undersampling and cluster-based undersampling) and oversampling techniques (random oversampling, SMOTE, Adasyn). We further used Tomek link to address class overlapping. Thereafter, we experiment with different baseline classifiers including Artificial Neural Networks (ANN) (Green and Choi, 1997; Kirkos et al., 2007; Feroz, Kwon, Pastena and Park, 2000; Kotsiantis et al., 2006; Ata et al., 2009; Perols, 2011; Ngai et al., 2011; Ravisankar et al., 2011; Lin et al.. 2015), Decision Tree (DT) (Kirkos et al., 2007; Kotsiantis et al., 2006; Ata et al., 2009; Perols, 2011; Ngai et al., 2011; Lin et al.. 2015), Naïve Bayes (NB) (Sharma et al., 2012), Bayesian Belief network (BBN) (Kirkos et al., 2007; Kotsiantis

et al., 2006; Ngai, Hu, Wong, Chen and Sun, 2011), SVM (Moepya et al, 2014) and Random Forest (RF) (Liu et al, 2015) to build predict models. Finally, we employ hybrid models (combining various sampling methods with bagging, boosting, cost-sensitive learning algorithm and stacking learning algorithms) to further improve model effectiveness.

The fourth and final phase is the model evaluation phase where we analyse the models' performance using appropriate performance measures, such as, recall, G-mean, F-measure. Weka presents a number of performance measures that can help us in determining the most appropriate learning algorithm. Most of these measures are derived from the confusion matrix. A detail discussion on these performance measures is presented in section 5.2.

A more detailed and operationalized diagram of the research methodology is presented in Section 5.2 (Figure 9).

## 5.2 Experimental Framework

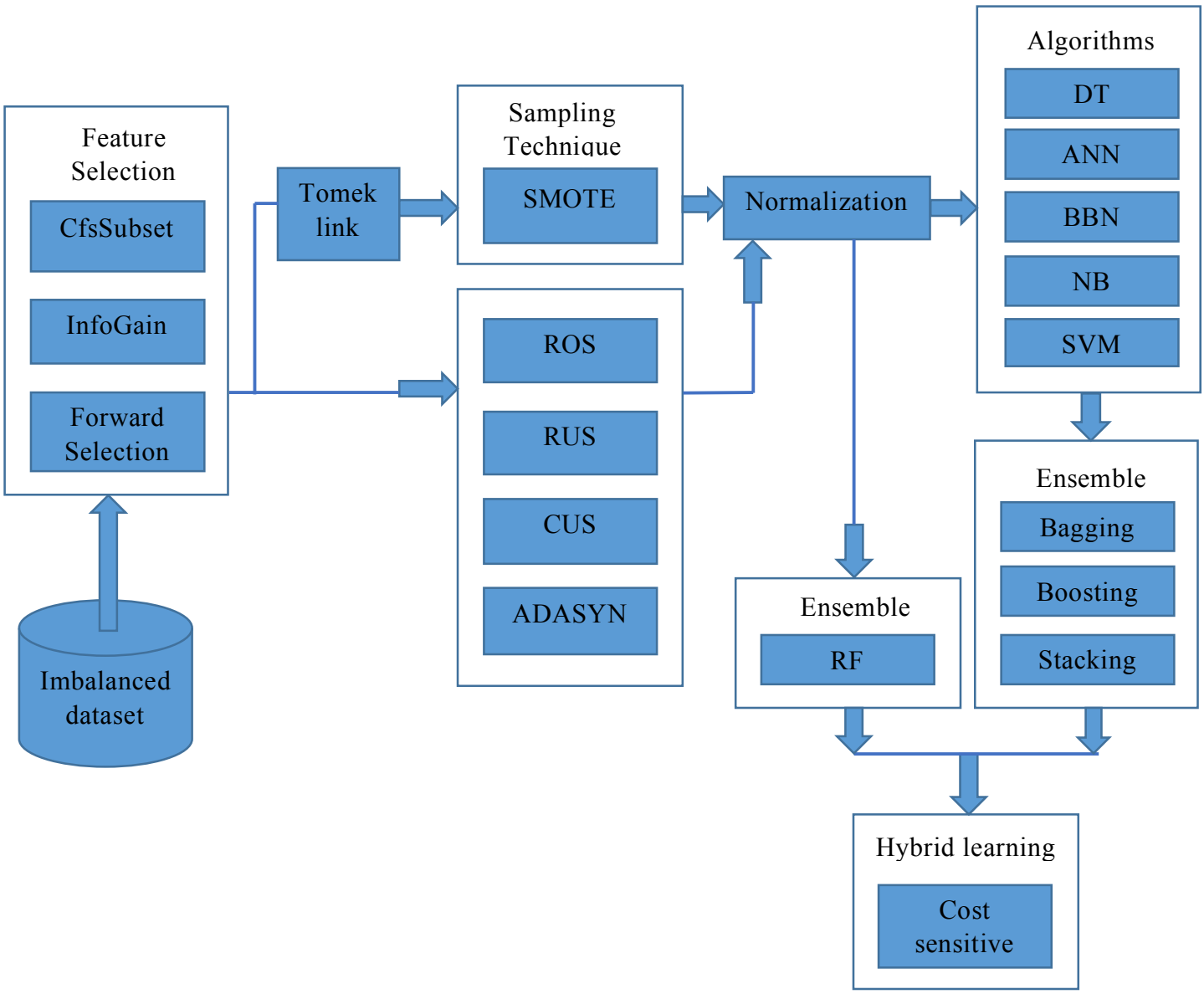
In the schematic diagram (Figure 9), we have presented the sequence of different steps undertaken to clean and pre-process the data, address class imbalance concerns and build predictive models. The framework use the following stages:

- Use of different feature selection techniques (namely, CfsSubset, InfoGain, and FwSelect) for attribute reduction, which will lead to more parsimonious predictive models.
- Use of various undersampling techniques (Random undersampling (RUS), Cluster-based undersampling (CUS)) and oversampling techniques (Random oversampling (ROS), SMOTE, ADASYN) to address the class imbalance issue. Earlier data mining

studies on financial restatement detection primarily relies on matching sample technique to obtain a balanced sample and generally do not employ other cluster based undersampling or synthetic oversampling techniques.

- Use of various classifiers (including Decision Tree (DT), Artificial Neural Networks (ANN), Naïve Bayes (NB), Support Vector Machine (SVM), and Bayesian Belief Network (BBN) classifiers) to predict financial restatements
- Use of ensemble models (employing Random Forest (RF), and other base classifiers with bagging, boosting, and stacking learning algorithms) and hybrid models (combining various sampling methods and classification algorithm with cost-sensitive learning) to further improve model effectiveness.

Figure 9: Outline of research methodology for financial restatement prediction




## 5.3 Performance Measures

In various studies, researchers use various performance measures to verify the correctness of a classification technique. In a binary classification problem, the models'

performance is analysed using appropriate performance measures, such as, recall, false positive rate (FP), accuracy, precision, specificity and area under the ROC curve (AUC). Most of these measures are derived from the confusion matrix. In case of an imbalanced dataset, if the class of interest is the minority class, then the recall of the minority class, G-mean, F-measure are the best performance measure because these values indicate the effectiveness of predicting minority instances. Figure 10 presents a confusion matrix and its components for a binary classification problem.

Figure 10: A confusion matrix

| | Actual Positive | Actual Negative |
|---|---|---|
| Predicted Positive | True Positive (TP) | False Positive (FP) |
| Predicted Negative | False Negative (FN) | True Negative (TN) |
| Total | P = TP + FN | N = TN + FP |

In order to compute the performance measures effectively, it is imperative to understand the confusion matrix elements. It contains four characteristics values. A brief description of these values are presented below:

*True positive (TP):* The tuples from the test set that are correctly classified as positive.

*True negative (TN):* The negative tuples from the test set that are correctly classified as negative.

*False positive (FP):* The negative tuples that are incorrectly classified as positive by the classifier.

*False negative (FN):* The positive tuples that are incorrectly classified as negative by the classifier.

Based on these characteristic values of the confusion matrix, the performance measures of a classifier are calculated.

*Sensitivity or Recall or the true positive (TP) rate*: It is the proportion of correctly identified positive instances whereas false positive (FP) rate is the proportion of falsely classified positive instances. In this study, Sensitivity measures the number of restatements instances that are correctly identified as a restatement by a particular model to the number of actual restatement instances.

$$Recall\ or\ Sensitivity = \frac{TP}{TP + FN} \qquad (x)$$

*False Positive (FP) rate*: As FP measure the negative tuples that are incorrectly classified as positive by the classifier. In this study, it measure the number of restatements instances that are incorrectly identified.

$$False\ positive\ rate = \frac{FP}{FP + TN} \qquad (xi)$$

*Accuracy*: The accuracy measures the true instances (i.e. correctly classified restatement and non-restatement instances) among the whole population. It can be described as the ratio of correctly classified instances to the total number of instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (xii)$$

*Precision*: It is the ratio of correctly identified restatement instances predicted as restatement instances to the total number of actual and predicted restatement instances.

$$Precision = \frac{TP}{TP + FP} \qquad (xiii)$$

*Specificity or true negative rate*: It is a measure of the proportion of the number of non-restatement instances predicted as non-restatement by a model to the total number of actual non-restatement instances.

$$Specificity = \frac{TN}{FP + TN} = 1 - false\ positive\ rate \qquad (xiv)$$

*Area under ROC curve*: An ROC curve is a plot in which the x- axis denotes the false positive rate (FPR) and y-axis denotes the true positive rate (TPR) of the classifier (Fawcett, 2006). It is a standard technique that denotes a classifier performance on the basis of true positive and false positive error rates (Swets, 1988). As Han et al., 2012 describe:

> "For a two-class problem, an ROC curve allows us to visualize the trade-off between the rate at which the model can accurately recognize positive cases versus the rate at which it mistakenly identifies negative cases as positive for different portions of the test set. Any increase in *TPR* occurs at the cost of an increase in *FPR*. The area under the ROC curve is a measure of the accuracy of the model" (p. 374).

Traditionally, we use area under the curve (AUC) as a performance metric for a ROC curve (Duda, et al., 2001; Bradley, 1997; Lee, 2000; Chawla et al., 2002). As the AUC shows the relationship between the true positive rate and the false positive rate, the larger the AUC, the

better is the classifier performance. However, as He and Garcia (2009) mentioned, in a highly imbalanced dataset, the ROC curve may not be a good measure because it may provide an overly optimistic performance of the model.

*F-Measure or F score*: F-measure is the harmonic mean of precision and recall. It combines precision and recall as a measure of effectiveness of classification in terms of ratio of the weighted average of the recall and precision (Fawcett, 2006).

$$F - measure = 2 \times \frac{precision \times recall}{precision + recall} \qquad (xv)$$

*Geometric mean or G-mean*: The G-mean performance measure is originally proposed by Kubat and Matwin, (1997). Many researchers employed G-mean as one of the performance measures (Su and Hsiao, 2007) for class imbalance situations. G-mean incorporates the relative performance of both minority and majority class identification. This gives a balanced view of a particular classifier performance. In certain classification models, the recall value could be high, but the specificity could be low. Therefore, just focussing on the recall value could be misleading. G-mean addresses this issue by considering both recall and specificity values in performance evaluation.

$$G - mean = \sqrt{Recall * Specificity} \qquad (xvi)$$

## 5.4 Chapter Summary

CRISP-DM methodology is a data mining process model that describes the structured approaches to plan and solve a data mining project. In this chapter, we describe the CRISP-

DM methodology and explain various phases of CRISP-DM methodology, namely, data collection, data pre-processing, model building and model evaluation. Further, we discuss the confusion matrix and the performance measures that are derived from the confusion matrix. We use these performance measures to evaluate the developed models.

# CHAPTER 6

# DATA COLLECTION AND PREPROCESSING

In this chapter, we discuss the first phase of CRISP-DM methodology. In this phase, we collect the relevant data from various the data sources, construct the final dataset for model building and preprocess all the relevant data for this study.

## 6.1 Data Sources

In this study, we use primarily two different data sources. *First*, financial restatement incidences are collected from Audit Analytics (AA) database. We restrict our sample from 2001 to 2014 financial year. *Second*, in order to develop the predictive models it is imperative to gather all relevant financial data. We collect relevant financial data from COMPUSTAT database. Thereafter, by using CIK identifier, we match the cases obtained from AA database with relevant COMPUSTAT entries. We present a more detail discussion on data download and filtering process in Section 5.2.

### 6.1.1 Various data sources on financial frauds/misstatements

Prior studies have used different data sources in order to identify fraud/ misstatement cases. However, these data sources differ in terms of scope and data coverage. As a result, the data structure and the instances of misstatements could differ quite significantly depending on data sources. Since the fraud/ misstatement cases are relatively rare, any small variation in instance count may affect the outcome of detection models. We provide a brief description of various data sources below.

***The Government Accountability Office (GAO) financial restatement database:*** The GAO database covers a relatively shorter time period (January 1997 to September 2005) but include a large number of restatements (2,309 cases). It includes all restatement cases relating to accounting irregularities without differentiating in terms of managerial intent, and materiality. One shortcoming of this database is that it includes information only on the first year of restatement without mentioning the overall restatement periods and it does not cover any restatement instances in the post-crisis period.

***Stanford Law Database on Shareholder Lawsuits:*** This database includes all shareholder lawsuit instances. However, one major challenge with using this database in the context of developing financial fraud/ misstatement detection model is that lawsuits are initiated not only for financial restatements but also for many other unrelated reasons.

***Accounting and Auditing Enforcement Releases (AAER) database:*** Dechow et al. (2011) present a useful discussion on AAER database, which is based Security Exchange Commission's (SEC) action against a firm or related parties due to rule violation.

> "The SEC takes enforcement actions against firms, managers, auditors, and other parties involved in violations of SEC and federal rules. At the completion of a significant investigation involving accounting and auditing issues, the SEC issues an AAER. The SEC identifies firms for review through anonymous tips and news reports. Another source is the voluntary restatement of the financial results by the firm itself, because restatements are viewed as a red flag by the SEC. The SEC also states that it reviews about one-third of public companies' financial statements each year and checks for compliance with GAAP" (p. 24).

One advantage of using AAER database is that identification of a misstatement does not depend on a researcher's judgement. These are determined by the SEC after a thorough investigation. The empirical results using AAER database, thus, can be easily replicable. However, AAER database does not include many restatement cases, especially the ones that are not investigated by SEC.

***Audit Analytics (AA) database:*** It is a commercial database which includes all restatement cases from 2000 to current period. Karpoff, Koester, Lee and Martin (2014) present a useful discussion on Audit Analytics data collection process:

> "Audit Analytics (AA) defines a restatement as an adjustment to previously issued financial statements as a result of an error, fraud, or GAAP misapplication [and] does not include restatements caused by adoption of new accounting principles or revisions for comparative purposes as a result of mergers and acquisitions. AA extracts its data principally from SEC Form 8-K or required amended periodic reports (Forms 10-K/A, 10-Q/A, 10KSB/A, 20-F/A, and 40-F/A). AA claims to analyze all 8-K and 8-K/A filings that contain "Item 4.02 - Non-Reliance on Previously Issued Financial Statements or a Related Audit Report or Completed Interim Review" (an item required by the SEC since August 2004). In addition, all amended Forms 10-K/A, 10-Q/A, 10KSB/A, 20-F/A, and 40-F/A are reviewed to determine if the amendment is due to a restatement, and all audit opinions are searched for derivatives of the word "restate" with the intent of detecting the so-called "stealth" restatements contained in periodic reports rather than event filings" (Appendix A and B, page 9, 10).

While all four data sources have been widely used in financial restatement studies (Karpoff et al., 2014), Audit Analytics database is the most suitable one for our study because of its extensive coverage of all restatement cases. In this study, since we focus on all restatement

cases (not just the fraudulent cases) that affect income statement or shareholder equity negatively, we need to reply on a comprehensive data sources on restatement cases, such as Audit analytics.

## 6.2 Data Preparation / filtering

Pre-processing a dataset is a crucial factor in the experimental setup. We use R, WEKA and STATA (a data analysis and statistical software), to preprocess and integrate different datasets collected from two different data sources. We merge the data collected from AA database and Compustat database. We go through several preprocessing steps for data pre-processing to obtain the final dataset for building the model. The steps are described below:

- For each restatement instance, there is a possibility that a firm can be affected over multiple period. Existing literatures suggest that we need to consider only 1$^{st}$ year of the restatement period (Hennes et al, 2014). We have addressed this issue during data construction process and made sure that subsequent restatement years (i.e. other than the 1$^{st}$ year) are removed from the dataset. This process is automated using Stata code.
- As presented in the literature review section, the usual notion is that all restatements have negative impacts. However, a closer look at the dataset suggests that there are some restatements that improves financial statements of affected year. So, it is important to distinguish those restatements and eliminate those cases. Therefore, in the dataset we considered only the 'adverse affect' restatements.
- In the Compustat database, there are some duplicates entries. We needed to address this issue. We delete the duplicates from the sample dataset with respect to year and CIK.

- In the Audit Analytics (AA) database there are some restatement instances which arise because of accounting regulation changes. To create a more meaning restatement dataset, we exclude those restatements (Hennes et al. 2008)
- A closer look at the dataset shows that some restatements can occur due to clerical errors. As per the existing literature, these restatement cases need to be excluded from the dataset. We automate this deletion using Stata.

Our final dataset includes 1,000 restatement instances and 19,157 non-restatement instances over a period of 2009-2014.

## 6.3 Related Attributes

We use a comprehensive list of attributes while developing relevant predictive models. Based on the existing literature, we first identify a large set of attributes (116) that are related to fraud/ financial restatement detection models (Dechow et al, 2011; Perols, 2011, Ravishankar et al., 2011; Green and Choi, 1997; Lin et al., 2003). A list of the initial set of 116 attributes is presented in Appendix B.

Our class variable has two categories; (i) cases without restatement incident in a year – denoted with a value '0', and cases with restatement incident in a year – denoted with a value '1'. In the final dataset we have 1,000 cases with restatement category and 19,157 cases with non-restatement category. It appears that the classes (restatement vs. non-restatement) are highly imbalanced. We address this issue in the dataset cleansing stage, as discussed in a later section (Section 6.4).

## 6.4 Data Pre-Processing

Data preparation is another important step in the experimental setup. During data cleansing, any non-required attributes or noise should be removed to run the classifier successfully. For example, the 'firm ID' and 'ticker symbol' attributes are removed from the initial dataset downloaded from COMPUSTAT database.

It is also very important to address the missing and mismatched values in the dataset. For example, in many instances the Tobin's Q (tob_q) values were missing in the dataset. These cases are replaced by '?' to make it readable for Weka. Similarly, for some of the instances the percentage sign from performance variables (e.g. roa_ebit) is removed by changing data type to keep the consistency with other data. Next, we winsorize the data at 1% and 99% percentiles to limit the outliers.

### 6.4.1 Normalization

The normalization is done by scaling the numeric attributes to the range of 0 to 1. This process changes the largest value for each attribute to 1 and the smallest value is 0. This technique is used when the distribution of the data is not Gaussian (a bell curve) or the distribution is unknown. In real dataset, some of the attributes may have larger values compare to the other attributes. In such cases, the classification result can be biased. By using min-max normalization we can eliminate the possibility of the bias. Also, by using normalization on numeric attributes of the dataset we bring both categorical and numerical attributes to the same scale. The min-max normalization technique scales all numeric attributes in the range [0,1], can be expressed as

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (xix)$$

In this study, we normalize all the numeric attributes in the dataset using Weka (Hall, 2009) unsupervised filter 'Normalize' and applying it to all the datasets.

## 6.5 Attribute Selection / Reduction

Ravisankar et al. (2011) show that we can obtain almost similar efficiency for a predictive model by carefully selecting a smaller set of attributes. To create an accurate predictive model, it is necessary to remove irrelevant and redundant attributes from the datasets. This will also results in reduction of algorithm learning time (Bryll et al., 2003). The data mining literature proposes various feature selection techniques to remove the redundant or irrelevant attributes. At times, it is difficult for a domain expert to know about the most important attributes. Elimination of wrong attributes may affect the performance of classification algorithm negatively (Han et al., 2012). As a result, it is desirable to use feature selection techniques to improve model performance. In this study, we use three feature selection techniques: CfsSubset Evaluation (Rajarajeswari and Somasundaram, 2012), Information Gain (InfoGain) Attribute Selection (Kim et al., 2016; Abbasi et al., 2012), and Stepwise Forward Selection (Han et al., 2012). In the CfsSubset Evaluation algorithm a subset of attributes is selected based on those attributes' predictive ability as well as their mutual redundancy. In other words, the attributes selected through CfsSubset Evaluation algorithm are highly correlated with the target class but show low intercorrelation among themselves. "Information gain (IG) measures the amount of information in bits about the class prediction, if the only information available is the presence of a feature and the

corresponding class distribution. Concretely, it measures the expected reduction in entropy (uncertainty associated with a random feature) (Mitchell, 1997)" (Roobaert et al., 2006; p. 463). In the stepwise forward selection approach, we add one variable at a time to the model. At each step, each variable that is not already in the model is tested for inclusion in the model. The most significant of these variables is added to the model. The significance is measured by P-value and compared with a pre-specified level (e.g. 0.05). Since there is no agreement in the literature on the best feature selection method, we use all three methods and evaluate our model performance. Lists of reduced attributes using CfsSubset evaluation, Information Gain (InfoGain) attribute selection, and Stepwise Forward selection are presented in Appendix C, D, and E. As we find, there are some overlaps between 3 selected feature sets. The common attributes are presented in Table 51 (Appendix H). The overlapping attributes are mainly observed between InfoGainEval and CfsSubsetEval feature selection methods. These two feature selection methods are more commonly used in data mining literature.

### 6.5.1 Forming different datasets

Earlier studies show that financial misrepresentation could lead to a substantial loss in terms of market value and reputation loss of firms (Karpoff et al., 2008). A mere look at the financial penalties imposed by the regulatory authorities (e.g. SEC), however, may not give us a holistic perspective. While actual penalty might seem lower, stock price of the firms that are accused of financial misrepresentation dip significantly in the subsequent period. Such instances are not desirable for the investors, employees and other stakeholders alike. Furthermore, Chakravarthy, Haan and Rajgopal (2014) point out that it requires a

significant level of efforts by the firms in rebuilding their reputation during the post financial misrepresentation period.

Therefore, it would be quite helpful for the investors, stakeholders and regulators if they can predict the financial misrepresentation (or, restatement) behaviour of a firm. This will allow the respective parties to take precautionary measures well in advance.

While there have been a number of studies on financial fraud detections, most of the studies worked on different sample period. More importantly, empirical studies on U.S. corporate fraudulent data in the post financial crisis are quite limited. The sample period of this study (2009 – 2014) will give an opportunity to examine the performance of fraud detection models in the post-financial crisis period. One implication of this development is that we may be able to come up with better predictive model in the post financial crisis period.

**CfsSubset**: It contains the data on the full sample period from 2009 to 2014 after correlation based feature selection. It includes all financial data obtained from COMPUSTAT dataset, industry dummy variable (12 categories), and a measure that denotes the industry concentration based on sales (hfi – Herfindahl Index). This data includes 26 attributes and all restatement cases on the full sample period (between 2009 and 2014 period). As we are interested in developing a predictive model based on the past financial information, we use lagged data (by one year) for all attributes. For example, if the restatement data is for year 2010, we use all financial data from year 2009. Failing to organize data in this way will induce biases in our analysis.

As the two classes are highly imbalanced in the dataset, we opted to employ various over and undersampling techniques to mitigate class imbalance problem. Thereafter, we use the randomized algorithm to shuffle the order of instances passed through it.

**InfoGain:** It contains 37 attributes and all restatement cases on the full sample period (between 2009 and 2014 period) using information gain subset evaluation method. Similar to the dataset CfsSubset, we use undersampling and over sampling techniques to mitigate the class imbalance issue.

**FwSelect:** It contains 15 attributes and all restatement cases on the full sample period (between 2009 and 2014 period) using stepwise forward selection procedure. Similar to the dataset CfsSubset, we use undersampling and over sampling techniques to mitigate the class imbalance issue.

In the process we prepare three different datasets using the feature selection techniques in this study. We summarize the dataset characteristics in Table 2.

Table 2: Different dataset characteristics

| **Dataset** | **No. of Instances** | **No. of restatement Instances** | **No. of non-restatement instances** | **No. of attributes (before attribute reduction)** | **No. of attributes (after attribute reduction)** |
|---|---|---|---|---|---|
| CfsSubset | 20,157 | 1,000 | 19,157 | 116 | 26 |
| InfoGain | 20,157 | 1,000 | 19,157 | 116 | 37 |
| FwSelect | 20,157 | 1,000 | 19,157 | 116 | 15 |

## 6.6 Chapter Summary

In this chapter, we present the data collection and different dataset formation process for this study. We collect relevant financial data from COMPUSTAT database and restatement data from Audit Analytics database. We discuss the detailed steps of data pre-processing. In order to build a reliable predictive model, we form three different datasets. We discuss attribute selection procedure and present the reduced attribute table with detail definition of the attribute set. We also present dataset characteristics values.

In the next chapter, we discuss the model building with various classifiers and present the experimental results.

# CHAPTER 7
# EVALUATION AND RESULTS

In this chapter, we present the experimental results of various models in detecting restatements for the real dataset. We build classification models for the sample period 2009 – 2014. As mentioned in chapter 6, we created three different datasets using various feature selection techniques. We present the results for minority class (financial restatement instances) and majority class (non-restatement instances) separately to evaluate model performance for each class prediction individually. We also focus on aggregated result (i.e. average performance metrics, such as G-mean) to evaluate the model performance in identifying both classes simultaneously. We applied various algorithms (e.g. random undersampling (RUS), Cluster based undersampling (CUS) (Sobhani et al., 2014), random oversampling (ROS), SMOTE, ADASYN (He et al., 2008), and Tomek links with SMOTE) to address class imbalance in the financial restatement dataset.

We perform classification employing six different choices of classifiers, DT, ANN, NB, RF, BBN and SVM using 10-fold cross validation. We also experiment different ensemble methods (bagging and boosting) with these classifiers and employ other meta-learning algorithms (stacking and cost-sensitive learning) to improve model performance.

We summarize the experimental results and report various performance measures including, recall, F-measure, G-mean, precision, and AUC to evaluate model effectiveness. However, as Japkowicz and Shah (2011) discussed, there are particular advantages and disadvantages associated with different performance measures. It appears that there is no

undisputed performance measure to validate the appropriateness of a learning algorithm and some measures are less suitable for highly skewed datasets. For example, Han, Kamber and Pei (2012) suggest that accuracy measure is not suitable for a highly skewed (i.e. high class imbalance) dataset because it can be dominated by majority class identification.

Further, Phua et al. (2005) posit that in financial restatement studies, a false negative error is usually more costly than a false positive error. Therefore, it is quite important for us to evaluate the model performance based on its ability to detect minority class instances. Accordingly, we focus on the minority recall value which emphasizes on how many minority class instances are accurately classified. While minority class recall value is a useful measure, it does not take into account the number of majority class instances incorrectly identified as a minority class instance. In order to overcome this drawback, we also focus on minority class F-measure that combines both minority class precision and recall in the measure.

However, focussing entirely on minority recall value or minority F-measure could be misleading. Some models can show a high success rate in identifying minority instances, yet can perform poorly in detecting majority class instances. Such model could be costly as the investors and regulators need to examine too many firms to explore plausible financial irregularities. Both minority recall value and minority F-measure ignore the correctly identified majority class instances. The number of correctly identified majority class instances can vary without impacting minority recall value or minority F-measure. In this context, we also explore G-mean since it focusses on the aggregate performance of both minority and majority class identification. This measure considers the success rate of both majority (i.e. non-restatement) and minority (i.e. restatement) classes simultaneously.

This chapter is organized in the following way. In the subsection 7.1, we present and discuss the experimental results. The results are organized for undersampling and oversampling techniques separately. However, for brevity some of the less promising results using random undersampling, random oversampling and basic SMOTE are not tabulated in this chapter. Further, for each sampling technique we present the results for three datasets (based on three different feature selection techniques) individually. In subsection 7.2 we summarize the experimental results using DT, ANN, NB, RF, BBN and SVM classifier. We further present and discuss the results of other meta-learning algorithms that are expected improve model performance.

## 7.1 Experimental Results

We evaluate the results using six classifiers. As mentioned in earlier subsection, DT, ANN, NB, BBN, SVM, RF classifier and their ensemble techniques are performed to build predictive models. A more detail background of these classifiers is presented in Chapter 2. In the following subsections we present the performance metrics of each classifier.

### 7.1.1 Cluster based undersampling (CUS)

#### 7.1.1.1 CUS-InfoGainEval

First, we present the experimental results for the dataset created by InfoGain feature selection technique. We performed 10 fold cross validation to avoid overfitting the model. Table 3 shows the summary of the results for restatement class. As discussed above we first

focus on the minority recall values and minority F-measures. The best minority recall is achieved by SVM model and SVM ensembles model. The respective recall values are 96.9% for SVM model, 96.7% for SVM bagging and 92.8% for SVM boosting. BBN and BBN bagging results also came close with 92.2% and 91.1% minority recall respectively. We find that ensemble methods do not lead to a performance improvement in this dataset. Ensemble methods are beneficial if additional information is generated through repeated sampling (in bagging) or multi-stage learning process (in boosting). Our results show that ensemble methods do not lead to any significant additional learning for the classifiers used in analyzing financial restatement datasets.

Table 3: CUS minority class performance measures for InfoGainEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.660 | 0.312 | 0.680 | 0.660 | 0.670 | 0.643 |
| ANN | 0.820 | 0.393 | 0.676 | 0.820 | 0.741 | 0.758 |
| RF | 0.837 | 0.417 | 0.668 | 0.837 | 0.743 | 0.771 |
| NB | 0.559 | 0.266 | 0.677 | 0.559 | 0.612 | 0.753 |
| BBN | 0.922 | 0.450 | 0.672 | **0.922** | 0.778 | 0.763 |
| SVM | 0.969 | 0.506 | 0.657 | **0.969** | 0.783 | 0.731 |
| DT bagging | 0.759 | 0.37 | 0.673 | 0.759 | 0.713 | 0.758 |
| ANN bagging | 0.845 | 0.401 | 0.678 | 0.845 | 0.753 | 0.773 |
| NB bagging | 0.618 | 0.304 | 0.671 | 0.618 | 0.643 | 0.755 |
| BBN bagging | 0.911 | 0.455 | 0.667 | **0.911** | 0.770 | 0.765 |
| SVM bagging | 0.967 | 0.497 | 0.661 | **0.967** | 0.785 | 0.740 |
| DT boosting | 0.700 | 0.335 | 0.676 | 0.700 | 0.688 | 0.751 |
| ANN boosting | 0.840 | 0.408 | 0.674 | 0.840 | 0.748 | 0.762 |
| NB boosting | 0.682 | 0.276 | 0.712 | 0.682 | 0.697 | 0.758 |
| BBN boosting | 0.790 | 0.356 | 0.690 | 0.790 | 0.737 | 0.772 |
| SVM boosting | 0.928 | 0.461 | 0.669 | **0.928** | 0.777 | 0.777 |

Another plausible reason could be the use of sub-optimal ensemble size. In order to explore this possibility further, we conduct additional experiments with varying ensemble sizes of 10, 20, 25, 30 and 50 for best resulting classifiers (i.e BBN and SVM). Table 4 shows

the results of the varying ensemble sizes for BBN and SVM. Our results indicate that BBN bagging with 25 iterations performs slightly better than the ensemble size of 10. However, with higher ensemble size variations, the minority recall performance improvement is insignificant. For BBN, G-mean increases by 2.8% and for SVM, it increases only by 1% as we change ensemble size from 10 to 25. On the other hand, the mean square error (MAE) percentage for the varying ensemble sizes almost remains the same. The difference is only 0.001 percent. Therefore, we decide to use 'ensemble size 10' in this study. Focusing on a smaller ensemble size would be computationally less costly. Our experimental results are in line with Fazelpour et al.'s, (2016) findings. This study also reports that changes in ensemble size not necessarily results into significant performance improvement.

Table 4: Result for varying bagging sizes - CUS InfoGain

| Ensemble size | Classifier | MAE | Precision | Recall | G-mean |
|---|---|---|---|---|---|
| 10 | | 0.281 | 0.667 | 0.911 | 0.712 |
| 20 | | 0.280 | 0.669 | 0.915 | 0.706 |
| 25 | BBN | 0.281 | 0.670 | 0.915 | 0.732 |
| 30 | | 0.281 | 0.668 | 0.912 | 0.705 |
| 50 | | 0.280 | 0.670 | 0.919 | 0.709 |
| 10 | | 0.267 | 0.661 | 0.967 | 0.692 |
| 20 | | 0.266 | 0.662 | 0.970 | 0.699 |
| 25 | SVM | 0.266 | 0.662 | 0.970 | 0.699 |
| 30 | | 0.267 | 0.662 | 0.970 | 0.699 |
| 50 | | 0.266 | 0.661 | 0.969 | 0.697 |

Based on Table 3 result, we find that a SVM classification model is quite successful in identifying restatement class (as denoted by recall value). It is quite encouraging to see that, SVM model can detect approximately 97% of restatement instances.

While minority class recall value is a useful measure, it does not take into account the number of majority class instances incorrectly identified as a minority class instance. We,

therefore, focus on minority class F-measure that combines both minority class precision and recall in the measure. From the previous Table 3, we find that F-measure values are quite low compared to minority recall values. It indicates that even the successful models (i.e. SVM and BBN) perform poorly in identifying majority class instances. In order to get a better perspective on this, we turn our focus on the majority recall and majority class F-measure in Table 5. We find that none of the classifiers perform well in predicting majority class instances. The best majority recall value (73.4%) is achieved by NB and the best majority class F-measure (70.9%) is achieved by NB boosting.

Table 5: CUS majority class performance measures for InfoGainEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.688 | 0.34 | 0.669 | 0.688 | 0.679 | 0.643 |
| ANN | 0.607 | 0.18 | 0.771 | 0.607 | 0.679 | 0.758 |
| RF | 0.583 | 0.163 | 0.782 | 0.583 | 0.668 | 0.771 |
| NB | 0.734 | 0.441 | 0.624 | 0.734 | 0.674 | 0.753 |
| BBN | 0.550 | 0.078 | 0.875 | 0.550 | 0.676 | 0.763 |
| SVM | 0.494 | 0.031 | 0.940 | 0.494 | 0.647 | 0.731 |
| DT bagging | 0.63 | 0.241 | 0.723 | 0.630 | 0.673 | 0.758 |
| ANN bagging | 0.599 | 0.155 | 0.794 | 0.599 | 0.683 | 0.773 |
| NB bagging | 0.696 | 0.382 | 0.645 | 0.696 | 0.670 | 0.755 |
| BBN bagging | 0.545 | 0.089 | 0.859 | 0.545 | 0.667 | 0.765 |
| SVM bagging | 0.503 | 0.033 | 0.939 | 0.503 | 0.655 | 0.74 |
| DT boosting | 0.665 | 0.300 | 0.688 | 0.665 | 0.676 | 0.751 |
| ANN boosting | 0.592 | 0.160 | 0.788 | 0.592 | 0.676 | 0.762 |
| NB boosting | 0.724 | 0.318 | 0.695 | 0.724 | 0.709 | 0.758 |
| BBN boosting | 0.644 | 0.210 | 0.754 | 0.644 | 0.695 | 0.772 |
| SVM boosting | 0.539 | 0.072 | 0.882 | 0.539 | 0.669 | 0.777 |

Both recall value and F-measure focus on single class and hence do not give us a holistic perspective. Therefore, we explore G-mean values at the aggregate level that accounts for performance of both minority and majority class identification. This measure considers the success rate of both majority (i.e. non-restatement) and minority (i.e.

restatement) classes simultaneously. In Table 6, we find that, the highest G-mean value is 71.3% for BBN boosting. We further find that the weighted average recall value and weighted average F-measure are less than 74% for all classifiers.

Table 6: CUS Weighted average performance measures for InfoGainEval

| | TP | FP | Precision | Recall | F-measure | AUC | G-mean |
|---|---|---|---|---|---|---|---|
| DT | 0.674 | 0.326 | 0.674 | 0.674 | 0.674 | 0.643 | 0.674 |
| ANN | 0.713 | 0.287 | 0.723 | 0.713 | 0.710 | 0.758 | 0.706 |
| RF | 0.710 | 0.290 | 0.725 | 0.710 | 0.705 | 0.771 | 0.699 |
| NB | 0.646 | 0.354 | 0.651 | 0.646 | 0.643 | 0.753 | 0.641 |
| BBN | 0.736 | 0.264 | 0.774 | 0.736 | 0.727 | 0.763 | 0.712 |
| SVM | 0.731 | 0.269 | 0.799 | 0.731 | 0.715 | 0.731 | 0.692 |
| DT bagging | 0.695 | 0.306 | 0.698 | 0.695 | 0.693 | 0.758 | 0.691 |
| ANN bagging | 0.722 | 0.278 | 0.736 | 0.722 | 0.718 | 0.773 | 0.711 |
| NB bagging | 0.657 | 0.343 | 0.658 | 0.657 | 0.656 | 0.755 | 0.656 |
| BBN bagging | 0.728 | 0.272 | 0.763 | 0.728 | 0.719 | 0.765 | 0.705 |
| SVM bagging | 0.735 | 0.265 | 0.800 | 0.735 | 0.721 | 0.740 | 0.697 |
| DT boosting | 0.682 | 0.318 | 0.682 | 0.682 | 0.682 | 0.751 | 0.682 |
| ANN boosting | 0.717 | 0.284 | 0.731 | 0.717 | 0.712 | 0.762 | 0.705 |
| NB boosting | 0.703 | 0.297 | 0.704 | 0.703 | 0.703 | 0.758 | 0.703 |
| BBN boosting | 0.717 | 0.283 | 0.722 | 0.717 | 0.716 | 0.772 | 0.713 |
| SVM boosting | 0.734 | 0.267 | 0.775 | 0.734 | 0.723 | 0.777 | 0.707 |

In summary, while we achieve very high level of minority recall value for InfoGain feature selection methods for the financial restatement dataset, the ability of the classifiers to detect majority class instances is dismal. However, our results are similar to the performance measures reported by Kim et al. (2016). This study employed a matching sample based undersampling technique and used logistic regression, SVM and BBN classifiers to develop financial restatement predictive models. The study's highest reported G-mean is 70% which is comparable to our results.

#### 7.1.1.2 CUS-CfsSubsetEval

In this subsection, we present the experimental results for the dataset created by CfsSubset feature selection technique. We performed 10 fold cross validation to avoid overfitting the model. Table 7 shows the summary of the results for restatement class. The best minority recall is achieved by SVM model and SVM ensembles model. The respective recall values are 96.9% for SVM model, 96.6% for SVM bagging and 93% for SVM boosting. RF also performed well in detecting minority cases with 88.8% minority recall value. All F-measures are less than 80% indicating that the classifiers are unable to detect majority class effectively.

In order to get a better perspective on this, we turn our focus on the majority recall and majority class F-measure in Table 8. We find that none of the classifiers perform well in predicting majority class instances. The best majority recall value (73.6%) is achieved by NB and the best majority class F-measure (69.6%) is achieved by DT boosting.

In Table 9, we report the G-mean values. We find that, the highest G-mean value is 71.8% for RF classifier. We further find that the weighted average recall value and weighted average F-measure are less than 74% for all classifiers. These results indicate that none of the classifiers do well in correctly identifying both minority and majority class instances simultaneously. In summary, while we achieve very high level of minority recall value for CfsSubsetEval feature selection method for the financial restatement dataset, the ability of the classifiers to detect majority class instances is not encouraging.

Table 7: CUS minority class performance measures for CfsSubsetEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.794 | 0.362 | 0.687 | 0.794 | 0.737 | 0.724 |
| ANN | 0.778 | 0.387 | 0.668 | 0.778 | 0.719 | 0.749 |
| RF | 0.888 | 0.420 | 0.679 | 0.888 | 0.769 | 0.784 |
| NB | 0.529 | 0.264 | 0.668 | 0.529 | 0.590 | 0.742 |
| BBN | 0.835 | 0.426 | 0.663 | 0.835 | 0.739 | 0.756 |
| SVM | 0.969 | 0.486 | 0.666 | **0.969** | **0.790** | 0.741 |
| DT bagging | 0.797 | 0.358 | 0.691 | 0.797 | 0.740 | 0.776 |
| ANN bagging | 0.818 | 0.38 | 0.683 | 0.818 | 0.744 | 0.769 |
| NB bagging | 0.582 | 0.283 | 0.673 | 0.582 | 0.672 | 0.742 |
| BBN bagging | 0.846 | 0.422 | 0.668 | 0.846 | 0.747 | 0.755 |
| SVM bagging | 0.966 | 0.484 | 0.660 | **0.966** | **0.789** | 0.745 |
| DT boosting | 0.741 | 0.338 | 0.687 | 0.741 | 0.713 | 0.765 |
| ANN boosting | 0.743 | 0.336 | 0.689 | 0.743 | 0.715 | 0.769 |
| NB boosting | 0.748 | 0.363 | 0.673 | 0.748 | 0.709 | 0.719 |
| BBN boosting | 0.744 | 0.372 | 0.667 | 0.744 | 0.703 | 0.762 |
| SVM boosting | 0.930 | 0.460 | 0.669 | **0.930** | **0.779** | 0.758 |

Table 8: CUS majority class performance measures for CfsSubsetEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.638 | 0.206 | 0.755 | 0.638 | 0.692 | 0.724 |
| ANN | 0.613 | 0.222 | 0.734 | 0.613 | 0.668 | 0.749 |
| RF | 0.580 | 0.112 | 0.837 | 0.580 | 0.685 | 0.784 |
| NB | 0.736 | 0.471 | 0.610 | 0.736 | 0.667 | 0.741 |
| BBN | 0.574 | 0.165 | 0.776 | 0.574 | 0.660 | 0.756 |
| SVM | 0.514 | 0.031 | 0.943 | 0.514 | 0.665 | 0.741 |
| DT bagging | 0.642 | 0.203 | 0.759 | 0.642 | 0.696 | 0.776 |
| ANN bagging | 0.620 | 0.182 | 0.773 | 0.620 | 0.688 | 0.769 |
| NB bagging | 0.717 | 0.418 | 0.631 | 0.717 | 0.671 | 0.742 |
| BBN bagging | 0.578 | 0.154 | 0.790 | 0.578 | 0.668 | 0.755 |
| SVM bagging | 0.516 | 0.034 | 0.938 | 0.516 | 0.665 | 0.745 |
| DT boosting | 0.662 | 0.259 | 0.719 | 0.662 | 0.689 | 0.765 |
| ANN boosting | 0.664 | 0.257 | 0.720 | 0.664 | 0.691 | 0.769 |
| NB boosting | 0.637 | 0.252 | 0.716 | 0.637 | 0.674 | 0.719 |
| BBN boosting | 0.628 | 0.256 | 0.710 | 0.628 | 0.667 | 0.762 |
| SVM boosting | 0.540 | 0.070 | 0.886 | 0.540 | 0.671 | 0.758 |

Table 9: CUS Weighted average performance measures for CfsSubsetEval

| | TP | FP | Precision | Recall | F-measure | AUC | G-mean |
|---|---|---|---|---|---|---|---|
| DT | 0.716 | 0.284 | 0.721 | 0.716 | 0.714 | 0.724 | 0.712 |
| ANN | 0.695 | 0.305 | 0.701 | 0.695 | 0.693 | 0.749 | 0.691 |
| RF | 0.734 | 0.266 | 0.758 | 0.734 | 0.727 | 0.784 | **0.718** |
| NB | 0.633 | 0.367 | 0.639 | 0.633 | 0.629 | 0.741 | 0.624 |
| BBN | 0.705 | 0.296 | 0.719 | 0.705 | 0.700 | 0.756 | 0.692 |
| SVM | 0.742 | 0.259 | 0.804 | 0.742 | 0.728 | 0.741 | **0.706** |
| DT bagging | 0.72 | 0.281 | 0.725 | 0.72 | 0.718 | 0.776 | 0.715 |
| ANN bagging | 0.719 | 0.281 | 0.728 | 0.719 | 0.716 | 0.769 | 0.712 |
| NB bagging | 0.649 | 0.351 | 0.652 | 0.649 | 0.648 | 0.742 | 0.646 |
| BBN bagging | 0.712 | 0.288 | 0.729 | 0.712 | 0.707 | 0.755 | 0.699 |
| SVM bagging | 0.741 | 0.259 | 0.802 | 0.741 | 0.727 | 0.745 | 0.706 |
| DT boosting | 0.702 | 0.298 | 0.703 | 0.702 | 0.701 | 0.765 | 0.700 |
| ANN boosting | 0.703 | 0.297 | 0.704 | 0.703 | 0.703 | 0.769 | 0.702 |
| NB boosting | 0.693 | 0.308 | 0.695 | 0.693 | 0.692 | 0.719 | 0.690 |
| BBN boosting | 0.686 | 0.314 | 0.689 | 0.686 | 0.685 | 0.762 | 0.684 |
| SVM boosting | 0.735 | 0.265 | 0.777 | 0.735 | 0.725 | 0.758 | 0.709 |

#### 7.1.1.3 CUS-FwSelecct

In this subsection, we present the experimental results for the dataset created by FwSelecct feature selection technique. We performed 10 fold cross validation to avoid overfitting the model. Table 10 shows the summary of the results for restatement class. Similar to the above mentioned feature selection techniques, the best minority recall is achieved by SVM model and SVM ensembles model. The respective recall values are 97.8% for SVM model, SVM bagging and 94.7% for SVM boosting. Other classifier and their ensembles results shows the minority recall is over 81%. All F-measures are less than 80% - indicating that the classifiers are unable to detect majority class effectively.

In order to get a better perspective on this, we turn our focus on the majority recall and majority class F-measure in Table 11. We find that none of the classifiers perform well

in predicting majority class instances. The best majority recall value (62%) and the best majority class F-measure (70%) are achieved by NB classifier.

In Table 12, we report the G-mean values. We find that, the highest G-mean value is 72% for NB classifier. We further find that the weighted average recall value and weighted average F-measure are less than 75% for all classifiers. These results indicate that none of the classifiers do well in correctly identifying both minority and majority class instances simultaneously. In summary, while we achieve very high level of minority recall value for FwSelect feature selection method for the financial restatement dataset, the ability of the classifiers to detect majority class instances is not encouraging.

Table 10: CUS minority class performance measures for FwSelect

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.867 | 0.424 | 0.672 | 0.867 | 0.757 | 0.739 |
| ANN | 0.884 | 0.423 | 0.677 | 0.884 | 0.767 | 0.767 |
| RF | 0.865 | 0.405 | 0.681 | 0.865 | 0.762 | 0.782 |
| NB | 0.849 | 0.380 | 0.691 | 0.849 | 0.762 | 0.761 |
| BBN | 0.822 | 0.410 | 0.667 | 0.822 | 0.737 | 0.746 |
| SVM | 0.978 | 0.490 | 0.667 | **0.978** | 0.793 | 0.744 |
| DT bagging | 0.810 | 0.379 | 0.682 | 0.810 | 0.740 | 0.765 |
| ANN bagging | 0.863 | 0.406 | 0.680 | 0.863 | 0.761 | 0.773 |
| NB bagging | 0.853 | 0.389 | 0.687 | 0.853 | 0.761 | 0.764 |
| BBN bagging | 0.826 | 0.410 | 0.668 | 0.826 | 0.739 | 0.747 |
| SVM bagging | 0.978 | 0.488 | 0.667 | **0.978** | 0.793 | 0.751 |
| DT boosting | 0.792 | 0.371 | 0.681 | 0.732 | 0.427 | 0.761 |
| ANN boosting | 0.810 | 0.384 | 0.679 | 0.810 | 0.739 | 0.755 |
| NB boosting | 0.852 | 0.389 | 0.687 | 0.852 | 0.76 | 0.778 |
| BBN boosting | 0.826 | 0.404 | 0.672 | 0.826 | 0.741 | 0.708 |
| SVM boosting | 0.947 | 0.456 | 0.675 | **0.947** | 0.788 | 0.785 |

Table 11: CUS majority class performance measures dataset FwSelect

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.576 | 0.133 | 0.813 | 0.576 | 0.674 | 0.739 |
| ANN | 0.577 | 0.116 | 0.833 | 0.577 | 0.682 | 0.767 |
| RF | 0.595 | 0.135 | 0.815 | 0.595 | 0.688 | 0.782 |
| NB | 0.620 | 0.151 | 0.804 | 0.620 | 0.700 | 0.761 |
| BBN | 0.590 | 0.178 | 0.768 | 0.590 | 0.667 | 0.746 |
| SVM | 0.510 | 0.022 | 0.958 | 0.510 | 0.666 | 0.744 |
| DT bagging | 0.621 | 0.019 | 0.766 | 0.621 | 0.686 | 0.765 |
| ANN bagging | 0.594 | 0.137 | 0.813 | 0.594 | 0.686 | 0.773 |
| NB bagging | 0.611 | 0.147 | 0.806 | 0.611 | 0.695 | 0.764 |
| BBN bagging | 0.590 | 0.174 | 0.772 | 0.590 | 0.669 | 0.747 |
| SVM bagging | 0.512 | 0.022 | 0.959 | 0.512 | 0.667 | 0.751 |
| DT boosting | 0.629 | 0.208 | 0.751 | 0.685 | 0.427 | 0.761 |
| ANN boosting | 0.616 | 0.190 | 0.764 | 0.616 | 0.682 | 0.755 |
| NB boosting | 0.611 | 0.148 | 0.804 | 0.611 | 0.694 | 0.778 |
| BBN boosting | 0.596 | 0.174 | 0.774 | 0.596 | 0.674 | 0.766 |
| SVM boosting | 0.544 | 0.053 | 0.911 | 0.544 | 0.681 | 0.785 |

Table 12: CUS Weighted average performance measures for FwSelect

| | TP | FP | Precision | Recall | F-measure | AUC | G-mean |
|---|---|---|---|---|---|---|---|
| DT | 0.722 | 0.279 | 0.742 | 0.722 | 0.715 | 0.739 | 0.707 |
| ANN | 0.731 | 0.270 | 0.755 | 0.731 | 0.724 | 0.767 | 0.714 |
| RF | 0.730 | 0.270 | 0.748 | 0.730 | 0.725 | 0.782 | 0.717 |
| NB | 0.735 | 0.266 | 0.747 | 0.735 | 0.731 | 0.761 | 0.726 |
| BBN | 0.706 | 0.294 | 0.718 | 0.706 | 0.702 | 0.746 | 0.696 |
| SVM | 0.744 | 0.256 | 0.812 | 0.744 | 0.730 | 0.744 | 0.706 |
| DT bagging | 0.716 | 0.284 | 0.724 | 0.716 | 0.713 | 0.765 | 0.709 |
| ANN bagging | 0.729 | 0.271 | 0.747 | 0.729 | 0.724 | 0.773 | 0.716 |
| NB bagging | 0.732 | 0.268 | 0.746 | 0.732 | 0.728 | 0.764 | 0.722 |
| BBN bagging | 0.708 | 0.292 | 0.720 | 0.708 | 0.704 | 0.747 | 0.698 |
| SVM bagging | 0.745 | 0.255 | 0.813 | 0.745 | 0.730 | 0.751 | 0.708 |
| DT boosting | 0.710 | 0.290 | 0.716 | 0.71 | 0.427 | 0.761 | 0.679 |
| ANN boosting | 0.713 | 0.287 | 0.721 | 0.713 | 0.710 | 0.755 | 0.706 |
| NB boosting | 0.731 | 0.269 | 0.745 | 0.731 | 0.727 | 0.778 | 0.722 |
| BBN boosting | 0.711 | 0.289 | 0.723 | 0.711 | 0.707 | 0.766 | 0.702 |
| SVM boosting | 0.746 | 0.255 | 0.793 | 0.746 | 0.735 | 0.785 | 0.718 |

### 7.1.2 Tomek links and SMOTE

#### 7.1.2.1 Tomek-SMOTE CfsSubsetEval

In this subsection, we present the experimental results for Tomek link + SMOTE for the dataset created by CfsSubset feature selection technique. Table 13 shows the summary of the results for the restatement class. Using Tomek link and SMOTE, the best minority recall is achieved by NB model and NB ensembles model. The minority recall values are 92.4%, 91.7% and 92.4% for NB model, NB bagging and NB boosting respectively. RF classifier yields the highest F-measure (92.4%) that incorporate both precision and recall values for the minority class instances. RF classifier also shows a reasonably high success rate in terms of minority recall (89%).

In order to get an idea on various classifiers' success rate in detecting majority class instances, next we turn our focus on the majority class recall value and majority class F-measure. We find that some of the models (RF, SVM) are highly successful in detecting majority class instances.

Both recall value and F-measure focus on single class and hence do not give us a holistic perspective. Therefore, we explore G-mean values at the aggregate level that accounts for performance of both minority and majority class identification. This measure considers the success rate of both majority (i.e. non-restatement) and minority (i.e. restatement) classes simultaneously. In Table 15, we find that, the highest G-mean value is 93.4% for RF classifier. We further find that the weighted average recall value and weighted average F-measure are 94.9% and 94.9% respectively for the RF classifier.

In summary, we achieve very high level of minority recall value for CfsSubsetEval feature selection methods for the financial restatement dataset. At the same time, the ability

of the classifiers to detect majority class instances is also very encouraging. The RF classifier shows a very high degree of predictive capability for both majority and minority class instances. These results show a clear improvement over the earlier large scale study on financial restatements by Kim et al. (2016). The highest reported G-mean value by Kim et al. (2016) is 70% compared to our value of 94.9% for RF classifier.

Table 13: Tomek Smote minority class performance measures for CfsSubsetEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.789 | 0.150 | 0.738 | 0.789 | 0.763 | 0.835 |
| ANN | 0.572 | 0.203 | 0.601 | 0.572 | 0.586 | 0.768 |
| RF | 0.890 | 0.019 | 0.962 | 0.890 | 0.924 | 0.986 |
| NB | 0.924 | 0.840 | 0.371 | 0.924 | 0.529 | 0.628 |
| BBN | 0.818 | 0.170 | 0.720 | 0.818 | 0.766 | 0.905 |
| SVM | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.500 |
| DT bagging | 0.833 | 0.059 | 0.882 | 0.833 | 0.857 | 0.959 |
| ANN bagging | 0.582 | 0.144 | 0.683 | 0.582 | 0.629 | 0.826 |
| NB bagging | 0.917 | 0.827 | 0.372 | 0.917 | 0.530 | 0.621 |
| BBN bagging | 0.832 | 0.146 | 0.753 | 0.832 | 0.790 | 0.917 |
| SVM bagging | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.500 |
| DT boosting | 0.893 | 0.056 | 0.896 | 0.893 | 0.894 | 0.975 |
| ANN boosting | 0.592 | 0.184 | 0.633 | 0.592 | 0.612 | 0.796 |
| NB boosting | 0.924 | 0.840 | 0.371 | 0.924 | 0.529 | 0.628 |
| BBN boosting | 0.791 | 0.126 | 0.770 | 0.791 | 0.780 | 0.897 |
| SVM boosting | 0.039 | 0.018 | 0.533 | 0.039 | 0.073 | 0.616 |

Table 14: Tomek Smote majority class performance measures for CfsSubsetEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.850 | 0.211 | 0.883 | 0.850 | 0.866 | 0.835 |
| ANN | 0.797 | 0.428 | 0.777 | 0.797 | 0.787 | 0.768 |
| RF | 0.981 | 0.110 | 0.943 | 0.981 | 0.962 | 0.986 |
| NB | 0.160 | 0.076 | 0.798 | 0.160 | 0.267 | 0.627 |
| BBN | 0.830 | 0.182 | 0.895 | 0.830 | 0.861 | 0.905 |
| SVM | 1.000 | 1.000 | 0.652 | 1.000 | 0.789 | 0.500 |
| DT bagging | 0.941 | 0.167 | 0.913 | 0.941 | 0.927 | 0.959 |
| ANN bagging | 0.856 | 0.418 | 0.793 | 0.856 | 0.823 | 0.826 |
| NB bagging | 0.173 | 0.083 | 0.796 | 0.173 | 0.285 | 0.621 |
| BBN bagging | 0.854 | 0.168 | 0.905 | 0.854 | 0.879 | 0.917 |
| SVM bagging | 1.000 | 1.000 | 0.652 | 1.000 | 0.789 | 0.500 |
| DT boosting | 0.944 | 0.107 | 0.943 | 0.944 | 0.944 | 0.975 |
| ANN boosting | 0.816 | 0.408 | 0.789 | 0.816 | 0.802 | 0.796 |
| NB boosting | 0.160 | 0.076 | 0.798 | 0.160 | 0.267 | 0.627 |
| BBN boosting | 0.874 | 0.209 | 0.887 | 0.874 | 0.880 | 0.897 |
| SVM boosting | 0.982 | 0.961 | 0.656 | 0.982 | 0.787 | 0.616 |

Table 15: Tomek Smote weighted average performance measures for CfsSubsetEval

| | TP | FP | Precision | Recall | F-measure | AUC | G-mean |
|---|---|---|---|---|---|---|---|
| DT | 0.829 | 0.190 | 0.832 | 0.829 | 0.830 | 0.835 | 0.819 |
| ANN | 0.718 | 0.350 | 0.716 | 0.718 | 0.717 | 0.768 | 0.675 |
| RF | 0.949 | 0.078 | 0.950 | 0.949 | 0.949 | 0.986 | **0.934** |
| NB | 0.426 | 0.342 | 0.649 | 0.426 | 0.358 | 0.627 | 0.384 |
| BBN | 0.826 | 0.178 | 0.834 | 0.826 | 0.828 | 0.905 | 0.824 |
| SVM | 0.652 | 0.652 | 0.424 | 0.652 | 0.514 | 0.500 | 0.000 |
| DT bagging | 0.903 | 0.129 | 0.903 | 0.903 | 0.902 | 0.959 | 0.885 |
| ANN bagging | 0.760 | 0.323 | 0.755 | 0.760 | 0.755 | 0.826 | 0.706 |
| NB bagging | 0.432 | 0.342 | 0.648 | 0.432 | 0.370 | 0.621 | 0.398 |
| BBN bagging | 0.846 | 0.161 | 0.852 | 0.846 | 0.848 | 0.917 | 0.843 |
| SVM bagging | 0.652 | 0.652 | 0.424 | 0.652 | 0.514 | 0.500 | 0.000 |
| DT boosting | 0.926 | 0.089 | 0.926 | 0.926 | 0.926 | 0.975 | 0.918 |
| ANN boosting | 0.738 | 0.330 | 0.735 | 0.738 | 0.736 | 0.796 | 0.695 |
| NB boosting | 0.426 | 0.342 | 0.649 | 0.426 | 0.358 | 0.627 | 0.384 |
| BBN boosting | 0.845 | 0.180 | 0.846 | 0.845 | 0.845 | 0.897 | 0.831 |
| SVM boosting | 0.653 | 0.632 | 0.613 | 0.653 | 0.538 | 0.616 | 0.196 |

#### 7.1.2.2 Tomek SMOTE InfoGainEval

In this subsection, we present the experimental results for Tomek link + SMOTE for the dataset created by InfoGain feature selection technique. Table 16 shows the summary of the results for the restatement class. Using Tomek link and SMOTE, the best minority recall is achieved by RF model. The minority recall values are 92.6%. RF classifier yields the highest F-measure (94.1%) that incorporate both precision and recall values for the minority class instances. NB classifier also shows a reasonably high success rate in terms of minority recall (92.4%).

In order to get an idea on various classifiers' success rate in detecting majority class instances, next we turn our focus on the majority class recall value and majority class F-measure. We find that some of the models (RF, SVM) are highly successful in detecting majority class instances.

Both recall value and F-measure focus on single class and hence do not give us a holistic perspective. Therefore, we explore G-mean values at the aggregate level that accounts for performance of both minority and majority class identification. This measure considers the success rate of both majority (i.e. non-restatement) and minority (i.e. restatement) classes simultaneously. In Table 18, we find that, the highest G-mean value is 95.2% for RF classifier. We further find that the weighted average recall value and weighted average F-measure are 96.1% and 96% respectively for the RF classifier.

In summary, we achieve very high level of minority recall value for InfoGainSubsetEval feature selection methods for the financial restatement dataset. At the same time, the ability of the classifiers to detect majority class instances is also very

encouraging. The RF classifier shows a very high degree of predictive capability for both majority and minority class instances.

Table 16: Tomek Smote minority class performance measures for InfoGainEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.812 | 0.107 | 0.795 | 0.812 | 0.803 | 0.873 |
| ANN | 0.469 | 0.147 | 0.621 | 0.469 | 0.534 | 0.751 |
| RF | 0.926 | 0.022 | 0.956 | 0.926 | 0.941 | 0.992 |
| NB | 0.924 | 0.801 | 0.372 | 0.924 | 0.530 | 0.666 |
| BBN | 0.734 | 0.317 | 0.543 | 0.734 | 0.624 | 0.781 |
| SVM | 0.023 | 0.009 | 0.563 | 0.023 | 0.043 | 0.507 |
| DT bagging | 0.862 | 0.057 | 0.887 | 0.862 | 0.874 | 0.964 |
| ANN bagging | 0.533 | 0.086 | 0.760 | 0.533 | 0.627 | 0.848 |
| NB bagging | 0.917 | 0.782 | 0.375 | 0.917 | 0.533 | 0.664 |
| BBN bagging | 0.751 | 0.303 | 0.559 | 0.751 | 0.641 | 0.799 |
| SVM bagging | 0.016 | 0.008 | 0.511 | 0.016 | 0.031 | 0.512 |
| DT boosting | 0.919 | 0.045 | 0.913 | 0.919 | 0.916 | 0.984 |
| ANN boosting | 0.480 | 0.126 | 0.662 | 0.480 | 0.557 | 0.781 |
| NB boosting | 0.924 | 0.801 | 0.372 | 0.924 | 0.530 | 0.666 |
| BBN boosting | 0.587 | 0.122 | 0.711 | 0.587 | 0.643 | 0.842 |
| SVM boosting | 0.220 | 0.111 | 0.503 | 0.220 | 0.306 | 0.660 |

Table 17: Tomek Smote majority class performance measures for InfoGainEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.893 | 0.188 | 0.903 | 0.893 | 0.898 | 0.873 |
| ANN | 0.853 | 0.531 | 0.758 | 0.853 | 0.803 | 0.751 |
| RF | 0.978 | 0.074 | 0.963 | 0.978 | 0.970 | 0.992 |
| NB | 0.199 | 0.076 | 0.837 | 0.199 | 0.322 | 0.666 |
| BBN | 0.683 | 0.266 | 0.834 | 0.683 | 0.751 | 0.781 |
| SVM | 0.991 | 0.977 | 0.664 | 0.991 | 0.795 | 0.507 |
| DT bagging | 0.943 | 0.138 | 0.930 | 0.943 | 0.937 | 0.964 |
| ANN bagging | 0.914 | 0.467 | 0.792 | 0.914 | 0.849 | 0.848 |
| NB bagging | 0.218 | 0.083 | 0.836 | 0.218 | 0.346 | 0.664 |
| BBN bagging | 0.697 | 0.249 | 0.845 | 0.697 | 0.764 | 0.799 |
| SVM bagging | 0.992 | 0.984 | 0.663 | 0.992 | 0.795 | 0.512 |
| DT boosting | 0.955 | 0.081 | 0.958 | 0.955 | 0.957 | 0.984 |
| ANN boosting | 0.874 | 0.520 | 0.766 | 0.874 | 0.817 | 0.781 |
| NB boosting | 0.199 | 0.076 | 0.837 | 0.199 | 0.322 | 0.666 |
| BBN boosting | 0.878 | 0.413 | 0.806 | 0.878 | 0.840 | 0.842 |
| SVM boosting | 0.889 | 0.780 | 0.690 | 0.889 | 0.777 | 0.660 |

Table 18: Tomek Smote weighted average performance measures for InfoGainEval

| | TP | FP | Precision | Recall | F-measure | AUC | G-mean |
|---|---|---|---|---|---|---|---|
| DT | 0.865 | 0.161 | 0.866 | 0.865 | 0.866 | 0.873 | 0.852 |
| ANN | 0.723 | 0.401 | 0.712 | 0.723 | 0.712 | 0.751 | 0.633 |
| RF | 0.961 | 0.056 | 0.960 | 0.961 | 0.960 | 0.992 | 0.952 |
| NB | 0.445 | 0.321 | 0.679 | 0.445 | 0.393 | 0.666 | 0.429 |
| BBN | 0.700 | 0.283 | 0.735 | 0.700 | 0.708 | 0.781 | 0.708 |
| SVM | 0.663 | 0.649 | 0.630 | 0.663 | 0.540 | 0.507 | 0.151 |
| DT bagging | 0.916 | 0.110 | 0.916 | 0.916 | 0.916 | 0.964 | 0.902 |
| ANN bagging | 0.785 | 0.338 | 0.782 | 0.785 | 0.774 | 0.848 | 0.698 |
| NB bagging | 0.455 | 0.320 | 0.680 | 0.455 | 0.409 | 0.664 | 0.447 |
| BBN bagging | 0.715 | 0.268 | 0.748 | 0.715 | 0.722 | 0.799 | 0.723 |
| SVM bagging | 0.661 | 0.653 | 0.611 | 0.661 | 0.536 | 0.512 | 0.126 |
| DT boosting | 0.943 | 0.068 | 0.943 | 0.943 | 0.943 | 0.984 | 0.937 |
| ANN boosting | 0.741 | 0.386 | 0.731 | 0.741 | 0.729 | 0.781 | 0.648 |
| NB boosting | 0.445 | 0.321 | 0.679 | 0.445 | 0.393 | 0.666 | 0.429 |
| BBN boosting | 0.779 | 0.314 | 0.774 | 0.779 | 0.773 | 0.842 | 0.718 |
| SVM boosting | 0.662 | 0.554 | 0.626 | 0.662 | 0.617 | 0.660 | 0.442 |

#### 7.1.2.3 Tomek SMOTE FwSelecct

In this subsection, we present the experimental results for Tomek link + SMOTE for the dataset created by forward feature selection technique. Table 19 shows the summary of the results for the restatement class. Using Tomek link and SMOTE, the best minority recall is achieved by RF and DT boosting model. The minority recall values are 88.1 and 88.6% respectively. RF classifier yields the highest F-measure (94.1%) that incorporate both precision and recall values for the minority class instances. DT bagging ensemble classifier also shows a reasonably high success rate in terms of minority recall (85.7%).

In order to get an idea on various classifiers' success rate in detecting majority class instances, next we turn our focus on the majority class recall value and majority class F-measure. We find that some of the models (RF, SVM) are highly successful in detecting majority class instances.

Both recall value and F-measure focus on single class and hence do not give us a holistic perspective. Therefore, we explore G-mean values at the aggregate level that accounts for performance of both minority and majority class identification. This measure considers the success rate of both majority (i.e. non-restatement) and minority (i.e. restatement) classes simultaneously. In Table 21, we find that, the highest G-mean value is 92.6% for RF classifier. We further find that the weighted average recall value and weighted average F-measure are 94.1% and 94.1% respectively for the RF classifier.

In summary, we achieve very high level of minority recall value for forward feature selection methods for the financial restatement dataset. At the same time, the ability of the classifiers to detect majority class instances is also very encouraging. The RF classifier shows a very high degree of predictive capability for both majority and minority class instances.

Table 19: Tomek Smote minority class performance measures for FwSelecct

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.791 | 0.151 | 0.744 | 0.791 | 0.767 | 0.846 |
| ANN | 0.455 | 0.206 | 0.550 | 0.455 | 0.498 | 0.700 |
| RF | 0.881 | 0.026 | 0.950 | **0.881** | 0.914 | 0.984 |
| NB | 0.882 | 0.729 | 0.401 | **0.882** | 0.552 | 0.638 |
| BBN | 0.521 | 0.229 | 0.557 | 0.521 | 0.538 | 0.730 |
| SVM | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.500 |
| DT bagging | 0.845 | 0.070 | 0.870 | **0.857** | 0.857 | 0.956 |
| ANN bagging | 0.371 | 0.124 | 0.623 | 0.371 | 0.465 | 0.739 |
| NB bagging | 0.882 | 0.727 | 0.402 | **0.882** | 0.552 | 0.638 |
| BBN bagging | 0.529 | 0.224 | 0.567 | 0.529 | 0.547 | 0.736 |
| SVM bagging | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.500 |
| DT boosting | 0.886 | 0.067 | 0.879 | **0.886** | 0.883 | 0.969 |
| ANN boosting | 0.418 | 0.162 | 0.588 | 0.418 | 0.488 | 0.720 |
| NB boosting | 0.882 | 0.729 | 0.401 | **0.882** | 0.552 | 0.638 |
| BBN boosting | 0.489 | 0.199 | 0.577 | 0.489 | 0.530 | 0.719 |
| SVM boosting | 0.103 | 0.057 | 0.502 | 0.103 | 0.171 | 0.627 |

Table 20: Tomek Smote majority class performance measures for dataet FwSelecct

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.849 | 0.209 | 0.880 | 0.849 | 0.864 | 0.846 |
| ANN | 0.794 | 0.545 | 0.724 | 0.794 | 0.758 | 0.700 |
| RF | 0.974 | 0.119 | 0.937 | 0.974 | 0.955 | 0.984 |
| NB | 0.271 | 0.118 | 0.805 | 0.271 | 0.406 | 0.638 |
| BBN | 0.771 | 0.479 | 0.744 | 0.771 | 0.757 | 0.730 |
| SVM | 1.000 | 1.000 | 0.643 | 1.000 | 0.783 | 0.500 |
| DT bagging | 0.930 | 0.155 | 0.915 | 0.930 | 0.923 | 0.956 |
| ANN bagging | 0.876 | 0.629 | 0.715 | 0.876 | 0.787 | 0.739 |
| NB bagging | 0.273 | 0.118 | 0.806 | 0.273 | 0.407 | 0.638 |
| BBN bagging | 0.776 | 0.471 | 0.748 | 0.776 | 0.762 | 0.736 |
| SVM bagging | 1.000 | 1.000 | 0.643 | 1.000 | 0.783 | 0.500 |
| DT boosting | 0.933 | 0.114 | 0.937 | 0.933 | 0.935 | 0.969 |
| ANN boosting | 0.838 | 0.582 | 0.722 | 0.838 | 0.776 | 0.72 |
| NB boosting | 0.271 | 0.118 | 0.805 | 0.271 | 0.406 | 0.638 |
| BBN boosting | 0.801 | 0.511 | 0.739 | 0.801 | 0.769 | 0.719 |
| SVM boosting | 0.943 | 0.897 | 0.655 | 0.943 | 0.773 | 0.627 |

Table 21: Tomek Smote weighted average performance measures for FwSelect

| | TP | FP | Precision | Recall | F-measure | AUC | G-mean |
|---|---|---|---|---|---|---|---|
| DT | 0.828 | 0.188 | 0.831 | 0.828 | 0.829 | 0.846 | 0.819 |
| ANN | 0.673 | 0.424 | 0.662 | 0.673 | 0.665 | 0.700 | 0.601 |
| RF | 0.941 | 0.085 | 0.942 | 0.941 | 0.941 | 0.984 | **0.926** |
| NB | 0.489 | 0.336 | 0.661 | 0.489 | 0.458 | 0.638 | 0.489 |
| BBN | 0.682 | 0.39 | 0.677 | 0.682 | 0.679 | 0.730 | 0.634 |
| SVM | 0.643 | 0.643 | 0.414 | 0.643 | 0.504 | 0.500 | 0.000 |
| DT bagging | 0.9 | 0.125 | 0.899 | 0.900 | 0.899 | 0.956 | 0.893 |
| ANN bagging | 0.696 | 0.449 | 0.683 | 0.696 | 0.673 | 0.739 | 0.570 |
| NB bagging | 0.49 | 0.335 | 0.662 | 0.490 | 0.459 | 0.638 | 0.491 |
| BBN bagging | 0.688 | 0.383 | 0.684 | 0.688 | 0.685 | 0.736 | 0.641 |
| SVM bagging | 0.643 | 0.643 | 0.414 | 0.643 | 0.504 | 0.500 | 0.000 |
| DT boosting | 0.916 | 0.097 | 0.916 | 0.916 | 0.916 | 0.969 | 0.909 |
| ANN boosting | 0.688 | 0.433 | 0.674 | 0.688 | 0.673 | 0.720 | 0.592 |
| NB boosting | 0.489 | 0.336 | 0.661 | 0.489 | 0.458 | 0.638 | 0.489 |
| BBN boosting | 0.69 | 0.399 | 0.681 | 0.690 | 0.684 | 0.719 | 0.626 |
| SVM boosting | 0.644 | 0.597 | 0.600 | 0.644 | 0.558 | 0.627 | 0.312 |

### 7.1.3 ADASYN

#### 7.1.3.1 ADASYN CfsSubsetEval

In this subsection, we present the experimental results for ADASYN for the dataset created by CfsSubset feature selection technique. Table 22 shows the summary of the results for the restatement class. Using ADASYN, the best minority recall is achieved by BBN boosting ensemble model and RF model. The minority recall values are 88% and 87.7% respectively. RF classifier yields the highest F-measure (93.4%) that incorporate both precision and recall values for the minority class instances. BBN bagging ensemble classifier also shows a reasonably high success rate in terms of minority recall (93.1%).

In order to get an idea on various classifiers' success rate in detecting majority class instances, next we turn our focus on the majority class recall value and majority class F-measure. We find that some of the models (RF, SVM) are highly successful in detecting majority class instances.

Both recall value and F-measure focus on single class and hence do not give us a holistic perspective. Therefore, we explore G-mean values at the aggregate level that accounts for performance of both minority and majority class identification. This measure considers the success rate of both majority (i.e. non-restatement) and minority (i.e. restatement) classes simultaneously. In Table 24, we find that, the highest G-mean value is 93.6% for RF classifier. We further find that the weighted average recall value and weighted average F-measure are 95.8% and 95.7% respectively for the RF classifier.

Table 22: ADASYN minority class performance measures for CfsSubsetEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.830 | 0.001 | 0.998 | 0.830 | 0.907 | 0.920 |
| ANN | 0.453 | 0.08 | 0.744 | 0.453 | 0.563 | 0.756 |
| RF | 0.877 | 0.000 | 1.000 | 0.877 | 0.934 | 0.961 |
| NB | 0.617 | 0.000 | 0.999 | 0.617 | 0.763 | 0.891 |
| BBN | 0.876 | 0.002 | 0.995 | 0.876 | 0.932 | 0.951 |
| SVM | 0.006 | 0.000 | 1.000 | 0.006 | 0.012 | 0.503 |
| DT bagging | 0.845 | 0.001 | 0.998 | 0.845 | 0.915 | 0.953 |
| ANN bagging | 0.460 | 0.002 | 0.990 | 0.460 | 0.628 | 0.836 |
| NB bagging | 0.596 | 0.000 | 0.999 | 0.596 | 0.747 | 0.911 |
| BBN bagging | 0.875 | 0.002 | 0.995 | 0.875 | 0.931 | 0.951 |
| SVM bagging | 0.006 | 0.000 | 1.000 | 0.006 | 0.012 | 0.505 |
| DT boosting | 0.876 | 0.010 | 0.978 | 0.876 | 0.924 | 0.955 |
| ANN boosting | 0.430 | 0.017 | 0.928 | 0.430 | 0.588 | 0.757 |
| NB boosting | 0.617 | 0.000 | 0.999 | 0.617 | 0.763 | 0.807 |
| BBN boosting | 0.880 | 0.008 | 0.983 | 0.880 | 0.929 | 0.949 |
| SVM boosting | 0.009 | 0.000 | 1.000 | 0.009 | 0.018 | 0.561 |

Table 23: ADASYN majority class Performance measures for CfsSubsetEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.999 | 0.170 | 0.920 | 0.999 | 0.958 | 0.920 |
| ANN | 0.920 | 0.547 | 0.764 | 0.920 | 0.835 | 0.756 |
| RF | 1.000 | 0.123 | 0.940 | 1.000 | 0.969 | 0.961 |
| NB | 1.000 | 0.383 | 0.835 | 1.000 | 0.910 | 0.815 |
| BBN | 0.998 | 0.124 | 0.940 | 0.998 | 0.968 | 0.951 |
| SVM | 1.000 | 0.994 | 0.661 | 1.000 | 0.796 | 0.503 |
| DT bagging | 0.999 | 0.155 | 0.926 | 0.999 | 0.961 | 0.953 |
| ANN bagging | 0.998 | 0.540 | 0.782 | 0.998 | 0.877 | 0.836 |
| NB bagging | 1.000 | 0.404 | 0.828 | 1.000 | 0.906 | 0.828 |
| BBN bagging | 0.998 | 0.125 | 0.939 | 0.998 | 0.968 | 0.951 |
| SVM bagging | 1.000 | 0.994 | 0.661 | 1.000 | 0.796 | 0.505 |
| DT boosting | 0.990 | 0.124 | 0.939 | 0.990 | 0.964 | 0.955 |
| ANN boosting | 0.983 | 0.570 | 0.770 | 0.983 | 0.863 | 0.757 |
| NB boosting | 1.000 | 0.383 | 0.835 | 1.000 | 0.910 | 0.807 |
| BBN boosting | 0.992 | 0.120 | 0.941 | 0.992 | 0.966 | 0.949 |
| SVM boosting | 1.000 | 0.991 | 0.662 | 1.000 | 0.797 | 0.561 |

In summary, we achieve very high level of minority recall value for CfsSubsetEval feature selection methods for the financial restatement dataset. At the same time, the ability

of the classifiers to detect majority class instances is also very encouraging. The RF classifier shows a very high degree of predictive capability for both majority and minority class instances. These results show a clear improvement over the earlier large scale study on financial restatements by Kim et al. (2016). The highest reported G-mean value by Kim et al. (2016) is 70% compared to our value of 94.9% for RF classifier.

Table 24: ADASYN weighted average performance measures for CfsSubsetEval

| | TP | FP | Precision | Recall | F-measure | AUC | G-mean |
|---|---|---|---|---|---|---|---|
| DT | 0.942 | 0.112 | 0.946 | 0.942 | 0.9400 | 0.920 | 0.911 |
| ANN | 0.761 | 0.389 | 0.758 | 0.761 | 0.743 | 0.756 | 0.646 |
| RF | 0.958 | 0.081 | 0.961 | **0.958** | **0.957** | 0.961 | **0.936** |
| NB | 0.869 | 0.253 | 0.891 | 0.869 | 0.860 | 0.841 | 0.785 |
| BBN | 0.956 | 0.083 | 0.959 | 0.956 | 0.956 | 0.951 | 0.935 |
| SVM | 0.662 | 0.656 | 0.776 | 0.662 | 0.530 | 0.503 | 0.077 |
| DT bagging | 0.947 | 0.103 | 0.95 | 0.947 | 0.945 | 0.953 | 0.919 |
| ANN bagging | 0.815 | 0.357 | 0.853 | 0.815 | 0.792 | 0.836 | 0.678 |
| NB bagging | 0.863 | 0.266 | 0.886 | 0.863 | 0.852 | 0.856 | 0.772 |
| BBN bagging | 0.956 | 0.083 | 0.958 | 0.956 | 0.955 | 0.951 | 0.934 |
| SVM bagging | 0.662 | 0.656 | 0.776 | 0.662 | 0.529 | 0.505 | 0.077 |
| DT boosting | 0.951 | 0.086 | 0.952 | 0.951 | 0.950 | 0.955 | 0.931 |
| ANN boosting | 0.795 | 0.382 | 0.824 | 0.795 | 0.770 | 0.757 | 0.650 |
| NB boosting | 0.869 | 0.253 | 0.891 | 0.869 | 0.860 | 0.807 | 0.785 |
| BBN boosting | 0.954 | 0.082 | 0.956 | 0.954 | 0.953 | 0.949 | 0.934 |
| SVM boosting | 0.663 | 0.654 | 0.777 | 0.663 | 0.532 | 0.561 | 0.095 |

#### 7.1.3.2 ADASYN InfoGainEval

In this subsection, we present the experimental results for ADASYN for the dataset created by InfoGain feature selection technique. Table 25 shows the summary of the results for the restatement class. Using ADASYN, the best minority recall is achieved by DT boosting ensemble model and RF model. The minority recall values are 89.4% and 88.9% respectively. RF classifier yields the highest F-measure (94%) that incorporate both precision

and recall values for the minority class instances. BBN boosting ensemble classifier also shows a reasonably high success rate in terms of minority recall (88.8%).

In order to get an idea on various classifiers' success rate in detecting majority class instances, next we turn our focus on the majority class recall value and majority class F-measure. We find that some of the models (RF, SVM) are highly successful in detecting majority class instances.

Both recall value and F-measure focus on single class and hence do not give us a holistic perspective. Therefore, we explore G-mean values at the aggregate level that accounts for performance of both minority and majority class identification. This measure considers the success rate of both majority (i.e. non-restatement) and minority (i.e. restatement) classes simultaneously. In Table 27, we find that, the highest G-mean value is 94.2% for RF classifier. We further find that the weighted average recall value and weighted average F-measure are 96.1% and 96.1% respectively for the RF classifier.

Table 25: ADASYN performance measures of the minority class for InfoGainEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.861 | 0.001 | 0.999 | 0.861 | 0.925 | 0.931 |
| ANN | 0.594 | 0.161 | 0.658 | 0.594 | 0.625 | 0.803 |
| RF | 0.889 | 0.001 | 0.998 | 0.889 | 0.940 | 0.960 |
| NB | 0.719 | 0.002 | 0.996 | 0.719 | 0.835 | 0.915 |
| BBN | 0.883 | 0.002 | 0.996 | 0.883 | 0.936 | 0.951 |
| SVM | 0.485 | 0.063 | 0.799 | 0.485 | 0.604 | 0.711 |
| DT bagging | 0.876 | 0.004 | 0.990 | 0.876 | 0.930 | 0.955 |
| ANN bagging | 0.611 | 0.066 | 0.829 | 0.611 | 0.704 | 0.866 |
| NB bagging | 0.701 | 0.002 | 0.994 | 0.701 | 0.822 | 0.927 |
| BBN bagging | 0.883 | 0.002 | 0.996 | 0.883 | 0.936 | 0.951 |
| SVM bagging | 0.484 | 0.063 | 0.800 | 0.484 | 0.603 | 0.726 |
| DT boosting | 0.894 | 0.019 | 0.96 | 0.894 | 0.926 | 0.961 |
| ANN boosting | 0.537 | 0.081 | 0.776 | 0.537 | 0.635 | 0.808 |
| NB boosting | 0.719 | 0.002 | 0.996 | 0.719 | 0.835 | 0.863 |
| BBN boosting | 0.888 | 0.010 | 0.978 | 0.888 | 0.931 | 0.948 |
| SVM boosting | 0.488 | 0.065 | 0.797 | 0.488 | 0.605 | 0.798 |

Table 26: ADASYN majority class performance measures for InfoGainEval

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.999 | 0.139 | 0.932 | 0.999 | 0.965 | 0.931 |
| ANN | 0.839 | 0.406 | 0.799 | 0.839 | 0.819 | 0.803 |
| RF | 0.999 | 0.111 | 0.945 | 0.999 | 0.971 | 0.960 |
| NB | 0.998 | 0.281 | 0.872 | 0.998 | 0.931 | 0.861 |
| BBN | 0.998 | 0.117 | 0.942 | 0.998 | 0.969 | 0.951 |
| SVM | 0.937 | 0.515 | 0.778 | 0.937 | 0.85 | 0.711 |
| DT bagging | 0.996 | 0.124 | 0.939 | 0.996 | 0.966 | 0.955 |
| ANN bagging | 0.934 | 0.389 | 0.822 | 0.934 | 0.874 | 0.866 |
| NB bagging | 0.998 | 0.299 | 0.865 | 0.998 | 0.927 | 0.882 |
| BBN bagging | 0.998 | 0.117 | 0.943 | 0.998 | 0.970 | 0.951 |
| SVM bagging | 0.937 | 0.516 | 0.777 | 0.937 | 0.850 | 0.726 |
| DT boosting | 0.981 | 0.106 | 0.947 | 0.981 | 0.963 | 0.96 |
| ANN boosting | 0.919 | 0.463 | 0.792 | 0.919 | 0.851 | 0.808 |
| NB boosting | 0.998 | 0.281 | 0.872 | 0.998 | 0.931 | 0.863 |
| BBN boosting | 0.990 | 0.112 | 0.944 | 0.990 | 0.966 | 0.948 |
| SVM boosting | 0.935 | 0.512 | 0.778 | 0.935 | 0.850 | 0.798 |

Table 27: ADASYN weighted average performance measures for InfoGainEval

| | TP | FP | Precision | Recall | F-measure | AUC | G-mean |
|---|---|---|---|---|---|---|---|
| DT | 0.952 | 0.092 | 0.955 | 0.952 | 0.951 | 0.931 | 0.927 |
| ANN | 0.755 | 0.322 | 0.751 | 0.755 | 0.752 | 0.803 | 0.706 |
| RF | 0.961 | 0.074 | 0.963 | **0.961** | **0.961** | 0.960 | **0.942** |
| NB | 0.903 | 0.185 | 0.915 | 0.903 | 0.898 | 0.880 | 0.847 |
| BBN | 0.959 | 0.078 | 0.961 | 0.959 | 0.958 | 0.951 | 0.939 |
| SVM | 0.782 | 0.36 | 0.785 | 0.782 | 0.766 | 0.711 | 0.674 |
| DT bagging | 0.955 | 0.083 | 0.957 | 0.955 | 0.954 | 0.955 | 0.934 |
| ANN bagging | 0.824 | 0.278 | 0.824 | 0.824 | 0.816 | 0.866 | 0.755 |
| NB bagging | 0.896 | 0.197 | 0.909 | 0.896 | 0.891 | 0.897 | 0.836 |
| BBN bagging | 0.959 | 0.077 | 0.961 | 0.959 | 0.958 | 0.951 | 0.939 |
| SVM bagging | 0.782 | 0.361 | 0.785 | 0.782 | 0.765 | 0.726 | 0.673 |
| DT boosting | 0.951 | 0.076 | 0.951 | 0.951 | 0.951 | 0.961 | 0.936 |
| ANN boosting | 0.788 | 0.332 | 0.787 | 0.788 | 0.777 | 0.808 | 0.702 |
| NB boosting | 0.903 | 0.185 | 0.915 | 0.903 | 0.898 | 0.863 | 0.847 |
| BBN boosting | 0.955 | 0.077 | 0.956 | 0.955 | 0.954 | 0.948 | 0.938 |
| SVM boosting | 0.782 | 0.359 | 0.785 | 0.782 | 0.766 | 0.798 | 0.675 |

#### 7.1.3.3 ADASYN FwSelect

In this subsection, we present the experimental results for ADASYN for the dataset created by forward feature selection technique. Table 28 shows the summary of the results for the restatement class. Using ADASYN, the best minority recall is achieved by RF model. The minority recall value is 86.8%. RF classifier yields the highest F-measure (92.6%) that incorporate both precision and recall values for the minority class instances. BBN and its ensemble classifier also shows reasonably high success rate in terms of minority recall (84.4%).

In order to get an idea on various classifiers' success rate in detecting majority class instances, next we turn our focus on the majority class recall value and majority class F-measure. We find that some of the models (RF, SVM) are highly successful in detecting majority class instances.

Both recall value and F-measure focus on single class and hence do not give us a holistic perspective. Therefore, we explore G-mean values at the aggregate level that accounts for performance of both minority and majority class identification. This measure considers the success rate of both majority (i.e. non-restatement) and minority (i.e. restatement) classes simultaneously. In Table 30, we find that, the highest G-mean value is 93% for RF classifier. We further find that the weighted average recall value and weighted average F-measure are 95.3% and 95.2% respectively for the RF classifier.

Table 28: ADASYN minority class performance measures for FwSelect

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.774 | 0.009 | 0.978 | 0.774 | 0.864 | 0.896 |
| ANN | 0.511 | 0.016 | 0.943 | 0.511 | 0.663 | 0.814 |
| RF | 0.868 | 0.004 | 0.992 | 0.868 | 0.926 | 0.956 |
| NB | 0.524 | 0.002 | 0.992 | 0.524 | 0.686 | 0.869 |
| BBN | 0.844 | 0.003 | 0.993 | 0.844 | 0.912 | 0.945 |
| SVM | 0.063 | 0.000 | 1.000 | 0.063 | 0.119 | 0.532 |
| DT bagging | 0.811 | 0.008 | 0.981 | 0.811 | 0.888 | 0.939 |
| ANN bagging | 0.545 | 0.013 | 0.956 | 0.545 | 0.694 | 0.876 |
| NB bagging | 0.522 | 0.001 | 0.994 | 0.522 | 0.685 | 0.895 |
| BBN bagging | 0.844 | 0.003 | 0.993 | 0.844 | 0.913 | 0.945 |
| SVM bagging | 0.062 | 0.000 | 1.000 | 0.062 | 0.118 | 0.535 |
| DT boosting | 0.845 | 0.021 | 0.954 | 0.845 | 0.896 | 0.944 |
| ANN boosting | 0.545 | 0.455 | 0.916 | 0.545 | 0.684 | 0.826 |
| NB boosting | 0.524 | 0.002 | 0.992 | 0.524 | 0.686 | 0.759 |
| BBN boosting | 0.846 | 0.005 | 0.990 | 0.846 | 0.912 | 0.945 |
| SVM boosting | 0.107 | 0.004 | 0.938 | 0.107 | 0.193 | 0.618 |

Table 29: ADASYN majority class performance measures for FwSelect

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.991 | 0.226 | 0.896 | 0.991 | 0.941 | 0.896 |
| ANN | 0.984 | 0.489 | 0.799 | 0.984 | 0.882 | 0.814 |
| RF | 0.996 | 0.132 | 0.937 | 0.996 | 0.966 | 0.956 |
| NB | 0.998 | 0.476 | 0.805 | 0.998 | 0.891 | 0.868 |
| BBN | 0.997 | 0.156 | 0.926 | 0.997 | 0.960 | 0.945 |
| SVM | 1.000 | 0.937 | 0.678 | 1.000 | 0.808 | 0.532 |
| DT bagging | 0.992 | 0.189 | 0.912 | 0.992 | 0.950 | 0.939 |
| ANN bagging | 0.987 | 0.455 | 0.810 | 0.987 | 0.890 | 0.876 |
| NB bagging | 0.999 | 0.478 | 0.805 | 0.999 | 0.645 | 0.895 |
| BBN bagging | 0.997 | 0.156 | 0.926 | 0.997 | 0.960 | 0.945 |
| SVM bagging | 1.000 | 0.938 | 0.678 | 1.000 | 0.808 | 0.535 |
| DT boosting | 0.979 | 0.155 | 0.926 | 0.979 | 0.952 | 0.944 |
| ANN boosting | 0.975 | 0.455 | 0.809 | 0.975 | 0.884 | 0.826 |
| NB boosting | 0.998 | 0.476 | 0.805 | 0.998 | 0.891 | 0.759 |
| BBN boosting | 0.995 | 0.154 | 0.927 | 0.995 | 0.960 | 0.945 |
| SVM boosting | 0.996 | 0.893 | 0.687 | 0.996 | 0.814 | 0.618 |

Table 30: ADASYN weighted average performance measures for FwSelect

| | TP | FP | Precision | Recall | F-measure | AUC | G-mean |
|---|---|---|---|---|---|---|---|
| DT | 0.918 | 0.153 | 0.924 | 0.918 | 0.915 | 0.896 | 0.876 |
| ANN | 0.825 | 0.330 | 0.847 | 0.825 | 0.808 | 0.814 | 0.709 |
| RF | 0.953 | 0.089 | 0.956 | 0.953 | 0.952 | 0.956 | **0.930** |
| NB | 0.839 | 0.316 | 0.868 | 0.839 | 0.822 | 0.869 | 0.723 |
| BBN | 0.945 | 0.105 | 0.949 | 0.945 | 0.944 | 0.945 | 0.917 |
| SVM | 0.685 | 0.621 | 0.786 | 0.685 | 0.576 | 0.532 | 0.251 |
| DT bagging | 0.931 | 0.128 | 0.935 | 0.931 | 0.929 | 0.939 | 0.897 |
| ANN bagging | 0.838 | 0.306 | 0.859 | 0.838 | 0.824 | 0.876 | 0.733 |
| NB bagging | 0.838 | 0.317 | 0.869 | 0.838 | 0.822 | 0.895 | 0.722 |
| BBN bagging | 0.945 | 0.104 | 0.949 | 0.946 | 0.944 | 0.945 | 0.917 |
| SVM bagging | 0.684 | 0.622 | 0.786 | 0.684 | 0.575 | 0.535 | 0.249 |
| DT boosting | 0.934 | 0.110 | 0.935 | 0.934 | 0.933 | 0.944 | **0.910** |
| ANN boosting | 0.830 | 0.310 | 0.845 | 0.830 | 0.816 | 0.826 | 0.545 |
| NB boosting | 0.839 | 0.316 | 0.868 | 0.839 | 0.822 | 0.759 | 0.723 |
| BBN boosting | 0.945 | 0.104 | 0.948 | 0.945 | 0.944 | 0.945 | **0.917** |
| SVM boosting | 0.697 | 0.593 | 0.772 | 0.697 | 0.605 | 0.618 | 0.326 |

## 7.2 Result Evaluation

In this subsection, we summarize the experimental results presented in section 7.1. As discussed earlier in this chapter, we focus on three performance measures: minority class recall value, minority class F-measure and G-mean while synthesizing various results. The first two measures focus on the classifier's success in predicting minority classes, while G-mean considers the success rate of both majority (i.e. non-restatement) and minority (i.e. restatement) classes simultaneously. Further, all performance measures are compared for three datasets that are obtained by employing three feature selection methods (CfsSubsetEval, InfoGain, FwSelect).

***Minority recall value:*** From Figure 11, we find that SVM and its ensembles has the highest recall value for the CUS approach. It appears that, RF also has achieved higher recall compared to the others. For the Tomek-SMOTE approach (Figure 12), RF, NB and ND ensembles achieves the highest recall value. On the other hand, for ADASYN approach, RF has the higher recall value (Figure 13).

Figure 11: CUS minority recall

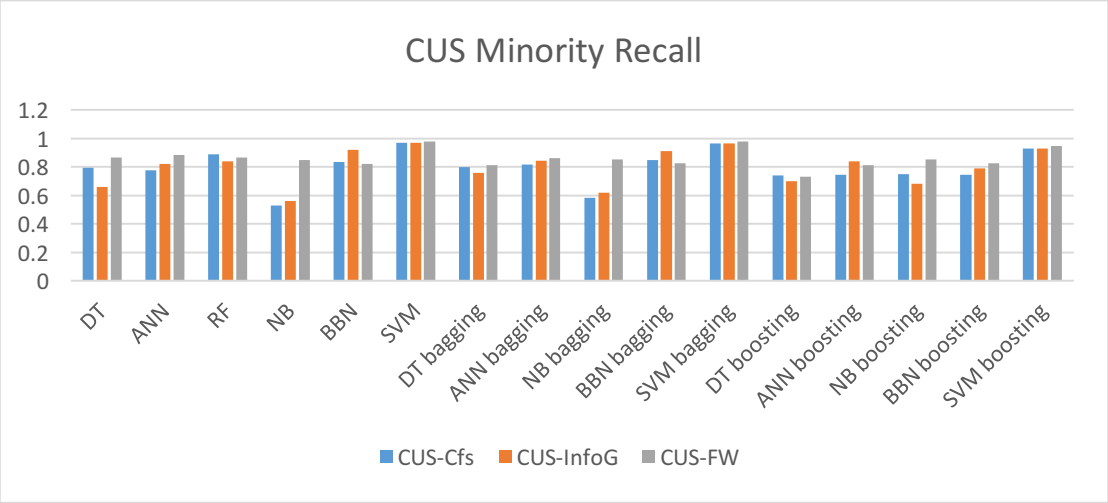


Figure 12: Tomek SMOTE minority recall

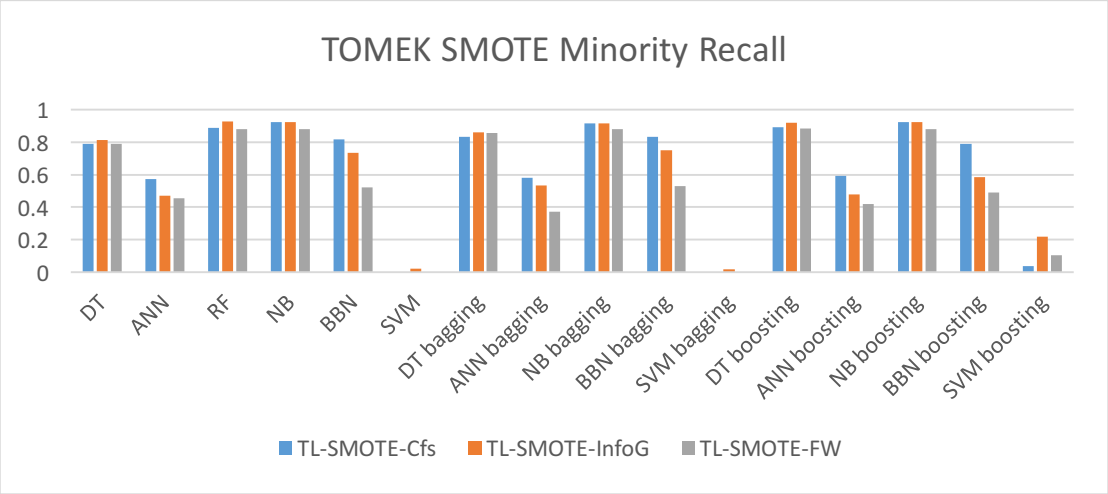
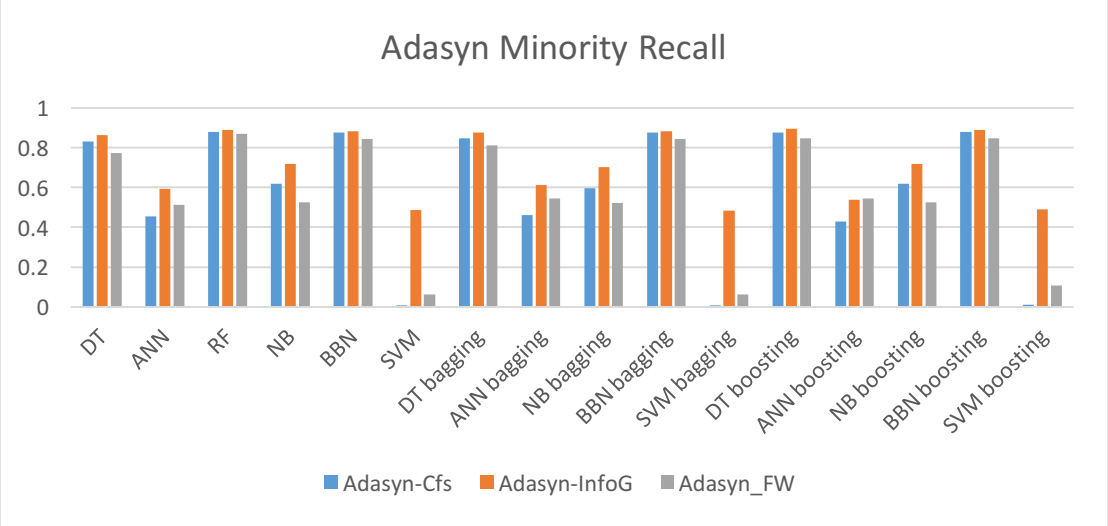



Figure 13: ADASYN minority recall




***F-measure:*** Similar to recall plot, Figure 14 suggest that for CUS approach, SVM achieved highest F-measure followed by RF. For the other two oversampling methods (Figure 15 and 16), RF attained the highest performance.

From these comparative analysis, we find that, in our financial restatement sample RF classifier shows the best performance in terms of predicting minority instances. It should

be noted that a particular feature selection technique does not yield the best result for all classifiers.

Figure 14: CUS F-measure

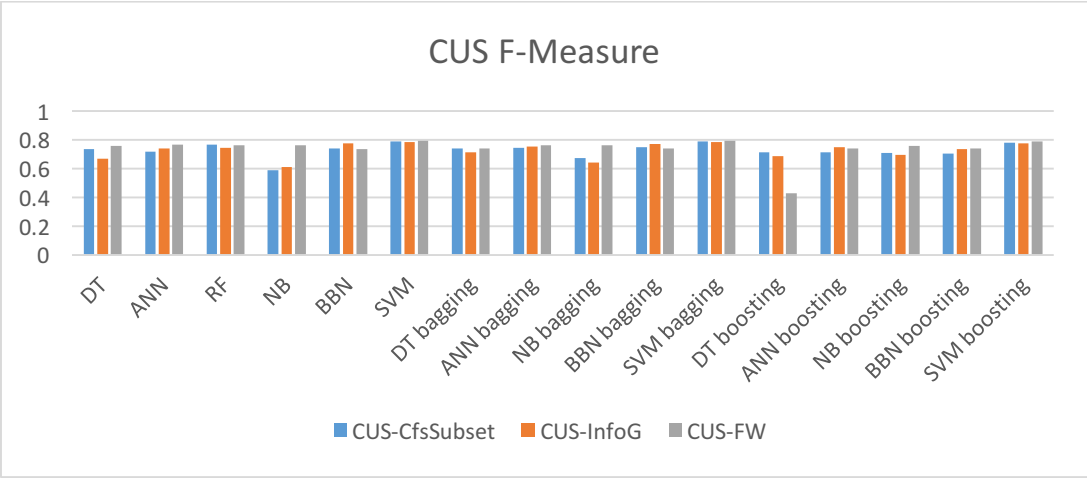



Figure 15: Tomek Smote F-measure

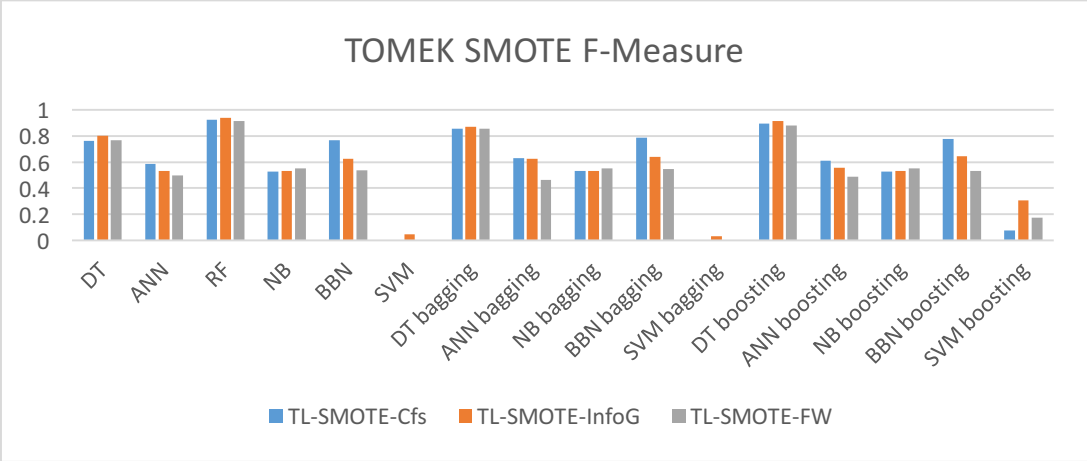
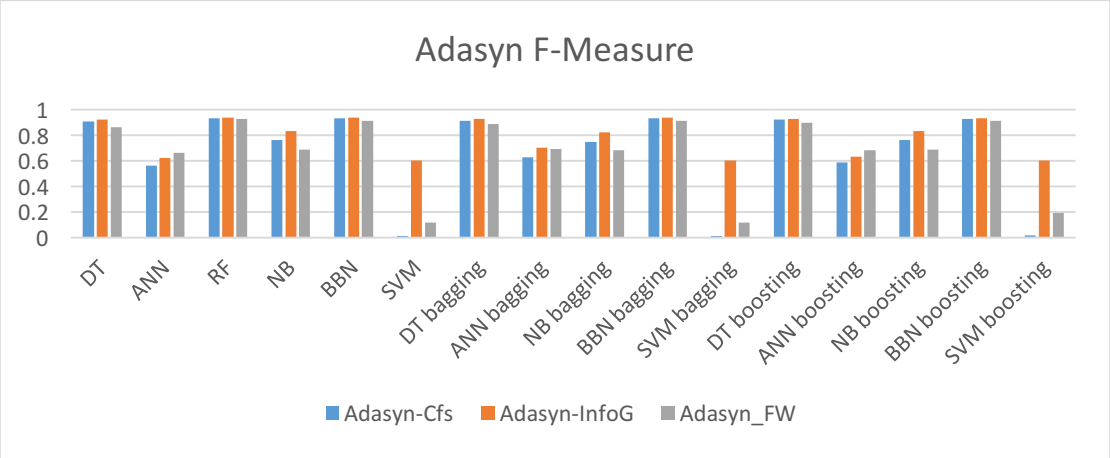


Figure 16: ADASYN F-measure




While minority recall value and F-measure are useful in evaluating a classifier's ability to predict minority class instances, these measures do not reflect the overall effectiveness of a classifier in identifying both minority and majority class instances simultaneously. Accordingly, we focus on G-mean value subsequently which considers both minority and majority class recall value concurrently.

***G-mean performance measure:*** Based on Figure 17 we find that, in general, G-mean values are lower for all classifiers in cluster based undersampling (CUS). On the other hand, we find encouraging trends in G-mean values in synthetic oversampling based class imbalance techniques or algorithms ('Tomek + SMOTE' and ADASYN). The results are summarized in Figure 18 and 19. Particularly, RF classifier show high values of G-mean in 'Tomek + SMOTE' (95.2%) and ADASYN (94.2%) with InfoGainEval feature selection technique. It should be noted that, RF classifier also perform quite well in detecting minority class instances. Using the dataset obtained through InfoGainEval feature selection approach, RF classifier shows minority recall values of 92.6% for 'Tomek Link + SMOTE' and 88.9% for

'ADASYN' techniques, respectively. These results show that RF classifier is quite effective in predicting both minority and majority classes.

Figure 17: CUS G-mean

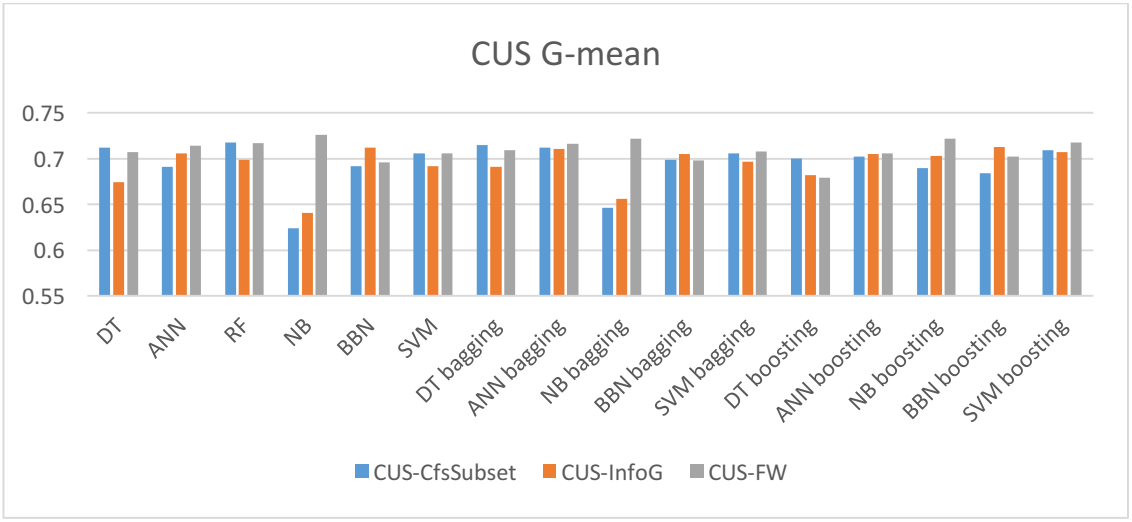



Figure 18: Tomek Smote G-mean

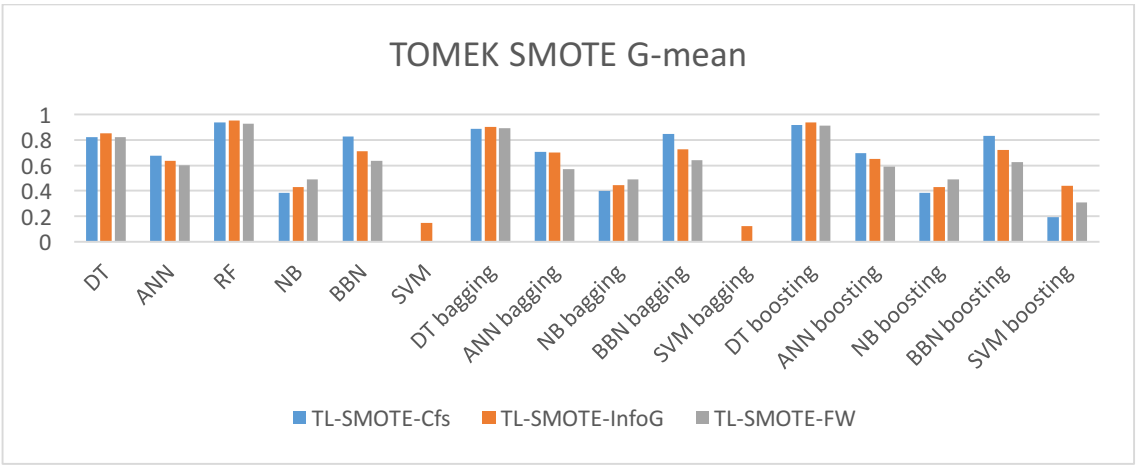
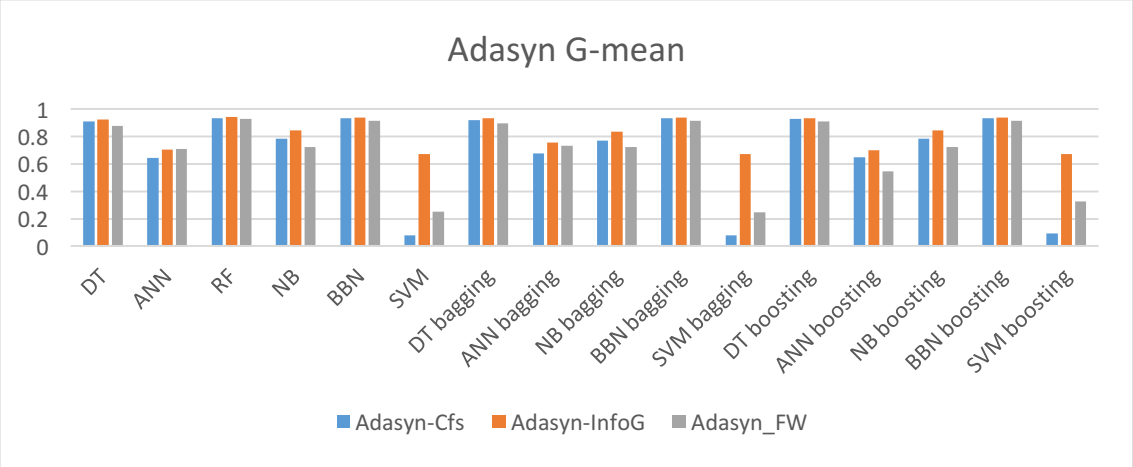


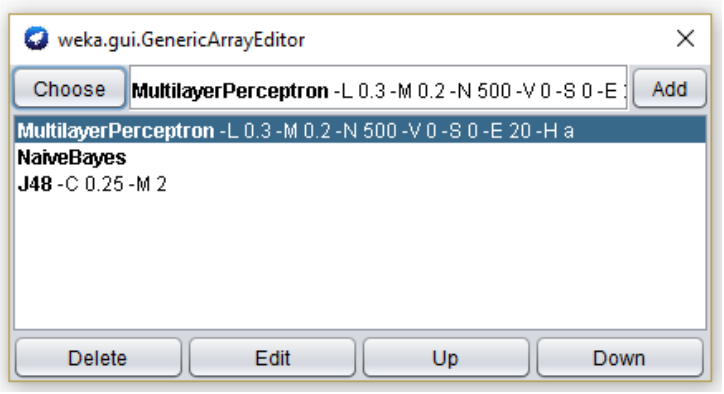
Figure 19: ADASYN G-mean




## 7.2.1 Stacking

We also use other meta-learning algorithms to find out whether we can improve the prediction model further. First, we employ a stacking algorithm by combining a set of base learners with a meta-learner. We employed various combinations of base learner and meta-learner. Among the all stacking experiments, the stacking model that achieved the best performance is shown in Figure 20, we include ANN, NB and DT as the base learners and SVM as the meta-learning algorithm.

Figure 20: Stacking framework



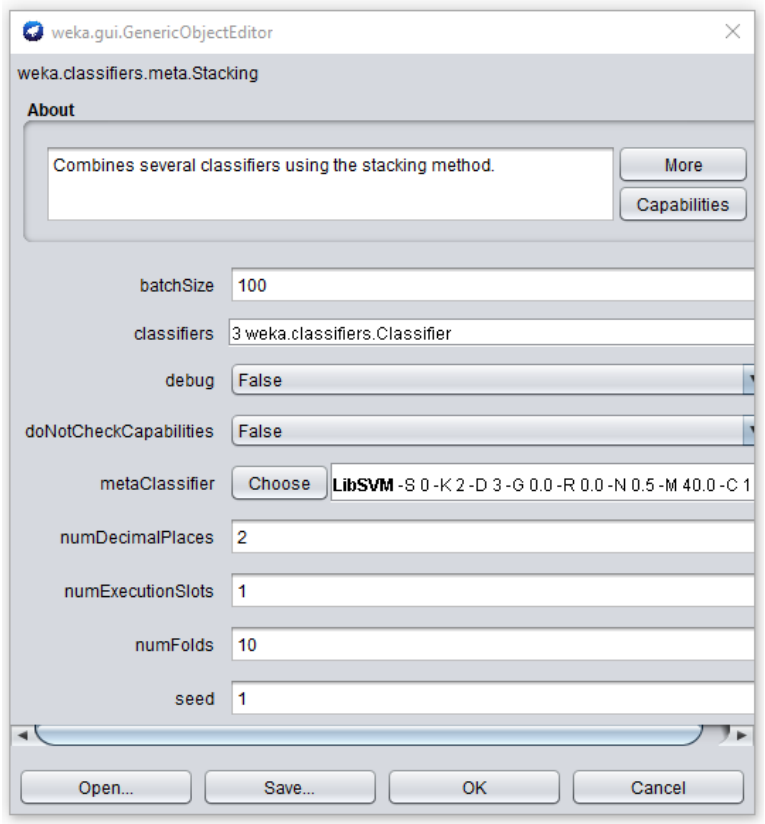



Table 31 presents the performance measures for the stacking algorithm using ADASYN (oversampling) and CUS (undersampling) approaches for infoGainEval dataset. It should be noted that, in this table we only present the results for highest performing models achieved by stacking combinations. The G-mean value for the stacking model is 92.7%. While the performance of the stacking model is encouraging, RF classifier (with ADASYN) outperforms this stacking model in terms of minority recall value, minority F-measure and G-mean values.

Table 31: Stacking results

| | TP | FP | Precision | Recall | F-measure | AUC | G-mean |
|---|---|---|---|---|---|---|---|
| InfoG-ADASYN | 0.861 | 0.001 | 0.999 | 0.861 | 0.925 | 0.930 | 0.927 |
| InfoG-CUS | 0.934 | 0.481 | 0.660 | 0.934 | 0.774 | 0.727 | 0.696 |

### 7.2.1 Cost-sensitive learning

Existing literature shows that financial frauds or restatements can lead to a significant level of shareholder wealth destruction. Hence, it is highly desirable to have effective predictive models to detect financial restatements. However, prediction models are not perfect and can lead to false negative and false positive cases. Such outcomes could be costly and should be addressed in the predictive models. Yet, we should recognize that cost of false negative outcome and false positive outcome are different. "In fraud detection, misclassification costs (false positive and false negative error costs) are unequal, uncertain, can differ from example to example, and can change over time. In fraud detection, a false negative error is usually more costly than a false positive error" (Phua el al. 2005, p. 4). Similarly Perols (2011) also observed that "It is more costly to classify a fraud firm as a non-fraud firm than to classify a non-fraud firm as a fraud firm" (p. 20). We address this issue with a cost sensitive learning algorithm included in Weka. Since RF model achieved the best overall performance as well as very good minority and majority recall performance, we investigate this classification algorithm using cost sensitive learning.

Table 32: Cost matrices

| Classified as --> | Cost matrix 1 | | Cost matrix 2 | | Cost matrix 3 | |
|---|---|---|---|---|---|---|
| | R | NR | R | NR | R | NR |
| Restatement (R) | 0 | 1 | 0 | 1 | 0 | 1 |
| Non-restatement (NR) | 5 | 0 | 10 | 0 | 20 | 0 |

In this context, we create three sets of cost matrices as shown in Table 32. These cost matrices represent three sets of relative error costs (false positive: false negative): (i) 1:5, (i) 1:10, and (iii) 1:20. In case of a false positive outcome, a non-restatement firm is incorrectly classified as a restatement firm; whereas, in case of a false negative outcome, a restatement firm is incorrectly classified as a non-restatement firm. Given that false negative cases are more damaging, we assign a higher cost weight for this category.

Table 33 presents the results for cost-sensitive learning for RF model with InfoGainSubset and CfsSubset datasets. We find that, in general, results for RF model improve with cost-sensitive learning algorithm. These results dominate other results obtained by using FwSelect dataset and 'Tomek + SMOTE' approach, hence are not shown in the table. The main improvement in RF model performance happens in the form of minority recall value. The RF classifier shows a very successful predictive capability for minority class instances (97.4%) and a very high G-mean value (95.3%) with cost-sensitive learning. However, as expected in cost-sensitive learning algorithm, an improvement in minority class value reduces the predictive capability of majority class instances (91.7%).

Table 33: Cost sensitive learning with RF

| | | Minority class | | | | | | | Average |
|---|---|---|---|---|---|---|---|---|---|
| | | Cost | TP | Specificity | Precision | Recall | F-measure | AUC | G-mean |
| Adasyn | InfoG | 5 | 0.897 | 0.990 | 0.979 | 0.897 | 0.936 | 0.961 | 0.942 |
| | | **10** | **0.906** | **0.973** | **0.946** | **0.906** | **0.926** | **0.961** | **0.939** |
| | | 20 | 0.922 | 0.900 | 0.827 | 0.922 | 0.872 | 0.961 | 0.911 |
| | Cfs | 5 | 0.889 | 0.998 | 0.996 | 0.889 | 0.939 | 0.962 | 0.942 |
| | | 10 | 0.895 | 0.989 | 0.976 | 0.895 | 0.934 | 0.963 | 0.941 |
| | | 20 | 0.913 | 0.936 | 0.88 | 0.913 | 0.896 | 0.963 | 0.924 |
| CUS | InfoG | 5 | 0.972 | 0.503 | 0.662 | 0.972 | 0.788 | 0.772 | 0.699 |
| | | 10 | 0.981 | 0.492 | 0.659 | 0.981 | 0.789 | 0.776 | 0.695 |
| | | 20 | 0.987 | 0.439 | 0.638 | 0.987 | 0.775 | 0.772 | 0.658 |
| | Cfs | 5 | 0.976 | 0.499 | 0.661 | 0.976 | 0.788 | 0.778 | 0.698 |
| | | 10 | 0.983 | 0.477 | 0.653 | 0.983 | 0.785 | 0.783 | 0.685 |
| | | 20 | 0.984 | 0.399 | 0.621 | 0.984 | 0.762 | 0.770 | 0.627 |
| Tomek Smote | InfoG | **5** | **0.974** | **0.917** | **0.858** | **0.974** | **0.913** | **0.990** | **0.953** |
| | | **10** | **0.983** | **0.833** | **0.751** | **0.983** | **0.852** | **0.988** | **0.905** |
| | | 20 | 0.991 | 0.661 | 0.600 | 0.991 | 0.747 | 0.986 | 0.809 |
| | Cfs | 5 | 0.967 | 0.876 | 0.807 | 0.967 | 0.879 | 0.984 | 0.920 |
| | | 10 | 0.981 | 0.745 | 0.673 | 0.981 | 0.798 | 0.980 | 0.855 |
| | | 20 | 0.990 | 0.558 | 0.545 | 0.990 | 0.703 | 0.976 | 0.743 |

## 7.3 Variable Importance (VI) and Decision Rules (DR)

In this subsection, we present some decision rules that can be helpful for the practitioners to detect restatement cases based on relevant attributes. Based on our experiments, we find that using InfoGainEval feature selection method, RF classifier shows very promising results for 'Tomek Link + SMOTE' techniques. Therefore, it would be ideal to develop decision rules based on our RF classifier results. However, RF classifier is an ensemble technique that develops prediction model(s) based on multiple decision trees. Each tree is trained on bagged data using random selection of features. Therefore, it is infeasible

to develop decision rules by examining each tree individually. In order to circumvent this problem, we rely on decision tree (DT) classifier, which generates one tree with various nodes and leafs, to develop a set of feasible and practical decision rules. We employ the following steps in developing the relevant rules.

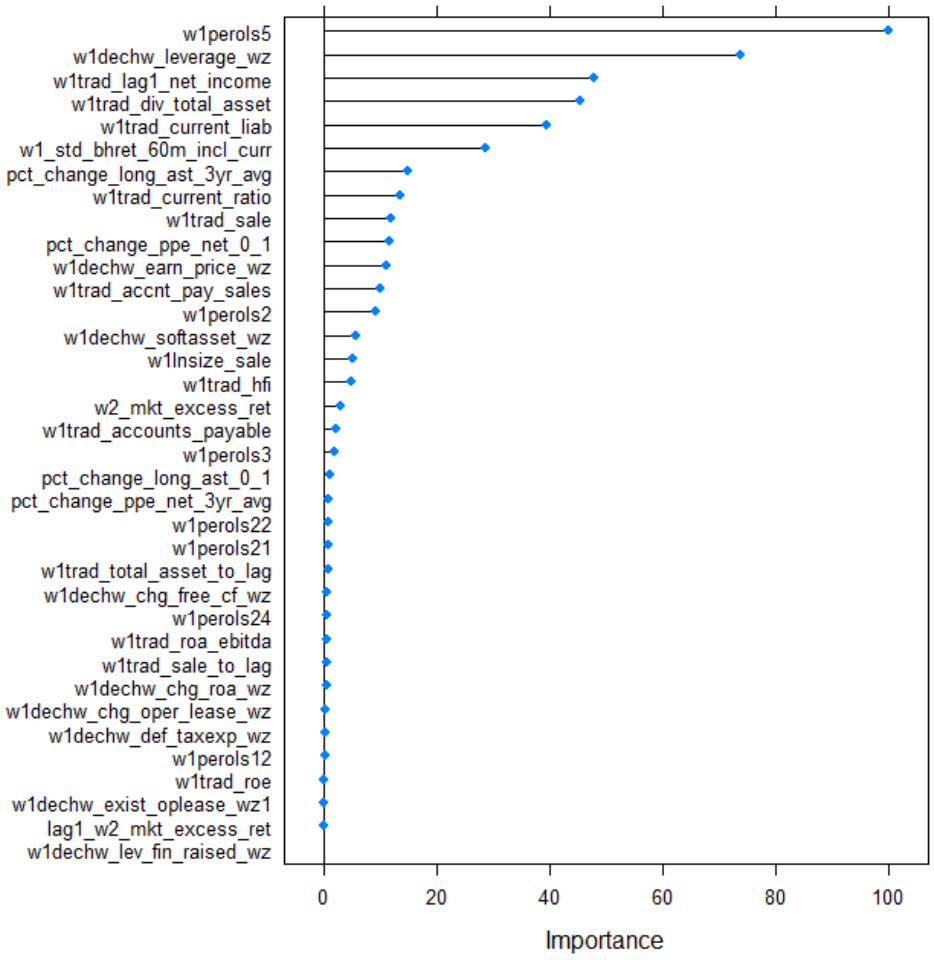
- First, we attempt to reduce the number of features (i.e. attributes) in order to have more parsimonious decision rules. Our dataset based on InfoGainEval feature selection method contains 37 attributes. Inclusion of all these attributes in the DT model might lead to complex decision rules. Therefore, we keep a subset of attributes based on their variable importance scores (Griselda et al., 2012).
- Second, we use DT classifier and develop prediction model with the reduced set of variables. Based on the corresponding decision tree (DT), we present a set of decision rules.

### 7.3.1 Variable Importance (VI)

We employ R software package to identify the variables that are of key importance in the prediction of restatement cases. We use importance index (Kashami and Mohaymany, 2011) in order to identify the key variables. The variable importance with 'important index value' for each individual variable is presented in Figure 21 below. Using this index, we identify 14 variables that have significant influence on restatement predictability. While there is no definite rule to select a sub-set of variables based on 'important index value', we rationalize this selection based on the trade-off associated with 'too few' or 'too many' variable selection. 'Too many' variable section will lead to a more complex tree, which will make it quite difficult to develop feasible decision rules. On the other hand, 'too few'

variable will lead to poor decision rules as the classifier will not be able to predict restatement cases effectively. Below we present a brief description of and discussion on these 14 variables.

Figure 21: Variable importance with InfoGain and Tomel+SMOTE

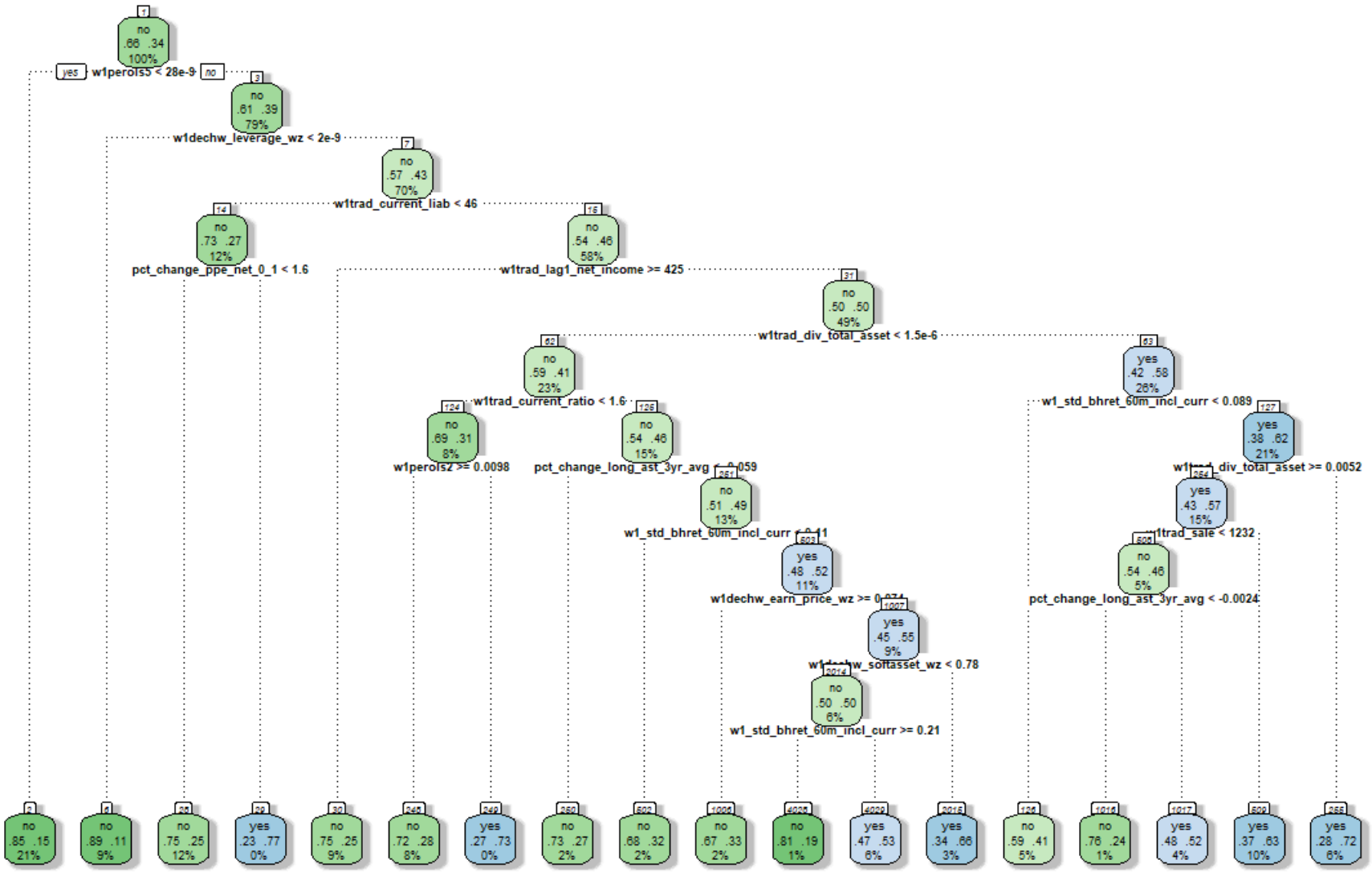



We find that 'Allowance for doubtful accounts to net sales' (Perols5) has the highest level of variable importance score. The firms that make more provisions for doubtful accounts have a higher propensity to restate their financial statements. 'w1dechw_leverage_wz' refers to the leverage (i.e. long-term debt to total asset) of the firm. In a firm with higher leverage, manages may get involved into earnings management to show

better performance to outside debtholders and in the process may violate accounting rules. This may lead to accounting restatements. 'w1trad_lag1_net_income' denotes last year's net income (i.e. firm's profit). If a firm is more profitable, they have less pressure to have aggressive accounting practices. This reduces the chance of restatement. 'w1trad_div_total_asset' is the dividend to total asset of the firm. Firms paying higher dividends have more pressure on their cash flows and may indulge into earnings management to improve financial results. This increases the probability of restatement. 'w1trad_current_liab' refers to the current liability (e.g. short term payment obligation) of the firm. The argument for this attribute is similar to the reasoning of leverage variable. Firms with more current liability feel more pressure to window dress its financial statements. 'w1_std_bhret_60m_incl_curr' refers to the standard deviation of last 60 months stock returns – it denotes firm risk. Less risky firms are likely to have a lower probability of restatement. 'pct_change_long_ast_3yr_avg' (3-year arithmetic average of percentage change in long term asset) and 'pct_change_ppe_net_0_1' (percentage change in plant, property and equipment (PPE)' refer to the growth of firm asset(s). Generally, growth firms adopt more aggressive accounting practices that increase the probability of restatement. 'w1trad_current_ratio' (calculated as current asset to current liability) denotes the short term financial strength of a firm. The higher the value of this ratio, the lower will be the chance of restatement. 'w1perols2' (accounts receivable to total assets) follows a similar logic. Higher accounts receivables alleviates short-term financing pressure of a firm. This reduces the need for accounting manipulation. 'w1trad_sale' represents the sales (i.e. revenue) level of a firm. Generally, firms with higher sales level has more financial resources. However, if the higher sales is accompanies with a high growth rate, it indicates firm complexity and

increases the chance of financial restatements. Higher sales may also be associated with revenue recognition problem that can increase the probability of restatement. 'w1dechw_earn_price_wz' represents the ratio of earnings per share to price per share. A higher value of this ratio indicates that the firm is not overvalued. Such firms are less likely to practice creative accounting and hence may not need to restate their financial statements. 'w1trad_accnt_pay_sale' refers to accounts payable to sales. A higher value of this attribute increases pressure on management and may lead to the practice of more aggressive accounting. 'w1dechw_softasset' represents the proportion of soft assets (i.e. intangible assets) in a firm. It is calculated as the ratio of (total asset minus fixed asset and cash & short-term investments) and average total asset. A higher level of soft asset increases managerial discretion in terms of accounting practices, which may lead to violation of accounting rules.

Figure 22: Decision tree for decision rules



### 7.3.2 Decision Rules (DR)

The decision tree (DT) obtained with our dataset has 18 terminal nodes, 11 of them has been identified as decision rules (DR). As we find, most of the rules (8 out of 11) are for the identification of non-restatement cases (NRS) and 3 rules are for the identification of restatements. While developing these rules, we rely on two other parameters (Population % and Probability %) to extract more meaningful rules and guidelines (Griselda et al. 2012). 'Population %' is the percentage of cases of a node in relation to the total number of cases analyzed; and class 'probability %' is the percentage of cases for the resulting class. Following Griselda et al. (2012), the minimum values used to make the rules more meaningful are: 'Population %' >=1% and 'Probability %' >=60%. All 11 rules are presented in Table 34. Some rules are easier to implement while others involve more nodes and splits. For example, decision rule 1 involves only two nodes. This rule implies that firms with less allocation to doubtful accounts ('w1perols5') have a lower probability of restatement. On the other hand, the decision rule 11 involves 11 nodes and 10 splits. Rule 11 is based on 10 attributes. We have discussed the rationale behind each attribute in sub-section 7.3.1. We discuss the logical sequence of Rule 11 below. (Other rules can be interpreted in a similar fashion).

Table 34: Decision rules

| Rule No. | Node | Decision Rule (IF..AND..AND) | THEN | Population % | Probability % |
|---|---|---|---|---|---|
| 1 | 2 | IF w1perols5 < $28e^{-9}$ | NRS | 21 | 85 |
| 2 | 6 | IF w1perols5 >= $28e^{-9}$ AND w1dechw_leverage_wz < $2e^{-9}$ | NRS | 9 | 89 |
| 3 | 28 | IF (w1perols5 >= $28e^{-9}$ AND w1dechw_leverage_wz >= $2e^{-9}$) AND w1trad_current_liab < 46 AND pct_change_ppe_net_0_1 < 1.6 | NRS | 12 | 75 |
| 4 | 30 | IF (w1perols5 >= $28e^{-9}$ AND w1dechw_leverage_wz >= $2e^{-9}$AND w1trad_current_liab >= 46) AND w1trad_lag1_net_income < 425 | NRS | 9 | 75 |
| 5 | 148 | IF (w1perols5 >= $28e^{-9}$ AND w1dechw_leverage_wz >= $2e^{-9}$ AND w1trad_current_liab >= 46 AND w1trad_lag1_net_income >= 425) AND w1trad_div_total_asset < $1.5e^{-6}$ AND w1trad_current_ratio < 1.6 AND w1perols2 < 0.0098 | NRS | 8 | 73 |
| 6 | 250 | IF (w1perols5 >= $28e^{-9}$ AND w1dechw_leverage_wz >= $2e^{-9}$ AND w1trad_current_liab >= 46 AND w1trad_lag1_net_income >= 425) AND w1trad_div_total_asset < $1.5e^{-6}$ AND w1trad_current_ratio >= 1.6 AND pct_change_long_ast_3yr_avg < 0.059 | NRS | 2 | 73 |
| 7 | 255 | IF (w1perols5 >= $28e^{-9}$ AND w1dechw_leverage_wz >= $2e^{-9}$ AND w1trad_current_liab >= 46 AND w1trad_lag1_net_income >= 425) AND w1trad_div_total_asset >= $1.5e^{-6}$) AND w1_std_bhret_60m_incl_curr >= 0.089 AND w1trad_div_total_asset >= 0.0052 | RS | 6 | 72 |
| 8 | 502 | IF (w1perols5 >= $28e^{-9}$ AND w1dechw_leverage_wz >= $2e^{-9}$ AND w1trad_current_liab >= 46 AND | NRS | 2 | 68 |

| | | | | | |
|---|---|---|---|---|---|
| | | w1trad_lag1_net_income >= 425 AND w1trad_div_total_asset < $1.5e^{-6}$ AND w1trad_current_ratio >= 1.6) AND pct_change_long_ast_3yr_avg >= 0.059 AND w1_std_bhret_60m_incl_curr < 0.41 | | | |
| 9 | 509 | IF (w1perols5 >= $28e^{-9}$ AND w1dechw_leverage_wz >= $2e^{-9}$ AND w1trad_current_liab >= 46 AND w1trad_lag1_net_income >= 425) AND w1trad_div_total_asset >= $1.5e^{-6}$ ) AND w1_std_bhret_60m_incl_curr >= 0.089 AND w1trad_div_total_asset < 0.0052 AND w1trad_sale >= 1232 | RS | 10 | 63 |
| 10 | 1006 | IF (w1perols5 >= $28e^{-9}$ AND w1dechw_leverage_wz >= $2e^{-9}$ AND w1trad_current_liab >= 46 AND w1trad_lag1_net_income >= 425 AND w1trad_div_total_asset < $1.5e^{-6}$ AND w1trad_current_ratio >= 1.6 AND pct_change_long_ast_3yr_avg >= 0.059 AND w1_std_bhret_60m_incl_curr >= 0.41) AND w1dechw_earn_price_wz < 0.074 | NRS | 2 | 67 |
| 11 | 2015 | IF (w1perols5 >= $28e^{-9}$ AND w1dechw_leverage_wz >= $2e^{-9}$ AND w1trad_current_liab >= 46 AND w1trad_lag1_net_income >= 425 AND w1trad_div_total_asset < $1.5e^{-6}$ AND w1trad_current_ratio >= 1.6 AND pct_change_long_ast_3yr_avg >= 0.059 AND w1_std_bhret_60m_incl_curr >= 0.41 AND w1dechw_earn_price_wz < 0.074 ) AND w1dechw_softasset_wz >= 0.78 | RS | 3 | 66 |

**Rule 11:**

IF (w1perols5 >= $28e^{-9}$ AND w1dechw_leverage_wz >= $2e^{-9}$ AND w1trad_current_liab >= 46 AND w1trad_lag1_net_income >= 425 AND w1trad_div_total_asset < $1.5e^{-6}$ AND w1trad_current_ratio >= 1.6 AND pct_change_long_ast_3yr_avg >= 0.059 AND w1_std_bhret_60m_incl_curr >= 0.41 AND w1dechw_earn_price_wz < 0.074) AND w1dechw_softasset_wz >= 0.78 THEN RS

**The logical sequence of rule 11:**

- Firms with a higher provision for doubtful accounts ('w1perols5'>= $28e^{-9}$) have a higher probability of restatement.
- Higher leverage ('w1dechw_leverage_wz'>= $2e^{-9}$) increases the chance of restatement.
- Higher current liability ('w1trad_current_liab'>= 46) increases the probability of restatement.
- A net income value ('w1trad_lag1_net_income'>= 425) of more than 425, leads to an equal probability for RS (restatement) and NRS (non-restatement).
- A lower dividend to total asset ratio ('w1trad_div_total_asset'< $1.5e^{-6}$) lowers the probability of restatement.
- A higher current ratio ('w1trad_current_ratio'>= 1.6) lowers the probability of restatement.
- A long-term asset growth value ('pct_change_long_ast_3yr_avg'>= 0.059) of more than 0.059, leads to an equal probability for RS (restatement) and NRS (non-restatement).

- A higher risk level (‘w1_std_bhret_60m_incl_curr’>= 0.41) increases the probability of restatement.
- An overvalued firm (‘w1dechw_earn_price_wz’< 0.074) has a higher risk of restatement.
- A firm with more soft assets (‘w1dechw_softasset_wz’ >= 0.78) has a higher chance of restatement.

## 7.4 Chapter Summary

In this chapter we present the experimental results pertaining to the detection of financial restatement instances using a real, comprehensive and novel dataset which includes approximately 1,000 restatement and 19,517 non-restatement instances over a period of 2009 to 2014. Our restatement dataset is highly skewed and highly biased towards non-restatement (majority) class. We applied various algorithms (e.g. random undersampling (RUS), Cluster based undersampling (CUS) (Sobhani et al., 2014), random oversampling (ROS), SMOTE, ADASYN (He et al., 2008), and Tomek links with SMOTE) to address class imbalance in the financial restatement dataset.

We perform classification employing six different choices of classifiers, DT, ANN, NB, RF, BBN and SVM using 10-fold cross validation and test the efficiency of various predictive models using minority class recall value, minority class F-measure and G-mean. We also experiment different ensemble methods (bagging and boosting) with these classifiers and employ other meta-learning algorithms (stacking and cost-sensitive learning) to improve model performance. While applying cluster-based undersampling technique, we find that various classifiers (e.g. SVM, BBN) show a high success rate in terms of minority class recall value. For

example, SVM classifier shows a minority recall value of 96% which is quite encouraging. However, the ability of these classifiers to detect majority class instances is dismal.

We find that some variations of synthetic oversampling such as 'Tomek Link + SMOTE' and 'ADASYN' show promising results in terms of both minority recall value and G-mean. Using InfoGainEval feature selection method, RF classifier shows minority recall values of 92.6% for 'Tomek Link + SMOTE' and 88.9% for 'ADASYN' techniques, respectively. The corresponding G-mean values are 95.2% and 94.2% for these two oversampling techniques, which show that RF classifier is quite effective in predicting both minority and majority classes. We find further improvement in results for RF classifier with cost-sensitive learning algorithm using 'Tomek Link + SMOTE' oversampling technique. Finally, we develop some decision rules to detect restatement firms based on a subset of important attributes.

To the best of our knowledge, only Kim et al. (2016) perform a data mining study using only pre-financial crisis restatement data. Kim et al. (2016) employed a matching sample based undersampling technique and used logistic regression, SVM and BBN classifiers to develop financial restatement predictive models. The study's highest reported G-mean is 70%. Our results with cluster based undersampling are similar to the performance measures reported by Kim et al. (2016). However, our synthetic oversampling based results show a better predictive ability. The RF classifier shows a very high degree of predictive capability for minority class instances (97.4%) and a very high G-mean value (95.3%) with cost-sensitive learning. Yet, we recognize that Kim et al. (2016) use a different restatement dataset (with pre-crisis restatement cases) and hence a direct comparison of results may not be justified.

# CHAPTER 8

# CONCLUSION AND FUTURE WORK

## 8.1 Conclusion

Data mining techniques and algorithms are widely used in developing predictive models and data visualization in various domains. Yet there are many unsettled issued in the data mining literature. For example, it appears that there is no consensus on the best data mining technique to develop predictive models. Similarly, we do not have any definite solution to address class imbalance issue. In this study, we focus on the suitability and robustness of various data mining techniques using a comprehensive financial restatement dataset. We also focus on how to address class imbalance issue in a large dataset. The results of this study are important and relevant for data mining as well as management literature. Moreover, the findings have practical significance.

In the context of data mining literature, it is quite interesting to study financial restatements for the following reasons: (i) identification of valid financial restatement cases is challenging, which may affect the reliability of predictive models, (ii) the restatement data is highly imbalanced that requires adequate attention in model building, (iii) financial restatement cases evolve over time which may affect the consistency of predictive models over different time periods, (iv) there are many financial and non-financial attributes that may affect financial restatement predictive models simultaneously. This requires careful implementation of data mining techniques to develop parsimonious models, and finally (v) there is no consensus in the data mining literature on the most suitable technique to detect

financial restatements. These characteristics make 'financial restatement dataset' as a suitable candidate to explore and address various challenges associated with data mining techniques and to get a holistic perspective.

Investors and regulators also pay a close attention to the financial restatement events. Earlier studies have shown that financial restatements could erode shareholders wealth quite significantly. Such events can also damage a firm's reputation considerably. Therefore, an early detection of suspect firms, which may make financial restatements or get involved with financial frauds in the subsequent periods, will be quite beneficial for the investors and regulators.

In this study, we developed predictive models based on all restatement cases (both intentional and unintentional restatements) using a real comprehensive and novel dataset which includes approximately 1,000 restatement and 19,517 non-restatement instances over a period of 2009 to 2014. To the best of our knowledge, no other study has developed predictive models for financial restatements using post-financial crisis events.

Our restatement dataset is highly skewed and highly biased towards non-restatement (majority) class. We applied various algorithms (e.g. random undersampling (RUS), Cluster based undersampling (CUS) (Sobhani et al., 2014), random oversampling (ROS), SMOTE, ADASYN (He et al., 2008), and Tomek links with SMOTE) to address class imbalance in the financial restatement dataset.

We perform classification employing six different choices of classifiers, DT, ANN, NB, RF, BBN and SVM using 10-fold cross validation and test the efficiency of various predictive models using minority class recall value, minority class F-measure and G-mean. We also experiment different ensemble methods (bagging and boosting) with these classifiers

and employ other meta-learning algorithms (stacking and cost-sensitive learning) to improve model performance. Our main findings are as follows:

- ***Cluster-based undersampling:*** We find that various classifiers (e.g. SVM, BBN) show a high success rate in terms of minority class recall value. For example, SVM classifier shows a minority recall value of 96% which is quite encouraging. These models will be useful for the regulators and investors who are mainly focussed on financial restatement detection. However, the ability of these classifiers to detect majority class instances is dismal.
- ***Synthetic oversampling:*** We find that some variations of synthetic oversampling such as 'Tomek Link + SMOTE' and 'ADASYN' show promising results in terms of both minority recall value and G-mean. RF classifier shows minority recall values of 92.6% (InfoGainEval) for 'Tomek Link + SMOTE' and 88.9% (InfoGainEval) for 'ADASYN' techniques, respectively. The corresponding G-mean values are 95.2% (InfoGainEval) and 94.2% (InfoGainEval) for these two oversampling techniques, which show that RF classifier is quite effective in predicting both minority and majority classes.
- ***Meta-learning algorithm:*** We find that, results for RF model improve further with cost-sensitive learning algorithm. The main improvement happens in the form of minority recall value. The RF classifier shows a very successful predictive capability for minority class instances (97.4%) and a very high G-mean value (95.3%) with cost-sensitive learning. However, as expected in cost-sensitive learning algorithm, an improvement in minority class value reduces the predictive capability of majority class instances (91.7%).

- ***Feature selection:*** In this study we have used three feature selection techniques: CfsSubsetEval, InfoGainEval, and FwSelect. While all three techniques lead to somewhat similar results, InfoGainEval feature selection technique yield marginally better results.
- ***Comparison with previous studies:*** To the best of our knowledge, only Kim et al. (2016) perform a data mining study using only pre-financial crisis restatement data. We have expanded their analysis by employing a wider range of classifiers and using various under and oversampling techniques to address class imbalance issue in our dataset. Our synthetic oversampling based results show a very high degree of predictive ability compared to Kim et al.'s (2016) results. Yet, we recognize that Kim et al. (2016) use a different restatement dataset (with pre-crisis restatement cases) and hence a direct comparison of results may not be fully justified.

In summary, the results of this study show a promising outcome and presents reliable predictive models that can guide the investors, regulators and auditors to detect a firm with the intention of financial fraudulent activity or the probability of material errors in financial statements.

## 8.2 Future Work

This study can be extended in different ways. As mentioned in the data mining literature, there is no consensus on the best model that is suitable for different domains and data structure. Therefore, our analysis could be extended to different domains and one can use different datasets and attributes to have a better insight into the suitability of various data

mining techniques. Also, we recognize that extant literature presents various undersampling and oversampling techniques, or custom-built kernel for SVM classifier, which are not examined in this study. One can explore different undersampling and oversampling techniques to address class imbalance issue for the better prediction of financial restatements. It appears that there is no consensus in the literature on how to address this issue, which is quite common in many domain specific datasets.

Also, we can explore other complementary classification techniques in the model building process (e.g. one class learning). In a future study, one can also explore a three-class model (intentional, unintentional and no restatement) with a comprehensive dataset for a better understanding of managerial intent and a firm's financial competency.

Another natural extension would be the inclusion of managerial variables (e.g. CEO characteristics, board characteristics) while developing predictive models. Finance and accounting research shows that managerial characteristics play an important role in corporate actions and decisions. Inclusion of managerial variables may lead to better predictive models.

Finally, we can extent the methodology and research framework of this study to other areas of financial irregularities (e.g. credit card fraud, insurance fraud) and significant financial events such as bankruptcy, mergers and acquisitions, strategic alliances, secondary equity offering, and corporate innovation to name a few.

# REFERENCES


Abbasi, A., Albrecht, C., Vance, A., & Hansen, J. (2012). Metafraud: A meta-learning framework for detecting financial fraud. *Mis Quarterly*, 36(4), 1293–1327.

Agrawal, A. Viktor, H. L. and Paquet, E. (2015). SCUT: Multi-class imbalanced data classification using SMOTE and Cluster-based Undersampling, , *In Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge management,* 226-234.

Akbani, R, Kwek, S. & Jakowicz, N. (2004). Applying support vector machines to imbalanced datasets, *In Proceedings of European Conference of Machine Learning*, 39–50

Albashrawi, M. (2016). Detecting financial fraud using data mining techniques: A decade review from 2004 to 2015. *Journal of Data Science*, 14(3), 553–569.

Aris, N. A., Arif, S. M. M., Othman, R., & Zain, M. M. (2015). Fraudulent financial statement detection using statistical techniques: The case of small medium automotive enterprise. The *Journal of Applied Business Research*, 14(4), 1469–1477.

Ata, H. A., & Seyrek, I. H. (2009). The use of data mining techniques in detecting fraudulent financial statements: An application on manufacturing firms. *The Journal of Faculty of Economics and Administrative Science*, 14(2), 157–170.

Banarescu, A., & Baloi, A. (2015). Preventing and detecting fraud through data analytics in auto insurance field. *Romanian Journal of Economics*, 40, 1(49), 89–114.

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKD Explorations Newsletter - Special Issue on Learning from Imbalanced Datasets, 6(1), 20–29.

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3), 235–255.

Bowen, R., Dutta, S., & Zhu, P. (2017). Financial constraints and accounting restatements. Ottawa, Canada: University of Ottawa Working Paper.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7), 1145–1159.

Brown, L. D., & Caylor, M. L. (2006). Corporate governance and firm valuation. Journal of Accounting and Public Policy, 25, 409–434. Burks, J. (2010). Disciplinary measures in response to restatements after Sarbanes-Oxley. *Journal of Accounting and Public Policy*, 29(3), 195–225.

Bryll, B., Gutierrez-Osuna, R. and Quek, F. (2003). Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern recognition*, 36(6), 1291–1302.

Burns, N., & Kedia, S. (2006). The impact of performance-based compensation on misreporting. *Journal of Financial Economics*, 79, 35–67.

Cecchini, M., Aytug, H., Koehler, G. J., & Pathak, P. (2010). Detecting management fraud in public companies. *Management Science*, 56(7), 1146–1160.

Chakravarthy, J., Haan, E. D., & Rajgopal, S. (2014). Reputation repair after a serious restatement. *The Accounting Review*, 89(4), 1329–1363.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step Data Mining guide. SPSS Inc. Retrieved from https://www.the-modeling-agency.com/crisp-dm.pdf

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.

Chen, S. (2016). Detection of fraudulent financial statements using the hybrid data mining approach. *SpringerPlus*. 5(89). DOI 10.1186/s40064-016-1707-6.

Chen, M. S., Han, J., & Yu, P. S. (1996). Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, *8*(6), 866-883.

Dechow, P. M., Ge, W., Larson, C. R., & Sloan, R. G. (2011). Predicting material accounting misstatements. *Contemporary Accounting Research*, 28(1), 17–82.

Deng, Q. (2010). Detection of fradulent financial statements based on Naive Bayes classifier. *In preceedings of the 5th International Conference on Computer Science & Education*, Hefei, China, Aug 24-27.

Desai, H., Hogan, C., & Wilkins, M. (2006). The reputational penalty for aggressive accounting: Earnings restatements and management turnover. *The Accounting Review*, 81(1), 83–112.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification (2nd ed). John Wiley & Sons. Farber, D. (2005). Restoring trust after fraud: Does corporate governance matter? *The Accounting Review*, 80(2), 539–561.

Elkan, C. (1997). Boosting and Naïve Bayesian learning. In *Technical Report CS97-557*, Dept. Computer Science and Engineering, University of California at San Diego, Sept. 1997.

Fawcett, T. (2006). An Introduction to ROC analysis. *Pattern Recognition Letters*, 27, 861–874.

Fazelpour, A., Khoshgoftaar, T. M., Dittman, D.J. and Napiltano, A. (2016) Investigating the Variation of Ensemble Size on Bagging-Based Classifier Performance in Imbalanced Bioinformatics Datasets. *In proceedings of IEEE 17th International Conference Information Reuse and Integration (IRI)*, 2016. Pittsburgh, PA, USA.

Feroz, E. H., Kwon, T. M., Pastena, V., & Park, K. J. (2000). The efficacy of red flags in predicting the SEC's targets: An artificial neural networks approach. *International Journal of Intelligent Systems in Accounting, Finance, and Management*, 9(3), 145–157.

Files, R., Swanson, E., & Tse, S. (2009). Stealth disclosure of accounting restatements. *The Accounting Review*, 84(5), 1495–1520.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Computer and System Sciences*, 55, 119–139.

GAO. (2002). Financial statement restatements: Trends, market impacts, regulatory responses, and remaining challenges. US General Accounting Office, http://www.gao.gov/special.pubs/gao-06-1079sp/.

Green, B. P., & Calderon, T. G. (1995). Analytical procedures and auditor's capacity to detect fraud. *Accounting Enquiries: A Research Journal*, 5(1), 1–47.

Green, B. P., & Choi, J. H. (1997). Assessing the risk of management fraud through neural network technology. *Auditing*, 16(1), 14.

Griselda, L., Juan, D. O., and Joaquin, A. (2012). Using decision trees to extract decision rules from police reports on road accidents. *Procedia-Social and Behavioral Sciences.* 53, 106–114.

Gupta, R., N. S., & Gill, N. S. (2012). A data mining framework for prevention and detection on financial statement fraud. *International Journal of Computer Applications*, 50(8), 7–14.

Hall, M. (2009). The WEKA data mining software: an update. *SIGKDD Explor.* 11, 10–18.

Han, J., Kamber, M., & Pei, J. (2012). Data mining: Concepts and techniques: Concepts and techniques (3rd ed.). MA: Morgan Kaufmann: Elsevier.

He, H., Bai, Y., Garcia, E. A. and Li, S. "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning", *In Proceedings of the International Joint Conference on Neural Networks*, 1322-1328.

He, H. and Garcia, E. A (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284.

Hennes, K. M., Leone, A. J., & Miller, B. P. (2008). The importance of distinguishing errors from irregularities in restatement research: The case of restatements and ceo/cfo turnover. *The Accounting Review*, 83(6), 1487–1519.

Hennes, K. M., Leone, A. J., & Miller, B. P. (2014). Determinants and market consequences of auditor dismissals after accounting restatements. *The Accounting Review*, 89(3), 1051–1082.

Holte, A., Holte, P. R., Acker, L. & Porter, B (1989). Concept learning and the problem of small disjuncts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*.

Huang, Y., & Scholz, S. (2012). Evidence on the association between financial restatements and auditor resignations. Accounting Horizons, 26(3), 439–464.

Japkowicz, N., & Shah, M. (2011). Evaluating learning algorithms: A classification perspective. New York, NY: Cambridge University Press 343.

Jo, T. and Japkowicz, N. (2004). Class Imbalances versus Small Disjuncts. *SIGKDD Explorations*, 6(1*),* June 2004, 40–49

Kaufman, L. and P.J. Rousseeuw, (1990). Finding Groups in Data. John Wiley & Sons, New York.

Kaminski, K., Wetzel, T., & Guan, L. (2004). Can financial ratios detect fraudulent financial reporting?, Managerial Auditing Journal, 19(1), 15–28.

Karpoff, J. M., Lee, D. S., & Martin, G. S. (2008). The cost to firms of cooking the books. *Journal of Financial and Quantitative Analysis*, 43(3), 581–612.

Karpoff, J. M., Koester, A., Lee, D. S., & Martin, G. S. (2014). Database Challenges in Financial Misconduct Research, University of Washington Working Paper. WA, USA: Seattle.

Kashani, A. T. and Mohaymany, A. S. (2011). Analysis of the traffic injury severity on two-lane, two-way rural roads based on classification tree models. *Safety Science.* 49. 1314–1320.

Kim, Y. J., Baik, B., & Cho, S. (2016). Detecting financial misstatements with fraud intention using multi-class cost-sensitive learning. *Expert Systems with Applications*, 62, 32–43.

Kirkos, E., Spathis, C., & Manolopoulos, Y. (2007). Data mining techniques for the detection of fraudulent financial statements. *Expert Systems with Applications*, 32(4), 995–1003.

Kotsiantis, S., Koumanakos, E., Tzelepis, D., & Tampakas, V. (2006). Forecasting fraudulent financial statements using data mining. *International Journal of Computational Intelligence*, 3(2), 104–110.

Kotu, V., Deshpande, B., (2015). Predictive analytics and data mining: Concepts and practice with RapidMiner (3rd ed.). MA: Morgan Kaufmann: Elsevier.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Prog. Artificial Intelligence.* 5. 221–232.

Kubat, M., Holte, R., & Matwin, S. (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*, 30, 195–215.

Longadge, R., Dongre, S.S. & Malik, L. (2013). Class imbalance problem in data mining: review. *International Journal of Computer Science and Network.* 2(1).

Lee, S. (2000). Noisy Replication in Skewed Binary Classification. *Computational Statistics and Data Analysis*, 34(2), 165–191.

Li, B., Yu, J., Zhang, J., & Ke, B. (2015). Detecting accounting frauds in publicly traded U.S. firms: A machine learning approach. *In Proceedings of the 7th asian conference on machine learning*.

Lin, C. C., Chiu, A. A., Huang, S. Y., and Yen, D. C. (2015). Detecting the financial statement fraud: The analysis of the differences between data mining techniques and experts' judgments. *Knowledge-Based Systems*, 89, 459–470.

Lin, W. C., Tsai, C. F., Hu, Y. H and Jhang, J. S. (2017). Clustering based undersampling in class imbalance data. *Information Sciences*. 409.17–26.

Liu, C., Chan, Y, Kazmi, S. H. A. and Fu, H. (2015). Financial fraud detection model: based on Random Forest. *International Journal of Economics and Finance*. 7(7). 178-188.

Lowd, D., & Domingos, P. (2005). Naive Bayes models for probability estimation. *In the Proceedings of the 22nd international conference on Machine learning*. Bonn, Germany. 529-536.

Mitchell. T, (1997). Machine Learning. McGraw-Hill, New York, 1997.

Moepya, S. O., Nelwamondo, F. V. and Van Der Walt, C. (2014). A support vector machine to detect financial statement fraud in South Africa: A first look. *Intelligent Information and Dadabase Systems*, 8398(2). 42-51.

Ngai, E., Hu, Y., Wong, Y., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, 50(3), 559–569.

Palmrose, Z.-V., Richardson, V., & Scholz, S. (2004). Determinants of market reactions to restatement announcements. *Journal of Accounting and Economics*, 37(1), 59–89.

Perols, J. (2011). Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing: A Journal of Practice and Theory*, 30(2), 19–50.

Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. Computing Research Repository, abs/1009.6119, 1–14.

Quinlan, J. R. (1996). Bagging, boosting, and C4.5. *In proceedings of. 1996 National. Conference of Artificial Intelligence (AAAI'96)*, 1, 725–730.

Ravisankar, P., Ravi, V., Rao, G. R., & Bose, I. (2011). Detection of financial statement fraud and feature selection using data mining techniques. *Decision Support System*, 50, 491–500.

Rajarajeswari, S. and Somasundaram, K. (2012). An Empirical study on feature selection for data classification. *International Journal of Advanced Computer Research.* 2(3), 111–115.

Sanchez, D., Vila, M. A., Cerda, L., & Serrano, J. M. (2009). Association rules applied to credit card fraud detection. *Expert Systems with Applications,* 36, 3630–3640.

Roobaert, D., Karakoulas, G. and Chawla, N. V.(2006): Information Gain, Correlation and Support Vector Machines*, StudFuzz* **207**, 463–470

Sobhani, P., Viktor, H., Matwin, S. (2014). Learning from imbalanced data using ensemble methods and cluster-based undersampling. In *PKDD nfMCD 2013: Lecture Notes in Computer Science*, 8983, Springer, 38-49.

Sharma, A., & Panigrahi, P. K. (2012). A review of financial accounting fraud detec- tion based on data mining techniques. *International Journal of Computer Applica- tions,* 39 (1), 37–47.

Srinivasan, S. (2005). Consequences of financial reporting failure for outside direc- tors: Evidence from accounting restatements and audit committee members. *Journal of Accounting Research,* 43 (2), 291–334.

Su, C. and Hsiao, Y. (2007). An evaluation of robustness of MTS for imbalance data. *IEEE Transaction on Knowledge and Data Engineering,* 19. 1321-1332.

Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science, 240*, 1285–1293.

Thiruvadi, S., & Patel, S. C. (2011). Survey of data-mining techniques used in Fraud detection and prevention. *Information Technology Journal,* 10, 710–716.

Tomek, I. (1976) An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(6), 448-452.

Turner, L., Dietrich, J. R., Anderson, K., & Bailey, A. J. (2001). Accounting restatements. Ohio State University Working paper (February 2001).

Vapnik, V., & Cortes, C. (1995). Support-vector networks. *Machine Learning,* 20 (3), 273–297.

Wallace, B. C., Small, K., Brodley, C. E., & Trikalinos, T. A. (2011). Class imbalance, redux. In *Proceedings of the 11th international conference on data mining,* (pp. 754–763). IEEE.

Weatherford, M. (2002). Mining for fraud. *IEEE Intelligent Systems,* 17 (4), 4–6.

Yen, S. J and Lee, Y. S. (2009). Cluster based undersampling approaches for imbalanced data distributions. *Expert Systems with Applications.* 36. 5718–5727.

Zhou, W., & Kapoor, G. (2011). Detecting evolutionary financial statement fraud. *Decision Support Systems,* 50 (3), 570–575

# APPENDIX A

Table 35. Summary of Data mining studies in financial restatement

| Sl. No | Author | Journal and Year of publication | Paper | Data sample | Database used | Main findings |
|---|---|---|---|---|---|---|
| 1 | Green and Choi | Auditing: A Journal of Practice and Theory, 1997 | Assessing the risk of management fraud through neural-network technology | 1982 - 1990 | SEC/AAER, Compustat | Green and Choi develop neural network fraud classification model employing endogenous financial data |
| 2 | Feroz et al. | International Journal of Intelligent Systems in Accounting, Finance and Management, 2000 | The Efficacy of Red Flags in Predicting the SEC's Targets | | SEC/AAER | The authors use ANN and logistic regression to predict possible fraudster and accounting manipulators. The ANN models classify the membership in target (investigated) versus control (non-investigated) firms with an average accuracy of 81%. |
| 3 | Bolton et al. | Statistical Science, 2002 | Statistical Fraud Detection: A Review | - | - | Bolton and Hand review statistical and data mining techniques used in several subgroups within fraud detection, such as, telecommunications fraud, credit card fraud, medical and scientific fraud, and as well as in other domains such as money laundering and intrusion detection. The authors describe the |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | available tools and techniques available for statistical fraud detection. |
| 4 | Kotsiantis et al. | International Journal of Computational Intelligence, 2006 | Forecasting Fraudulent Financial Statements using data Mining | 2001 - 2002 | Athens Stock Exchange | Kotsiantis et al. (2006) examine the performance of Neural Networks, Decision Tree, Bayesian Belief Network and K-Nearest Neighbour to detect financial fraud. They proposed and implemented a hybrid decision support system to identify the factors which would be used to determine the probability of fraudulent financial statements. According to the authors, the proposed stacking variant method achieves better performance than other described methods (BestCV, Grading, and Voting). |
| 5 | Kirkos et al. | Expert Systems with Applications, 2007 | Data Mining techniques for the detection of fraudulent financial statements | - | Athens financial and taxation databases and Athens Stock Exchange | Kirkos et al. explore the performance of Neural Networks, Decision Tree and Bayesian Belief Network to detect financial statement fraud. Among these three models, Bayesian Belief Network showed the best performance. |
| 6 | Ata et al. | The Journal of Faculty of Economics and Administrative Sciences, 2009 | The use of data mining techniques in detecting fraudulent financial statements: An application on | 2005 | Istanbul Stock Exchange | Ata et al. use decision tree (DT) and neural networks for the detection of fraudulent financial statements. The authors selected the variables that are related to liquidity, financial situation and the profitability of the firms in the selected dataset. The authors |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | manufacturing firms | | | conclude that neural network did better prediction of the fraudulent activities than decision tree method. |
| 7 | Cecchini et al. | Management Science, 2010 | Detecting management fraud in public companies | 1991 - 2003 | SEC/ AAER | The authors provide a methodology based on support vector machine and domain specific kernel to implicitly map the financial attributes to ratio and year over year changes of the ratio. Their methodology correctly achieve 80% of fraud cases using publicly available data and also correctly label 90.6% of the non-fraud cases with the same model. Their experiment shows that their model outperforms F-score model by Dechow et al. (2011) and the neural network model by Green and Choi (1997). |
| 8 | Ravisankar et al. | Decision Support Systems, 2011 | Detection of financial statement fraud and feature selection using data mining techniques | | Various Chinese stock exchange | The authors apply logistic regression, SVM, Genetic Programing (GP), Probabilistic Neural Network (PNN), multilayer feed forward Neural Network (MLFF) and group method of data handling (GMDH) to predict the financial fraud. The authors use the t-statistics to achieve feature selection on the dataset. They conducted the experiment with or without the feature selection on different classifiers. Their study results show that PNN has outperformed all the other techniques in both cases. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 9 | Zhou et al. | Decision Support Systems, 2011 | Detecting evolutionary financial statement fraud | - | - | In order to improve the efficiency of the existing data mining framework, Zhou and Kapoor propose an adaptive learning framework. According to Zhou and Kapoor, one way to improve the efficiency of data mining technique is to use adaptive learning framework (ALF). This framework proposes the inclusion of many exogenous variables, governance variables and management choice variables that will contribute towards the adaptive nature of financial fraud detection techniques. |
| 10 | Dechow et al. | Contemporary Accounting Research, 2011 | Predicting material accounting misstatements | 1982 - 2005 | SEC Accounting and Auditing Enforcement Releases (AAER) | Dechow et al. develop a comprehensive database of financial misstatements and develop F- score model to predict misstatement. |
| 11 | Perols | Auditing: A Journal of Practice and Theory, 2011 | Financial Statement Fraud Detection: An Analysis of Statistical and Machine Learning Algorithms | 1998 - 2005 | SEC/ AAER | The authors compare the performance of six popular statistical and machine learning models in detecting financial statement fraud under different assumptions of misclassification costs and ratios of fraud firms to non-fraud firms. They show that logistic regression and SVM outperform ANN, bagging, C4.5, and stacking. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 12 | Ngai et al. | Decision Support Systems, 2011 | The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature | - | - | Ngai et al. present a comprehensive literature review on the data mining techniques. They show that Logistic Regression model, NN, the Bayesian belief Network and DT are the main data mining techniques that are commonly used in financial fraud detection. |
| 13 | Thiruvadi et al. | Information Technology Journal, 2011 | Survey of Data-mining Techniques used in Fraud Detect ion and Prevention | - | - | Thiruvadi and Patel classify the frauds into management fraud, customer fraud, network fraud and computer-based fraud and discuss data mining techniques to fraud detection and prevention. |
| 14 | Sharma et al | International Journal of Computer Applications, 2012 | A review of financial accounting fraud detection based on data mining techniques | - | - | The authors present a comprehensive review of the exiting literatures on financial fraud detection using the data mining techniques. Their finding shows that regression analysis, DT, Bayesian network, NN are the most commonly used data mining techniques in fraud detection. |
| 15 | Abbasi et al. | MIS Quarterly, 2012 | MetaFraud: a meta learning framework | 1995 - 2010 | SEC Accounting and Auditing Enforcement | They propose a meta-learning framework to enhance financial model fraud detection. This framework integrates business intelligence methods into a meta-learning artifact. This |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | for detecting financial fraud | | Releases (AAER) | framework uses organizational and industry contextual information, quarterly and annual data and more robust classification methods using stacked generalization and adaptive learning. |
| 16 | Lin et al. | Knowledge-Based Systems, 2015 | Detecting the financial statement fraud: The analysis of the differences between data mining techniques and experts' judgments | 1998 - 2010 | Taiwan Securities and Futures Bureau and the group litigation cases announced by the Securities and Futures Investors Protection Center | The authors employ logistic regression, DT (CART) and ANN to detect financial statement fraud. The authors concluded that ANN and CART provide better accuracy (over 90%) than the logistic model. |
| 17 | Aris et al. | The Journal of Applied Business Research, 2015 | Fraudulent Financial Statement Detection Using Statistical Techniques: The Case Of | 2010 - 2012 | A small medium automotive company in Malaysia | The authors evaluate the possibility of fraudulent financial statements using statistical analysis, such as, financial ratio, Altman Z-score and Beneish model. They analyzed the financial statement data in a small medium automotive company in Malaysia and suggests further investigation by the management. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | Small Medium Automotive Enterprise | | | |
| 18 | Albashrawi | Journal of Data Science, 2016 | Detecting Financial Fraud using Data Mining Techniques: A Decade Review from 2004 to 2015 | - | - | Albashrawi present a recent comprehensive literature survey based on 40 articles on data mining applications in corporate fraud, such as, financial statement fraud, automobile insurance fraud, accounting fraud, health insurance fraud and credit card fraud. Their detail study suggest that logistic regression, decision tree, SVM, NN and Bayesian networks have been widely used to detect financial fraud. However, these techniques may not always yield best classification results. |
| 19 | Li et al. | Proceedings of the 7th Asian Conference on Machine Learning, Journal of Machine Learning Research Workshops and Conference Proceedings, 2015 | Detecting Accounting Frauds in Publicly Traded U.S. Firms: A Machine Learning Approach | 1982 - 2010 | AAER, Compustat | The authors evaluate machine learning techniques in detecting accounting fraud. They employ linear and non-linear SVM and ensemble methods to handle class imbalance problem. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 20 | Kim et al. | Expert Systems with Applications, 2016 | Detecting financial misstatements with fraud intention using multi-class cost sensitive learning | 1997 - 2005 | GAO, Compustat | Kim et al. develop multi-class financial misstatement detection model to detect the misstatements with fraud intension. They investigate three groups; 1) intentional restatement or fraud, 2) un-intentional restatement or error, and 3) no restatement. The authors develop three multi-class classifier model, multinomial logistic regression, support vector machine and Bayesian networks, using cost-sensitive learning to predict fraudulent intention in financial misstatements. |

# APPENDIX B

Table 36. List of attributes

| Attribute | Description | Reference |
|---|---|---|
| dechw_ab_chg_orderbklg | (OB - OBt-1) / OBt-1 - (SALE - SALEt-1) / SALEt-1 | Dechow et al.(2011) |
| dechw_actual_issuance | IF SSTK>0 or DLTIS>0 THEN 1 ELSE 0 | Dechow et al.(2011) |
| dechw_book_mkt_value | CEQ / (CSHO * PRCC_F) | Dechow et al.(2011) |
| dechw_rsst_accru | RSST Accruals = (ΔWC + ΔNCO + ΔFIN)/Average total assets, where WC = (ACT - CHE) - (LCT - DLC); NCO = (AT - ACT - IVAO) - (LT - LCT - DLTT); FIN = (IVST + IVAO) - (DLTT + DLC + PSTK) | Dechow et al.(2011) |
| dechw_chg_free_cf | (IB - RSST Accruals) / Average total assets - (IBt-1 - RSST Accrualst-1) /Average total assetst-1 | Dechow et al.(2011) |
| dechw_chg_invt | (INVT- INVTt-1)/Average total assets | Dechow et al.(2011) |
| dechw_chg_oper_lease | ((MRC1/1.1+ MRC2/1.1^2+ MRC3/1.1^3+ MRC4/1.1^4+ MRC5/1.1^5) - (MRC1t-1/1.1+ MRC2t-1/1.1^2+ MRC3t-1/1.1^3+ MRC4 -1/1.1^4+ MRC5t-1/1.1^5) )/Average total assets | Dechow et al.(2011) |
| dechw_chg_receiv | (RECT- RECTt-1)/Average total assets | Dechow et al.(2011) |
| dechw_chg_roa | IB / Average total assets - IBt-1 / Average total assetst-1 | Dechow et al.(2011) |
| dechw_def_taxexp | TXDI / ATt-1 | Dechow et al.(2011) |
| dechw_demand_fin | IF ((OANCF-(CAPXt-3+CAPXt-2+ CAPXt-1)/ 3) /(ACT) < -0.5 THEN 1 ELSE 0 | Dechow et al.(2011) |
| dechw_earn_price | IB / (CSHO x PRCC_F) | Dechow et al.(2011) |
| dechw_exist_oplease | IF (MRC1 > 0 OR MRC2 > 0 OR MRC3 > 0 OR MRC4 > 0 OR MRC5 > 0 THEN 1 ELSE 0 | Dechow et al.(2011) |
| dechw_lev_fin_raised | FINCF / Average total assets | Dechow et al.(2011) |
| dechw_leverage | DLTT / AT | Dechow et al.(2011) |

| | | |
|---|---|---|
| dechw_per_chg_cashmargin | ((1-(COGS+(INVT-INVTt-1))/(SALE-(RECT-RECTt-1)))-(1-(COGSt-1+(INVTt-1-INVTt-2))/(SALEt-1-(RECTt-1-RECTt-2)))) /(1-(COGSt-1+(INVTt-1-INVTt-2))/(SALEt-1-(RECTt-1-RECTt-2))) | Dechow et al.(2011) |
| dechw_per_chg_cashsale | ((SALE - (RECT - RECTt-1)) - (SALEt-1 - (RECTt-1 - RECTt-2))) /(SALEt-1 - (RECTt-1 - RECTt-2)) | Dechow et al.(2011) |
| dechw_softasset | (AT-PPENT-CHE)/Average total assets | Dechow et al.(2011) |
| dechw_fm_ind_unexp_em_prod | FIRM((SALE/EMP - SALEt-1/EMPt-1)/(SALEt-1/EMPt-1)) - INDUSTRY((SALE/EMP - SALEt-1/EMPt-1)/(SALE t-1/EMPt-1)) | Dechow et al.(2011) |
| dechw_wc_accrual | (((ACT - ACTt-1) - (CHE - CHEt-1)) - ((LCT - LCTt-1) - (DLC - DLCt-1) - (TXP - TXPt-1)) - DP)/Average total assets | Dechow et al.(2011) |
| perols1 | Accounts receivable to sales | Perols (2011) |
| perols2 | Accounts receivable to total assets | Perols (2011), Ravisankar et al. (2011) |
| perols3 | Allowance for doubtful accounts | Perols (2011) |
| perols4 | Allowance for doubtful accounts to accounts receivable | Perols (2011) |
| perols5 | Allowance for doubtful accounts to net sales | Perols (2011) |
| perols6 | Altman Z score | Perols (2011) |
| perols7 | Big four auditor | Perols (2011) |
| perols8 | Current minus prior year inventory to sales | Perols (2011) |
| perols9 | Days in receivables index | Perols (2011) |
| perols10 | Debt to equity | Perols (2011), Ravisankar et al. (2011) |
| perols11 | Declining cash sales dummy | Perols (2011) |
| perols12 | Fixed assets to total assets | Perols (2011) |
| perols13 | Four year geometric sales growth rate | Perols (2011) |
| perols14 | Gross margin | Perols (2011) |
| perols15 | Holding period return in the violation period | Perols (2011) |
| perols16 | Industry ROE minus firm ROE | Perols (2011) |
| perols17 | Inventory to sales | Perols (2011) |
| perols18 | Net sales | Perols (2011) |

| | | |
|---|---|---|
| perols19 | Positive accruals dummy | Perols (2011) |
| perols20 | Prior year ROA to total assets current year | Perols (2011) |
| perols21 | Property plant and equipment to total assets | Perols (2011) |
| perols22 | Sales to total assets | Perols (2011) |
| perols24 | Times interest earned | Perols (2011) |
| perols25 | Total accruals to total assets | Perols (2011) |
| perols26 | Total debt to total assets | Perols (2011) |
| perols27 | Total discretionary accrual | Perols (2011) |
| perols28 | Value of issued securities to market value | Perols (2011) |
| perols29 | Whether accounts receivable>1.1 of last year's | Perols (2011) |
| perols31 | Whether gross margin percent > 1.1 of last year's | Perols (2011) |
| perols32 | Whether LIFO | Perols (2011) |
| perols33 | Whether new securities were issued | Perols (2011) |
| perols34 | Whether SIC code larger (smaller) than 2999 (4000) | Perols (2011) |
| lag1_mkt_excess_ret | Lag of (firm's yearly stock returns minus stock index's (i.e. S&P500) yearly stock returns) | Dechow et al.(2011) |
| long_term_debt | Total long term debt | In light of Ravisankar et al. (2011) |
| total_asset | Total asset | Ravisankar et al. (2011) |
| inv_total_asset | Ratio: Inventory to Total asset | Ravisankar et al. (2011) |
| accounts_recev | Account receivables | Green and Choi (1997); Lin et al. (2015); Ravisankar et al. (2011) |
| accnt_rec_sales | Ratio: Account receivable to Total sales | Green and Choi (1997); Feroz et al. (2000); Lin et al. (2015); Kaminski et al. (2004) |
| current_ratio | Ratio: Current asset to Current liability | Ravisankar et al. (2011) |
| inv_current_liab | Ratio: Inventory to Current liability | Ravisankar et al. (2011) |
| tob_q | Tobin's Q; represents the market value to book value ratio of the firm | Brown and Caylor (2006) |
| acc_recev_to_lag | Ratio: Account receivable to Last year's account receivables | Ravisankar et al. (2011) |

| | | |
|---|---|---|
| acc_payable_to_lag | Ratio: Account payable to Last year's account payable | In light of Ravisankar et al. (2011) |
| total_asset_to_lag | Ratio: Total asset to Last year's total asset | Ravisankar et al. (2011) |
| sale_to_lag | Ratio: Total sales to Last year's total sales | In light of Ravisankar et al. (2011) |
| fyear | Fiscal year; represents the respective financial year of the firm | Dechow et al.(2011) |
| trad_current_asset | Current Assets (e.g. inventory, accounts receivables) | In light of Ravisankar et al. (2011) |
| trad_current_liab | Current Liability (e.g. short term payment obligation) | In light of Ravisankar et al. (2011) |
| trad_inventory | Total inventory | In light of Ravisankar et al. (2011) |
| trad_current_total_asset | Ratio: Current asset to Total asset | In light of Ravisankar et al. (2011) |
| trad_accounts_payable | Account payables | In light of Ravisankar et al. (2011) |
| compu_ff48 | Fama-french 48 sectors | Conventional financial/accounting variable |
| roe_higher_indavg48 | Dummy = 1 if firm return on equity (ROE) >= industry average ROE | Conventional financial/accounting variable |
| roa_ebitda3yr_higher_indavg48 | Dummy = 1 if firm Return on Asset to EBITDA (3-yr average) ratio is >= Industry Return on Asset to EBITDA (3-yr average) ratio; EBITDA = Earnings before interest, tax, depreciation & amortization | Conventional financial/accounting variable |
| tsr_crsp_higher_indavg48 | Dummy = 1 if firm total shareholder return (TSR) >= Industry average total shareholder return (TSR); where, TSR = buy and hold return for last 12-monthly returns. | Conventional financial/accounting variable |
| tsr_crsp3yr_higher_indavg48 | Dummy = 1 if firm total shareholder return (TSR) - 3-yr avg >= Industry average total shareholder return (TSR) - 3-yr avg; where, TSR = buy and hold return for last 12-monthly returns. | Conventional financial/accounting variable |

| | | |
|---|---|---|
| lnsize_tasset | Log of total asset | Conventional financial/accounting variable |
| lnsize_tasset_3yr_avg | Log of total asset - 3-yr average | Conventional financial/accounting variable |
| lnsize_mktval | Log of market value of equity | Conventional financial/accounting variable |
| lnsize_sale | Log of total sale | Conventional financial/accounting variable |
| mkt_excess_ret | firm's yearly stock returns minus stock index's (i.e. S&P500) yearly stock returns | Conventional financial/accounting variable |
| std_bhret_48m_incl_curr | Standard deviation of last 48 months stock returns (denotes risk) | Conventional financial/accounting variable |
| std_bhret_60m_incl_curr | Standard deviation of last 60 months stock returns (denotes risk) | Conventional financial/accounting variable |
| pct_change_ppe_net_0_1 | Percentage change in plant, property and equipment (PPE) net | Conventional financial/accounting variable |
| pct_change_ppe_net_3yr_avg | 3 year arithmetic average of yearly percentage change in PPE | Conventional financial/accounting variable |
| pct_change_long_ast_0_1 | Yearly percentage change in long term asset | Conventional financial/accounting variable |
| pct_change_long_ast_3yr_avg | 3 year arithmetic average of percentage change in long term asset | Conventional financial/accounting variable |
| pct_change_capex_0_1 | Percentage change capex between current and one lagged year | Conventional financial/accounting variable |
| pct_change_capex_3yr_avg | 3 year arithmetic average of percentage change in capital expenditure (capex) | Conventional financial/accounting variable |
| trad_mkt_val_equity | Market value of equity - calculated as: stock price * outstanding shares | Conventional financial/accounting variable |
| trad_ltd_to_equity | Ratio: Long term debt to  common equity (CEQ) | Conventional financial/accounting variable |

| | | |
|---|---|---|
| trad_ltd_to_asset | Ratio: Total long term debt to Total asset | Conventional financial/accounting variable |
| trad_ni | Net Income | Conventional financial/accounting variable |
| trad_roa | Return on asset (calculated as: Net Income / Total asset) | Conventional financial/accounting variable |
| trad_roa_ebit | Return on asset (calculated as: EBIT / Total asset); EBIT = Earnings before interest & tax | Conventional financial/accounting variable |
| trad_roa_ebitda | Return on asset (calculated as: EBITDA / Total asset); EBITDA = Earnings before interest, tax, depreciation & amortization | Conventional financial/accounting variable |
| trad_roe | Return on equity (calculated as: Net income / total market value of equity) | Conventional financial/accounting variable |
| trad_r_and_d | R&D expenditure | Conventional financial/accounting variable |
| trad_sale | Total sales | Conventional financial/accounting variable |
| trad_cash | Total cash holding in a year | Conventional financial/accounting variable |
| trad_cash_shrt_invest | Total cash holding and short term investment in a year | Conventional financial/accounting variable |
| trad_cash_to_tasset | Ratio: Total cash holding to Total asset | Conventional financial/accounting variable |
| trad_accnt_pay_sales | Ratio: Account payable to Total sales | Conventional financial/accounting variable |
| trad_intangible_assets | Intangible assets (e.g. goodwill, patent) | Conventional financial/accounting variable |
| trad_intan_total_asset | Ratio: Intangible asset to Total assets | Conventional financial/accounting variable |
| trad_ppe_gross | Gross value of Plant, Property and Equipment (Gross PPE) | Conventional financial/accounting variable |

| | | |
|---|---|---|
| trad_ppe_net | Net value of Plant, Property and Equipment (Net PPE) | Conventional financial/accounting variable |
| trad_ppe_net_total_asset | Ratio: Net value of Plant, Property and Equipment (Net PPE) to Total asset | Conventional financial/accounting variable |
| trad_auditor_code | auditor identification | Conventional financial/accounting variable |
| trad_ocf_tasset_new | Ratio: Operating cash flow to Total asset | Conventional financial/accounting variable |
| trad_ocf_mequity_new | Ratio: Operating cash flow to Total market value of equity | Conventional financial/accounting variable |
| trad_mkt_to_bookvalue | Ratio: market value of common equity to book value of common equity | Conventional financial/accounting variable |
| trad_div_total_asset | Ratio: Total dividend to Total asset | Conventional financial/accounting variable |
| trad_hfi | Herfindahl index; represents the market concentration in that industry | Conventional financial/accounting variable |
| trad_lag1_net_income | Net income of last year (lagged time period) | Conventional financial/accounting variable |
| trad_lag1_total_asset | Total asset of last year (lagged time period) | Conventional financial/accounting variable |

# APPENDIX C

Table 37: List of reduced attributes by CfsSubset selection

| Attribute | Description |
|---|---|
| dechw_actual_issuance | IF SSTK>0 or DLTIS>0 THEN 1 ELSE 0 |
| dechw_book_mkt_value | CEQ / (CSHO * PRCC_F) |
| dechw_rsst_accru | RSST Accruals = (ΔWC + ΔNCO + ΔFIN)/Average total assets, where WC = (ACT - CHE) - (LCT - DLC); NCO = (AT - ACT - IVAO) - (LT - LCT - DLTT); FIN = (IVST + IVAO) - (DLTT + DLC + PSTK) |
| dechw_chg_invt | (INVT- INVTt-1)/Average total assets |
| dechw_chg_oper_lease | ((MRC1/1.1+ MRC2/1.1^2+ MRC3/1.1^3+ MRC4/1.1^4+ MRC5/1.1^5) - (MRC1t-1/1.1+ MRC2t-1/1.1^2+ MRC3t-1/1.1^3+ MRC4 -1/1.1^4+ MRC5t-1/1.1^5) )/Average total assets |
| dechw_def_taxexp | TXDI / ATt-1 |
| dechw_earn_price | IB / (CSHO x PRCC_F) |
| dechw_softasset | (AT-PPENT-CHE)/Average total assets |
| perols3 | Allowance for doubtful accounts |
| perols12 | Fixed assets to total assets |
| perols13 | Four year geometric sales growth rate |
| perols14 | Gross margin |
| lag1_mkt_excess_ret | Lag of (firm's yearly stock returns minus stock index's (i.e. S&P500) yearly stock returns) |
| mkt_excess_ret | firm's yearly stock returns minus stock index's (i.e. S&P500) yearly stock returns |
| std_bhret_60m_incl_curr | Standard deviation of last 60 months stock returns (denotes risk) |
| total_asset_to_lag | Ratio: Total asset to Last year's total asset |
| pct_change_ppe_net_0_1 | Percentage change in plant, property and equipment (PPE) net |
| pct_change_ppe_net_3yr_avg | 3 year arithmetic average of yearly percentage change in PPE |
| pct_change_long_ast_0_1 | Yearly percentage change in long term asset |
| trad_current_asset | Current Assets (e.g. inventory, accounts receivables) |
| trad_accnt_pay_sales | Ratio: Account payable to Total sales |
| trad_div_total_asset | Ratio: Total dividend to Total asset |

| | |
|---|---|
| trad_hfi | Herfindahl index; represents the market concentration in that industry |
| trad_lag1_net_income | Net income of last year (lagged time period) |
| trad_lag1_total_asset | Total asset of last year (lagged time period) |

# APPENDIX D

Table 38: List of attributes for InfoGain feature selection

| Attribute | Description |
|---|---|
| dechw_chg_free_cf | (IB - RSST Accruals) / Average total assets - (IBt-1 - RSST Accrualst-1) /Average total assetst-1 |
| dechw_chg_oper_lease | ((MRC1/1.1+ MRC2/1.1^2+ MRC3/1.1^3+ MRC4/1.1^4+ MRC5/1.1^5) - (MRC1t-1/1.1+ MRC2t-1/1.1^2+ MRC3t-1/1.1^3+ MRC4 -1/1.1^4+ MRC5t-1/1.1^5) )/Average total assets |
| dechw_chg_roa | IB / Average total assets - IBt-1 / Average total assetst-1 |
| dechw_def_taxexp | TXDI / ATt-1 |
| dechw_earn_price | IB / (CSHO x PRCC_F) |
| dechw_exist_oplease | IF (MRC1 > 0 OR MRC2 > 0 OR MRC3 > 0 OR MRC4 > 0 OR MRC5 > 0 THEN 1 ELSE 0 |
| dechw_lev_fin_raised | FINCF / Average total assets |
| dechw_leverage | DLTT / AT |
| dechw_softasset | (AT-PPENT-CHE)/Average total assets |
| perols2 | Accounts receivable to total assets |
| perols3 | Allowance for doubtful accounts |
| perols5 | Allowance for doubtful accounts to net sales |
| perols12 | Fixed assets to total assets |
| perols21 | Property plant and equipment to total assets |
| perols22 | Sales to total assets |
| perols24 | Times interest earned |
| lag1_mkt_excess_ret | Lag of (firm's yearly stock returns minus stock index's (i.e. S&P500) yearly stock returns) |
| trad_current_liab | Current Liability (e.g. short term payment obligation) |
| current_ratio | Ratio: Current asset to Current liability |
| trad_accounts_payable | Account payables |
| trad_roa_ebitda | Return on asset (calculated as: EBITDA / Total asset); EBITDA = Earnings before interest, tax, depreciation & amortization |
| trad_roe | Return on equity (calculated as: Net income / total market value of equity) |
| trad_sale | Total sales |
| lnsize_sale | Log of total sale |
| mkt_excess_ret | firm's yearly stock returns minus stock index's (i.e. S&P500) yearly stock returns |

| | |
|---|---|
| trad_roe | Return on equity (calculated as: Net income / total market value of equity) |
| trad_accnt_pay_sales | Ratio: Account payable to Total sales |
| trad_div_total_asset | Ratio: Total dividend to Total asset |
| std_bhret_60m_incl_curr | Standard deviation of last 60 months stock returns (denotes risk) |
| pct_change_ppe_net_0_1 | Percentage change in plant, property and equipment (PPE) net |
| pct_change_ppe_net_3yr_avg | 3 year arithmetic average of yearly percentage change in PPE |
| pct_change_long_ast_0_1 | Yearly percentage change in long term asset |
| pct_change_long_ast_3yr_avg | 3 year arithmetic average of percentage change in long term asset |
| sale_to_lag | Ratio: Total sales to Last year's total sales |
| trad_hfi | Herfindahl index; represents the market concentration in that industry |
| trad_lag1_net_income | Net income of last year (lagged time period) |
| total_asset_to_lag | Ratio: Total asset to Last year's total asset |

# APPENDIX E

Table 39: List of reduced attributes after stepwise forward selection

| Attribute | Description |
|---|---|
| roa_ebitda3yr_higher_indavg48 | Dummy = 1 if firm Return on Asset to EBITDA (3-yr average) ratio is >= Industry Return on Asset to EBITDA (3-yr average) ratio; EBITDA = Earnings before interest, tax, depreciation & amortization |
| std_bhret_60m_incl_curr | St. dev of last 60 months of return (denotes risk) |
| dechw_actual_issuance | IF SSTK>0 or DLTIS>0 THEN 1 ELSE 0 |
| dechw_chg_oper_lease | ((MRC1/1.1+ MRC2/1.1^2+ MRC3/1.1^3+ MRC4/1.1^4+ MRC5/1.1^5) - (MRC1t-1/1.1+ MRC2t-1/1.1^2+ MRC3t-1/1.1^3+ MRC4 -1/1.1^4+ MRC5t-1/1.1^5) )/ Average total assets |
| dechw_leverage | DLTT / AT |
| dechw_per_chg_cashmargin | ((1-(COGS+(INVT-INVTt-1))/(SALE-(RECT-RECTt-1)))-(1-(COGSt-1+(INVTt-1-INVTt-2))/(SALEt-1-(RECTt-1-RECTt-2)))) / (1-(COGSt-1+(INVTt-1-INVTt-2))/(SALEt-1-(RECTt-1-RECTt-2))) |
| dechw_softasset | (AT-PPENT-CHE)/Average total assets |
| trad_div_total_asset | Ratio: Total dividend to Total asset |
| trad_lag1_net_income | Net income of last year (lagged time period) |
| perols12 | Fixed assets to total assets |
| perols13 | Four year geometric sales growth rate |
| perols20 | Prior year ROA to total assets current year |
| perols21 | Property plant and equipment to total assets |
| perols31 | Whether gross margin percent > 1.1 of last year's |

# APPENDIX F

Table 40: Smote Performance measures of the minority class for dataset FwSelect

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.639 | 0.132 | 0.669 | 0.639 | 0.654 | 0.822 |
| ANN | 0.560 | 0.132 | 0.639 | 0.560 | 0.597 | 0.826 |
| RF | 0.722 | 0.073 | 0.806 | 0.722 | 0.762 | 0.916 |
| NB | 0.796 | 0.480 | 0.409 | 0.796 | 0.541 | 0.710 |
| BBN | 0.674 | 0.217 | 0.564 | 0.674 | 0.614 | 0.808 |
| SVM | 0.375 | 0.094 | 0.624 | 0.375 | 0.468 | 0.640 |
| DT bagging | 0.684 | 0.090 | 0.761 | 0.684 | 0.720 | 0.894 |
| ANN bagging | 0.582 | 0.127 | 0.656 | 0.582 | 0.616 | 0.837 |
| NB bagging | 0.795 | 0.476 | 0.411 | 0.795 | 0.542 | 0.711 |
| BBN bagging | 0.673 | 0.216 | 0.566 | 0.673 | 0.615 | 0.810 |
| SVM bagging | 0.303 | 0.073 | 0.633 | 0.303 | 0.410 | 0.675 |
| DT boosting | 0.689 | 0.100 | 0.741 | 0.689 | 0.714 | 0.888 |
| ANN boosting | 0.567 | 0.131 | 0.644 | 0.567 | 0.603 | 0.818 |
| NB boosting | 0.543 | 0.211 | 0.518 | 0.543 | 0.530 | 0.714 |
| BBN boosting | 0.625 | 0.166 | 0.611 | 0.625 | 0.618 | 0.815 |

Table 41: Smote Performance measures of the majority class for dataset FwSelect

| | TP | FP | Precision | Recall | F-measure | AUC |
|---|---|---|---|---|---|---|
| DT | 0.868 | 0.361 | 0.852 | 0.868 | 0.860 | 0.822 |
| ANN | 0.868 | 0.440 | 0.825 | 0.868 | 0.846 | 0.826 |
| RF | 0.927 | 0.278 | 0.889 | 0.927 | 0.908 | 0.916 |
| NB | 0.520 | 0.204 | 0.859 | 0.520 | 0.648 | 0.710 |
| BBN | 0.783 | 0.326 | 0.852 | 0.783 | 0.816 | 0.808 |
| SVM | 0.906 | 0.626 | 0.776 | 0.906 | 0.836 | 0.640 |
| DT bagging | 0.910 | 0.317 | 0.873 | 0.910 | 0.891 | 0.894 |
| ANN bagging | 0.873 | 0.419 | 0.833 | 0.873 | 0.852 | 0.837 |
| NB bagging | 0.524 | 0.205 | 0.860 | 0.524 | 0.651 | 0.711 |
| BBN bagging | 0.784 | 0.327 | 0.852 | 0.784 | 0.817 | 0.810 |
| SVM bagging | 0.927 | 0.697 | 0.761 | 0.927 | 0.836 | 0.675 |
| DT boosting | 0.900 | 0.311 | 0.874 | 0.900 | 0.887 | 0.888 |
| ANN boosting | 0.869 | 0.434 | 0.828 | 0.869 | 0.848 | 0.818 |
| NB boosting | 0.789 | 0.457 | 0.805 | 0.789 | 0.797 | 0.714 |
| BBN boosting | 0.834 | 0.375 | 0.842 | 0.834 | 0.838 | 0.815 |

Table 42: Smote weighted average performance measures for dataset FwSelect

| | TP | FP | Precision | Recall | F-measure | AUC | G-mean |
|---|---|---|---|---|---|---|---|
| DT | 0.800 | 0.294 | 0.798 | 0.800 | 0.799 | 0.822 | 0.745 |
| ANN | 0.777 | 0.349 | 0.770 | 0.777 | 0.773 | 0.826 | 0.697 |
| RF | 0.867 | 0.217 | 0.865 | 0.867 | 0.865 | 0.916 | 0.818 |
| NB | 0.601 | 0.285 | 0.727 | 0.601 | 0.616 | 0.710 | 0.643 |
| BBN | 0.751 | 0.294 | 0.767 | 0.751 | 0.756 | 0.808 | 0.726 |
| SVM | 0.749 | 0.469 | 0.731 | 0.749 | 0.728 | 0.640 | 0.583 |
| DT bagging | 0.843 | 0.250 | 0.840 | 0.843 | 0.841 | 0.894 | 0.789 |
| ANN bagging | 0.787 | 0.333 | 0.781 | 0.787 | 0.783 | 0.837 | 0.713 |
| NB bagging | 0.604 | 0.285 | 0.728 | 0.604 | 0.619 | 0.711 | 0.645 |
| BBN bagging | 0.751 | 0.294 | 0.767 | 0.751 | 0.757 | 0.810 | 0.726 |
| SVM bagging | 0.743 | 0.513 | 0.723 | 0.743 | 0.710 | 0.675 | 0.530 |
| DT boosting | 0.838 | 0.249 | 0.835 | 0.838 | 0.836 | 0.888 | 0.787 |
| ANN boosting | 0.780 | 0.344 | 0.773 | 0.780 | 0.776 | 0.818 | 0.702 |
| NB boosting | 0.717 | 0.384 | 0.721 | 0.717 | 0.719 | 0.714 | 0.655 |
| BBN boosting | 0.773 | 0.313 | 0.774 | 0.773 | 0.773 | 0.815 | 0.722 |

# APPENDIX G

Table 43: Parameter specification for all the Bagging Algorithms used in this study

| **Parameter** | **Value** |
|---|---|
| Bag size % of training set size | 100 |
| Number of instances to process | 100 |
| Number of threads/execution slots | 1 |
| Number of classifiers | 10 |

Table 44: Parameter specification for all the Boosting Algorithms used in this study

| **Parameter** | **Value** |
|---|---|
| Weight Threshold | 100 |
| Number of instances to process | 100 |
| Resampling | False |
| Number of classifiers | 10 |

Table 45: Parameter specification for ANN Algorithms used in this study

| **Parameter** | **Value** |
|---|---|
| Nodes | sigmoid |
| Hidden layer | (attribute+classes)/2 |
| Validation set size | 0 |
| Training time | 500 |
| Learning rate | 0.3 |
| Number of instances to process | 100 |

Table 46: Parameter specification for BBN Algorithms used in this study

| **Parameter** | **Value** |
|---|---|
| Estimator | SimpleEstimator |
| Search Algorithm | K2 |
| Number of instances to process | 100 |

Table 47: Parameter specification for NB Algorithms used in this study

| **Parameter** | **Value** |
|---|---|
| Kernel estimator use | False |
| Number of instances to process | 100 |

Table 48: Parameter specification for DT Algorithms used in this study

| **Parameter** | **Value** |
|---|---|
| Number of folds | 3 |
| Minimum number of instances per leaf | 2 |
| Pruned | True |
| Number of instances to process | 100 |

Table 49: Parameter specification for RF Algorithms used in this study

| **Parameter** | **Value** |
|---|---|
| Bag size % of training set size | 100 |
| Number of threads/execution slots | 1 |
| Number of classifiers | 10 |
| Maximum depth of tree | unlimited |
| Minimum number of instances per leaf | 2 |
| Number of iterations | 100 |
| Number of features | int(log_2( no. of predictors) + 1) |
| Number of instances to process | 100 |

Table 50: Parameter specification for SVM Algorithms used in this study

| Parameter | Value |
|---|---|
| Type | C-SVC |
| Kernel | Radial basis |
| Gamma | 1/max_index |
| Cost | 1 |
| Degree of the kernel | 3 |
| Tolerance of termination | 0.001 |
| Number of instances to process | 100 |

# APPENDIX H

Table 51: List of overlapped attributes by various feature selection

| **CfsSubset - InfoGain** | **InfoGain - FwSelect** | **FwSelect- CfsSubset** | **All – 3 Feature Selection** |
|---|---|---|---|
| perols3 | perols21 | trad_div_total_asset | Perols12 |
| total_asset_to_lag | dechw_leverage | std_bhret_60m_incl_curr | std_bhret_60m_incl_curr |
| trad_accnt_pay_sales | | perols13 | trad_lag1_net_income |
| trad_hfi | | dechw_chg_oper_lease | |
| std_bhret_60m_incl_curr | | dechw_actual_issuance | |
| pct_change_long_ast_0_1 | | | |
| pct_change_ppe_net_0_1 | | | |
| pct_change_ppe_net_3yr_avg | | | |
| mkt_excess_ret | | | |
| lag1_mkt_excess_ret | | | |
| dechw_chg_oper_lease | | | |
| dechw_def_taxexp | | | |
| dechw_earn_price | | | |