2014

# Cost-Sensitive Learning-based Methods for Imbalanced Classification Problems with Applications

Talayeh Razzaghi
*University of Central Florida*

# COST-SENSITIVE LEARNING-BASED METHODS FOR IMBALANCED CLASSIFICATION PROBLEMS WITH APPLICATIONS

by

TALAYEH RAZZAGHI
B.S. University of Tehran, 2005
M.S. Sharif University of Technology, 2007

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2014

Major Professor: Petros Xanthopoulos

# ABSTRACT

Analysis and predictive modeling of massive datasets is an extremely significant problem that arises in many practical applications. The task of predictive modeling becomes even more challenging when data are imperfect or uncertain. The real data are frequently affected by outliers, uncertain labels, and uneven distribution of classes (imbalanced data). Such uncertainties create bias and make predictive modeling an even more difficult task. In the present work, we introduce a cost-sensitive learning method (CSL) to deal with the classification of imperfect data. Typically, most traditional approaches for classification demonstrate poor performance in an environment with imperfect data. We propose the use of CSL with Support Vector Machine, which is a well-known data mining algorithm. The results reveal that the proposed algorithm produces more accurate classifiers and is more robust with respect to imperfect data. Furthermore, we explore the best performance measures to tackle imperfect data along with addressing real problems in quality control and business analytics.

*To My Beloved Aunt, Mina*

# ACKNOWLEDGMENTS

My journey toward fulfillment of the doctoral studies at the University of Central Florida was one of the most valuable experiences in my life. It would not have been possible without all those who encouraged and supported me during this process. I would like to to gratefully acknowledge to them here.

First and foremost, I would like to express the deepest appreciation to my adviser Dr. Petros Xanthopoulos whose help, stimulating ideas and encouragement helped me in working on this problem and writing this dissertation. It was truly a pleasure for me to work under his supervision. I would especially like to thank him for his mentoring contributions towards my growth as a researcher.

I would like to thank my committee members, Professor Waldemar Karwowski, Dr. Jennifer Pazour, and Professor Mikusinski, for being on my Dissertation committee and valuable comments. I would like to especial thank to Dr. Pazour to always allow me to feel comfortable sharing my thoughts with her and use her precious advice for academic life and job search experiences. In addition, my sincere thanks goes to Dr. Onur Seref at the Business Information Technology, Virginia Tech for his motivation and stimulating discussions.

My warm thanks to my first mentor and Master's thesis advisor, Professor Farhad Kianfar from the Industrial Engineering department at Sharif University of Technology, for inspiring me to study higher education abroad and for his invaluable support. I am very thankful to Ms. Liz Stavely, Maria Bull, Li Muyuan, Yilling He, Serina Haddad, and Halil Bozkurt for being always so kind, helpful and supportive. It has been a true pleasure working in the same environment with them.

And last but not least, my special thanks to my parents and my lovely sister, Tarlan, for their unconditional support and love throughout my life. I would especially like to thank my dear aunt Mina and his husband, Ali Tarkhagh, for the love and support that they continuously gave me especially throughout my graduate studies.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1: INTRODUCTION

Data mining (DM) has emerged as one of the most important research areas in recent decades. DM aims to extract useful knowledge from data and identify significant patterns. Today, the size of the data in different domains is continuously growing due to advanced computational technology and the reduced cost of storage. The real data are frequently affected by outliers, uncertain labels, and uneven distribution of classes (imbalanced data). We call this type of data imperfect data. This work will help to address the unavoidable difficulty of data uncertainty that occurs in many real-world problems. In particular, we study the cost-sensitive learning-based adaption of the Support Vector Machine to deal with the imperfect data. As it will be discussed later, the Support Vector Machine is an excellent learning algorithm for binary data classification and regression.

## A Brief Overview

In this section, we review the historical background and methodological aspects of DM and, in particular, classification or supervised learning. DM, as an interdisciplinary field, is the intersection of artificial intelligence, machine learning, and statistics. In addition, the progress in this field has created a strong connection with mathematical optimization. Mathematical optimization provides a powerful and effective tool for data mining. Mangasarian's team developed a large margin classifier through a linear programming formulation in the 1960s (Fung & Mangasarian, 2001). Charnes et al. (1985) proposed Data Envelopment Analysis through a fractional programming formulation. Between the 1980s and the 1990s, Glover developed various linear programming models which solve discriminant problems with a small sample size dataset (Freed & Glover, 1986). Subsequently, the researcher and his collaborators further expanded this research initiative into classification problems through multiple criteria linear programming and quadratic programming. The Support Vector Machine method initiated by Vapnik (2000) is based on a quadratic programming

formulation. Clustering algorithms are formulated as a concave minimization problem. However, most DM techniques are combinatorial in nature and can be formulated as discrete optimization problems, which lead to NP-hard optimization problems (Xanthopoulos et al., 2013). To date, research on adoption of optimization techniques to tackle data mining problems has been extremely popular.

The data is defined as a set of samples/instances/observations and their features/attributes, where a feature/attribute is described as a characteristic of a sample. Generally, DM can be divided into certain categories according to the analysis of the knowledge discovery process, which is listed below:

- Data preprocessing and preparation methods
- Data visualization
- Machine learning (Supervised learning, semi supervised learning, and unsupervised learning)

We intend to briefly explain these categories. Data preprocessing or preparation is necessary in all knowledge discovery tasks. Data preprocessing mainly includes outlier detection, data normalization, data cleaning, sampling, and feature selection and extraction algorithms. Appropriate data preprocessing significantly improves the performance of learning algorithms. The original data set is divided into training and test data subsets (sometimes validation data might be used). The learning model is performed using the training data and then evaluated with the test data. Standard random sampling method repetitively performs learning through using different training/test data sets. The average performance of all learned models is reported. This procedure is known as cross-validation. There are a number of cross validation techniques. In k-Fold cross validation, the dataset is divided into k folds in which training is performed on $k-1$ folds and testing is performed on one fold. In the cross validation process, each subset is only used once for testing. While k is

equal to the number of samples in the dataset, the leave-one-out approach can be used which is a special type of k-Fold cross validation.

Outlier detection is an important problem for mining purposes. An outlier is a data point that is significantly different from the remaining data points (Hawkins, 1980). Generally, either the class or attributes of the data can be affected by outliers or noise. In this work, we mainly focus on class noise, which has significantly detrimental impacts on the classifier. Class noise refers to data points that are identified by an incorrect class label. Class noise can occur for various reasons, such as subjectivity, human error in entering data, instrument imperfections (Liu et al., 2013), and lack of sufficient information used to label each data point (Brodley & Friedl, 2011). Outlier detection algorithms usually use some evaluation measures in order to report the outlierness of an observation, e.g. the sparsity of the region around data points, distance based on nearest neighbor, or the fitness of primary data distribution (Aggarwal, 2013). An accurate data model may result in better performance measures for data mining algorithms.

Feature selection techniques have been drawing increasing attention in data mining. Feature selection provides the most relevant subset of features. Recently, a wide variety of feature selection methods have been proposed, such as filter methods, wrapper methods, and embedded methods (Wang et al., 2013). Moreover, feature selection has been widely used in biomedical research (Balabin & Smirnov, 2011; Warren Liao, 2011; Peng et al., 2010). For better understanding of feature selection methods, we refer the reader to Omar et al. (2013) and Bolón-Canedo et al. (2013).

Feature extraction generates small number of features from the original set of features. By applying feature selection and extraction techniques on a large feature set, subsets of lower dimensionality are obtained, which significantly reduces of the original number of features. This procedure is called dimensionality reduction and leads into the improvement of performance measures and processing time. There are several methods of feature extraction reported in the literature for character recognition (Due Trier et al., 1996). For further study, we refer interested readers to

Pradeep et al. (2011).

Data visualization is useful for analysis of the high-dimensional data in a low-dimensional space. For this purpose, dimensionality reduction techniques have been widely implemented. Multidimensional scaling (MDS) (Cox & Cox, 2010) and principal component analysis (PCA) (Jolliffe, 1986) are conventional linear methods for dimensionality reduction. Manifold learning (Tenenbaum et al., 2000; Roweis & Saul, 2000) and self–organizing maps (SOMs) (Jphonen & Maps, 1995) is a non–linear dimensionality reduction methods. Recently, several dimensionality reduction methods have been developed in data mining community (Tenenbaum et al., 2000; Saul & Roweis, 2003; Belkin & Niyogi, 2003; Moody & Healy, 2014). Network representation can facilitate understanding of the dynamics that govern a system.

Machine learning is the core of data mining which concerns the construction of algorithms that can be learned from data. For example, a machine learning algorithm is trained on email messages to detect spam and non-spam messages.

Unsupervised learning, sometimes known as clustering, extracts hidden structure in unlabeled data. Clustering methods aim to detect homogeneous groups or clusters using unlabeled data in the training set. This makes a distinction between unsupervised learning from supervised learning. For example, an unsupervised learning algorithm can be applied to classify medical images (Srivastava et al., 2013).

Supervised learning, sometimes known as classification, is significantly crucial for automated data driven knowledge discovery. In a supervised learning model, each training data is a pair includes an input value and a targeted output value. The main objective is to separate a set of data into classes or sub-categories and then to predict the class of a new observation. The mathematical function, implemented by a classification algorithm is known as a classifier.

Semi supervised learning lies at the intersection of supervised and unsupervised learning techniques.The main idea in semi supervised learning is to exploit unlabeled data during a supervised

learning procedure. We note that acquisition of unlabeled data is relatively inexpensive compared to acquisition of a fully labeled training set. For a complete overview of semi supervised techniques, we refer the reader to Zhu (2006), Hady & Schwenker (2013), and Richarz et al. (2014).

There are several commercial problem solving environments like SAS (http://www.sas.com/), SPSS (http://www.spss.com/), and Statistica (http://www.statsoft.com/). Some of data mining techniques can be easily implemented in some popular technical computing programming languages such as Matlab and Mathematica. Moreover, many open source environment can be found online like Weka (http://www.cs.waikato.ac.nz/ml/weka/), R (http://www.r-project.org/), and Python (http://www.python.org/).

## A Brief History of Imbalanced Classification

Today, imbalanced classification has been drawing increasing attention in the data mining community. Imbalanced classification occurs in problems where the class size of given examples is not equal (Japkowicz, 2000). For example, in a cancer diagnostic problem, the main objective is to identify individuals stricken with cancer, and such events are relatively rare compared to normal cases. In network intrusion detection, cyber attacks on the system are very rare. In general, imbalanced classification problems can be found in many areas, including security surveillance (Wu et al., 2003), disease diagnosis (Huang & Du, 2005), bioinformatics (Al-Shahib et al., 2005), geomatics (Kubat et al., 1998), telecommunications (Fawcett & Provost, 1997; Tang et al., 2006), risk management (Ezawa et al., 1996; Groth & Muntermann, 2011), manufacturing (Adam et al., 2011), quality estimation (Lee et al., 2005; Suresh et al., 2009), tornado detection (Trafalis et al., 2013), and power systems (Hu et al., 2008). There are several review papers dedicated to classification of imbalanced datasets (Japkowicz, 2000; Japkowicz & Stephen, 2002; Guo et al., 2008; He & Garcia, 2009; Su et al., 2009; Sun et al., 2009; Chawla, 2010). In the literature, imbalanced classification problems can also be known as skewed class distribution problems or as small/ rare

class learning problems (Sun et al., 2009; Weiss, 2004). In the case of binary classification, the number of examples in one class may greatly outnumber the other class. The class with fewer examples is the so-called minority class and the class with more examples is defined as the majority class. In many application areas (e.g. fraud detection, computer intrusion detection, oil spill detection, defect product detection), the detection of the minority class is more critical than the majority class. Napierala & Stefanowski (2012) show that the distribution of data is much more influential in constructing classifiers than the size of data. They analyze the influence of different types of datasets on six various classifiers and compare the sensitivity of each classification method based on different performance measures. A preferred classification algorithm is one that yields a greater identification rate on the rare event (e.g. disease type). These classification algorithms are evaluated through performance measures. However, certain performance measures are more appropriate than others when the classification problem is imbalanced. For example, classification accuracy, which is the percent of the correctly classified training samples over the total number of training samples, has been found to be a weak performance measure (Chawla, 2010; He & Garcia, 2009). This is because the error of the minority class is not well reflected in the overall accuracy. Usage of inappropriate measures might yield the wrong understanding of the classifier performance. Even the scheme of class imbalance into the classification problem initiated serious challenges that needed to be studied. Moreover, class noise and outliers make the problem of imbalanced classification more difficult. In the next section, we will explain the necessary concepts to understand the imbalanced classification in the presence of class noise.

### *Imbalanced Classification with Class Noise*

The classification of imbalanced data can be even more difficult in the presence of class noise and outliers. Class noise or outliers refer to data points that are identified by incorrect class labels. Clearly, uncertainty associated with class noise is inherent in these real-life problems, thus noisy data management is extremely necessary. Imbalanced classification with class noise is common in

various domains. Some examples are as follow:

- **Defect Detection:** In manufacturing, a non-defected product might be labeled as a defected one. For example, data collected by hardware technologies (e.g. sensors) might be influenced by the environmental variables or temporal malfunctioning (Zhong et al., 2005) and this might affect the labeling of the data. The misdetection of defect-prone products can dramatically lower customer satisfaction, jeopardize the reputation of manufacturing companies, and finally result in a huge loss of money.

- **Fraud Detection:** A fraud transaction recorded as a non-fraud transaction is another example of class noise. In some cases, an authorized transaction may appear an unauthorized transaction, such as buying something from geographically unknown locations.

- **RFID Network Systems:** One of the main drawbacks of the RFID technology is that it sometimes produces unreliable data streams corrupted by outliers (Jeffery et al., 2006; Nie et al., 2009). Recently, RFID network systems have been extensively used in many applications such as security and access control, transportation, supply chain tracking, and healthcare (Ma, 2012). It is reported that the misreading rate can be up to 30–40% (Jeffery et al., 2006).

- **Medical Diagnosis:** The clinical data are collected in different formats, such as MRI scans and ECG time-series. In general, abnormal patterns in such data indicate disease symptoms. Misdiagnosis or error in labeling patients frequently takes place and negatively affects medical decision making (Pechenizkiy et al., 2006). A wrong prediction of the nonexistence of cancer or cancer existence may lead to the patient risk, unnecessary anxiety and extra medical tests.

- **Earth Science:** A considerable amount of data on the topic of climate change, weather patterns, or land cover patterns is accumulated using a range of technological tools like

satellites or remote sensing. Abnormal patterns in such data are useful to detect hidden human or environmental trends which may be the reason for such anomalies.

- **Spam Filtering:** While it may be tolerable to misclassify a few spam emails (thereby allowing them into the inbox), it is much more undesirable to incorrectly label a legitimate email as a junk mail (Tang et al., 2006).

Other applications can be found in marketing and customer behavior modeling (Casillas & Martínez-López, 2009). There are several techniques to tackle these problems. We will discuss the conventional outlier detection techniques and embedded algorithms in the next section.

### Embedded Outlier Detection and Classification vs. Conventional Outlier Detection

Most conventional outlier detection techniques first detect outliers and then tend to remove them from the original data set. Thus, the remaining data is used for training. However, there is no work in the literature that detects outliers and classifies the data set in an embedded formulation. Even the characteristic of unbalancedness makes the classification problem a more challenging task. Cost-sensitive learning is the main algorithmic approach for imbalanced classification problems. However, cost-sensitive based approaches are highly sensitive to training datasets with outliers (Chawala et al., 2002; Batuwita & Palade, 2010; Wang et al., 2012). Therefore, our main research focus is on how to improve a cost-sensitive learning algorithm to deal with imbalanced data and outliers simultaneously. Cost-sensitive learning assigns different weights to each data point based on its importance to the model and solves the weighted classification problem. The SVM adaptation to the cost-sensitive learning framework is termed WSVM (also known as Fuzzy SVM in some works) and was originally proposed by Lin & Wang (2002) and further applied and studied in subsequent works (Fan & Ramamohanarao, 2005; Bao et al., 2005; Huang & Du, 2005; Hwang et al., 2011; Zhang et al., 2012). WSVM's advantage is that the cost coefficient is directly factored into the SVM problem, providing an exact optimal solution.

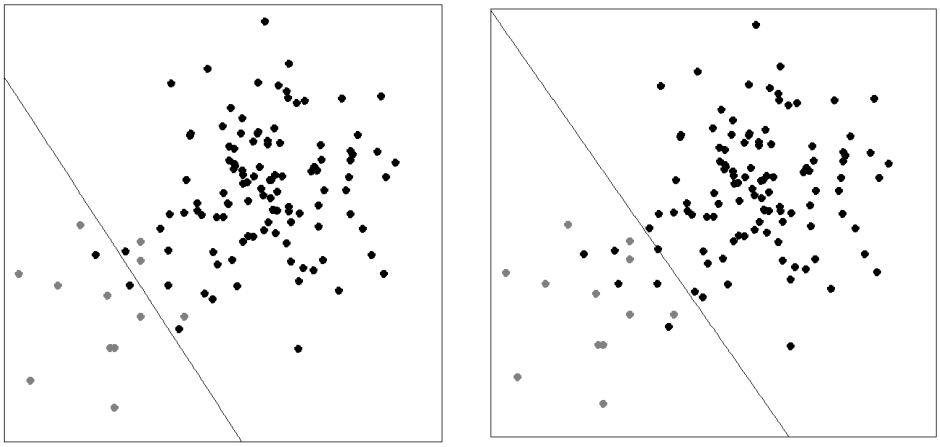
SVM classifier misclassifies a subset of data points in the minority class (See Figure 1.1 in the left). The WSVM classifier detects all the data points in the minority class; however, it might misclassify a few data points of the majority class (See Figure 1.1 ). We note that the correct classification of the minority class examples is often more important. WSVM results in poor classification performance in the presence of outliers. Figure 1.2 shows that the decision boundary of WSVM is shifted toward outliers and tends to misclassify the data points in the majority class. Since the minority class data points are associated with higher weights, then the outliers receive high weights and contribute strongly in training. To solve this problem, it is necessary to propose a method that decreases the effects of outliers on the classification of imbalanced datasets.

## Dissertation Goal and Structure

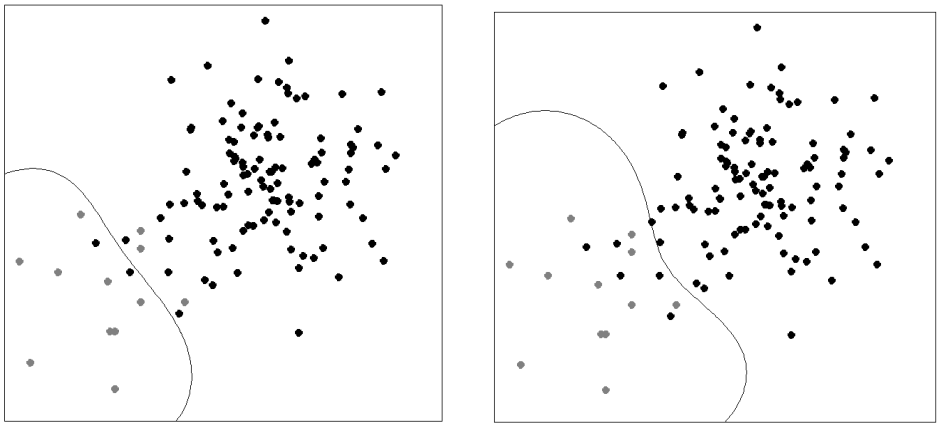
Although outliers are rare events, their detection is extremely important compared to other events. The main objective of this research is to explore the techniques to tackle highly imbalanced noisy data and provide an efficient cost-sensitive learning (CSL) method as a solution. One of the contributions of this work is to develop an effective embedded formulation of CSL in order to simultaneously deal with imbalanced data and outliers through setting the following goals:

1. To find the applications of imbalanced and noisy classification in the real world
2. To propose an efficient CSL algorithm in binary and multi-class imbalanced environments
3. To improve model selection in the CSL algorithm for imbalanced data
4. To provide a guideline to help the decision makers to efficiently classify uncertain or imperfect data

Particularly, when the outliers belong to the minority class, traditional classification techniques might result in poor classification performances. On the other hand, another concern regarding the



(a) Linear SVM (b) Linear WSVM

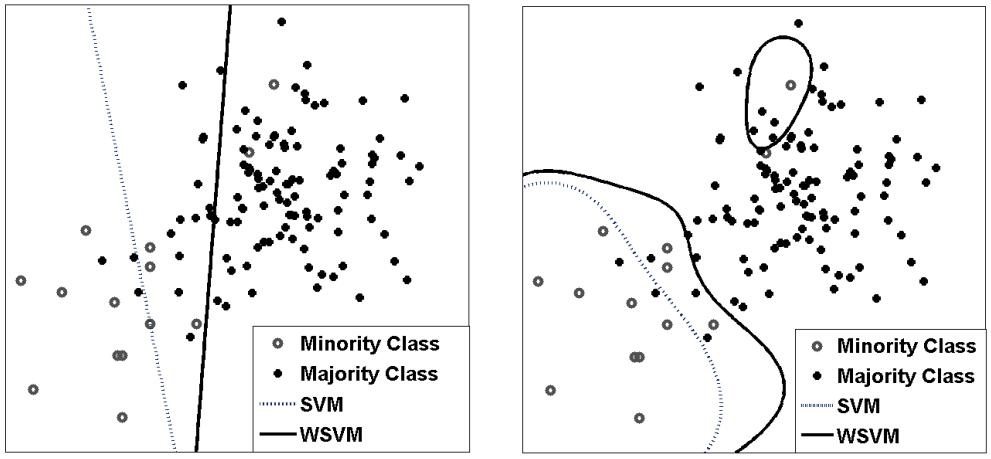


(c) Non-linear SVM (d) Non-linear WSVM

Figure 1.1: Imbalanced data classification without outliers using linear (a-b) and RBF (c-d) kernel functions. The black and gray points show the majority and minority class data points respectively.

use of data mining algorithms (e.g. SVM) is that all the parameters of the algorithm need to be tuned during the training process. Frequently, users must carry out an exhaustive search to find the best value for the parameters and this might even gets worse if the parameters increase. Therefore, an effective and adjusted method is needed for parameter selection. In this work, we propose an embedded model that at the same time can automatically perform model section, outlier detection, and classification. Then, we evaluate the proposed model on the simulated and benchmark datasets.

Furthermore, as a real case, we develop an effective and useful CSL algorithm in quality control. Quality control is one of the most important topics in the field of industrial engineering. The




(a) Linear SVM and WSVM (b) Non-linear SVM and WSVM

Figure 1.2: Imbalanced data classification with outliers using linear (a) and RBF (b) kernel functions.

manual inspection and evaluation of quality control data is a tedious task that requires the undivided attention of specialized personnel. Over the last two decades, control chart pattern recognition (CCPR) problems have received a lot of attention (Hachicha & Ghorbel, 2012). Current state-of-the-art control monitoring methodology includes K charts, which are based on Support Vector Machine (SVM). Although K charts have some profound benefits, their performance deteriorates when the learning examples for the normal class greatly outnumber the ones for the abnormal class. Since imbalanced problems represent the vast majority of the real life control pattern classification problems, the original SVM formulation needs to be tailored in order to address this deficiency. This is another contribution of this work. To the best of the authors' knowledge, there is not a CCPR algorithm that takes into consideration the imbalanced nature of the abnormal pattern detection problems. Furthermore, there have not been sufficient computational studies that evaluate the applicability of generic SVM in highly imbalanced environments. For instance, in the quality field, the traditional control charts assign equal importance to all data points, which is not necessarily optimal and might give poor classification performance.

The rest of this work is organized as follows: In section 2, we give an overview of the main algorithmic approaches, the state-of-the-art performance evaluation measures for imbalanced classification and outlier detection, and several applications in quality control and business analytics. In section 3, we introduce the primal and dual formulations for the Support Vector Machine (SVM), the Weighted Support Vector Machine (WSVM), and the Weighted Relaxed Support Vector Machine (WRSVM) along with their theoretical properties. We also describe an adjusted method for parameter selection based on nested uniform designs. In Section 4, we provide a comparative study of SVM and WSVM using both simulated and real data from quality control in a highly imbalanced environment. Then, we present the comparative computational results for WRSVM against SVM, WSVM and other related classification methods for different types of imbalanced problems and noise levels. We finally conclude our work and provide directions for future research in Section 5.

# CHAPTER 2: LITERATURE REVIEW

This section provides an overview of imbalanced classification and outlier detection techniques. We first explain the state-of-art algorithms to solve imbalanced classification problems and proper evaluation measures. We discuss outlier detection techniques in a more detail. Finally, we present common techniques for imbalanced classification for two main application fields: quality control and business analytics.

## Imbalanced Classification Techniques

Several classification techniques have been proposed and applied in the literature for imbalanced classification problems. These techniques can be classified in two major categories: resampling and cost-sensitive learning. However, there are ensemble algorithms which builds an ingratiation of classifiers. Typically, these algorithms are ensemble of cost-sensitive learning or resampling algorithms. The objective of using ensemble learning is to improve the classification performance.

### *Resampling*

Resampling techniques are among the most popular preprocessing methods. Under this framework data points are added (oversampling) or removed (undersampling) to create a balanced problem. The Synthetic Minority Oversampling Technique (SMOTE) belongs to this category (Chawala et al., 2002). However, resampling methods become inefficient for highly imbalanced problems with limited minority class examples and when data distribution are unknown (Elazmeh et al., 2006). In fact, oversampling often suffers from induced bias or overfitting, whereas through undersampling it is possible to lose valuable information by removing data (Nitesh et al., 2002;

Estabrooks et al., 2004; Akbani et al., 2004; Chawla et al., 2005; Tang et al., 2009; Liu et al., 2009).

### *Cost-Sensitive Learning*

Cost-sensitive learning algorithms assign weights to data examples based on their importance. They are equivalent to resampling technique and combine both undersampling and oversampling. Many popular classification algorithms can be adapted under this framework. The SVM adaptation is termed weighted support vector machine (also termed Fuzzy SVM) which was originally proposed by Lin & Wang (2002) and further applied and studied in subsequent works (Zhang et al., 2011; An & Liang, 2013; Ke et al., 2013). Their advantage is that the cost coefficient is directly factored into the SVM problem providing an exact optimal solution. Assume that a dataset is represented by a set of data point $J = \{(x_i, y_i)\}_{i=1}^{l}$ where $(x_i, y_i) \in R^{n+1}$, $l$ and $n$ are the number of samples and features, respectively, and each $x_i$ is a sample with $n$ features and a class label $y_i \in \{+1, -1\}$. The costs for two classes (minority and majority) are represented with $C^+$ and $C^-$. The weighted SVM classifies the data points by identifying a separating hyperplane whose distance is maximum with respect to the data points of each class. The separation hyperplane defined by the parameters $w$ and $b$ can be obtained by solving the following convex optimization problem.

$$\min \ \frac{1}{2}\|w\|^2 + C^+ \sum_{\{i|y_i=+1\}}^{n_+} \xi_i + C^- \sum_{\{j|y_j=-1\}}^{n_-} \xi_j \tag{2.1a}$$

$$\text{s.t. } y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i \qquad i = 1, \ldots, l \tag{2.1b}$$

$$\xi_i \geq 0 \qquad i = 1, \ldots, l \tag{2.1c}$$

where $\phi$ is the kernel function $\phi : R^n \to R^m$ where $m \gneq n$, i.e. each training sample $x_i$ is mapped into a higher dimensional space by the function $\phi$. The slack variables $\xi_i \in \{1, \ldots, l\}$ are added to the objective function whose goal is to allow but penalize misclassified points.

In (Zhao et al., 2007), a cost/benefit sensitive algorithm is presented to classify rare events in online data (called Statistical Online Cost-Sensitive Classification) and the results of the model demonstrate that this method performs better than any other cost-insensitive online algorithms.

The SVM adaptation to the cost-sensitive learning framework is termed WSVM (also found as Fuzzy SVM in some studies) which was originally proposed by and further applied and studied in subsequent works (Fan & Ramamohanarao, 2005; Bao et al., 2005; Huang & Du, 2005; Hwang et al., 2011; Zhang et al., 2012). Their advantage is that the cost coefficient is directly factored into the SVM problem providing an exact optimal solution.

However, it is often difficult to determine costs in reality and might need more knowledge and domain experts' involvement (Han et al., 2009). It is suggested to differ the cost ratio until an acceptable objective function value is found (Weiss, 2004). Some researchers have used the heuristic algorithms to set the parameters. In Sun et al. (2006), Genetic Algorithms are used to search the optimal misclassification cost for each class in a multiclass classification problem. However, this problem is more discussed in section 3.

Cost-sensitive algorithms outperforms traditional data mining algorithms. However, various standard classification algorithms have been developed, but they often result in poor performance in detection of minority class. For instance, experiments on training imbalanced classification data sets in (Anand et al., 1993) using the neural network showed that the error for samples in the major class was decreased quickly, and the error for smaller class increased significantly. The experiments also demonstrated that the rate of the error decreasing for the smaller class is too low by using ANN and takes too much iteration to obtain an acceptable solution. For detailed explanation of the performance of ANN with imbalanced classification problems, we refer the reader to (Carvajal et al., 2004; Japkowicz & Stephen, 2002). Not only does the ANN result in misclassification of imbalanced data, but also it has been demonstrated that the traditional SVM and decision tree classification suffer from this deficiency for imbalanced data (Japkowicz & Stephen, 2002).

Another methods implemented as cost-sensitive learning to overcome the imbalanced data is introduced by other researchers (Wu & Chang, 2005; Imam et al., 2006). Imam et al. (2006) proposed a so-called z-svm which implements a weighted strategy for the positive support class with the objective of maximizing G-mean to adjust the hyperplane and reduce skew towards the minority class. The advantage of this work is that the model avoids pre-selection of parameters and auto-adjust the decision hyperplane. Wu & Chang (2005) developed a cost-sensitive algorithm with adjusting the class boundary and the kernel matrix on the basis of the data distribution.

### *Ensemble Learning*

Ensemble learning is a well-established method that combines the outputs of multiple base learners. The intuition behind the algorithms is that they modify the generalization ability of individual classifiers by assembling many sub-classifiers (Bishop & Nasrabadi, 2006). Averaging the outputs of base models reduces the bias among classification models. Several ensemble methods has been proposed such as Bagging (Breiman, 2001), Boosting (Freund & Schapire, 1995), and Stacking (Wolpert, 1992).

*Bagging algorithms*, first proposed by Breiman (Breiman, 2001), constructs an ensemble of multiple base classifiers by random uniformly sampling from the original training data set. For example, the random forest algorithm uses random decision trees with bagging which results in high classification performance. However, most techniques modify the bagging method by combining it with resampling techniques. Since the original data is imbalanced, the bagging will not change the class distribution considerably in the training sample (Błaszczyński et al., 2013). Exactly Balanced Bagging (EBBag) is the simplest version of bagging method which implements undersampling to exactly balance the cardinality of the minority and the majority class in each sample (Chang et al., 2003), thus the entire minority class is trained with randomly chosen subsets of the majority class (Figure 2.1).

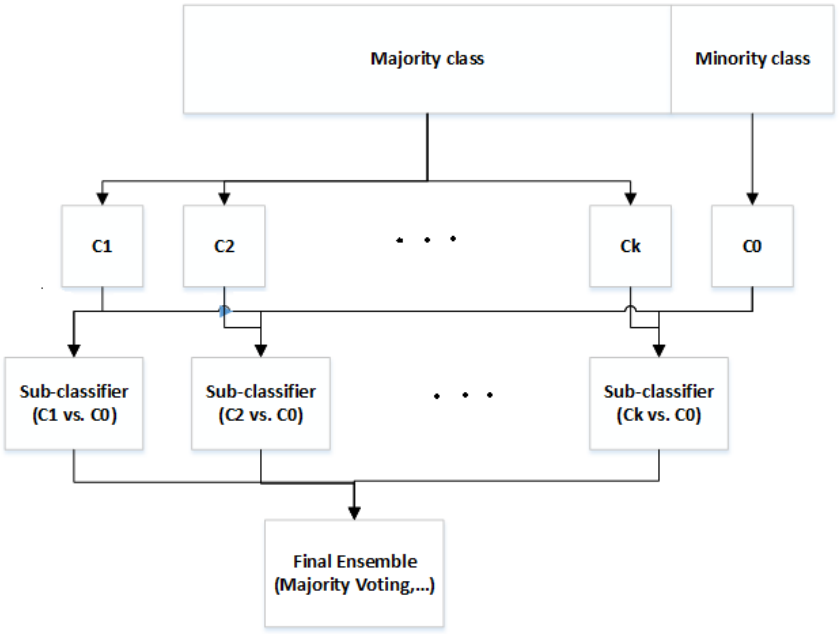



Figure 2.1: Bagging Algorithm

*Boosting algorithms* are powerful successive ensemble learning algorithms (Freund & Schapire., 1997; Schapire & Singer, 1999). Boosting algorithms is just another classification algorithm that can be adapted into the cost-sensitive framework. The boosting algorithms such as Ada-boost performs learning iteratively and assigns weights to each example in a way the misclassified examples from previous learning are assigned with higher weights than the correctly classified examples (Schapire et al., 1998; Changrampadi et al., 2012). But in bagging algorithms, all training examples are equally weighted.

*Stacking algorithms* is different from bagging and boosting algorithms in a way that the individual weak learners are not the same. Stacking consists of a two-level structure: base-level classifiers and meta-level classifiers (Wolpert, 1992). The base-level classifiers are obtained with the training datasets and produce their predictions. Then the predictions are considered as the input of the meta-classifier to construct the nal decision. For Further detail about ensemble learning techniques, we refer interested readers to the reference (Duda et al., 2001). In addition to these

techniques, other hybrid techniques have been implemented in the literature to overcome the issue of imbalanced data. Padmaja et al. (2011) develop an extreme outlier elimination with hybrid sampling technique and k Reverse Nearest Neighbors for fraud detection. The hybrid sampling technique propped by them consists of a combination of SMOTE and random undersampling.

### Performance Measures for Imbalanced Classification

Classification performance measures can be obtained, directly or indirectly, from the confusion matrix. For a classification problem with $k$ classes, the confusion matrix is a square matrix $C \in \mathbb{R}^{k \times k}$, with each of its entries $c_{ij}$, denoting the percentage of the samples that belong to the class $i$ and classified to the class $j$. For the special case of binary classification (positive and negative), the confusion matrix is as follows:

Table 2.1: Confusion matrix for binary classification

| | Predicted Positive | Predicted Negative |
|---|---|---|
| Actual Positive | TP (True Positive) | FN (False Negative) |
| Actual Negative | FP (False Positive) | TN (True Negative) |

where TP, FP, FN, TN stand for true positives, false positives, false negatives and true negatives correspondingly. In this matrix, diagonal elements represent accurately classified examples and the off-diagonal elements the misclassified data for each class. A typical performance measure for classification is the so-called accuracy, which is calculated as the correctly classified samples over the total number of training samples. However, for imbalanced classification problems this might not be a good performance indicator, since the majority class dominates the behavior of this metric. More specifically, naive decision rules can yield high classification accuracy. For example, the rule *"Assign all data point to the positive (majority) class"* will yield 95% classification accuracy in an imbalanced problem where 95% of the data belong to the positive class and 5% to the negative.

Alternatively, sensitivity and specificity can be used. They are defined as

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \quad \text{Specificity} = \frac{TN}{TN + FP}. \tag{2.2}$$

For the previous "toy" example, the discussed naive classifier would have $95\%$ sensitivity and $0\%$ specificity. Still, sensitivity is manipulated by the majority (positive) class. However, the specificity is not and therefore, it is a more appropriate measure for this purpose. The space spanned by sensitivity and specificity is termed Receiver Operator Characteristic (ROC) space. The ROC space provides a good visual representation of the classifier 2.2. A combined measure frequently used for imbalanced data is the geometric mean of sensitivity and specificity (often abbreviated *G-mean*) defined by

$$\text{G-mean} = \sqrt{Sensitivity * Specificity} \tag{2.3}$$

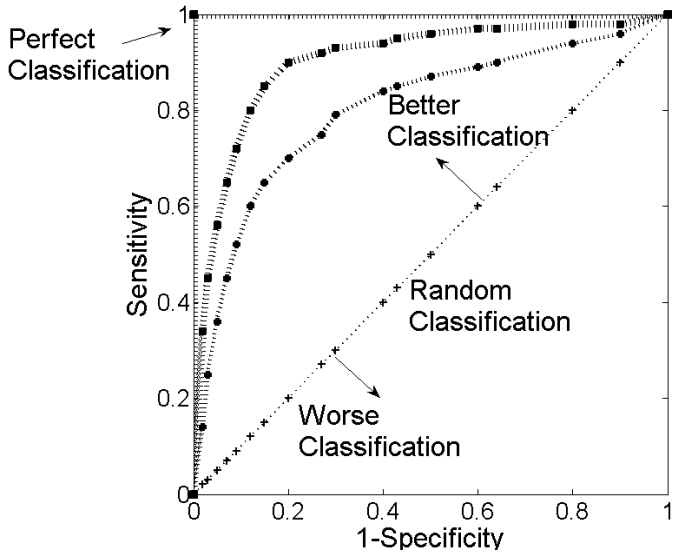



Figure 2.2: ROC curve showing four classifiers

There are other metrics used in the literature, including precision and recall or hit rate which is

the ratio of true positive to the sum of true positive and false positive (Duman et al., 2012) and lift which is highly related to accuracy, but it is well used in marketing practice (Ling & Li, 1998). For a comprehensive review of classification performance measures we refer the reader to (Sokolova & Lapalme, 2009).In this study, sensitivity, specificity and *G-mean* are used as performance measures.

## Outlier Detection Techniques

Outliers are objects that are significantly different from the rest of the data. They are often a good indication of abnormal behavior in the system (Hawkins, 1980; Han & Kamber, 2006). Outlier detection has been widely used for detection of anomalies in fraud detection (Bolton & Hand, 2002), network intrusion detection (Lane & Brodley, 1999; Gogoi et al., 2011), criminal detection in e-commerce (Chiu & Fu, 2003), wireless sensor networks noise detection (Zhang et al., 2010), and detection of outliers in surface acoustic wave sensor (Jha & Yadava, 2011).

Outlier detection is a quite active research area and a wide variety of methods have been developed so far (Knox & Ng, 1998; Knorr et al., 2000; Hodge & Austin, 2004; Cateni et al., 2008; Xi, 2008; Ben-Gal, 2010; Suri et al., 2011; Niu et al., 2011; Zimek et al., 2012). Typically, two common approaches exists to deal with outliers in the literature. The first approach is to detect the outliers and remove them from the analysis while the second approach is to detect them and contribute them in the learning process but with indicating how much they are important in the study. Generally, outlier detection techniques can be classified as,

- Probabilistic and Statistical-based Models
- Distance-based Outlier Models
- Density-based Outlier Models
  - LOF: Local Outlier Factor

  - LOCI: Local Correlation Integral

- Clustering-based Outlier Models

The appropriate choice for an outlier detection model is usually data set specific. Therefore, a good understanding of the data (e.g. as data type, outlier type, and outlier degree) significantly helps to choose the model. For example, when the distribution of data is linear, the most suitable model for identifying outliers would be a regression-based model.

*Probabilistic and statistical-based models* model the data using a statistical distribution, and then outliers are determined based on how they relate to the proposed model. (Hawkins, 1980; Bamett & Lewis, 1994). The choice of the data distribution with which the modeling is performed is the primary assumption. A major drawback with probabilistic models is that sometimes the particular kind of distribution which fit into the data may not be an appropriate distribution. Moreover, over-fitting might occur when the number of model parameters increases in the statistical model. In such cases, detecting of outliers from the normal data becomes extremely challenging. The data points are assumed as outliers while their values are either too large or too small. It is very important to identify the statistical tails of the underlying distribution. Typically, the normal distribution is the easiest way for this purpose and a common rule of thumb is that those data points deviating more than three times the standard deviation from the mean of a normal distribution are assumed to be outliers. This rule is also known as the "$3.\sigma$-rule". Visual techniques such as box plots and histograms can also help to detect these extreme values.

*Density-based models* identify the data points in low dense regions as outliers. Breunig et al. (2000) determined an outlier score to any data point, so-called as Local Outlier Factor (LOF). The LOF score is calculated based on the distance of each point from its local neighborhood. Aggarwal (2010) has proposed a general density-based approach for handling uncertain data and outliers. The Local Outlier Factor (LOF) is a measure of the outlierness of a data point. The LOF method was initially proposed in (Breunig et al., 2000) as a density-based method because of its ability to adjust

for the variations in the diverse densities. Other visual technique to evaluate outliers is the LOCI method which is a local density-based method for outlier analysis (Papadimitriou et al., 2003).

*Distance-based models* are very popular among outlier detection techniques for a wide variety of data domains, and identify outlier scores based on nearest neighbor distances. These methods assume that the k-nearest neighbor distances of outlier data points are much larger than normal data points (Aggarwal, 2013).

*Clustering-based models* identify very small subsets as clustered outliers. In this approach, ouliers are those clusters which include extremely less data points than other clusters. Clustering-based algorithms are unsupervised learning algorithms. Additionally, clustering-based methods have this advantages that after learning the clusters, new data points can be put into the system and tested for outliers. Van Cutsem & Gath (1993) proposed a fuzzy clustering model for outlier detection. Jiang et al. (2001) presented a two-phase method such that in the first phase a modified k-means algorithm and in the second phase an Outlier-Finding Process is implemented. Outliers were selected as very small clusters through using minimum spanning trees. Loureiro et al. (2004) proposed an outlier detection technique based on hierarchical clustering. The presence of outliers was identified by the size of the resulting clusters. A similar approach is described in (Almeida et al., 2007). Acuna & Rodriguez (2004) implemented the Partitioning Around Medoids algorithm followed by the technique. If the separation between clusters is large enough, then the objects in that cluster are detected as outliers. The desired number of clusters should be determined by the decision maker. Yoon et al. (2007) presented a k-means clustering algorithm to detect outliers. The disadvantage of k-means methods is their sensitivity to outliers, and because of this reason, sometimes they may not provide accurate results.

*Active learning* is an iterative procedure aims to label some of the examples in each iteration. A number of important examples are identified in each iteration, in a way that addition of labels helps further classification. The labels for these examples are provided by human experts. These additional data with labels are then used in learning the classifier. The first iteration uses an unsu-

pervised learning approach due to unlabeled data. This approach is carried out iteratively until the addition of further examples no longer improves the classification performance. This method can be helpful in situations in which a small number of labeled data points are available to begin with.

There are other supervised learning techniques for outlier detection. For example, Ghazikhani et al. (2012) used Support Vector Data Description (SVDD) for outlier detection. Schubert et al. (2014) have addressed unsupervised outlier detection techniques for spatial, video, and network datasets Ben-Gal (2010) categorized the outlier detection techniques from a different point of view: parametric and nonparametric techniques. Unlike parametric techniques, non-parametric methods rely on the concept of distance in order to estimate the separation between two data points.

For a more detailed review about outlier detection techniques, we refer the reader to (Hodge & Austin, 2004; Agyemang et al., 2006; Gogoi et al., 2011).

### *Outlier Detection Evaluation Measures*

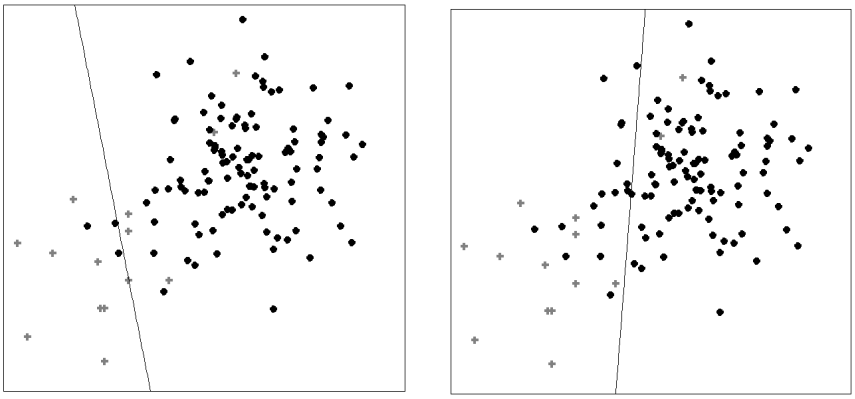
In general, most outlier detection algorithms are evaluated on the basis of several measures of the outlierness of a data point, such as the sparsity of the region around data points, distance badsed on nearest neighbor, or the fitness of primary data distribution (Aggarwal, 2013).

Typical performance measures to compare and evaluate outlier detection techniques are the detection rate, precision, recall, the ROC curves, the area under the ROC curves (AUC) (Provost & Fawcett, 2001). These metrics are calculated on the basis of the confusion matrix (Table 2.1).

### Classification with Imbalanced Data in the Presence of Outliers

Classification of imbalanced data in the presence of outliers is a very challenging task. Since most traditional classification techniques are highly sensitive to outliers (See Figure 2.3) (Zhao

et al., 2012; Batuwita & Palade, 2010). Not only cost-sensitive learning techniques but also pre-processing and ensemble learning techniques suffer from sensitivity toward outliers or noise (Wang et al., 2012; Chawala et al., 2002). There are several studies to deal with imbalanced classification and outlier detection in a separate way. In most cases, the outliers are detected and removed through using outlier detection/elimination techniques and then the remaining imbalanced data are employed in the learning process. For example, Batuwita & Palade (2010) have identified outliers by fuzzy membership values and then incorporated them in learning of imbalanced data with the use of a fuzzy support vector machine (FSVM) method. Furthermore, there are several studies which used the combination of two or more methods to deal with imbalanced and noisy data classification (Zhao et al., 2012). For example, Zhao et al. (2012) used the combination of FSVM with the kernel modification method on the basis of Riemannian metric.



(a) No outliers (b) Outliers

Figure 2.3: Imbalanced data classification in the presence of outliers. It can be observed that the classifier is greatly influenced by the outliers and the decision boundary is shifted to the right.

## Control Chart Pattern Recognition

Over the years, several abnormal patterns have been reported in real industrial problems, each of them reflecting a different underlying fault mechanism. In an early publication of Western Electric Company (1958), 15 normal and abnormal patterns are identified, one normal, seven *basic*

abnormal and seven *composite* abnormal. Examples of basic patterns, namely 1) normal (N), 2) up trend (Ut) 3) down trend (Dt), 4) up shift (Us), 5) down shift (Ds), 6) cyclic (C), 7) systematic (S), 8) stratification (F) patterns are illustrated in Figure 2.4 (a)–(g), whereas their mathematical model description can be found in the Appendix. The *composite* abnormal patterns are formed from linear combination of basic ones and they are more rare in practical applications.

Up trend and down trend patterns are associated with tool wear and malfunction in the crank case manufacturing operations (El-Midany et al., 2010). The Shift patterns may occur due to variations of material, machine or operator i.e. defect detection in a musical signal obtained from a broken disc or instrument (Davy et al., 2006; El-Midany et al., 2010). The power supply voltage variability is often indicated by cyclic patterns (Kawamura et al., 1988). Cyclic patterns also arise in manufacturing processes, such as frozen orange juice packing (Hwarng, 1995). Jang et al. (2003) describe anomalies in automotive body assembly process as up/down trends, cyclic, and systematic patterns. Jin & Shi (2001) detect stamping tonnage abnormal signals by detecting up/down trend patterns. Cook & Chiu (1998) and Chinnam (2002) identify that the abnormal control chart patterns of paper making and viscosity data are up/down trend whereas, Zorriassatine et al. (2005) uses up trend patterns with a fault state in an end-milling process. Since each pattern uniquely characterizes a certain type of malfunction, with respect to a specific application, methods for efficient identification of abnormal patterns are necessary in order to improve fault diagnosis/repair decision making.

Early CCPR studies propose basic statistical heuristics for mean and variance shift detection (Swift, 1987). Knowledge based expert systems and artificial neural networks for CCPR were also employed in the seminal works of Hwarng & Hubele (1992); Hwarng (1995); Hwarng & Hubele (1993a) and Hwarng & Hubele (1993b). Other CCPR algorithms include principal component analysis (PCA) (Aparisi, 1996), time series modeling (Alwan & Roberts, 1988), regression (Mandel, 1969), and correlation analysis techniques (Al-Ghanim & Kamat, 1995; Yang & Yang, 2005). Moreover, there are artificial intelligence-based CCPR approaches, such as the expert sys-

tem (Alexander, 1987; Cheng & Hubele, 1992) and the artificial neural network (ANN) (Pugh, 1989; Cheng, 1997; Cheng & Cheng, 2009). Soft computing/ data mining techniques are also used in CCPR based on the literature including clustering (Ghazanfari et al., 2008), neurofuzzy approaches (Chang & Aw, 1996; Taylan & Darrab, 2012), fuzzy-clustering (Zarandi & Alaeddini, 2010), decision trees (Wang et al., 2008) and support vector machines (Camci et al., 2008; Kumar et al., 2006; Sukchotrat et al., 2009).

Computational studies show that $K$ charts perform better than $T^2$ charts when the data is not normally distributed. Sun & Tsung (2003) proposed the complementary use of those two control chart types based on the underlying data distribution assumption. Camci et al. (2008) proposed a robust approach for $K$ charts along with a heuristic method for tuning the kernel parameters. The SVM based charts are based on quadratic programming and have been proved to have minimum generalization errors. The classifier is obtained as an exact solution to the convex optimization problem for large datasets. These characteristics makes them a popular choice over other heuristic based classifiers. (Burges, 1998; Byvatov et al., 2003; Suykens et al., 2002). The previous works are classified in Table 4.12.

*Average Run Length (ARL) Based Measures*

In addition to data mining based evaluation we employed ARL based measures. The ARL many "faulty samples" does a process need to produce, on average, in order to make sure that an anomaly has been detected. In the CCPR framework we use the Average Target Pattern Run Length (ATPRL) (Hwarng & Hubele, 1991) and consists of the average number of samples needed for discovering an abnormal pattern. Since ATPRL can only be computed for discovered abnormal patterns one needs to take into account the rate of abnormal pattern discovery. For this, here we use the *Average Run Length Index* (ARLIDX) (Hwarng & Hubele, 1991) which equals to the fraction of ATPRL divided by the discovery rate of abnormal patterns. It is worth noting that when classification accuracy equal $100\%$ the two measures, ATPRL and ARLIDX, are equivalent.

Table 2.2: Literature review on CCPR using support vector machine method: M: Multivariate, C: Correlated, NC: Non-correlated

| **Author(s) (Year)** | **Performance measures** | **Validation data Assumption** | **Data** | **Input Representation** | **Benchmark Comparison** |
|---|---|---|---|---|---|
| Chinnam (2002) | Error Visu | Real data | M,C & NC | Raw | $T^2$ charts |
| Sun & Tsung (2003) | Visu | Real data | M i.i.d | Raw | $T^2$ charts |
| Kumar et al. (2006) | Visu | Real data | M & C | Raw | $T^2$ charts |
| Zhang et al. (2007) | Acc Error | Real data | M & C | Raw | $T^2$ charts |
| Camci et al. (2008) | Acc Error | Real data | M & NC | Raw | $T^2$ charts MLP & SVM |
| Cheng et al. (2009) | Acc | Real data | M & C | Feature | $T^2$ charts |
| Sukchotrat et al. (2010) | Error | Simulation | M & C | Raw | $T^2$ charts |
| Chongfuangprinya et al. (2011) | Error Acc ARL | Simulation | M & C | Raw | PoC, $T^2$ charts |
| Lin et al. (2011) | ARL | Simulation | M & | Raw | LVQN, BPN |

The ARL based measures are important especially for applications where the production of each sample is cost and labor intensive. Ultimately one wants to detect an anomaly with the lower ATPRL possible.

The class imbalance problem has a widespread range of applications in business. The following examples explain certain business cases that imbalanced classification problems occur[1]

- **Fraud detection**. The rate of fraud event is growing extremely along with the development of modern technology and communication, yielding the loss of millions of dollars each year. Finding a solution for this problem is tremendously expensive for numerous business associations. Organizations try to identify fraud by monitoring the suspicious transactions. Though, there are more reliable users than fraudulent examples in transaction information. There are different types of fraud based on the financial institution's products and technologies (Yue et al., 2007) including, transaction products: credit and debit cards and checks, technologies: ATM and Internet, and so on. The detection of credit card fraud, telecommunication or cellular fraud, online banking fraud, and insurance fraud has significant importance, since these types of fraud are more likely to happen.

  - Credit card fraud detection: A range of techniques has been used to address this problem such as k reverse nearest neighbors (kRNNs) concept for eliminating extreme outliers and hybrid sampling technique (Padmaja et al., 2007).
  - Online banking fraud detection: the majority of online banking fraud problems deal with online banking transaction data sets with these characteristics and challenges: (1) highly imbalanced large data set; (2) real time detection; (3) dynamic fraud behavior; (4) weak forensic evidence; and (5) diverse customer behavior patterns. The techniques have been used in the literature for imbalanced classification problems include: Cost-sensitive neural network (Wei et al., 2012) and random Forests (Breiman, 2001). Wei et al. (2012) designed cost-sensitive neural network for the online banking scenario

[1]Razzaghi, T., Xanthopoulos, P., and Otero, A. (2013). "Imbalanced Classification: Methods and Applications in Business Analytics". In Encyclopedia of Business Analytics and Optimization (pp. Accepted). *IGI Global*.

which is a modified neural network-based scoring method. A decision forest is a modified version of classic decision tree methods for imbalanced data in building a scoring model, which consists of multiple strong decision trees.

  - Insurance fraud detection: There are few papers tackling insurance fraud detection with the techniques including, backpropagation (BP), together with naive Bayesian (NB) and C4.5 algorithms on preprocessed data with minority oversampling (Phua et al., 2004), hybrid undersampling approach along with kRNN and K-means algorithms (Vasu & Ravi, 2011).
  - Telecommunication/Cellular fraud detection (Fawcett & Provost, 1997; Walters & Wilkinson, 1994): In the United States, the telecommunications industry loses a huge amount of money each year (Steward, 1997).

- **Customer relationship management (CRM)**. Identifying probable contributors or customers is of great importance for a company's sales, profits, and improvement. In recent years, academic researchers and customer data analyzers have focused on developing the related databases and data analysis techniques. Olson (2007) reviewed the applications of data mining in CRM. However, there are few works dedicated to classification of imbalanced data in CRM (Kim et al., 2012; Tu et al., 2011). In fact, most datasets in the real world are more likely to be imbalanced while a binary variable is used for prediction (i.e. 1 for purchase and 0 for no purchase); and the proportion of 1 in the datasets is too small. The techniques commonly used include the combined approach of SVM with undersampling(Kim et al., 2012), cost-based version of bayesian network classification (Tu et al., 2011), and Weighted random forests (Burez & Van den Poel, 2009). One of the interesting topics in CRM is churn prediction which has become one of the main challenges of many companies (Alberts et al., 2006; Chandar et al., 2006).

  - Churn prediction. Customer churn is the tendency of customers to terminate a service offered by a company (e.g. bank, financial institution and so on) in a given period of

time. A churn prediction model will help a company to identify the customers at risk. Customer churn is a frequently rare event in service industries (Gupta et al., 2006) and due to the imbalance in the data distribution; churn prediction is a crucial yet challenging problem to address. Different classification techniques have been applied for the imbalanced churn prediction problem, such as weighted random forest and logistic regression (Burez & Van den Poel, 2009), a random forests together with the sampling techniques and cost-sensitive learning (Xie et al., 2009), hybrid undersampling approach along with kRNN and K-means algorithms (Vasu & Ravi, 2011).

- **Marketing**. Database marketing models is one of the popular classification problems in the business domain, which aim to classify customers into buyers and non-buyers usually through using probabilistic models. (Duman et al., 2012; Cui et al., 2008; Duman et al., 2012; Ling & Li, 1998). Recently database marketers have increasingly adopted new methods and models.

- **Risk management**.There is few works addressed imbalanced classification in risk management. The initial work done by Ezawa et al. (1996) implemented Bayesian network model learning for predicting uncollectibles in telecommunications risk-management using imbalanced datasets. A more recent work (Wei et al., 2012) have implemented an online banking risk management system using a risk scoring method. In their system, a voting method is combined the scores from three models of contrast pattern mining, cost-sensitive neural network and decision forest.

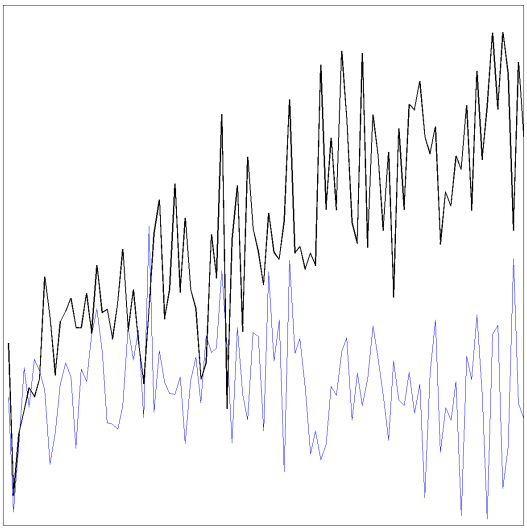
- **Stock market prediction**. Financial time series are intrinsically noisy and non-stationary (Bao et al., 2005). The information which is not involved in the model is considered as noise. Since, usually comprehensive information from the past behavior of financial markets is unavailable to thoroughly obtain the relation between future and past prices. For prediction of stock prices and stock selection, the ANN method has been widely used, but there is always a drawback with this method where the data is imbalanced and includes noise and

outliers. In these problems, a range of specific methods for imbalanced prediction problem is needed such as fuzzy support vector machine regression.

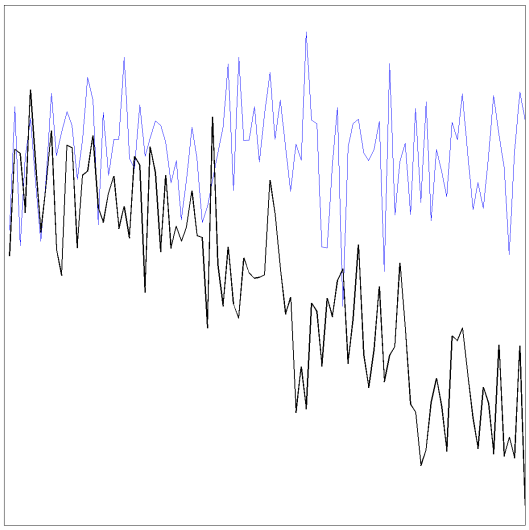
Some previous studies that addressed the imbalanced classification problems in business applications are classified in Table 2.3.

Table 2.3: Imbalanced classification problems in business applications

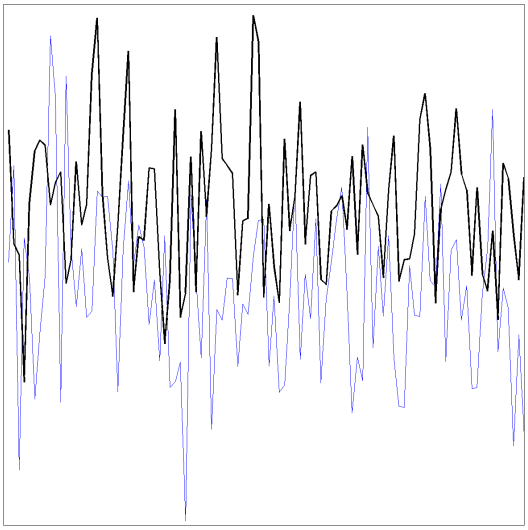
| Authors /Year | Imbalanced Classification | Business Domain | Performance Measure |
|---|---|---|---|
| Phua et al. (2004) | Back propagation combined with Naive Bayesian, C4.5, & oversampling | Fraud detection | Accuracy |
| Padmaja et al. (2007) | K reverse nearest neighbors & hybrid resampling technique | Fraud detection | True positive rate & true negative rate |
| Perols (2011) | Logistic regression, SVM, ANN, C4.5, & stacking | Fraud detection | Estimated relative costs of misclassification (ERC) |
| Wei et al. (2012) | Cost-sensitive neural network, decision forest, & contrast pattern mining | Fraud detection | Accuracy |
| Tu et al. (2011) | Cost-based bayesian network | CRM | AUC & sensitivity |
| Kim et al. (2012) | SVM with random undersampling | CRM | Accuracy, sensitivity & specificity |
| Burez & Van den Poel (2009) | Undersampling, gradient boosting, & weighted random forests | Churn prediction | AUC & lift |
| Xie et al. (2009) | Balanced random forests, resampling techniques, & cost-sensitive learning | Churn prediction | Lift curve & Top-Decile lift |
| Duman et al. (2012) | Logistic regression, ANN, Chi-squared automatic & interaction detector algorithm | Marketing | Accuracy, AUC & precision (Hit rate) |
| Vasu & Ravi (2011) | Hybrid undersampling with KRNN & K-means | Insurance fraud detection & churn prediction | Sensitivity, specificity, AUC & accuracy |
| Ezawa et al. (1996) (1996) | Bayesian network learning | Risk management | ROC |
| Bao et al. (2005) | Fuzzy support vector machines regression (FSVR) | Stock market predication | Normalized mean -squared error (NMSE) |



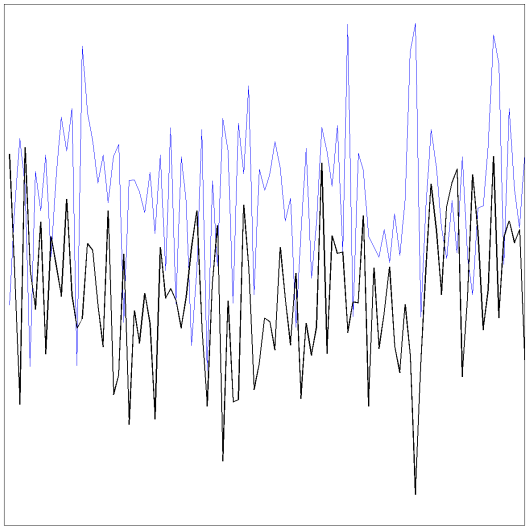
(a) Up trend



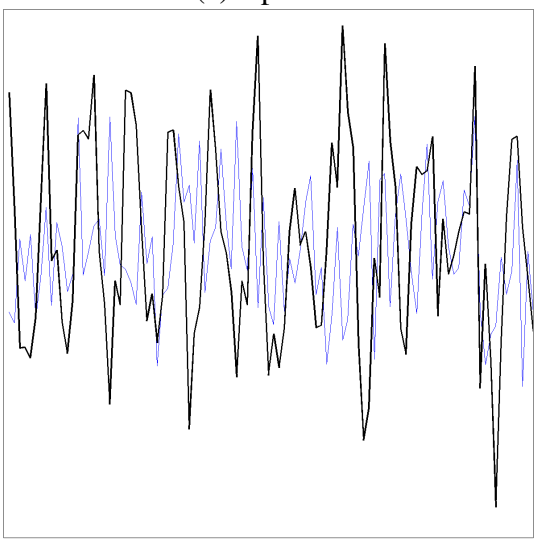
(b) Down trend



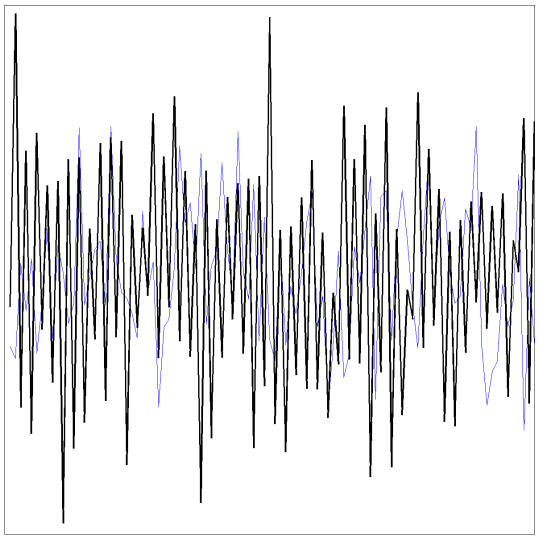
(c) Up shift



(d) Down shift

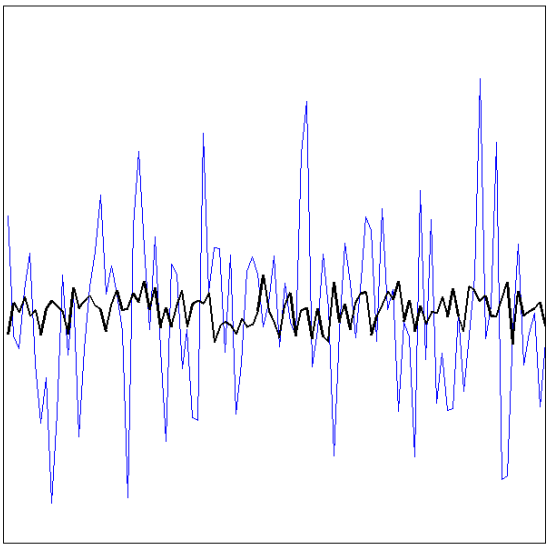


(e) Cyclic



(f) Systematic

Figure 2.4: Examples of six abnormal patterns (bold) plotted versus an example of normal one



(g) Stratification

Figure 2.5: Examples of stratification abnormal pattern (bold) plotted versus an example of normal one

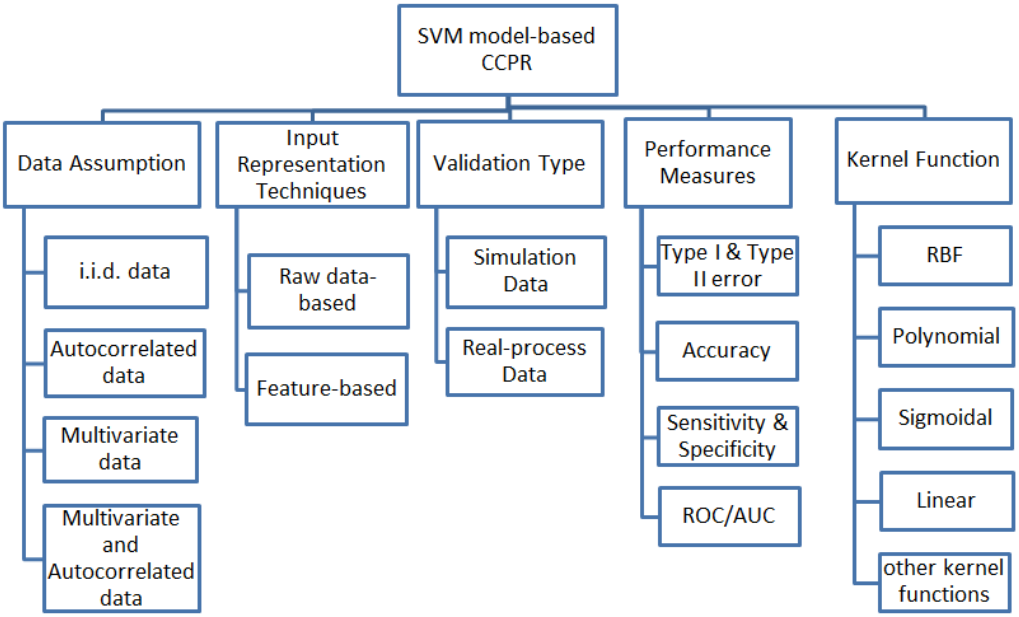



Figure 2.6: Conceptual scheme for classification of imbalanced data

# CHAPTER 3: METHODOLOGY

## Support Vector Machines

Support Vector Machine (SVM) is a popular supervised learning algorithm originally proposed by Vapnik (2000). It has been used in many real-world problems such as text categorization (Joachims, 1998; Pilászy, 2005), image classification (Chapelle et al., 1999; Foody & Mathur, 2004), bioinformatics (including protein classification and cancer classification) (Leslie et al., 2002; Zavaljevski et al., 2002; Guyon et al., 2002) and hand-written character recognition (Bahlmann et al., 2002). Originally, the SVM is designed to solve binary classification problems, but multi-class extensions are also available.

Originally SVM has been developed to solve linearly separable problems. However, it is possible to generalize them in order to classify non-linear problems by employing the *kernel trick* (Cristianini & Shawe-Taylor, 2000). The intuition behind *kernel trick* is that the original data points are projected into a higher dimensional feature space in which they can be separated by a linear classifier. The projection of a linear classifier on the feature space can be non-linear in the original space. In order to use SVM, each data point is required to be a real value. If the attributes are categorical, then they should be transformed into numeric values.

Assume that a dataset is represented by a set $\mathcal{J} = \{(x_i, y_i)\}_{i=1}^{l}$ where $(x_i, y_i) \in \mathbb{R}^{n+1}$, $l$ and $n$ are the number of samples and features, respectively, and each $x_i$ is a sample with n features and a class label $y_i \in \{-1, 1\}$. The SVM classifies the data points by identifying a separating hyperplane whose distance is maximum with respect to the data points of each class. The separation hyperplane defined by the parameters $w$ and $b$ can be obtained by solving the

following convex optimization problem (Cortes & Vapnik, 1995):

$$\min \ \frac{1}{2} \|w\|^2 \tag{3.1a}$$

$$\text{s.t.} \ \ y_i(w^T \phi(x_i) + b) \geq 1 \qquad\qquad i = 1, \ldots, l \tag{3.1b}$$

where $\phi$ is the kernel function $\phi : \mathbb{R}^n \to \mathbb{R}^m$ where $m \geq n$, i.e. each training sample $x_i$ is mapped into a higher dimensional space by the function $\phi$. For linear SVM, we have $\phi(x_i) = x_i$. Then, the class $y_u$ of an arbitrary unknown point $x_u$ is assigned based on the following rule:

$$y_u = sgn\{w^T \phi(x_u) + b\}, \tag{3.2}$$

where $sgn\{\cdot\}$ is the sign function. This formulation is known as *hard margin* SVM because it requires that the two classes to be separable through a classification hyperplane. If the classification problem is non-separable, then Problem 3.1 is infeasible. In this case, slack variables $\xi_i$, $i \in \{1, \ldots, l\}$ are added to the objective function whose goal is to allow but penalize misclassified points. This approach is known as *soft margin* SVM and the corresponding quadratic programming problem can be formulated as (Cortes & Vapnik, 1995):

$$\min \ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{l} \xi_i \tag{3.3a}$$

$$\text{s.t.} \ \ y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \qquad\qquad i = 1, \ldots, l \tag{3.3b}$$

$$\xi_i \geq 0 \qquad\qquad i = 1, \ldots, l \tag{3.3c}$$

The parameter $C$ controls the magnitude of penalization. The soft margin formulation converges to the hard margin as $C \to +\infty$. Many algorithms, such as sequential minimal optimization (SMO), operate on the Lagrangian dual problem instead of Problem 3.3 for faster and more stable

convergence. The Lagrangian dual of 3.3 will be:

$$\max \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{3.4a}$$

$$\text{s.t.} \sum_{j=1}^{l} \alpha_i y_i = 0 \qquad i = 1, \ldots, l \tag{3.4b}$$

$$0 \leq \alpha_i \leq C \qquad i = 1, \ldots, l \tag{3.4c}$$

where $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is the kernel function that measures the similarity between two arbitrary points.

### Weighted Support Vector Machines

In previous formulation (3.4a-3.4c), all data points are given the same importance in the training process. This might not be desirable especially in the case that one class contains outliers or in the case that one class contains considerably less point than the other. For this reason, a modified version of soft margin SVM has been proposed for making the training process more flexible. Suppose we are given a set of labeled samples with corresponding weights represented by the set $\mathcal{J}' = \{(x_i, y_i, s_i)\}_{i=1}^{l}$. Each training sample $(x_i, y_i) \in \mathbb{R}^{n+1}$ is associated to a given label $y_i \in \{-1, 1\}$ and the corresponding weight $0 \leq s_i \leq 1$ with $i = 1, \ldots, l$. The optimal hyperplane is again identified from the solution of the optimization problem (Lin & Wang, 2002).

$$\min \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{l} s_i \xi_i \tag{3.5a}$$

$$\text{s.t. } y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \qquad i = 1, \ldots, l \tag{3.5b}$$

$$\xi_i \geq 0 \qquad i = 1, \ldots, l \tag{3.5c}$$

Through WSVM one can assign different weights to each data sample based on a predetermined importance measure. This provides a more flexible scheme compared to SVM where the overall penalization magnitude $C$ is the only parameter. For the special case of imbalanced binary classification, Veropoulos et al. (1999) proposed the usage of different costs associated with the positive ($C^+$) and negative ($C^-$) class

$$\min \ \frac{1}{2}\|w\|^2 + C^+ \sum_{\{i|y_i=+1\}}^{n_+} \xi_i + C^- \sum_{\{j|y_j=-1\}}^{n_-} \xi_j \tag{3.6a}$$

$$\text{s.t. } y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i \qquad i = 1, \ldots, l \tag{3.6b}$$

$$\xi_i \geq 0 \qquad i = 1, \ldots, l \tag{3.6c}$$

The problem 6 to find the optimal hyperplane is a Quadratic programming problem, which can transformed into the Lagrangian dual with the Kuhn-Tucker conditions. The Lagrangian dual is given by:

$$\max \ \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i\alpha_j y_i y_j K(x_i, x_j) \tag{3.7a}$$

$$\text{s.t. } \sum_{j=1}^{l} \alpha_i y_i = 0 \qquad i = 1, \ldots, l \tag{3.7b}$$

$$0 \leq \alpha_i \leq C^+ \qquad \text{if } y_i = +1 \text{ and } i = 1, \ldots, l \tag{3.7c}$$

$$0 \leq \alpha_i \leq C^- \qquad \text{if } y_i = -1 \text{ and } i = 1, \ldots, l \tag{3.7d}$$

The role of the weighting parameters $C^+$ and $C^-$ is to assign different "importance" to the misclassification of the positive and negative class. In this way the minority class becomes more important in terms of objective function value. As it is shown in Figure 2.6, WSVM classifies correctly minority class examples.

Let $\boldsymbol{x}_i \in \mathbb{R}^n$ denote sample $i$ and $y_i \in \{-1, 1\}$ its class label. Let the set of sample indices for the positive and negative classes be $I^+$ and $I^-$, respectively, and let $I = I^+ \cup I^-$. The RSVM model is given in Formulation (3.8). The fundamental idea behind RSVM is to provide a restricted amount $n\ \Upsilon$ of unpenalized (free) slack for samples that may hinder the classification performance, where $\Upsilon$ is a parameter that determines average slack per sample. Free slack is distributed to samples via the variables $\upsilon$ in Formulation (3.8), however, without differentiating between positive and negative classes, which may become problematic with unbalanced data. Moreover, the penalty term applies equally to positive and negative classes in RSVM regardless of relative class sizes.

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\upsilon}} \ \frac{1}{2}\langle \boldsymbol{w} \cdot \boldsymbol{w} \rangle + \frac{C}{2n} \sum_{i \in I} {\xi_i}^2 \tag{3.8a}$$

$$\text{s.t. } y(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i - \upsilon_i, \ \ \forall i \in I \tag{3.8b}$$

$$\sum_{i \in I} \upsilon_i \leq n\ \Upsilon \tag{3.8c}$$

$$\upsilon_i \geq 0, \ \ \forall i \in I \tag{3.8d}$$

The WRSVM model[1], which is given in Formulation (3.9), combines the cost-sensitive ap-

[1]Şeref, O., Razzaghi, T., and Xanthopoulos, P., "A Weighted Relaxed Support Vector Machine Method". Submitted to *Expert Systems with Applications, 2014.*

proach of WSVM and the relaxation approach of RSVM.

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\upsilon}} \ \frac{1}{2}\langle \boldsymbol{w}\cdot\boldsymbol{w}\rangle + \frac{C}{2n^+}\sum_{i\in I^+}{\xi_i}^2 + \frac{C}{2n^-}\sum_{i\in I^-}{\xi_i}^2 \tag{3.9a}$$

$$\text{s.t. } \langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle + b \geq 1-\xi_i-\upsilon_i, \ \ \forall i\in I^+ \tag{3.9b}$$

$$-\langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle - b \geq 1-\xi_i-\upsilon_i, \ \ \forall i\in I^- \tag{3.9c}$$

$$\sum_{i\in I^+}\upsilon_i \leq n^+\ \Upsilon \tag{3.9d}$$

$$\sum_{i\in I^-}\upsilon_i \leq n^-\ \Upsilon \tag{3.9e}$$

$$\upsilon_i \geq 0, \ \ \forall i\in I \tag{3.9f}$$

In Formulation (3.9), $n^+$ and $n^-$ are the sizes of the *majority* and *minority* class, respectively. Free slack for sample $i$ is denoted with the variable $\upsilon_i$ in constraints (3.9b) for positive samples and in (3.9c) for negative samples. Due to imbalance, we provide separate amounts of total free slack for the positive and the negative classes in constraints (3.9d) and (3.9e), respectively, parameterized by $\Upsilon$, which is the free slack provided per sample.

The Lagrangian function for Formulation (3.9) can be written as,

$$\mathcal{L}(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\upsilon},\boldsymbol{\alpha},\beta^+,\beta^-,\boldsymbol{\lambda}) = \tag{3.10a}$$

$$\begin{aligned}
&\tfrac{1}{2}\langle \boldsymbol{w}\cdot\boldsymbol{w}\rangle + \frac{C}{2n^+}\textstyle\sum_{i\in I^+}^{n}{\xi_i}^2 + \frac{C}{2n^-}\sum_{i\in I^-}{\xi_i}^2 \\
&-\sum_{i=1}^{n}\alpha_i^+\left(\langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle + b - 1 + \xi_i + \upsilon_i\right) \\
&-\sum_{i=1}^{n}\alpha_i^-\left(-\langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle - b - 1 + \xi_i + \upsilon_i\right) \\
&-\beta^+\left(n^+\Upsilon - \textstyle\sum_{i\in I^+}\upsilon_i\right) - \sum_{i\in I^+}\lambda_i\upsilon_i \\
&-\beta^-\left(n^-\Upsilon - \textstyle\sum_{i\in I^-}\upsilon_i\right) - \sum_{i\in I^-}\lambda_i\upsilon_i,
\end{aligned} \tag{3.10b}$$

where $\boldsymbol{\alpha}$, $\beta$ and $\boldsymbol{\lambda}$ are the Lagrangian multipliers. Since Formulation (3.9) is a convex problem, its Wolfe dual can be obtained from the following stationary first order conditions of the primal variables $\boldsymbol{w}$ , $b$, $\boldsymbol{\xi}$, and $\boldsymbol{\upsilon}$.

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \boldsymbol{w} - \textstyle\sum_{i \in I^+} \alpha_i \boldsymbol{x}_i + \sum_{i \in I^-} \alpha_i \boldsymbol{x}_i = 0 \tag{3.11a}$$

$$\frac{\partial \mathcal{L}}{\partial b} = - \textstyle\sum_{i \in I^+} \alpha_i + \sum_{i \in I^-} \alpha_i = 0 \tag{3.11b}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = \tfrac{C}{n^+} \xi_i - \alpha_i = 0, \forall i \in I^+ \tag{3.11c}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = \tfrac{C}{n^-} \xi_i - \alpha_i = 0, \forall i \in I^- \tag{3.11d}$$

$$\frac{\partial \mathcal{L}}{\partial \upsilon_i} = \beta^+ - \alpha_i - \lambda_i = 0, \forall i \in I^+ \tag{3.11e}$$

$$\frac{\partial \mathcal{L}}{\partial \upsilon_i} = \beta^- - \alpha_i - \lambda_i = 0, \forall i \in I^- \tag{3.11f}$$

Substituting the equivalent expressions for $\boldsymbol{w}$ , $b$, $\boldsymbol{\xi}$, $\boldsymbol{\upsilon}$ from equations (3.11a) - (3.11e) back in expression (3.10b), the Wolfe dual can be written as follows:

$$\begin{aligned} \max_{\boldsymbol{\alpha}, \beta^+, \beta^-} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \boldsymbol{x}_i \cdot \boldsymbol{x}_j \rangle \\ & - \frac{n^+}{2C} \sum_{i \in I^+} \alpha_i^2 - \frac{n^-}{2C} \sum_{i \in I^-} \alpha_i^2 \\ & - n^+ \Upsilon \beta^+ - n^- \Upsilon \beta^- \end{aligned} \tag{3.12a}$$

$$\text{s.t.} \quad - \sum_{i \in I^+} \alpha_i + \sum_{i \in I^-} \alpha_i = 0 \tag{3.12b}$$

$$0 \leq \alpha_i \leq \beta^+ \quad \forall i \in I^+ \tag{3.12c}$$

$$0 \leq \alpha_i \leq \beta^- \quad \forall i \in I^-. \tag{3.12d}$$

The dot products $\langle \boldsymbol{x}_i \cdot \boldsymbol{x}_j \rangle$ in Formulation (3.12) can be replaced by a kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for nonlinear classification. The optimal hyperplane $\boldsymbol{w}$ can be found using Equation (3.11a).

**Lemma 1.** *If $||\boldsymbol{w}|| > 0$ in the optimum solution of Formulation* (3.9)*, then the "total free slack" constraints, $\sum_{i \in I^+} \upsilon_i \leq n^+ \Upsilon$ and $\sum_{i \in I^-} \upsilon_i \leq n^- \Upsilon$, are always binding, i.e., the total free slack for both classes are always consumed.*

*Proof.* The proof follows from the complementary slackness conditions that $\beta^+ > 0$ and $\beta^- > 0$ for any solution such that $||\boldsymbol{w}|| > 0$ due to Equation (3.11a) and constraints (3.12c) and (3.12d). Thus, constraints (3.9d) and (3.9e) are binding, i.e., total free slack amounts $n^+ \Upsilon$ and $n^- \Upsilon$ are always consumed completely. □

We now show that there are two margins parallel to the separating hyperplane, which are determined by the maximum penalty for each class, $\xi^+_{max} = \max_{i \in I^+}\{\xi_i\}$ and $\xi^-_{max} = \max_{i \in I^-}\{\xi_i\}$. Samples behind the boundary of this margin do not require free slack, whereas samples beyond this boundary claim some of the free slack reserved for their respective class.

**Theorem 1.** *Let $\xi^+_{max} = \max_{i \in I^+}\{\xi_i\}$ and $\xi^-_{max} = \max_{i \in I^-}\{\xi_i\}$ be the maximum penalties for the positive and negative classes, respectively, in the optimum solution to Formulation* (3.9)*. Then,*

1. *$\upsilon_i = 0$ for any sample such that $\xi_i < \xi^+_{max}$, $\forall i \in I^+$ and $\xi_i < \xi^-_{max}$, $\forall i \in I^-$, and*
2. *$\xi_i = \xi^+_{max}$ for $\upsilon_i > 0$, $i \in I^+$ , and $\xi_i = \xi^-_{max}$ for $\upsilon_i > 0$, $i \in I^+$,.*

*Proof.* Without loss of generality, assume that there exist a sample $i \in I^+$ such that $\xi_i < \xi^+_{max}$ and $\upsilon_i > 0$ in the optimal solution to Formulation (3.9). Let $I^+_{max}$ be the set of samples with penalty $\xi_j = \xi^+_{max}$ for all $j \in I^+_{max}$. Let $|I^+_{max}| = k$. Let the penalty difference between samples in $I^+_{max}$ and the next highest penalty of a sample be $\delta = \xi^+_{max} - \xi_{t^*}$, where $t^* = \arg\max\{\xi_t : t \in I^+ \setminus I^+_{max}\}$. From Lemma 1, one can shift $\delta_{min} = min\{\upsilon_i, \delta\}$ amount of the free slack from sample $i$ over to the samples in $I^+_{max}$ such that the new penalty values for the samples in $I^+_{max}$ are $\xi'_j = \xi_j - \delta_j$,

where $\delta_{min} = \sum_{j \in I^+_{max}} \delta_j$. Let $\Delta Z$ be the reduction in the total penalty. Then,

$$\begin{aligned}
\Delta Z &= k \, {\xi^+_{max}}^2 + \xi_i^2 \\
&\quad - \left( \textstyle\sum_{j \in I^+_{max}} (\xi^+_{max} - \delta_j)^2 + (\xi_i + \delta_{min})^2 \right) && \text{(3.13a)} \\
&= 2\delta_{min}(\xi^+_{max} - \xi_i) - \delta^2_{min} - \sum_{j \in A_{max}} \delta_j^2 && \text{(3.13b)} \\
&\geq 2\delta_{min}(\xi^+_{max} - (\xi_i + \delta_{min})) > 0, && \text{(3.13c)}
\end{aligned}$$

which contradicts with the optimality of the solution. The reduction in (3.13b) is maximized when $\delta_j = \frac{\delta_{min}}{k}$ for all $j \in A_{max}$ with a new maximum penalty value $\xi^{+\prime}_{max} = \xi_{max} - \frac{\delta_{min}}{k}$. From Lemma 1, the corresponding new free slack values are $\upsilon'_j = \upsilon_j + \frac{\delta_{min}}{k}$ for all $j \in A_{max}$. This shows that all free slack is consumed by the samples with maximum penalty, thus proving item (2) for the positive class case. The proof for the negative class is along the same lines as the positive class. □

From Theorem 1, the samples with positive free slack have the maximum penalties of each class. This implies functional margins of $\gamma^+_r = 1 - \xi^+_{max}$ for the positive class and $\gamma^-_r = 1 - \xi^-_{max}$ for the negative class. Note that constraints (3.9b) and (3.9c) with penalty $\xi_{max} > 0$ are binding, and therefore, free slack for these constraints, and the corresponding samples would be equal to $\upsilon_j = \max\{0, \gamma_r - y_j(\langle \boldsymbol{w} \cdot \boldsymbol{x}_j \rangle + b)\}$, where $\gamma_r = \gamma^+_r$ for $i \in I^+$ and $\gamma_r = \gamma^-_r$ for $i \in I^-$. In other words, the functional distance from a sample beyond this margin and the margin itself is not penalized. Without loss of generality, replacing sample $\boldsymbol{x}_j$ that has a positive slack with a new sample $\boldsymbol{x}'_j = \boldsymbol{x}_j - \upsilon_j(\boldsymbol{w}/||\boldsymbol{w}||)$ does not change the optimal solution. This is equivalent to pushing such samples back to the margins $\gamma^+_r$ and $\gamma^-_r$ for the positive and negative classes, respectively, along the direction of the normal vector $\boldsymbol{w}$. This result implies that some of the most influential support vectors are relaxed due to the free slack and are pushed back to a distance from the hyperplane, which is determined by the maximum penalty for each class.

From constraints (3.12c) and (3.12d) and equations (3.11e) and (3.11f), we can estblish the equivalences $\beta^+ = \frac{C}{n^+}\xi^+_{max}$ and $\beta^- = \frac{C}{n^-}\xi^-_{max}$. Note that variables $\beta^+$ and $\beta^-$ are upperbounds and they are penalized in the objective function by the total free slack amounts $n^+\Upsilon$ and $n^-\Upsilon$, respectively. Thus, higher amounts of free slack will impose lower values of $\beta^+$ and $\beta^-$, which are direcly proportional to maximum penalties $\xi^+_{max}$ and $\xi^-_{max}$. Such a change will imply that the support vectors are receding further toward their classes with increasing total free slack amounts.

Without loss of generality, for a positive sample $i$ such that $0 < \alpha_i < \beta^+$, constraint (3.9b) is tight for sample $i$ and Theorem 1 implies that $\upsilon_i = 0$. Then, the bias $b$ for the separating hyperplane can be calculated as follows,

$$b = y_i \left(1 - \frac{n^+}{C}\alpha_i\right) - \sum_{j \in I} y_j \alpha_j \langle \boldsymbol{x}_i \cdot \boldsymbol{x}_j \rangle. \tag{3.14}$$

For excessive amounts of free slack for the positive class, we may have $\upsilon_i > 0$ for all $i \in I^+$. In this case all constraints (3.9b) are tight. Then using Lemma 1, we can calculate the bias $b$ as follows,

$$b = 1 - \frac{1}{C}\sum_{i \in I^+} \alpha_i - \Upsilon - \frac{1}{n^+}\sum_{i \in I^+}\sum_{j \in I} y_i y_j \alpha_j \langle \boldsymbol{x}_i \cdot \boldsymbol{x}_j \rangle. \tag{3.15}$$

Similar arguments can be used with the negative samples.

## Model Selection for Support Vector Machines

The SVM, WSVM and WRSVM algorithms have certain parameters that need to be tuned during the training phase; in particular, cost and kernel parameters for SVM and WSVM and in addition the total free slack parameter ($\Upsilon$) for WRSVM. For this, we use an adapted nested uniform design model selection algorithm. Uniform designs (UD) algorithms have been proposed for supervised learning model selection Huang et al. (2007) and have been identified as more

efficient and robust compared to the uniform grid search model selection **?**. The intuition behind this approach is to search the parameter space by exploring the points that minimize a *discrepancy* function between their empirical distribution and the theoretical uniform distribution. This process can be applied iteratively in a nested manner in order to identify the close-to-optimal parameter set. The optimal points are selected based on some classification performance measure. Here, we use the $L_2$-discrepancy measure. For a set of points $P = \{x_1, \ldots, x_n\}$ over a parameter space $\mathcal{A} \subset \mathbb{R}^m$, where $m$ is the number of parameters, it is defined by,

$$D_2(\mathcal{A}, P) = \left[\int_{\mathcal{A}} |F(x) - F_e(x)|^2 dx\right]^{1/2}, \tag{3.16}$$

where $F(x)$ is the uniform cumulative density function (c.d.f) and $F_e(x)$ is the empirical c.d.f. This technique is a multiple stage procedure that performs a parameter space search in order to determine the near-optimal parameter. Since the test instances are imbalanced we select the optimal parameter set based on the highest *G-mean* value. Formally the model selection process can be described as follows:

1. Choose parameter search domain and number of levels (factors) for each parameter.
2. Choose a suitable UD table **?** to accommodate the number of parameters and levels. Here the UD tables are built under the centered $L_2$-discrepancy.
3. From the UD table, randomly determine the run order of experiments and estimate performance measure for each one of them (here *G-mean*).
4. Refine the search around the point with the highest performance measure value by applying steps 1–3.

# CHAPTER 4: RESULTS

We present our result in this section into two subsections. In the first subsection, we explain the results of WSVM and SVM for control chart pattern recognition when data are highly imbalanced. In the second subsection, we explain the results of WRSVM compared to the state-of-art techniques when data is imbalanced and noisy. For this section, we show the performance of the proposed method on benchmark datasets.

## Imbalanced Support Vector Machine for Control Chart Pattern Recognition

### *Binary Classification*

In this section, we present experimental results between SVM and the proposed WSVM [1]. Experiments on both SVM and WSVM were conducted with LIBSVM-3.12 and LIBSVM-weights-3.12 (Chang & Lin, 2011). The LIBSVM was interfaced in MATLAB and the rest of the script was developed in it as well. All experiments are performed on an Intel core i5, 2.3 GHz with 4Gb of RAM in a 64-bit platform. For each classification problem, we generate a total of 1000 data points and for cross validation purposes, $90\%$ of the data was used for training and the rest $10\%$ was used for testing. All data are normalized prior to classification, so that they have zero mean and unitary standard deviation (*zscore()* function in MATLAB was used). Data based on different normal and abnormal patterns are generated (for the mathematical models see appendix). However, there are several kernel functions in the literature for this particular problem, we chose the *Radial Basis Function* (RBF) kernel. This is because the natural distribution of the classes is either normally distributed (for control data) and very close to normally distributed (for abnormal data). The RBF kernel is effective in producing spherical and ellipsoidal decision areas which makes it

[1] Razzaghi, T. and Xanthopoulos, P. (2014), A Weighted Support Vector Machine Method for Control Chart Pattern Recognition. *Industrial and Computer Engineering,* 70, pp. 134-149.

an appropriate kernel candidate for the problem under consideration. For this kernel function, the similarity between two data points $x_i$ and $x_j$ is given by:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma \geq 0. \tag{4.1}$$

For each class, the weights are estimated as the inverse of the class size:

$$C^+ = \frac{C}{n^+}, \;\; C^- = \frac{C}{n^-} \tag{4.2}$$

where $n^+$ and $n^-$ are the size of normal and abnormal class, and $C^+$ and $C^-$ are weights corresponding to the normal and abnormal classes respectively. Note that for balanced problems the weights become equal ($C^+ = C^-$) and the algorithm reduces to SVM. This weight strategy has been employed in a number of previous studies (Liu et al., 2005; Du & Chen, 2005; Huang & Du, 2005; Hwang et al., 2011). We studied the behavior of our classifiers for different values of abnormal trend pattern values as well as for different window lengths $w$ (different number of features). Our goal is to study and implement WSVM and SVM for CCPR and at the same time identify the set of parameters that generate the most challenging problems and also determine what is the minimum number of features (minimum $w$) for which the classifier is trusted. In some sense parameter $w$ is related to the *Average Run Length (ARL)* since it shows how much data do we need in order for the abnormal patterns to be discoverable. In particular, the following experiments are conducted:

1. We compare WSVM against SVM for each abnormal pattern and for different window lengths $w$ and different pattern parameters with respect to *G-mean*. Through this experiment, we identify the parameter values for which, a) the problems easily solvable for both algorithms, b) the problems cannot be solved by neither algorithm and c) the rest (partially separable problems). For this test we consider highly imbalanced problems where $97.5\%$ of the data belong to the normal class and only $2.5\%$ belong to the abnormal.

2. For selected separable (Sep), partially separable (Ps) and inseparable (Is) problems, we perform a statistical test the performance between SVM and WSM. We observe that although the mechanism of generation of the various classes is different, the statistical test failed to reject the null hypothesis except for the case of separable problems where both algorithms perform equally well.

3. For selected instances of (Sep), (Ps) and (Is) problems, we perform detailed and *G-mean* analysis for different types of imbalanced problems. We observe that WSVM performance is highly robust for difference imbalanced instances whereas SVM highly depends on the imbalance ratio. The results are consistent with earlier imbalanced classification literature.

4. We measure the time needed for training and testing of the proposed algorithms in order to determine whether this scheme can be of practical usefulness. In addition to the *ARL* represented by the parameter $w$ it is important to understand whether there is a computational bottleneck. It turns out that the proposed algorithms can handle large amounts of data in reasonable amount of time.

5. We perform a multi-class comparison between multiclass-WSVM and multiclass-SVM with a total of seven classes (one normal and six abnormal). Results are compared in terms of the *confusion matrix* of each classifier. Computational running time for training and testing is reported as well.

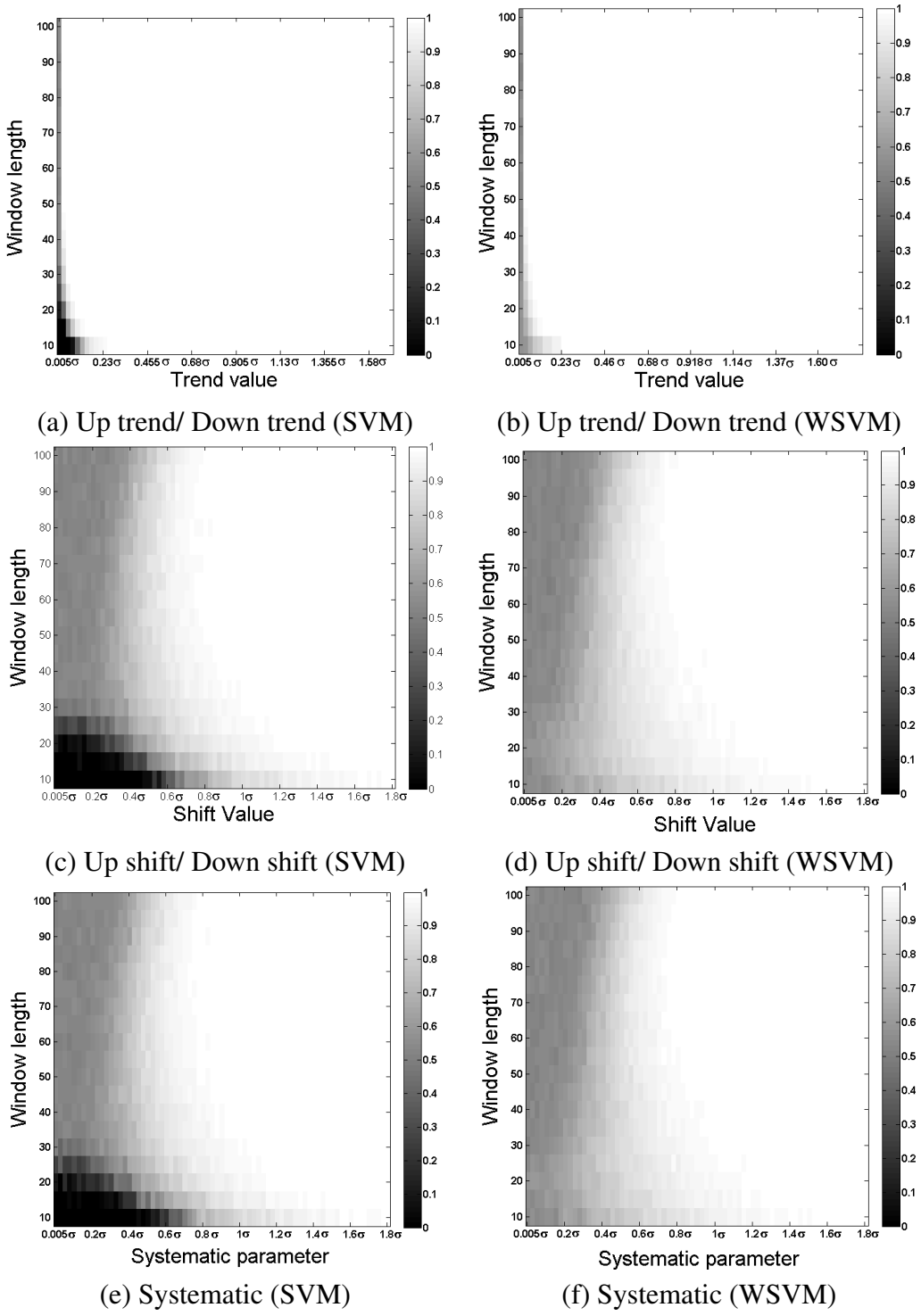
The parameter selection C and $\gamma$ for SVM and WSVM was performed through a uniform grid search over the parameter space. The two classification schemes (SVM and WSVM) were compared in terms of their *G-mean* for different values of the window $w$ and abnormal parameter values Table B.2.

We consider a wide range of window lengths ($w$) and a wide range of abnormal pattern parameters. In general higher $w$ yields less challenging problems however this requires more data since $w$ is the time window of the process. The results are shown in Figures 4.6 & 4.2. The tables with

the all the numerical values along with the sensitivity, specificity and classification accuracy for these experiments can be found in the supplementary material section of this paper. On the other side higher deviation from normal data yields also to easier problems as expected. Ultimately one would like to detect slide parameter shifts with the smallest $w$ possible. It turns out that for these instances WSVM provides a considerable improvement over SVM. We observe that overall in each pattern there are some problems that both SVM and WSVM can solve (white areas), some problems that are partially solvable (gray areas) and some problems that are inseparable (black areas). We can see that WSVM, for most of the cases can solve better the instances where SVM cannot solve. These are typically the ones with low $w$ and low pattern parameter. We can also see that for small pattern parameter changes problems remain partially solvable for both classifiers. This means that the problem remains challenging for small parameter shifts. However such shifts are less likely to cause a severe malfunction. Overall we see that WSVM performs the same or better compared to SVM and further more its behavior is more robust since the performance does not change dramatically with the changes of the parameters. In addition, we observe a symmetric behavior of the algorithms for symmetric trend and shift patterns. In practice, the choice of $w$ depends on the nature of the imbalanced problem as well as the magnitude of parameter shift that one wishes to detect.

Table 4.1: Summary of parameter range for computational experiments

| **Name** | **Symbol** | **Range** |
|---|---|---|
| Window length | $w$ | $[10, 100]$ |
| Process mean (all patterns) | $\mu$ | 0 |
| Standard deviation of normal process | $\sigma$ | 1 |
| Slope (Up/Down trend pattern) | $\lambda$ | $[0.005\sigma, 0.605\sigma]$ |
| Shift (Up/Down shift pattern) | $\omega$ | $[0.005\sigma, 1.805\sigma]$ |
| Standard deviation (Stratification pattern) | $\acute{\varepsilon_t}$ | $[0.005\sigma, 0.8\sigma]$ |
| Cyclic parameter (Cyclic pattern) | $\alpha$ | $[0.005\sigma, 1.805\sigma]$ |
| Systematic parameter (Systematic pattern) | $k$ | $[0.005\sigma, 1.805\sigma]$ |




(a) Up trend/ Down trend (SVM)

(b) Up trend/ Down trend (WSVM)

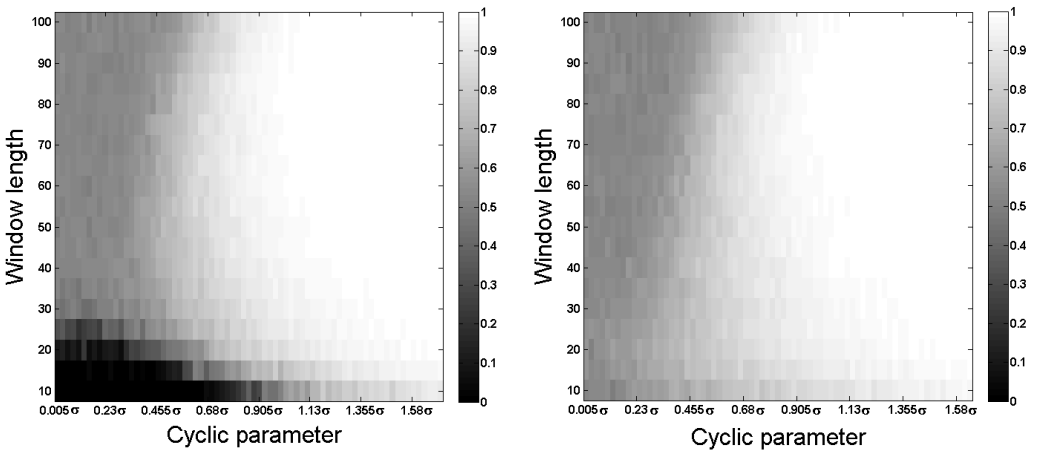
(c) Up shift/ Down shift (SVM)

(d) Up shift/ Down shift (WSVM)

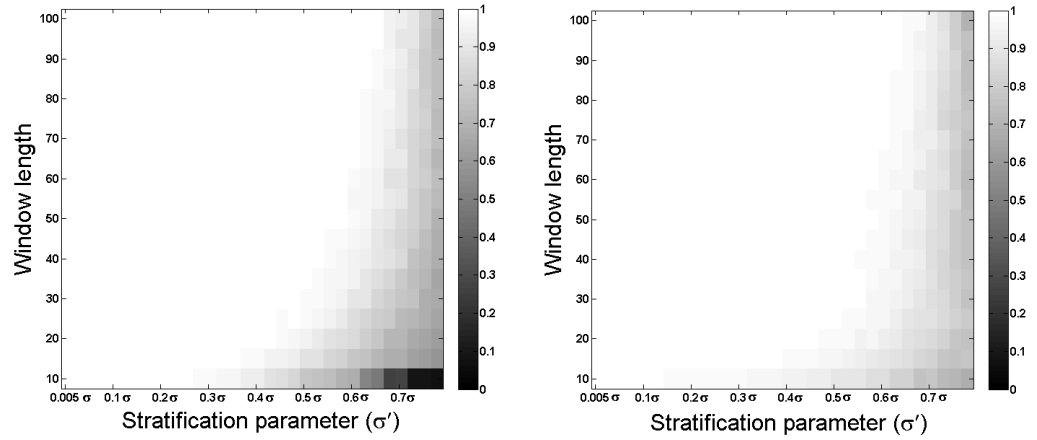
(e) Systematic (SVM)

(f) Systematic (WSVM)

Figure 4.1: Geometric mean of sensitivity for different parameters window lengths and patterns for highly imbalanced data.




(a) Cyclic (SVM) (b) Cyclic (WSVM)




(a) Stratification (SVM) (b) Stratification(WSVM)

Figure 4.2: Boundary obtained for inseparable, partially separable, and separable classification problems for cyclic and stratification patterns

Another necessary aspect to consider is the computational running time of the algorithm. In a practical setting the training phase can occur off-line on the historical data and only the testing or prediction phase will be conducted on-line to generated data. However, if computational time allows, it would be beneficial to retrain the algorithm on-line as new data are generated real time. We recorded the time required for training and testing as a function of the data size (Figure 4.4) and abnormal pattern parameter (Figure 4.3 and Table 4.5). As expected the training and testing time for a fixed data size is negligible (order $\sim 10^{-2}$ sec) compared to the order of training ($\sim 10$ sec) and the order of testing ($\sim 1$ sec). It is noteworthy that these computational times are

based on a non embedded implementation developed for research in MATLAB environment. A potential industrial embedded implementation will propably have much lower computational times. Therefore the times reported in Figure 4.4 & 4.3 and Table 4.5 should be seen as upper bound for a real time implementation.

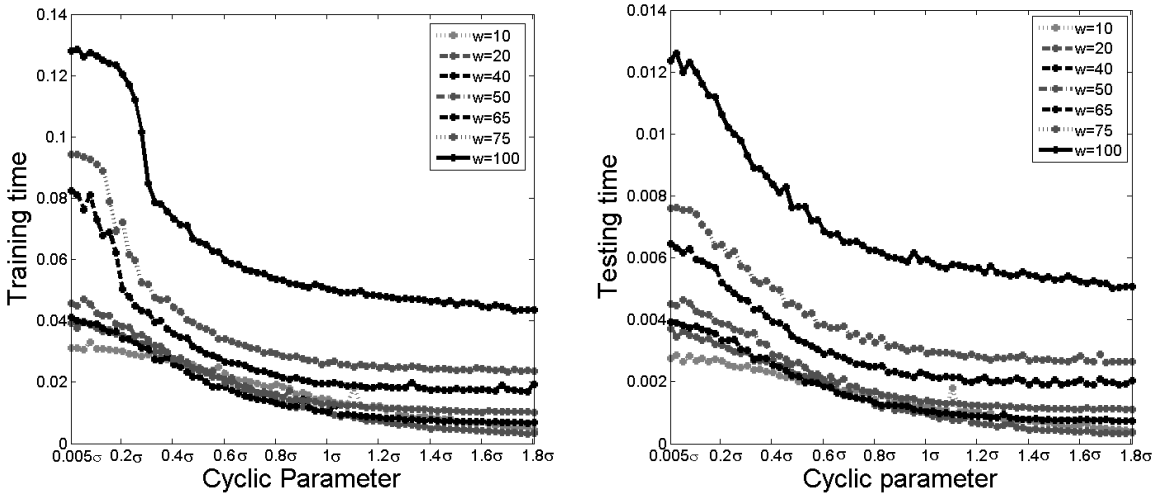



Figure 4.3: WSVM training and testing time vs. abnormal parameter for cyclic patterns. The computation time decreases as the value of the parameter increases. This is expected since higher parameter values make the problem less challenging (more separable).

Table 4.2: The maximum and minimum training and testing time of WSVM for different abnormal patterns

| | Training time | | Testing time | |
|---|---|---|---|---|
| Abnormal pattern | min | max | min | max |
| Uptrend | 0.0007 | 0.0748 | 0.0001 | 0.0174 |
| Downtrend | 0.0006 | 0.0747 | 0.0001 | 0.0172 |
| Upshift | 0.0017 | 0.1594 | 0.0002 | 0.0160 |
| Downshift | 0.0016 | 0.1325 | 0.0002 | 0.0209 |
| Systematic | 0.0014 | 0.1293 | 0.0001 | 0.0234 |
| Cyclic | 0.0028 | 0.1323 | 0.0003 | 0.0248 |
| Stratification | 0.0059 | 0.0218 | 0.0006 | 0.0026 |

Next we select some representative problems from (Sep), (Ps) and (Is) classes and compare them with respect to *accuracy*, *sensitivity*, *specificity* and *G-means*. Sensitivity and accuracy are greatly driven by the majority class and specificity is mostly affected by the correct classification

of the minority class. However, *G-mean* is a measure that considers both sensitivity and specificity and can be a trusted performance metric of imbalanced classification. We can see that *G-mean* consistently improves for all the problems and all the patterns under consideration (Table 4.12). We perform a statistical test in order to measure the significance of WSVM improvement. Our null hypothesis is that the *G-mean* of SVM and WSVM are equal and the alternative hypothesis that they are not. The t-test rejects the null hypothesis in most of the instances except for the separable cases where both SVM and WSVM perform equally well (less challenging instances). We note that in many cases SVM achieves high sensitivity and zero specificity. This implies *null classification* meaning that all the test samples are classified in the same class and is indicative of poor performance. This has been noted in previous studies (Weiss, 2004) and it is in accordance with previous computational results (Anand et al., 2010; Hwang et al., 2011).

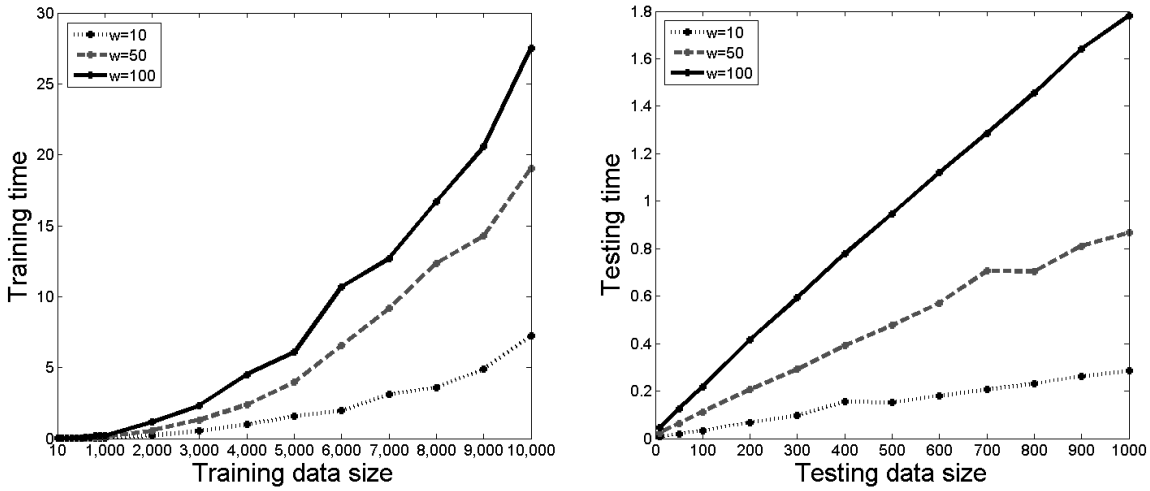



Figure 4.4: WSVM training and testing time vs. training size for cyclic pattern

Next we compare the performance of SVM and WSVM for different imbalanced ratios and representative problems of different difficulty level ((Sep),(Ps) and (Is)). Parameters for each problem are the same as these in Table 4.12. Results are shown in Tables 4.4, 4.5 and 4.6. We consider the size of imbalanced normal and abnormal data as $(50 + r)\%$ and $(50 - r)\%$ respectively. For all instances, and regardless the difficulty level of the problem we can see that WSVM demonstrates

a more robust behavior for various values of $r$. On the other side SVM performs well for low $r$ (more balanced problems and rapidly decreases as $r$ decreases (with only exception the (Sep) problems). When $r$ is low then the two formulations become equivalent and thus the performance is very similar.

Table 4.3: Sensitivity (Sen), Specificity (Spe), Accuracy (Acc), and G-means (G) of SVM and WSVM over all six abnormal patterns for different problems with three types including separable(Se), partially separable (Ps), and inseparable(Is). We define these three types based on SVM classification performance.

| Pattern | SVM | | | | WSVM | | | | | Parameters | Type |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sen | Spe | Acc | G | Sen | Spe | Acc | G | P-value | | |
| Ut | **1.00** | 0.00 | **0.93** | 0.00 | 0.64 | **0.54** | 0.63 | **0.57** | $< 10^{-4}$ | (w=10, $\lambda$=0.005) | Is |
| | **0.99** | 0.44 | **0.95** | 0.66 | 0.88 | **0.83** | 0.88 | **0.86** | $< 10^{-4}$ | (w=15, $\lambda$=0.06) | Ps |
| | **1.00** | 0.99 | **1.00** | 0.99 | 1.00 | **0.99** | 1.00 | **1.00** | 0.085 | (w=15, $\lambda$=0.16) | Se |
| Dt | **1.00** | 0.00 | **0.92** | 0.00 | 0.60 | **0.45** | 0.59 | **0.52** | $< 10^{-4}$ | (w=10, $\lambda$=0.005) | Is |
| | **0.99** | 0.38 | **0.94** | 0.62 | 0.88 | **0.81** | 0.87 | **0.84** | $< 10^{-4}$ | (w=15, $\lambda$=0.06) | Ps |
| | **1.00** | 0.96 | **1.00** | 0.98 | 1.00 | **0.98** | 1.00 | **0.99** | 0.534 | (w=15, $\lambda$=0.16) | Se |
| Us | **1.00** | 0.00 | **0.93** | 0.00 | 0.61 | **0.51** | 0.60 | **0.54** | $< 10^{-4}$ | (w=10, $\omega$=0.105) | Is |
| | **0.99** | 0.40 | **0.94** | 0.63 | 0.90 | **0.77** | 0.89 | **0.83** | $< 10^{-4}$ | (w=20, $\omega$=0.43) | Ps |
| | **1.00** | 0.98 | **1.00** | 0.99 | 1.00 | **0.99** | 1.00 | **1.00** | 0.085 | (w=20, $\omega$=1.205) | Se |
| Ds | **1.00** | 0.00 | **0.90** | 0.00 | 0.64 | **0.57** | 0.63 | **0.59** | $< 10^{-4}$ | (w=10, $\omega$=0.105) | Is |
| | **0.97** | 0.46 | **0.93** | 0.65 | 0.91 | **0.72** | 0.89 | **0.81** | $< 10^{-4}$ | (w=20, $\omega$=0.43) | Ps |
| | **1.00** | 0.98 | **1.00** | 0.99 | 1.00 | **1.00** | 1.00 | **1.00** | 0.163 | (w=20, $\omega$=1.205) | Se |
| S | **1.00** | 0.00 | **0.93** | 0.00 | 0.76 | **0.74** | 0.76 | **0.74** | $< 10^{-4}$ | (w=10, k=0.405) | Is |
| | **1.00** | 0.21 | **0.94** | 0.46 | 0.85 | **0.70** | 0.84 | **0.77** | $< 10^{-4}$ | (w=15, k=0.455) | Ps |
| | **1.00** | 0.98 | **1.00** | 0.99 | 1.00 | **0.98** | 1.00 | **0.99** | 0.055 | (w=15, k=1.405) | Se |
| C | **1.00** | 0.00 | **0.94** | 0.00 | 0.73 | **0.50** | 0.71 | **0.59** | $< 10^{-4}$ | (w=15, $\alpha$=0.23) | Is |
| | **0.99** | 0.19 | **0.93** | 0.43 | 0.87 | **0.72** | 0.86 | **0.79** | $< 10^{-4}$ | (w=20, $\alpha$=0.48) | Ps |
| | **1.00** | 0.94 | **0.99** | 0.97 | 1.00 | **0.98** | 1.00 | **0.99** | 0.596 | (w=20, $\alpha$=1.605) | Se |
| Str | **1.00** | 0.00 | **0.92** | 0.00 | 0.63 | **0.77** | 0.64 | **0.69** | $< 10^{-4}$ | (w=10, $\acute{\varepsilon_t}$=0.78) | Is |
| | **1.00** | 0.11 | **0.93** | 0.33 | 0.89 | **0.96** | 0.89 | **0.92** | $< 10^{-4}$ | (w=15, $\acute{\varepsilon_t}$=0.58) | Ps |
| | **0.99** | 0.92 | **0.98** | 0.95 | 0.96 | **1.00** | 0.96 | **0.98** | 0.021 | (w=15, $\acute{\varepsilon_t}$=0.43) | Se |

Table 4.4: G-mean of SVM and WSVM of all patterns in Inseparable (Is) problems for different imbalanced ratio. The majority class contains $(50 + r)\%$ of the data and the minority $(50 - r)\%$.

| | Ut | | Dt | | Us | | Ds | | S | | C | | Str | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $r$ | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM |
| 5 | 0.42 | 0.54 | 0.43 | 0.52 | 0.48 | 0.54 | 0.49 | 0.59 | 0.72 | 0.74 | 0.58 | 0.62 | 0.70 | 0.67 |
| 10 | 0.25 | 0.54 | 0.24 | 0.51 | 0.38 | 0.56 | 0.34 | 0.54 | 0.71 | 0.74 | 0.52 | 0.60 | 0.69 | 0.68 |
| 15 | 0.07 | 0.53 | 0.05 | 0.52 | 0.12 | 0.55 | 0.07 | 0.55 | 0.68 | 0.74 | 0.47 | 0.61 | 0.66 | 0.67 |
| 20 | 0.00 | 0.52 | 0.00 | 0.53 | 0.12 | 0.55 | 0.04 | 0.56 | 0.65 | 0.73 | 0.35 | 0.62 | 0.46 | 0.68 |
| 25 | 0.00 | 0.53 | 0.00 | 0.53 | 0.00 | 0.56 | 0.00 | 0.55 | 0.59 | 0.73 | 0.17 | 0.62 | 0.12 | 0.67 |
| 30 | 0.00 | 0.54 | 0.00 | 0.51 | 0.00 | 0.55 | 0.00 | 0.55 | 0.55 | 0.75 | 0.09 | 0.59 | 0.00 | 0.70 |
| 35 | 0.00 | 0.51 | 0.00 | 0.53 | 0.00 | 0.57 | 0.00 | 0.56 | 0.37 | 0.72 | 0.03 | 0.62 | 0.00 | 0.69 |
| 40 | 0.00 | 0.56 | 0.00 | 0.52 | 0.00 | 0.55 | 0.00 | 0.56 | 0.20 | 0.72 | 0.00 | 0.62 | 0.00 | 0.69 |
| 45 | 0.00 | 0.57 | 0.00 | 0.52 | 0.00 | 0.57 | 0.00 | 0.59 | 0.00 | 0.74 | 0.00 | 0.69 | 0.00 | 0.69 |

Table 4.5: G-mean of SVM and WSVM of all patterns in Partially separable (Ps) problems for different imbalanced ratio. The majority class contains $(50 + r)\%$ of the data and the minority $(50 - r)\%$.

| | Ut | | Dt | | Us | | Ds | | S | | C | | Str | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $r$ | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM |
| 5 | 0.84 | 0.84 | 0.85 | 0.84 | 0.82 | 0.82 | 0.81 | 0.83 | 0.80 | 0.81 | 0.76 | 0.76 | 0.92 | 0.92 |
| 10 | 0.84 | 0.85 | 0.84 | 0.85 | 0.81 | 0.82 | 0.80 | 0.82 | 0.78 | 0.80 | 0.74 | 0.76 | 0.92 | 0.92 |
| 15 | 0.83 | 0.85 | 0.83 | 0.84 | 0.80 | 0.83 | 0.79 | 0.83 | 0.78 | 0.80 | 0.72 | 0.75 | 0.92 | 0.92 |
| 20 | 0.82 | 0.84 | 0.82 | 0.86 | 0.79 | 0.83 | 0.78 | 0.82 | 0.77 | 0.80 | 0.72 | 0.77 | 0.92 | 0.93 |
| 25 | 0.81 | 0.85 | 0.82 | 0.85 | 0.76 | 0.84 | 0.74 | 0.83 | 0.73 | 0.79 | 0.68 | 0.77 | 0.90 | 0.92 |
| 30 | 0.77 | 0.84 | 0.79 | 0.85 | 0.73 | 0.82 | 0.75 | 0.81 | 0.66 | 0.80 | 0.62 | 0.77 | 0.90 | 0.92 |
| 35 | 0.76 | 0.84 | 0.77 | 0.85 | 0.70 | 0.81 | 0.72 | 0.82 | 0.62 | 0.78 | 0.61 | 0.77 | 0.86 | 0.92 |
| 40 | 0.72 | 0.85 | 0.66 | 0.85 | 0.68 | 0.81 | 0.67 | 0.82 | 0.60 | 0.79 | 0.49 | 0.75 | 0.80 | 0.92 |
| 45 | 0.66 | 0.86 | 0.62 | 0.84 | 0.63 | 0.83 | 0.65 | 0.81 | 0.46 | 0.77 | 0.43 | 0.79 | 0.33 | 0.92 |

Table 4.6: G-mean of SVM and WSVM of all patterns in Separable (Se) problems for different imbalanced ratio. The majority class contains $(50 + r)\%$ of the data and the minority $(50 - r)\%$.

| | Ut | | Dt | | Us | | Ds | | S | | C | | Str | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $r$ | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM | SVM | WSVM |
| 5 | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 0.99 | 0.99 | 0.98 | 0.98 |
| 10 | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 | 0.99 | 1.00 | 1.00 | 0.99 | 0.99 | 0.98 | 0.98 |
| 15 | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 0.99 | 0.99 | 0.98 | 0.98 |
| 20 | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 0.99 | 0.99 | 1.00 | 0.99 | 0.99 | 0.99 | 0.98 | 0.97 |
| 25 | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 0.99 | 0.99 | 0.99 | 1.00 | 0.99 | 0.99 | 0.98 | 0.98 |
| 30 | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.98 | 0.98 |
| 35 | 1.00 | 1.00 | 0.99 | 1.00 | 0.99 | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 0.99 | 0.99 | 0.99 | 0.98 |
| 40 | 1.00 | 1.00 | 0.99 | 1.00 | 0.99 | 1.00 | 0.99 | 1.00 | 0.99 | 1.00 | 0.99 | 0.99 | 0.99 | 0.98 |
| 45 | 0.99 | 1.00 | 0.98 | 0.99 | 0.99 | 1.00 | 0.99 | 1.00 | 0.99 | 0.99 | 0.97 | 0.99 | 0.99 | 0.98 |

### *Multi-Class Classification*

We extend the control chart pattern recognition from binary classification to multi-class classification and compare the results of WSVM with regular SVM for highly imbalanced datasets. In this classification test, there are seven classes: Normal, Uptrend, Downtrend, Up shift, Down shift, Cyclic, Systematic and Stratification patterns. Although the generic version of SVM can accom-

modate only two classes it is possible to generalize the classifier to multi class by constructing all the classifier pairs (one-against-one policy) and then use a *majority voting* scheme for assigning a new point to its class. This method has been found to perform well however the computational time of the model is expected to increase as the number of the classes increase. To study highly imbalanced classification problem, we generate $1000$ data which consist of $951$ normal data (approximately $95\%$ of total data), and $49$ abnormal data (approximately $5\%$ of the total data). For each abnormal control chart pattern, 7 examples were generated. Similar to binary classification, we consider the weight of each class as inverse of the class size (Veropoulos et al., 1999)

$$C_i = \frac{C}{n_i} \qquad\qquad i = 1, 2, \ldots, m. \tag{4.3}$$

Where $n_i$ and $C_i$ are the class size and weight related to the class $i$ respectively, $i = 1, 2, \ldots, m$ and $m$ is the number of classes. The parameters $C$ and $\gamma$ were tuned during the training process through the same parameter grid search as in the binary case. The parameters for abnormal patterns are selected from Table B.2.

We evaluate the performance of WSVM versus SVM for a partially separable problem with the following abnormal pattern characteristics: $\lambda$=0.58 (Downtrend), $\lambda$=0.93 (Uptrend), $k$=1.53 (Systematic), $\omega$=0.63 (Downshift), $\omega$=0.38 (Upshift), $\alpha$=0.405 (Cyclic), and $\acute{\varepsilon}$=0.805 (Stratification). We show the performance for three representative window ($w$) parameter values (10, 50, 100). The parameters of each problem were chosen randomly for the parameter range shown in Table B.2. the results are the average over ten such randomly selected problems. More computational experiments were conducted and can be found in the supplementary material section of the paper. Since the problem contains more than two classes sensitivity specificity and G-mean are not defined. For this we provide the full confusion matrix that demonstrates the exact classification accuracy for each class sepearetly. As expected WSVM performs better in identifying the examples from the minority classes although it sacrifices some of the normal class accuracy. For short windows, SVM fails to detect examples from four classes and for medium window for two. In general, a

problem becomes less challenging as $w$ increases. This is consistent with the binary classification examples.

Table 4.7: Classification results for multi-class SVM and WSVM for CCPR with window length=10 and highly imbalanced data. Rows are related to predicted class labels and the columns are related to real labels.

| | | N | Dt | Ut | S | Ds | Us | C | Str |
|---|---|---|---|---|---|---|---|---|---|
| SVM | N | **1.00** | 0.00 | 0.00 | 0.05 | 1.00 | 1.00 | 1.00 | 1.00 |
| | Dt | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ut | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | S | 0.00 | 0.00 | 0.00 | 0.95 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ds | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Us | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | C | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Str | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| WSVM | N | 0.65 | 0.00 | 0.00 | 0.00 | 0.15 | 0.19 | 0.13 | 0.31 |
| | Dt | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ut | 0.00 | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | S | 0.00 | 0.00 | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ds | 0.06 | 0.00 | 0.00 | 0.00 | **0.75** | 0.00 | 0.00 | 0.08 |
| | Us | 0.10 | 0.00 | 0.00 | 0.00 | 0.00 | **0.77** | 0.00 | 0.00 |
| | C | 0.07 | 0.00 | 0.00 | 0.00 | 0.00 | 0.04 | **0.70** | 0.00 |
| | Str | 0.11 | 0.00 | 0.00 | 0.00 | 0.10 | 0.00 | 0.17 | **0.61** |

The rows and columns of the *confusion matrix* show the predicted class and the real class respectively. Moreover, the diagonal elements show the accurate classification percentage. However, the accuracy for normal class has decreased slightly for multi-class WSVM, the fact is that the classification accuracy has increased for other patterns. In other words, results show that multi-class WSVM has higher performance than multi-class SVM for highly imbalanced data. The multi-class SVM assigns most of data to the majority class. In this highly imbalanced classification problem, we can observe that the behavior of multi-class SVM is very close to naive classification rules.

In addition as previous we computed the computational time for training and testing for the multiclass problem (Figure 4.5). This experiment was conducted for three window lengths 10, 50, 100.

Table 4.8: Classification results for multi-class SVM and WSVM for CCPR with window length=50 and highly imbalanced data

| | | N | Dt | Ut | S | Ds | Us | C | Str |
|---|---|---|---|---|---|---|---|---|---|
| SVM | N | **1.00** | 0.00 | 0.00 | 0.00 | 0.40 | 1.00 | 0.47 | 1.00 |
| | Dt | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ut | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | S | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ds | 0.00 | 0.00 | 0.00 | 0.00 | 0.60 | 0.00 | 0.00 | 0.00 |
| | Us | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | C | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.53 | 0.00 |
| | Str | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| WSVM | N | 0.98 | 0.00 | 0.00 | 0.00 | 0.20 | 0.37 | 0.27 | 0.37 |
| | Dt | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ut | 0.00 | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | S | 0.00 | 0.00 | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ds | 0.00 | 0.00 | 0.00 | 0.00 | **0.80** | 0.00 | 0.00 | 0.00 |
| | Us | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.63** | 0.00 | 0.00 |
| | C | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.73** | 0.00 |
| | Str | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.63** |

Table 4.9: Classification results for multi-class SVM and WSVM for CCPR with window length=100 and highly imbalanced data

| | | N | Dt | Ut | S | Ds | Us | C | Str |
|---|---|---|---|---|---|---|---|---|---|
| SVM | N | **1.00** | 0.00 | 0.00 | 0.00 | 0.25 | 0.28 | 0.47 | 1.00 |
| | Dt | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ut | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | S | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ds | 0.00 | 0.00 | 0.00 | 0.00 | 0.75 | 0.00 | 0.00 | 0.00 |
| | Us | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.72 | 0.00 | 0.00 |
| | C | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.53 | 0.00 |
| | Str | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| WSVM | N | 1.00 | 0.00 | 0.00 | 0.00 | 0.23 | 0.22 | 0.15 | 0.48 |
| | Dt | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ut | 0.00 | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | S | 0.00 | 0.00 | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 |
| | Ds | 0.00 | 0.00 | 0.00 | 0.00 | **0.77** | 0.00 | 0.00 | 0.00 |
| | Us | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.78** | 0.00 | 0.00 |
| | C | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.85** | 0.00 |
| | Str | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.52** |

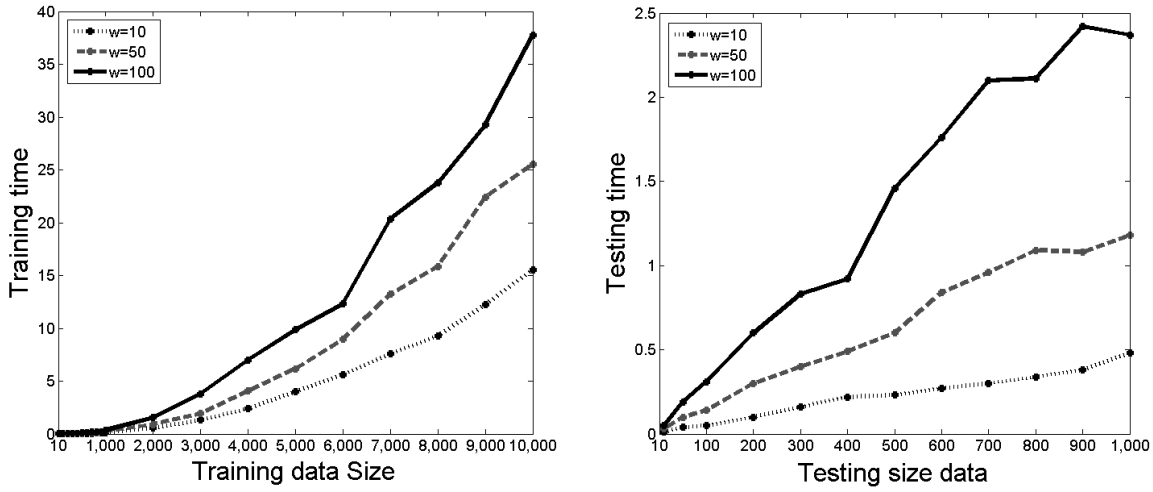



Figure 4.5: WSVM training and testing time vs. training size for multi-class classification

The trends observed are similar to binary classification however the seven classes do not add an high additional computational "overhead" compared to binary classification. Confusion matrices for more study is written in appendix.

Last we compare the proposed algorithm for a real life application from wafer manufacturing industry. In wafer manufacturing. Electronics manufacturing usually involve a large number of steps ($> 250$) which can induce defects to the final product. quality control is performed by recording the different frequencies that are emitted by the plasma during the process. The dataset is composed out of 1000 training samples (of length152 each) and 6174 testing samples of the same length (Olszewski, 2001; Keogh et al., 2011). The training samples are imbalanced (903 are majority and 97 minority). We performed cross validation on training data and then we used the model developed from training for testing on the larger testing dataset (Table 4).

Furthermore, we perform sensitivity analysis by evaluating prediction performance for different values of penalty parameter $C$. Results are shown in Table 4:

Table 4.10: Training and testing performance for the wafer dataset

| | | Sensitivity | Specificity | Gmean | Accuracy |
|---|---|---|---|---|---|
| Training | SVM | **0.9996** | 0.9160 | 0.9156 | **0.9913** |
| | WSVM | 0.9967 | **0.9350** | **0.9319** | 0.9905 |
| Testing | SVM | **0.9971** | 0.9654 | 0.9811 | **0.9937** |
| | WSVM | 0.9895 | **0.9895** | **0.9895** | 0.9895 |

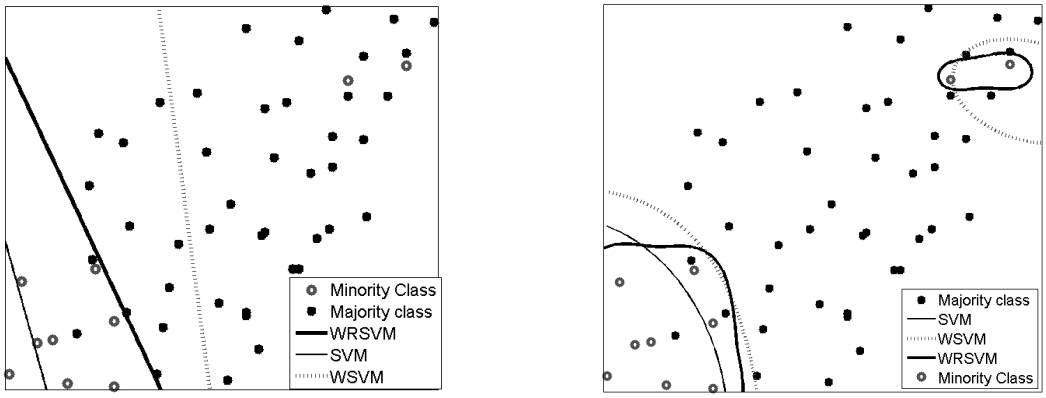
Table 4.11: Testing performance for wafer dataset (With bold color is denoted the highest *G-mean* score)

| | SVM | WSVM |
|---|---|---|
| $C$ | Gmean | Gmean |
| 0.1 | 0.8756 | **0.9684** |
| 1 | 0.9790 | **0.9895** |
| 10 | **0.9811** | **0.9811** |
| 100 | **0.9811** | **0.9811** |
| 1000 | **0.9811** | **0.9811** |



In this section, we present computational results to demonstrate the performance of WRSVM in comparison with its standard counterparts. However, we first provide a visual account of the solution obtained from WRSVM in contrast with SVM and WSVM on a toy dataset. Figure 4.6 illustrates the results of WRSVM on a typical dataset and the control boundary in two-dimensional cases. The black nonlinear curve shows WRSVM boundary, the black dashed and gray solid curves shows WSVM and SVM boundaries respectively. In Figure 4.6, SVM boundary has a heavy bias towards the majority class, causing the minority class samples to be misclassified, whereas WRSVM is greatly influenced by the outliers and cause the majority class samples to be misclassified. WRSVM establishes a balance between these two extreme behaviors by simultaneously using relative weights for different class sizes and reducing the influence of outliers by relaxing them using free slack. The same behavior can clearly be observed in both linear and nonlinear

classification in Figure 4.6.




(a) Linear WRSVM vs. SVM and WSVM (b) Non-linear WRSVM vs. SVM and WSVM

Figure 4.6: Linear and non-linear WRSVM classifier vs. SVM and WSVM classifiers

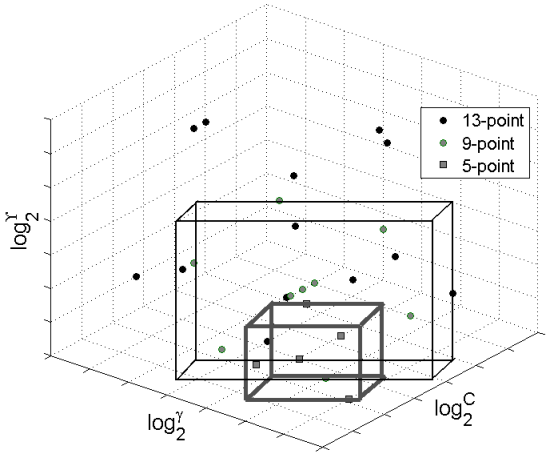



Figure 4.7: The nested UD model selection with a 13-point UD at the 1st iteration, a 9-point UD at 2nd iteration and a 5-point UD at 3rd iteration

We adopt a 13- and 9-point run design for the first and second stages of the nested UD for all three methods, which has been found to be adequate for UCI data Huang et al. (2007). We use 5 points for the third stage of the nested UD for WRSVM. In Figure 4.7 we show the 13-, 9-, 5-point nested UD sampling pattern for the regularization parameter $C$, bandwidth parameter $l$ for RBF kernel function, and average free slack per sample $\Upsilon$.

*Comparative Evaluation*

Next, we apply WRSVM to University of California, Irvine (UCI) benchmark datasets [2] for binary classification and compare the experimental results of the proposed method with both with standard SVM and WSVM as well as other supervised learning methods that have been found to be robust in imbalanced problems with outliers. SVM and WSVM models are solved using LIBSVM-3.12 and LIBSVM-weights-3.12, respectively Chang & Lin (2011), and the WRSVM model is solved using CPLEX 12.3 [3], while data processing and further scripting is done in MATLAB 2009b [4]. We use a typical 10-fold cross validation setup. We create outliers on the training set by flipping the class label the farthest majority-class samples to minority class labels. All data are normalized prior to classification, so that they have zero mean and unitary standard deviation. We note that in each iteration of the 10-fold cross-validation, model selection based on the nested UD is performed on the training data, and the test data is only used to calculate performance measures such as sensitivity, specificity, G-mean and accuracy.

We use the RBF kernel, which is the most commonly used kernel on the UCI benchmark datasets in the literature. The similarity between two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in RBF kernel is given by,

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-l\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2), l \geq 0. \tag{4.4}$$

For each class, the weights are assigned proportional to the inverse of the class size, $\frac{C}{2n^+}$ and $\frac{C}{2n^-}$, where $n^+$ and $n^-$ are the sizes of the minority and the majority classes, respectively. This weight strategy has been used in a number of previous studies (Liu et al., 2005; Du & Chen, 2005; Huang & Du, 2005; Hwang et al., 2011).

[2] A. Frank and A. Asuncion, machine learning repository [http://archive. ics. uci. edu/ml]. Irvine, CA: University of California," School of Information and Computer Science, vol. 213, 2010.

[3] ILOG CPLEX: High-performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming," World Wide Web, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

[4] http://www.mathworks.com/products/matlab/

Table 4.12: UCI and IDA data sets used with changing positive/negative class sizes with respect to imbalance ratio.

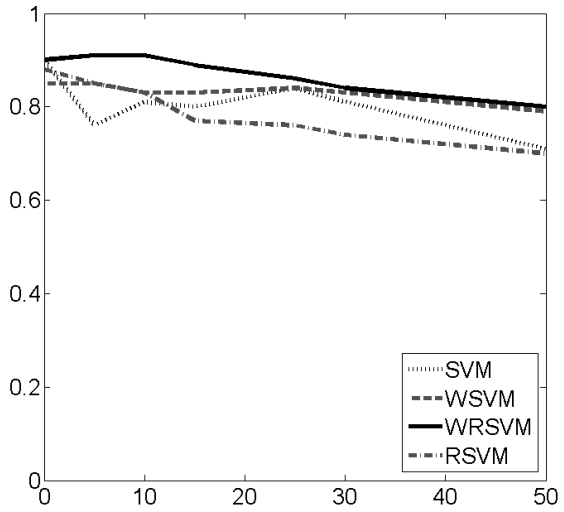
| Dataset | $r_{imb}$ | # Pos. | # Neg. | Total |
|---|---|---|---|---|
| German | 0.9 | 77 | 700 | 777 |
| | 0.925 | 56 | 700 | 756 |
| | 0.95 | 36 | 700 | 736 |
| | 0.975 | 17 | 700 | 717 |
| Diabetes | 0.9 | 55 | 500 | 555 |
| | 0.925 | 40 | 500 | 540 |
| | 0.95 | 26 | 500 | 526 |
| | 0.975 | 13 | 500 | 513 |
| Thyroid | 0.9 | 16 | 150 | 166 |
| | 0.925 | 12 | 150 | 162 |
| | 0.95 | 8 | 150 | 158 |
| | 0.975 | 4 | 150 | 154 |
| Breast | 0.9 | 40 | 357 | 397 |
| | 0.925 | 29 | 357 | 386 |
| | 0.95 | 19 | 357 | 376 |
| | 0.975 | 9 | 357 | 366 |
| Heart | 0.9 | 23 | 212 | 235 |
| | 0.925 | 17 | 212 | 229 |
| | 0.95 | 11 | 212 | 223 |
| | 0.975 | 5 | 212 | 217 |
| Credit | 0.9 | 42 | 383 | 425 |
| | 0.925 | 31 | 383 | 414 |
| | 0.95 | 20 | 383 | 403 |
| | 0.975 | 9 | 383 | 392 |
| Ringnorm | 0.9 | 26 | 237 | 263 |
| | 0.925 | 19 | 237 | 256 |
| | 0.95 | 15 | 237 | 252 |
| | 0.975 | 7 | 237 | 244 |
| Twonorm | 0.9 | 17 | 155 | 172 |
| | 0.925 | 12 | 155 | 167 |
| | 0.95 | 8 | 155 | 163 |
| | 0.975 | 3 | 155 | 158 |
| Waveform | 0.9 | 15 | 140 | 155 |
| | 0.925 | 11 | 140 | 151 |
| | 0.95 | 7 | 140 | 147 |
| | 0.975 | 3 | 140 | 143 |
| Banana | 0.9 | 33 | 300 | 333 |
| | 0.925 | 24 | 300 | 324 |
| | 0.95 | 15 | 300 | 315 |
| | 0.975 | 7 | 300 | 307 |

The proposed methodology is examined for different imbalance ratios ($r_{imb}$) of majority class to all data chosen as 90%, 92.5%, 95%, and 97.5%. Similarly different outliers to all data ratios of 0%, 1%, 2%,..., 50% are chosen. Table 4.12 details the datasets with different number of samples in majority and minority classes for different imbalance ratios.

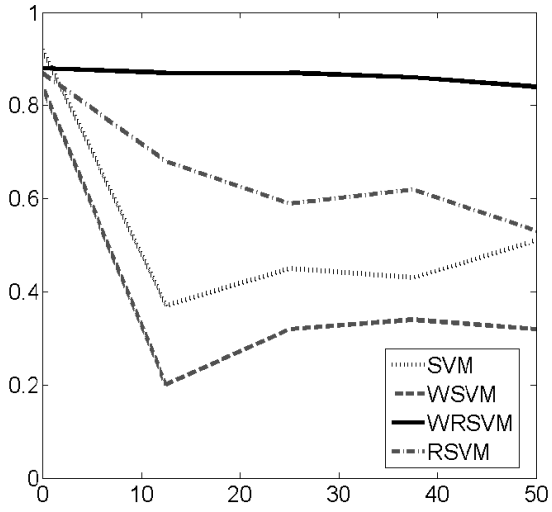
The extent to which classification algorithms are affected by outliers of the minority class highly depends on the distribution and the geometry of classes. The detailed comparison of our proposed method against SVM and WSVM is given in Figure 4.8. The horizontal axis in each plot shows percent of the outliers between 0% and 50%, and the vertical axis shows G-mean. The left and right columns of the plots have 90% and 97.5% imbalance ratios, respectively. G-mean reduces slightly with increasing percentage of outliers in each plot, where the reduction in SVM is more pronounced than the others in all plots. In the Heart and Diabetes datasets WRSVM has a slight advantage over WSVM, and significantly better than SVM. In the Credit dataset WRSVM is clearly better than SVM and WSVM. It is also interesting to observe that WRSVM produces better G-mean values for the datasets with higher imbalance, which confirms that our mathematical model is a good fit for imbalanced datasets.

The comparative results of WRSVM in terms of G-mean values is given against FSVM Lin & Wang (2002), WSVM, SVM, Naïve Bayes (NB), C4.5 and 5NN methods for a low outlier ratio of 14% in Table 4.13 and a high outlier ratio of 30% in Table 4.14, and for imbalance ratios of 0.90, 0.925, 0.95, and 0.975 for each dataset in both tables. It is known that NB and 5NN (Anyfantis et al., 2007), as well as C4.5 (Quinlan, 1986; Khoonsari & Motie, 2012) methods are known to be robust to outliers in. The highest values are marked in boldface across the three methods for their respective outlier levels.

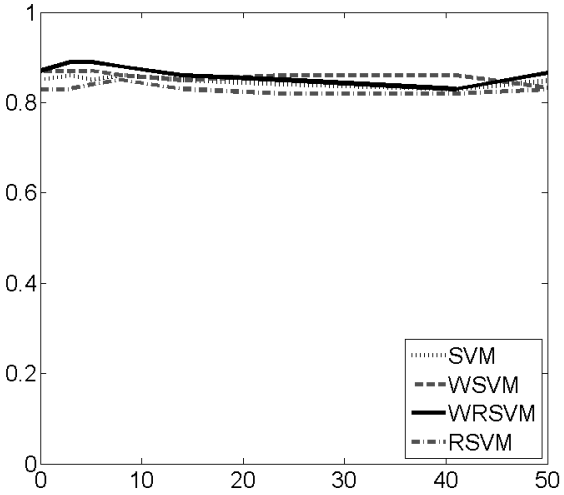
It is clear from the accumulation of boldface results under the WRSVM in both tables that WRSVM performs better than the other methods in general for both low and high outlier ratios. More specifically, WRSVM produces the highest G-mean values in *22 out of 40* dataset/$r_{imb}$ combinations for low outlier ratio followed by NB with *11 out of 40*. For high outlier ratios, WRSVM produces the highest G-mean values in *22 out of 40* dataset/$r_{imb}$ followed by SVM and NB with *9 out of 40*.



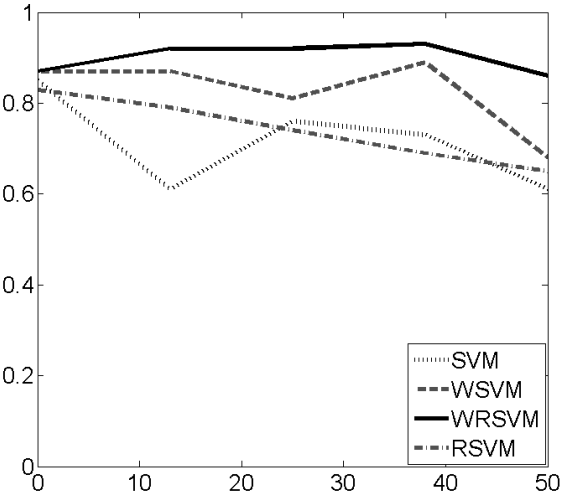

(a) Heart (90%)



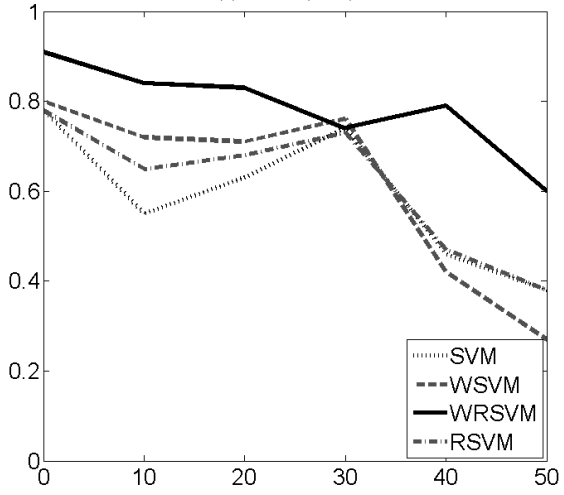

(b) Heart (97.5%)



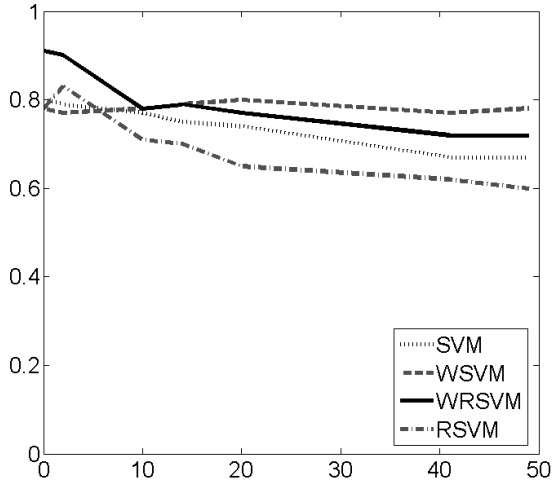

(c) Credit (90%)




(d) Credit (97.5%)




(e) Diabetes (90%)




(f) Diabetes (97.5%)

Figure 4.8: G-mean vs. the outlier ratio for Heart, Credit, and Diabetes data with imbalance ratios of 90% and 97.5% for left and right columns of plots, respectively.

Table 4.13: Comparative G-mean results for WRSVM against WSVM, FSVM, RSVM, SVM, NB, C4.5 and 5NN on on UCI datasets for different imbalanced case with low outlier ratio (average (standard dev.))

| Dataset | $r_{imb}$ | WRSVM | FSVM | RSVM | WSVM | SVM | NB | C4.5 | 5NN |
|---|---|---|---|---|---|---|---|---|---|
| German | 0.90 | **0.85** (0.018) | 0.77 (0.022) | 0.78 (0.040) | 0.75 (0.012) | 0.76 (0.022) | 0.52 (0.009) | 0.37 (0.057) | 0.39 (0.010) |
| | 0.925 | **0.83** (0.052) | **0.83** (0.032) | 0.81 (0.079) | 0.77 (0.037) | 0.78 (0.023) | 0.45 (0.006) | 0.30 (0.067) | 0.33 (0.015) |
| | 0.95 | **0.85** (0.050) | 0.81 (0.050) | 0.77 (0.072) | 0.83 (0.082) | 0.81 (0.035) | 0.44 (0.014) | 0.06 (0.030) | 0.32 (0.017) |
| | 0.975 | **0.88** (0.030) | 0.46 (0.127) | 0.53 (0.123) | 0.84 (0.074) | 0.73 (0.140) | 0.33 (0.032) | 0.00 (0.000) | 0.00 (0.000) |
| Diabetes | 0.90 | **0.89** (0.042) | 0.79 (0.020) | 0.82 (0.042) | 0.78 (0.012) | 0.80 (0.018) | 0.64 (0.006) | 0.64 (0.039) | 0.47 (0.014) |
| | 0.925 | **0.91** (0.022) | 0.78 (0.032) | 0.84 (0.061) | 0.79 (0.045) | 0.76 (0.028) | 0.56 (0.006) | 0.28 (0.077) | 0.37 (0.023) |
| | 0.95 | **0.86** (0.061) | 0.76 (0.053) | 0.81 (0.051) | 0.79 (0.010) | 0.72 (0.066) | 0.51 (0.006) | 0.48 (0.065) | 0.01 (0.013) |
| | 0.975 | **0.87** (0.034) | 0.68 (0.089) | 0.62 (0.079) | 0.68 (0.081) | 0.63 (0.0132) | 0.45 (0.023) | 0.31 (0.124) | 0.00 (0.009) |
| Thyroid | 0.90 | 0.98 (0.006) | 0.97 (0.010) | 0.97 (0.019) | 0.96 (0.021) | **0.99** (0.008) | 0.86 (0.003) | 0.98 (0.008) | 0.85 (0.005) |
| | 0.925 | 0.98 (0.011) | **0.99** (0.006) | 0.96 (0.026) | 0.98 (0.013) | 0.96(0.011) | 0.91(0.002) | 0.98 (0.005) | 0.78 (0.016) |
| | 0.95 | **0.98** (0.010) | 0.90 (0.049) | 0.94 (0.019) | 0.89 (0.072) | 0.96 (0.010) | 0.92 (0.005) | 0.95 (0.031) | 0.60 (0.013) |
| | 0.975 | **0.93** (0.028) | 0.00 (0.000) | 0.81 (0.012) | 0.77 (0.046) | 0.78 (0.036) | **0.93** (0.025) | 0.60 (0.000) | 0.00 (0.129) |
| Breast | 0.90 | 0.95 (0.014) | **0.97** (0.020) | 0.95 (0.019) | 0.95 (0.009) | **0.97** (0.007) | 0.92 (0.007) | 0.96 (0.009) | 0.86 (0.005) |
| | 0.925 | 0.93 (0.013) | **0.97** (0.008) | 0.93 (0.012) | 0.92 (0.007) | 0.96 (0.011) | 0.90 (0.008) | 0.94 (0.009) | 0.79 (0.007) |
| | 0.95 | 0.92 (0.018) | **0.97** (0.010) | 0.89 (0.031) | 0.93 (0.011) | 0.96 (0.014) | 0.91 (0.012) | 0.94 (0.019) | 0.78 (0.006) |
| | 0.975 | 0.95 (0.010) | **0.97** (0.035) | 0.91 (0.011) | 0.83 (0.039) | 0.94 (0.006) | 0.92 (0.009) | 0.96 (0.018) | 0.48 (0.010) |
| Heart | 0.90 | **0.91** (0.012) | 0.85 (0.053) | 0.80 (0.048) | 0.85 (0.024) | 0.79 (0.067) | 0.77 (0.014) | 0.88 (0.020) | 0.53 (0.014) |
| | 0.925 | **0.92** (0.014) | 0.60 (0.075) | 0.80 (0.033) | 0.88(0.022) | 0.76 (0.092) | 0.78 (0.026) | 0.82 (0.031) | 0.39 (0.038) |
| | 0.95 | **0.90** (0.007) | 0.23 (0.093) | 0.72 (0.113) | 0.76 (0.047) | 0.73 (0.100) | 0.81 (0.021) | 0.60 (0.089) | 0.26 (0.037) |
| | 0.975 | 0.89 (0.003) | 0.49 (0.109) | 0.88 (0.013) | 0.82 (0.071) | 0.87 (0.043) | **0.90** (0.022) | 0.00 (0.000) | 0.00 (0.000) |
| Credit | 0.90 | **0.89** (0.010) | 0.89 (0.016) | 0.85 (0.016) | 0.86 (0.017) | 0.86 (0.021) | 0.67 (0.007) | 0.85 (0.015) | 0.80 (0.004) |
| | 0.925 | **0.91** (0.008) | 0.88 (0.008) | 0.86 (0.024) | 0.89 (0.012) | 0.87 (0.020) | 0.69 (0.008) | 0.80 (0.019) | 0.78 (0.006) |
| | 0.95 | **0.92** (0.015) | 0.87 (0.015) | 0.84 (0.047) | 0.87 (0.046) | 0.86 (0.025) | 0.67 (0.014) | 0.81 (0.019) | 0.77 (0.014) |
| | 0.975 | **0.93** (0.020) | 0.86 (0.039) | 0.79 (0.043) | 0.82 (0.095) | 0.59 (0.181) | 0.66 (0.013) | 0.22 (0.157) | 0.49 (0.033) |
| Ringnorm | 0.90 | 0.90 (0.105) | 0.99 (0.009) | 0.98 (0.009) | 0.97 (0.018) | 0.98 (0.008) | **1.00** (0.001) | 0.94 (0.009) | 0.00 (0.000) |
| | 0.925 | 0.97 (0.009) | 0.95 (0.051) | 0.94 (0.042) | 0.96 (0.007) | 0.98 (0.007) | **1.00** (0.003) | 0.89 (0.011) | 0.00 (0.000) |
| | 0.95 | 0.93 (0.013) | 0.82 (0.106) | 0.92 (0.004) | 0.88 (0.063) | 0.94 (0.015) | **0.99** (0.006) | 0.88 (0.024) | 0.00 (0.000) |
| | 0.975 | 0.92 (0.009) | 0.52 (0.124) | 0.88 (0.013) | 0.78 (0.044) | 0.92 (0.006) | **0.97** (0.011) | 0.54 (0.160) | 0.00 (0.000) |
| Twonorm | 0.90 | 0.95 (0.011) | 0.96 (0.028) | 0.94 (0.035) | 0.98 (0.005) | 0.97 (0.010) | **0.99** (0.004) | 0.92 (0.021) | 0.81 (0.012) |
| | 0.925 | 0.92 (0.008) | 0.79 (0.119) | 0.92 (0.007) | 0.93 (0.014) | 0.94 (0.014) | **0.96** (0.007) | 0.88 (0.019) | 0.65 (0.016) |
| | 0.95 | 0.94 (0.011) | 0.48 (0.116) | 0.94 (0.008) | 0.93 (0.033) | **0.95** (0.013) | **0.95** (0.019) | 0.87 (0.000) | 0.72 (0.023) |
| | 0.975 | **0.82** (0.000) | 0.00 (0.000) | 0.81 (0.012) | 0.75 (0.073) | 0.70 (0.075) | **0.82** (0.000) | 0.31 (0.114) | 0.00 (0.000) |
| Waveform | 0.90 | 0.94 (0.010) | 0.90 (0.014) | 0.94 (0.011) | 0.89 (0.019) | 0.94 (0.008) | 0.92 (0.010) | **0.95** (0.016) | 0.74 (0.007) |
| | 0.925 | 0.91 (0.008) | 0.82 (0.044) | 0.91 (0.013) | 0.90 (0.005) | **0.93** (0.012) | 0.91 (0.029) | 0.92 (0.022) | 0.69 (0.030) |
| | 0.95 | 0.92 (0.006) | 0.76 (0.081) | 0.92 (0.009) | 0.89 (0.062) | **0.93** (0.011) | **0.93** (0.015) | 0.85 (0.048) | 0.73 (0.022) |
| | 0.975 | 0.82 (0.009) | 0.00 (0.000) | 0.80 (0.017) | 0.80 (0.034) | **0.85** (0.014) | 0.82 (0.000) | 0.08 (0.077) | 0.00 (0.000) |
| Banana | 0.90 | **0.96** (0.008) | 0.91 (0.010) | 0.92 (0.011) | 0.92 (0.010) | 0.88 (0.017) | 0.81 (0.006) | 0.93 (0.009) | 0.92 (0.006) |
| | 0.925 | **0.95** (0.009) | 0.92 (0.018) | 0.89 (0.017) | 0.93 (0.017) | 0.84 (0.025) | 0.80 (0.004) | 0.91 (0.017) | 0.83 (0.007) |
| | 0.95 | **0.95** (0.004) | 0.87 (0.026) | 0.91 (0.013) | 0.89 (0.032) | 0.88 (0.016) | 0.85 (0.003) | 0.90 (0.007) | 0.85 (0.004) |
| | 0.975 | **0.92** (0.023) | 0.69 (0.058) | 0.84 (0.019) | 0.40 (0.126) | 0.76 (0.036) | 0.60 (0.113) | 0.55 (0.044) | 0.75 (0.005) |

*Outlier Detection Performance*

Since WRSVM assigns free slack through Formulation 3.12 by giving priority to the points that further most from the rest of the class. the amount of free slack received by each point can be used as a score of "outlierness" of each point. In that respect WRSVM can be seen as an embedded outlier detection and supervised classification scheme.

For each dataset used in this paper we estimated the percentage of *induced* outliers that received free slack as well as the percentage of non outlier data points that also received free slack.

Table 4.14: Comparative G-mean results for WRSVM against WSVM, FSVM, RSVM, SVM, NB, C4.5 and 5NN on on UCI datasets for different imbalanced case with high outlier ratio

| Dataset | $r_{imb}$ | WRSVM | FSVM | RSVM | WSVM | SVM | NB | C4.5 | 5NN |
|---|---|---|---|---|---|---|---|---|---|
| German | 0.90 | 0.77 (0.021) | 0.79 (0.026) | 0.77 (0.044) | 0.73 (0.013) | **0.86** (0.017) | 0.53 (0.006) | 0.45 (0.081) | 0.39 (0.007) |
| | 0.925 | 0.77 (0.030) | **0.85** (0.025) | 0.81 (0.052) | 0.76 (0.056) | 0.80 (0.070) | 0.45 (0.007) | 0.28 (0.060) | 0.33 (0.016) |
| | 0.95 | 0.79 (0.029) | 0.83 (0.046) | 0.78 (0.065) | **0.85** (0.098) | 0.84 (0.046) | 0.44 (0.011) | 0.06 (0.048) | 0.32 (0.015) |
| | 0.975 | **0.85** (0.035) | 0.62 (0.098) | 0.65 (0.068) | 0.81 (0.097) | 0.71 (0.104) | 0.34 (0.030) | 0.00 (0.011) | 0.00 (0.000) |
| Diabetes | 0.90 | **0.87** (0.040) | 0.82 (0.032) | 0.82 (0.040) | 0.77 (0.015) | 0.79 (0.021) | 0.64 (0.006) | 0.64 (0.025) | 0.47 (0.010) |
| | 0.925 | **0.89** (0.050) | 0.81 (0.026) | 0.85 (0.039) | 0.79 (0.046) | 0.76 (0.060) | 0.56 (0.008) | 0.28 (0.134) | 0.36 (0.024) |
| | 0.95 | **0.87** (0.042) | 0.79 (0.047) | 0.78 (0.063) | 0.78 (0.013) | 0.76 (0.045) | 0.51 (0.008) | 0.47 (0.074) | 0.01 (0.010) |
| | 0.975 | **0.86** (0.026) | 0.68 (0.117) | 0.61 (0.143) | 0.73 (0.039) | 0.61 (0.082) | 0.47 (0.024) | 0.23 (0.083) | 0.00 (0.009) |
| Thyroid | 0.90 | **0.98** (0.011) | 0.96 (0.014) | 0.96 (0.021) | 0.96 (0.013) | **0.98** (0.009) | 0.86 (0.003) | **0.98** (0.007) | 0.85 (0.004) |
| | 0.925 | **0.98** (0.012) | **0.98** (0.014) | 0.97 (0.013) | **0.98** (0.018) | 0.96 (0.011) | 0.91 (0.004) | **0.98** (0.008) | 0.77 (0.020) |
| | 0.95 | **0.98** (0.012) | 0.88 (0.047) | 0.93 (0.034) | 0.92 (0.028) | 0.95 (0.009) | 0.92 (0.005) | **0.98** (0.013) | 0.61 (0.013) |
| | 0.975 | **0.92** (0.030) | 0.00 (0.000) | 0.81 (0.017) | 0.80 (0.024) | 0.79 (0.026) | **0.92** (0.033) | 0.55 (0.095) | 0.00 (0.000) |
| Breast | 0.90 | 0.94 (0.017) | 0.96 (0.012) | 0.95 (0.008) | 0.95 (0.010) | **0.98** (0.012) | 0.92 (0.005) | 0.96 (0.012) | 0.86 (0.004) |
| | 0.925 | 0.92 (0.013) | **0.96** (0.006) | 0.93 (0.010) | 0.92 (0.010) | **0.96** (0.010) | 0.91 (0.007) | 0.93 (0.009) | 0.78 (0.007) |
| | 0.95 | 0.92 (0.012) | **0.96** (0.011) | 0.90 (0.017) | 0.93 (0.011) | **0.96** (0.013) | 0.90 (0.007) | 0.94 (0.014) | 0.78 (0.010) |
| | 0.975 | 0.95 (0.011) | **0.98** (0.011) | 0.90 (0.031) | 0.85 (0.050) | 0.94 (0.023) | 0.92 (0.013) | 0.97 (0.023) | 0.49 (0.009) |
| Heart | 0.90 | **0.91** (0.012) | 0.86 (0.036) | 0.82 (0.057) | 0.83 (0.018) | 0.84 (0.077) | 0.77 (0.016) | 0.90 (0.017) | 0.53 (0.014) |
| | 0.925 | **0.91** (0.038) | 0.80 (0.094) | 0.82 (0.050) | 0.88 (0.016) | 0.78 (0.080) | 0.78 (0.023) | 0.82 (0.030) | 0.39 (0.023) |
| | 0.95 | **0.90** (0.008) | 0.39 (0.126) | 0.78 (0.080) | 0.78 (0.045) | 0.76 (0.089) | 0.80 (0.020) | 0.60 (0.083) | 0.26 (0.047) |
| | 0.975 | **0.89** (0.000) | 0.00 (0.000) | 0.88 (0.008) | 0.78 (0.092) | 0.87 (0.037) | 0.91 (0.023) | 0.00 (0.000) | 0.00 (0.000) |
| Credit | 0.90 | **0.89** (0.011) | **0.89** (0.012) | 0.86 (0.013) | 0.85 (0.017) | 0.86 (0.017) | 0.67 (0.004) | 0.85 (0.011) | 0.79 (0.007) |
| | 0.925 | **0.90** (0.013) | 0.88 (0.015) | 0.85 (0.019) | 0.89 (0.008) | 0.86 (0.027) | 0.70 (0.007) | 0.79 (0.019) | 0.78 (0.005) |
| | 0.95 | **0.91** (0.025) | 0.88 (0.009) | 0.86 (0.044) | 0.88 (0.045) | 0.88 (0.021) | 0.66 (0.008) | 0.83 (0.019) | 0.77 (0.016) |
| | 0.975 | **0.92** (0.027) | 0.86 (0.058) | 0.81 (0.062) | 0.89 (0.067) | 0.61 (0.155) | 0.65 (0.023) | 0.28 (0.167) | 0.49 (0.030) |
| Ringnorm | 0.90 | 0.96 (0.042) | 0.99 (0.008) | 0.97 (0.007) | 0.97 (0.015) | 0.99 (0.004) | **1.00** (0.002) | 0.93 (0.011) | 0.00 (0.000) |
| | 0.925 | 0.97 (0.009) | 0.98 (0.010) | 0.96 (0.012) | 0.96 (0.007) | 0.98 (0.006) | **1.00** (0.003) | 0.90 (0.003) | 0.00 (0.000) |
| | 0.95 | 0.93 (0.008) | 0.80 (0.067) | 0.92 (0.008) | 0.84 (0.084) | 0.94 (0.014) | **0.99** (0.007) | 0.86 (0.029) | 0.00 (0.000) |
| | 0.975 | 0.92 (0.006) | 0.36 (0.174) | 0.90 (0.011) | 0.75 (0.079) | 0.92 (0.005) | **0.97** (0.014) | 0.54 (0.098) | 0.00 (0.000) |
| Twonorm | 0.90 | 0.95 (0.008) | 0.95 (0.032) | 0.96 (0.007) | 0.96 (0.013 | 0.96 (0.008) | **0.98** (0.006) | 0.93 (0.019) | 0.81 (0.008) |
| | 0.925 | 0.92 (0.005) | 0.78 (0.080) | 0.90 (0.011) | 0.93 (0.012) | 0.94 (0.011) | **0.97** (0.008) | 0.87 (0.025) | 0.65 (0.025) |
| | 0.95 | 0.94 (0.007) | 0.61 (0.163) | 0.94 (0.011) | 0.93 (0.032) | 0.95 (0.005) | **0.95** (0.008) | 0.87 (0.000) | 0.72 (0.024) |
| | 0.975 | **0.82** (0.000) | 0.00 (0.000) | 0.81 (0.010) | 0.75 (0.070) | 0.77 (0.043) | **0.82** (0.000) | 0.33 (0.161) | 0.00 (0.000) |
| Waveform | 0.90 | 0.93 (0.010) | 0.90 (0.011) | 0.93 (0.010) | 0.88 (0.015) | **0.94** (0.011) | 0.92 (0.017) | **0.94** (0.017) | 0.75 (0.016) |
| | 0.925 | 0.91 (0.007) | 0.85 (0.033) | 0.92 (0.014) | 0.90 (0.004) | **0.94** (0.015) | 0.92 (0.014) | 0.91 (0.017) | 0.68 (0.021) |
| | 0.95 | 0.92 (0.007) | 0.77 (0.073) | 0.92 (0.007) | 0.90 (0.038) | **0.94** (0.011) | 0.93 (0.012) | 0.87 (0.027) | 0.73 (0.018) |
| | 0.975 | 0.83 (0.014) | 0.00 (0.000) | 0.80 (0.020) | 0.81 (0.026) | **0.87** (0.035) | 0.82 (0.000) | 0.08 (0.072) | 0.00 (0.000) |
| Banana | 0.90 | **0.95** (0.009) | 0.92 (0.015) | 0.92 (0.015) | 0.93 (0.015) | 0.89 (0.029) | 0.80 (0.005) | **0.95** (0.011) | 0.92 (0.007) |
| | 0.925 | **0.95** (0.012) | 0.94 (0.013) | 0.90 (0.020) | 0.92 (0.014) | 0.85 (0.037) | 0.80 (0.003) | 0.90 (0.014) | 0.83 (0.004) |
| | 0.95 | **0.95** (0.016) | 0.92 (0.013) | 0.91 (0.015) | 0.89 (0.032) | 0.87 (0.016) | 0.85 (0.003) | 0.90 (0.003) | 0.85 (0.004) |
| | 0.975 | **0.93** (0.018) | 0.71 (0.053) | 0.85 (0.015) | 0.43 (0.152) | 0.77 (0.031) | 0.75 (0.005) | 0.65 (0.068) | 0.55 (0.034) |

Results are shown in Table 4.15. The first percentage is a measure of sensitivity whereas the second a measure of specificity. Overall the proposed technique is able to detect the vast majority of induced outliers while keeping the false positive percentage to relatively low levels ($< 15\%$). This outlier detection technique is a by product of the classification training and does not require any further computational effort.

Table 4.15: Amount of outlier and non outlier data that receive free slack. Ideally we want all the outlier data points (100%) to receive free slack and on the other side no non outlier points (0%) to receive free slack.

| **Dataset** | **% of non outliers receiving free slack** | **% of outliers % receiving free slack** |
|---|---|---|
| German | 13.15 | 100 |
| Diabetes | 6.12 | 100 |
| Thyroid | 2.94 | 50 |
| Breast | 0 | 100 |
| Heart | 10 | 100 |
| Credit | 5.40 | 100 |
| Ringnorm | 0 | 100 |
| Twonorm | 6.66 | 100 |
| Waveform | 0 | 100 |
| Banana | 4.41 | 100 |

# CHAPTER 5: CONCLUSION

In this work, we presented promising cost-sensitive learning techniques to deal with imperfect data along with the real problems in quality control and business analytics. In particular, we introduced the WRSVM in which the penalization cost is weighted to deal with the imbalanced data while a restricted amount of free slack is used to diminish the influence of outliers and misclassified points. We demonstrated that the WRSVM produces considerably superior results than the standard SVM and WSVM techniques in most datasets both with low and high outlier ratio. We reported G-mean for selected datasets to show the effect of free slack. Computational experiments show that the proposed approach achieve comparative or better performance than the SVM and WSVM, as well as other robust conventional classification methods in most cases for both low and high outlier ratios.

We particularly studied a real problem in quality control in this thesis. Detecting abnormal patterns is an important task that has practical value related to diagnostic and maintenance operations. In this work, we compared SVM against WSVM for the imbalanced CCPR problem. We tested the two algorithms for several normal and abnormal classification problems as well as multi-class classification in highly imbalanced environment. Comparison demonstrated that WSVM is better in terms of specificity and G-mean, two measures that are used for imbalanced classification problems. However, sensitivity and classification accuracy for WSVM drops, which is a compromise for correctly detecting the rare abnormal patterns. Therefore, the choice of the algorithm and the associated parameters is dominated by the proportion of available historical data, the cost of acquisition of new data and the minimal abnormal pattern alterations that ones wishes to detect.

Smart feature selection might need to be employed in order to improve classification accuracy whereas alternative imbalanced classification techniques are worthwhile to be explored. Re-sampling methods by themselves introduce biases and might not be optimal, however, their paired usage with cost-sensitive methods has to be examined in future research. Some research

works have started looking at this combined preprocessing strategy for other imbalanced problems (Anand et al., 2010; Akbani et al., 2004). Their potential usage has to be explored for CCPR as well. The potential application of clustering as preprocessing is another interesting research direction (Jo & Japkowicz, 2004). In this work, we focused our efforts in test involving fixed data time series of various $w$ length. In the future, we will focus on stream data mining and attempt an on-line classifier with an incremental real time retraining. Current study results are encouraging enough in terms of accuracy, average run length and computational time. Another important aspect for the verification and validation of the proposed methods is the testing through real case studies and datasets. However, the lack of real data is common in the majority of CCPR literature. As it is pointed out in the review paper of Hachicha & Ghorbel (2012) approximately 95.59% of CCPR literature uses simulated time series data for CCPR algorithm validation.

Finally, we believe that the future CCPR research should focus more on multi-class generalization. Since, in reality, one is interested to discriminate not only the normal versus abnormal problem but have as much information as possible about the abnormality, multi-class CCPR need to receive more attention in future works. The vast majority of previous research focuses on the binary problem with only few exceptions (Ghanem et al., 2010; Shao, 2012). In this work, we presented very promising multi-class results under a highly imbalanced environment, however, additional investigation and computational testing needs to be conducted in the future.

# APPENDIX A: MATHEMATICAL MODELS OF CONTROL CHART PATTERNS

The western electric company (1958) first documented several typical control chart patterns. These patterns were subsequently used in a large number of CCPR publications(Hwarng & Hubele, 1992; Cheng, 1997; Al-Ghanim, 1997; Al-Assaf, 2004; Assaleh & Al-assaf, 2005; Gauri & Chakraborty, 2008; Cheng et al., 2009; Guh, 2010; Shao, 2012). These simulated control charts ($a(t)$) consist of three major components, namely a constant term $\mu$, a random and normally distributed term $\varepsilon_t$, and a function $d(t)$ that models a particular abnormal pattern. This term is zero for in-control data The mathematical model for all components considered in this study can be written as:

$$a(t) = \mu + \varepsilon_t + d(t) \tag{A.1}$$

Without loss of generality, we use $\mu = 0$ and $\varepsilon_t \sim N(0, 1)$ which is consistent with previous works (Cheng et al., 2009; Yang, 2010; Shao, 2012). In particular the form of $d(t)$ for each particular pattern is as follows:

(a) *Up/ Down trends*

$$d(t) = \lambda t \tag{A.2}$$

Where $\lambda$ is the trend slope in terms of $\sigma_\varepsilon$. The parameter $\lambda > 0$ is chosen for up trends and $\lambda < 0$ for downtrends.

(b) *Up/Down shifts*

$$d(t) = \omega \tag{A.3}$$

Where parameter $\omega$ denotes the shift magnitude. Similarly to the trend patterns $\omega > 0$ for up shift and $\omega < 0$ for down shift.

(c) *Cyclic trends*

$$d(t) = \alpha \sin\left(\frac{2\pi t}{\Omega}\right) \tag{A.4}$$

Where $\alpha$ is the amplitude of the cyclic patterns, and $\Omega$ is the cyclic pattern period. For this

paper, we fix $\Omega = 8$ and treat $\alpha$ as parameter similar to previous works (Cheng et al., 2009; Shao, 2012).

(d) *Systematic trends*

$$d(t) = k(-1)^t \tag{A.5}$$

Where k is magnitude of the systematic pattern.

(e) *Stratification trends*

$$d(t) = \acute{\varepsilon_t} \tag{A.6}$$

is another abnormal pattern related to shift in the process standard deviation ($\varepsilon_t$). Parameter $\acute{\varepsilon_t}$ is a fraction of the natural process standard deviation.

# APPENDIX B: A PRACTICAL GUIDE TO WEIGHTED SUPPORT VECTOR MACHINE TOOLBOX FOR CONTROL CHART PATTERN RECOGNITION

For the evaluation of WSVM algorithm, we have developed a toolbox in MATLAB, termed WSVMToolbox. It features:

- Implementations of SVM (Vapnik, 2000) and WSVM (Veropoulos et al., 1999) techniques for time series data
- A guide to generate simulated data for different abnormal patterns, routines to pre-process data sets
- An experiment result format and functions for calculation of Sensitivity, Specificity, Accuracy, and G-mean for imbalanced classification

We note that experiments on both SVM and WSVM are conducted with LIBSVM-3.12 and LIBSVM-weights-3.12 (Chang & Lin, 2011). The whole script is developed in MATLAB and LIBSVM is interfaced in it. Therefore before the implementation of this toolbox, we suggest the user to download LIBSVM from http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Proposed Procedure

The WSVMToolbox conducts the following procedure:

- Generate data in the format of time series with specific window lengths
- Perform data preprocessing
- Use the RBF kernel

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma \geq 0. \tag{B.1}$$

- Use model selection techniques to find the best parameter C and $\gamma$
- Implement the optimal or near optimal parameters for C and $\gamma$ to train the training dataset

- Report G-mean, Sensitivity, Specificity, and Accuracy for test dataset

We explain this procedure in a more detail in the next sections.

*Data Generation*

Data based on different normal and abnormal patterns are generated using GenData.m and GenDataMulti.m functions for binary and multiclass classification. For binary classification, the user should first select the abnormal data type and set the input parameters. We provide a table for abnormal parameter type as following,

Table B.1: Abnomal pattern types symbols used in WSVMToolbox

| **Abnormal Pattern** | **Symbol** |
|---|---|
| Up trend | 1 |
| Down trend | 2 |
| Up Shift | 3 |
| Down shift | 4 |
| Cyclic | 5 |
| Systematic | 6 |
| Stratification | 7 |

Other input parameters are imbalanced ratio(r), window length(w), parameter of abnormal pattern (t) for binary classification. The imbalanced ratio (r) in the code is determined as,

The input parameters for muticlass classification is slightly different from binary classification. They consist of all abnormal parameters, the size of minority(n) and majority(m) class. The abnormal parameters are given using muticlass.mat. Furthermore, we can select window w and abnormal parameter values from Table B.2. All input parameters should be given in Main.m file for both binary and muticlass classification.

Table B.2: Imbalanced ratio symbols used in WSVMToolbox

| r | imbalanced ratio |
|---|---|
| 0 | %50 |
| 5 | %55 |
| 10 | %60 |
| 15 | %65 |
| 20 | %70 |
| 25 | %75 |
| 30 | %80 |
| 35 | %85 |
| 40 | %90 |
| 45 | %95 |

*Data Preprocessing*

Data preprocessing and scaling before applying any data mining algorithm is the key step. Scaling makes all features in the same numeric ranges. We suggest normalize all data prior to classification, so that they have zero mean and unit ary standard deviation (zscore() function in MATLAB is used). This step can be found in the first line of wsvmmodel.m and wsvmmodel-multi.m for binary and muticlass classification respectively.

*Model Selection*

The SVM and WSVM algorithms have certain parameters that need to be tuned during the training phase: C and $\gamma$ (RBF kernel). For this, we use the "grid search" model selection using cross-validation. We use exponentially growing sequences of C and $\gamma$ to identify good parameters, such as C $\in \{2^{-5}, 2^{-4},...,2^{15}\}$, $\gamma \in \{2^{-15}, 2^{-13}, ..., 2^{3}\}$. These parameter sequence sets are also suggested in Chang & Lin (2011).

*WSVM Training and Testing*

The learning process is conducted in 10-fold cross validation loop using wsvmmodel.m (binary) and wsvmmodelmulti.m (multiclass classification). For cross validation purposes, 90% of the data is used for training and the rest 10% is used for testing. The output for binary classification is sensitivity, specificity, accuracy, and G-mean. For multi-class classification, the output is the accuracy and confusion matrix table. The diagonal elements of confusion matrix table show the accurate classification percentage.

In order to run the main.m, the user should put all files including LIBSVM and LIBSVM-weights .mex files in the same folder as WSVMToolbox.

# LIST OF REFERENCES


Acuna, E., & Rodriguez, C. (2004). A meta analysis study of outlier detection methods in classification, technical paper, department of mathematics, university of puerto rico at mayaguez. *proceedings IPSI*.

Adam, A., Chew, L., Shapiai, M., Jau, L., Ibrahim, Z., & Khalid, M. (2011). A hybrid artificial neural network-naive bayes for solving imbalanced dataset problems in semiconductor manufacturing test process. In *Hybrid Intelligent Systems (HIS), 2011 11th International Conference on*, (pp. 133–138). IEEE.

Aggarwal, C. C. (2010). *Managing and Mining Uncertain Data: 3, A.*, vol. 3. Springer.

Aggarwal, C. C. (2013). *Outlier analysis*. Springer.

Agyemang, M., Barker, K., & Alhajj, R. (2006). A comprehensive survey of numeric and symbolic outlier mining techniques. *Intelligent Data Analysis*, *10*(6), 521–538.

Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. *Machine Learning: ECML 2004*, (pp. 39–50).

Al-Assaf, Y. (2004). Multi-resolution wavelets analysis approach for the recognition of concurrent control chart patterns. *Quality Engineering*, *17*(1), 11–21.

Al-Ghanim, A. (1997). An unsupervised learning neural algorithm for identifying process behavior on control charts and a comparison with supervised learning approaches. *Computers & Industrial Engineering*, *32*(3), 627–639.

Al-Ghanim, A., & Kamat, S. (1995). Unnatural pattern recognition on control charts using correlation analysis techniques. *Computers & Industrial Engineering*, *29*(1-4), 43–47.

Al-Shahib, A., Breitling, R., & Gilbert, D. (2005). Feature selection and the class imbalance problem in predicting protein function from sequence. *Applied Bioinformatics*, *4*(3), 195–203.

Alberts, L., Peeters, I., Braekers, R., & Meijer, C. (2006). *Churn Prediction in the Mobile Telecommunications Industry*. Ph.D. thesis, Maastricht University.

Alexander, S. (1987). The application of expert systems to manufacturing process control. *Computers & Industrial Engineering*, *12*(4), 307–314.

Almeida, J., Barbosa, L., Pais, A., & Formosinho, S. (2007). Improving hierarchical cluster analysis: A new method with outlier detection and automatic clustering. *Chemometrics and Intelligent Laboratory Systems*, *87*(2), 208–217.

Alwan, L., & Roberts, H. (1988). Time-series modeling for statistical process control. *Journal of Business & Economic Statistics*, (pp. 87–95).

An, W., & Liang, M. (2013). Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises. *Neurocomputing*, *110*, 101–110.

Anand, A., Pugalenthi, G., Fogel, G., & Suganthan, P. (2010). An approach for classification of highly imbalanced data using weighting and undersampling. *Amino Acids*, *39*(5), 1385–1391.

Anand, R., Mehrotra, K., Mohan, C., & Ranka, S. (1993). An improved algorithm for neural network classification of imbalanced training sets. *Neural Networks, IEEE Transactions on*, *4*(6), 962–969.

Anyfantis, D., Karagiannopoulos, M., Kotsiantis, S., & Pintelas, P. (2007). Robustness of learning techniques in handling class noise in imbalanced datasets. In *Artificial Intelligence and Innovations 2007: from Theory to Applications*, (pp. 21–28). Springer.

Aparisi, F. (1996). Hotelling's t2 control chart with adaptive sample sizes. *International Journal of Production Research*, *34*(10), 2853–2862.

Assaleh, K., & Al-assaf, Y. (2005). Features extraction and analysis for classifying causable patterns in control charts. *Computers & Industrial Engineering*, *49*(1), 168–181.

Bahlmann, C., Haasdonk, B., & Burkhardt, H. (2002). Online handwriting recognition with support vector machines-a kernel approach. In *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, (pp. 49–54). IEEE.

Balabin, R. M., & Smirnov, S. V. (2011). Variable selection in near-infrared spectroscopy: Benchmarking of feature selection methods on biodiesel data. *Analytica chimica acta*, *692*(1), 63–72.

Bamett, V., & Lewis, T. (1994). Outliers in statistical data. *The VLDB Jourrzal, New York*, *3*, 401–444.

Bao, Y., Liu, Z., Guo, L., & Wang, W. (2005). Forecasting stock composite index by fuzzy support vector machines regression. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, vol. 6, (pp. 3535–3540). IEEE.

Batuwita, R., & Palade, V. (2010). Fsvm-cil: fuzzy support vector machines for class imbalance learning. *Fuzzy Systems, IEEE Transactions on*, *18*(3), 558–571.

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, *15*(6), 1373–1396.

Ben-Gal, I. (2010). Outlier detection. *Data Mining and Knowledge Discovery Handbook*, (pp. 117–130).

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, vol. 1. springer New York.

Błaszczyński, J., Stefanowski, J., & Idkowiak, Ł. (2013). Extending bagging for imbalanced data. In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, (pp. 269–278). Springer.

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowledge and information systems*, *34*(3), 483–519.

Bolton, R., & Hand, D. (2002). Statistical fraud detection: A review. *Statistical Science*, (pp. 235–249).

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breunig, M., Kriegel, H., Ng, R., & Sander, J. (2000). Lof: identifying density-based local outliers. In *ACM Sigmod Record*, vol. 29, (pp. 93–104). ACM.

Brodley, C. E., & Friedl, M. A. (2011). Identifying mislabeled training data. *arXiv preprint arXiv:1106.0219*.

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, *36*(3), 4626–4636.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121–167.

Byvatov, E., Fechner, U., Sadowski, J., & Schneider, G. (2003). Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *Journal of Chemical Information and Computer Sciences*, *43*(6), 1882–1889.

Camci, F., Chinnam, R., & Ellis, R. (2008). Robust kernel distance multivariate control chart using support vector principles. *International Journal of Production Research*, *46*(18), 5075–5095.

Carvajal, K., Chacón, M., Mery, D., & Acuna, G. (2004). Neural network method for failure detection with skewed class distribution. *Insight-Non-Destructive Testing and Condition Monitoring*, *46*(7), 399–402.

Casillas, J., & Martínez-López, F. J. (2009). Mining uncertain data with multiobjective genetic fuzzy systems to be applied in consumer behaviour modelling. *Expert Systems with Applications*, *36*(2), 1645–1659.

Cateni, S., Colla, V., & Vannucci, M. (2008). Outlier detection methods for industrial applications. *Advances in Robotics, Automation and Control*, (pp. 265–282).

Chandar, M., Laha, A., & Krishna, P. (2006). Modeling churn behavior of bank customers using predictive data mining techniques. In *National conference on soft computing techniques for engineering applications (SCT-2006)*, (pp. 24–26).

Chang, C., & Lin, C. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 27.

Chang, E. Y., Li, B., Wu, G., & Goh, K. (2003). Statistical learning for effective visual information retrieval. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 3, (pp. III–609). IEEE.

Chang, S., & Aw, C. (1996). A neural fuzzy control chart for detecting and classifying process mean shifts. *International Journal of Production Research*, *34*(8), 2265–2278.

Changrampadi, M. H., Yun, Y., & Gu, I. Y. (2012). Multi-class ada-boost classification of object poses through visual and infrared image information fusion. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, (pp. 2865–2868). IEEE.

Chapelle, O., Haffner, P., & Vapnik, V. (1999). Support vector machines for histogram-based image classification. *Neural Networks, IEEE Transactions on*, *10*(5), 1055–1064.

Charnes, A., Cooper, W. W., Golany, B., Seiford, L., & Stutz, J. (1985). Foundations of data envelopment analysis for pareto-koopmans efficient empirical production functions. *Journal of Econometrics*, *30*(1), 91–107.

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

Chawla, N. (2010). Data mining for imbalanced datasets: An overview. *Data Mining and Knowledge Discovery Handbook*, (pp. 875–886).

Chawla, N., Hall, L., & Joshi, A. (2005). Wrapper-based computation and evaluation of sampling

methods for imbalanced datasets. In *Proceedings of the 1st International Workshop on Utility-based Data Mining*, (pp. 24–33). ACM.

Cheng, C. (1997). A neural network approach for the analysis of control chart patterns. *International Journal of Production Research*, *35*(3), 667–697.

Cheng, C., Cheng, H., & Huang, K. (2009). A support vector machine-based pattern recognizer using selected features for control chart patterns analysis. In *Industrial Engineering and Engineering Management, 2009. IEEM 2009. IEEE International Conference*, (pp. 419–423). IEEE.

Cheng, C., & Hubele, N. (1992). Design of a knowledge-based expert system for statistical process control. *Computers & Industrial Engineering*, *22*(4), 501–517.

Cheng, H., & Cheng, C. (2009). Control chart pattern recognition using wavelet analysis and neural networks. *Journal of Quality Vol*, *16*(5), 311.

Chinnam, R. (2002). Support vector machines for recognizing shifts in correlated and other manufacturing processes. *International Journal of Production Research*, *40*(17), 4449–4466.

Chiu, A., & Fu, A. (2003). Enhancements on local outlier detection. In *Database Engineering and Applications Symposium, 2003. Proceedings. Seventh International*, (pp. 298–307). IEEE.

Cook, D., & Chiu, C. (1998). Using radial basis function neural networks to recognize shifts in correlated manufacturing process parameters. *IIE Transactions*, *30*(3), 227–234.

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

Cox, T. F., & Cox, M. A. (2010). *Multidimensional scaling*. CRC Press.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press.

Cui, G., Wong, M., Zhang, G., & Li, L. (2008). Model selection for direct marketing: performance criteria and validation methods. *Marketing Intelligence & Planning*, *26*(3), 275–292.

Davy, M., Desobry, F., Gretton, A., & Doncarli, C. (2006). An online support vector machine for abnormal events detection. *Signal Processing*, *86*(8), 2009–2025.

Du, S., & Chen, S. (2005). Weighted support vector machine for classification. In *Systems, Man and Cybernetics,IEEE International Conference*, vol. 4, (pp. 3866–3871). IEEE.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification. *New York: John Wiley, Section*, *10*, l.

Due Trier, Ø., Jain, A. K., & Taxt, T. (1996). Feature extraction methods for character recognition-a survey. *Pattern recognition*, *29*(4), 641–662.

Duman, E., Ekinci, Y., & Tanrıverdi, A. (2012). Comparing alternative classifiers for database marketing: The case of imbalanced datasets. *Expert Systems with Applications*, *39*(1), 48–53.

El-Midany, T., El-Baz, M., & Abd-Elwahed, M. (2010). A proposed framework for control chart pattern recognition in multivariate process using artificial neural networks. *Expert Systems with Applications*, *37*(2), 1035–1042.

Elazmeh, W., Japkowicz, N., & Matwin, S. (2006). Evaluating misclassifications in imbalanced data. *Machine Learning: ECML 2006*, (pp. 126–137).

Estabrooks, A., Jo, T., & Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, *20*(1), 18–36.

Ezawa, K., Singh, M., & Norton, S. (1996). Learning goal oriented bayesian networks for telecommunications risk management. In *Machine Learning-International Workshop Conference*, (pp. 139–147). Morgan Kaufmann Publishers, Inc.

Fan, H., & Ramamohanarao, K. (2005). A weighting scheme based on emerging patterns for weighted support vector machines. In *Granular Computing, 2005 IEEE International Conference on*, vol. 2, (pp. 435–440). IEEE.

Fawcett, T., & Provost, F. (1997). Adaptive fraud detection. *Data mining and knowledge discovery*, *1*(3), 291–316.

Foody, G., & Mathur, A. (2004). A relative evaluation of multiclass image classification by support vector machines. *Geoscience and Remote Sensing, IEEE Transactions on*, *42*(6), 1335–1343.

Freed, N., & Glover, F. (1986). Evaluating alternative linear programming models to solve the two-group discriminant problem. *Decision Sciences*, *17*(2), 151–162.

Freund, Y., & Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, (pp. 23–37). Springer.

Freund, Y., & Schapire., R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Science*, *55(1)*, 119–139.

Fung, G., & Mangasarian, O. L. (2001). Proximal support vector machine classifiers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 77–86). ACM.

Gauri, S., & Chakraborty, S. (2008). Improved recognition of control chart patterns using artificial neural networks. *International Journal of Advanced Manufacturing Technology*, *36*(11), 1191–1201.

Ghanem, A., Venkatesh, S., & West, G. (2010). Multi-class pattern classification in imbalanced data. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, (pp. 2881–2884). IEEE Computer Society.

Ghazanfari, M., Alaeddini, A., Niaki, S., & Aryanezhad, M. (2008). A clustering approach to identify the time of a step change in shewhart control charts. *Quality and Reliability Engineering International*, *24*(7), 765–778.

Ghazikhani, A., Monsefi, R., & Yazdi, H. (2012). Svbo: Support vector-based oversampling

for handling class imbalance in k-nn. *20th Iranian Conference on Electrical Engineering, (ICEE2012)*.

Gogoi, P., Bhattacharyya, D., Borah, B., & Kalita, J. (2011). A survey of outlier detection methods in network anomaly identification. *The Computer Journal*, *54*(4), 570–588.

Groth, S., & Muntermann, J. (2011). An intraday market risk management approach based on textual analysis. *Decision Support Systems*, *50*(4), 680–691.

Guh, R. (2010). Simultaneous process mean and variance monitoring using artificial neural networks. *Computers & Industrial Engineering*, *58*(4), 739–753.

Guo, X., Yin, Y., Dong, C., Yang, G., & Zhou, G. (2008). On the class imbalance problem. In *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, vol. 4, (pp. 192–201). IEEE.

Gupta, S., Hanssens, D., Hardie, B., Kahn, W., Kumar, V., Lin, N., Ravishanker, N., & Sriram, S. (2006). Modeling customer lifetime value. *Journal of Service Research*, *9*(2), 139–155.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1), 389–422.

Hachicha, W., & Ghorbel, A. (2012). A survey of control-chart pattern-recognition literature (1991-2010) based on a new conceptual classification scheme. *Computers & Industrial Engineering*, *63*, 204–222.

Hady, M. F. A., & Schwenker, F. (2013). Semi-supervised learning. In *Handbook on Neural Information Processing*, (pp. 215–239). Springer.

Han, J., & Kamber, M. (2006). *Data mining: concepts and techniques*. Morgan Kaufmann.

Han, S., Yuan, B., & Liu, W. (2009). Rare class mining: progress and prospect. In *Pattern Recognition, 2009. CCPR 2009. Chinese Conference on*, (pp. 1–5). IEEE.

Hawkins, D. M. (1980). *Identification of outliers*, vol. 11. Springer.

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

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, *22*(2), 85–126.

Hu, G., Zhu, F., & Ren, Z. (2008). Power quality disturbance identification using wavelet packet energy entropy and weighted support vector machines. *Expert Systems with Applications*, *35*(1), 143–149.

Huang, C., Lee, Y., Lin, D., & Huang, S. (2007). Model selection for support vector machines via uniform design. *Computational Statistics & Data Analysis*, *52*(1), 335–346.

Huang, Y., & Du, S. (2005). Weighted support vector machine for classification with uneven training class sizes. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, vol. 7, (pp. 4365–4369). IEEE.

Hwang, J., Park, S., & Kim, E. (2011). A new weighted approach to imbalanced data classification problem via support vector machine with quadratic cost function. *Expert Systems with Applications*, *38*, 8580–8585.

Hwarng, H. (1995). Multilayer perceptions for detecting cyclic data on control charts. *International journal of production research*, *33*(11), 3101–3117.

Hwarng, H. B., & Hubele, N. F. (1991). X-bar chart pattern recognition using neural nets. In *ASQC Quality Congress Transactions*, vol. 45, (pp. 884–889). Milwaukee.

Hwarng, H. B., & Hubele, N. F. (1992). Boltzmann machines that learn to recognize patterns on control charts. *Statistics and computing*, *2*(4), 191–202.

Hwarng, H. B., & Hubele, N. F. (1993a). Back-propagation pattern recognizers for x control charts: methodology and performance. *Computers & Industrial Engineering*, *24*(2), 219–235.

Hwarng, H. B., & Hubele, N. F. (1993b). X control chart pattern identification through efficient off-line neural network training. *IIE transactions*, *25*(3), 27–40.

Imam, T., Ting, K., & Kamruzzaman, J. (2006). z-svm: An svm for improved classification of imbalanced data. *AI 2006: Advances in Artificial Intelligence*, (pp. 264–273).

Jang, K., Yang, K., & Kang, C. (2003). Application of artificial neural network to identify non-random variation patterns on the run chart in automotive assembly process. *International Journal of Production Research*, *41*(6), 1239–1254.

Japkowicz, N. (2000). The class imbalance problem: Significance and strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence*, vol. 1, (pp. 111–117).

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis*, *6*(5), 429–449.

Jeffery, S. R., Garofalakis, M., & Franklin, M. J. (2006). Adaptive cleaning for RFID data streams. In *Proceedings of the 32nd international conference on Very large data bases*, (pp. 163–174). VLDB Endowment.

Jha, S., & Yadava, R. (2011). Detection of outliers in surface acoustic wave (saw) chemical sensor array responses by one-class support vector machine. In *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*, (pp. 896–901). IEEE.

Jiang, M., Tseng, S., & Su, C. (2001). Two-phase clustering process for outliers detection. *Pattern recognition letters*, *22*(6), 691–700.

Jin, J., & Shi, J. (2001). Automatic feature extraction of waveform signals for in-process diagnostic performance improvement. *Journal of Intelligent Manufacturing*, *12*(3), 257–268.

Jo, T., & Japkowicz, N. (2004). Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter*, *6*(1), 40–49.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, (pp. 137–142).

Jolliffe, I. (1986). Principal component analysis. 1986. *Spring-verlag, New York*.

Jphonen, Y., & Maps, S.-O. (1995). Springer-verlag new york. *Inc., Secaucus, NJ*.

Kawamura, A., Chuarayapratip, R., & Haneyoshi, T. (1988). Deadbeat control of pwm inverter with modified pulse patterns for uninterruptible power supply. *Industrial Electronics, IEEE Transactions*, *35*(2), 295–300.

Ke, H. X., Liu, G. D., & Pan, G. B. (2013). Fuzzy support vector machine for PolSAR image classification. *Advanced Materials Research*, *639*, 1162–1167.

Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L., & Ratanamahatana, C. (2011). The ucr time series classification/clustering. Homepage: www.cs.ucr.edu/ eamonn/time series data/.

Khoonsari, P. E., & Motie, A. (2012). A comparison of efficiency and robustness of id3 and c4. 5 algorithms using dynamic test and training data sets. *International Journal of Machine Learning and Computing*, *2*(5), 540–543.

Kim, G., Chae, B., & Olson, D. (2012). A support vector machine (svm) approach to imbalanced datasets of customer responses: comparison with other customer response models. *Service Business*, (pp. 1–16).

Knorr, E., Ng, R., & Tucakov, V. (2000). Distance-based outliers: algorithms and applications. *The VLDB Journal*, *8*(3), 237–253.

Knox, E., & Ng, R. (1998). Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the International Conference on Very Large Data Bases*. Citeseer.

Kubat, M., Holte, R., & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, *30*(2), 195–215.

Kumar, S., Choudhary, A., Kumar, M., Shankar, R., & Tiwari, M. (2006). Kernel distance-based robust support vector methods and its application in developing a robust k-chart. *International Journal of Production Research*, *44*(1), 77–96.

Lane, T., & Brodley, C. (1999). Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information and System Security (TISSEC)*, *2*(3), 295–331.

Lee, D., Song, J., Song, S., & Yoon, E. (2005). Weighted support vector machine for quality estimation in the polymerization process. *Industrial & engineering chemistry research*, *44*(7), 2101–2105.

Leslie, C., Eskin, E., & Noble, W. (2002). The spectrum kernel: A string kernel for svm protein classification. In *Proceedings of the Pacific Symposium on Biocomputing, Hawaii, USA*, vol. 7, (pp. 566–575).

Lin, C., & Wang, S. (2002). Fuzzy support vector machines. *Neural Networks, IEEE Transactions on*, *13*(2), 464–471.

Ling, C., & Li, C. (1998). Data mining for direct marketing: Problems and solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, (pp. 73–79).

Liu, B., Xiao, Y., Cao, L., Hao, Z., & Deng, F. (2013). SVDD-based outlier detection on uncertain data. *Knowledge and information systems*, *34*(3), 597–618.

Liu, S., Jia, C., & Ma, H. (2005). A new weighted support vector machine with ga-based parameter selection. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference*, vol. 7, (pp. 4351–4355). IEEE.

Liu, X., Wu, J., & Zhou, Z. (2009). Exploratory undersampling for class-imbalance learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *39*(2), 539–550.

Loureiro, A., Torgo, L., & Soares, C. (2004). Outlier detection using clustering methods: A data cleaning application. In *Proceedings of KDNet Symposium on Knowledge-based systems for the Public Sector*.

Ma, X. (2012). Rfid-based business process and workflow management in healthcare: design and implementation. *Wayne State University Dissertations*, (p. 551).

Mandel, B. (1969). The regression control chart. *Journal of Quality Technology*, *1*(1), 1–9.

Moody, J. W., & Healy, K. (2014). Data visualization in sociology. *Annual Review of Sociology*, *40*(1).

Napierala, K., & Stefanowski, J. (2012). Identification of different types of minority class examples in imbalanced data. *Hybrid Artificial Intelligent Systems*, (pp. 139–150).

Nie, Y., Li, Z., Peng, S., & Chen, Q. (2009). Probabilistic modeling of streaming rfid data by using correlated variable-duration hmms. In *Software Engineering Research, Management and Applications, 2009. SERA'09. 7th ACIS International Conference on*, (pp. 72–77). IEEE.

Nitesh, V., Kevin, W., Lawrence, O., Hall, W., & Philip, K. (2002). Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Niu, Z., Shi, S., Sun, J., & He, X. (2011). A survey of outlier detection methodologies and their applications. *Artificial Intelligence and Computational Intelligence*, (pp. 380–387).

Olson, D. (2007). Data mining in business services. *Service Business*, *1*(3), 181–193.

Olszewski, R. T. (2001). Generalized feature extraction for structural pattern recognition in time-series data. Tech. rep., DTIC Document.

Omar, N., Jusoh, F., Ibrahim, R., & Othman, M. (2013). Review of feature selection for solving classification problems. *Journal of Information System Research and Innovation*, *3*.

Padmaja, T., Bapi, R., & Krishna, P. (2011). Unbalanced sequential data classification using extreme outlier elimination and sampling techniques. *Pattern Discovery Using Sequence Data Mining: Applications and Studies*, (p. 83).

Padmaja, T., Dhulipalla, N., Bapi, R., & Krishna, P. (2007). Unbalanced data classification using extreme outlier elimination and sampling techniques for fraud detection. In *Advanced Computing and Communications, 2007. ADCOM 2007. International Conference on*, (pp. 511–516). IEEE.

Papadimitriou, S., Kitagawa, H., Gibbons, P., & Faloutsos, C. (2003). Loci: Fast outlier detection using the local correlation integral. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, (pp. 315–326). IEEE.

Pechenizkiy, M., Tsymbal, A., Puuronen, S., & Pechenizkiy, O. (2006). Class noise and supervised learning in medical domains: The effect of feature extraction. In *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, (pp. 708–713). IEEE.

Peng, Y., Wu, Z., & Jiang, J. (2010). A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics*, *43*(1), 15–23.

Phua, C., Alahakoon, D., & Lee, V. (2004). Minority report in fraud detection: classification of skewed data. *ACM SIGKDD Explorations Newsletter*, *6*(1), 50–59.

Pilászy, I. (2005). Text categorization and support vector machines. In *the Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence*. Citeseer.

Pradeep, J., Srinivasan, E., & Himavathi, S. (2011). Diagonal based feature extraction for handwritten alphabets recognition system using neural network. *International Journal of Computer Science & Information Technology (IJCSIT)*, *3*(1), 27–38.

Provost, F., & Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning*, *42*(3), 203–231.

Pugh, G. (1989). Synthetic neural networks for process control. *Computers & Industrial Engineering*, *17*(1), 24–26.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81–106.

Richarz, J., Vajda, S., Grzeszick, R., & Fink, G. A. (2014). Semi-supervised learning for character recognition in historical archive documents. *Pattern Recognition*, *47*(3), 1011–1020.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

Saul, L. K., & Roweis, S. T. (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, *4*, 119–155.

Schapire, R., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine learning*, *37*(3), 297–336.

Schapire, R. E., Freund, Y., Bartlett, P., Lee, W. S., et al. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, *26*(5), 1651–1686.

Schubert, E., Zimek, A., & Kriegel, H.-P. (2014). Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery*, *28*(1), 190–237.

Shao, X. (2012). Recognition of control chart patterns using decision tree of multi-class svm. *Advances in Intelligent Systems*, (pp. 33–41).

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437.

Srivastava, A., Singhai, J., & Bhattacharya, M. (2013). Collaborative rough-fuzzy clustering: An application to intensity non-uniformity correction in brain mr images. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, (pp. 1–6). IEEE.

Steward, S. (1997). Lighting the way in' 97. *Cellular Business*, *23*.

Su, P., Mao, W., Zeng, D., Li, X., & Wang, F. (2009). Handling class imbalance problem in cultural modeling. In *Intelligence and Security Informatics, 2009. ISI'09. IEEE International Conference on*, (pp. 251–256). IEEE.

Sukchotrat, T., Kim, S., & Tsung, F. (2009). One-class classification-based control charts for multivariate process monitoring. *IIE Transactions*, *42*(2), 107–120.

Sun, R., & Tsung, F. (2003). A kernel-distance-based multivariate control chart using support vector methods. *International Journal of Production Research*, *41*(13), 2975–2989.

Sun, Y., Kamel, M., & Wang, Y. (2006). Boosting for learning multiple classes with imbalanced class distribution. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, (pp. 592–602). IEEE.

Sun, Y., Wong, A., & Kamel, M. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognitionand Artificial Intelligence*, *23*(4), 687–719.

Suresh, S., Venkatesh Babu, R., & Kim, H. (2009). No-reference image quality assessment using modified extreme learning machine classifier. *Applied Soft Computing*, *9*(2), 541–552.

Suri, N., Murty, M., & Athithan, G. (2011). Data mining techniques for outlier detection. *Visual Analytics and Interactive Technologies: Data, Text, and Web Mining Applications*, (p. 19).

Suykens, J., De Brabanter, J., Lukas, L., & Vandewalle, J. (2002). Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, *48*(1-4), 85–105.

Swift, J. A. (1987). Development of a knowledge-based expert system for control-chart pattern recognition and analysis. Tech. rep., Oklahoma State Univ., Stillwater, OK (USA).

Tang, Y., Krasser, S., Judge, P., & Zhang, Y. (2006). Fast and effective spam sender detection with granular svm on highly imbalanced mail server behavior data. In *Collaborative Computing: Networking, Applications and Worksharing*, (pp. 1–6). IEEE.

Tang, Y., Zhang, Y., Chawla, N., & Krasser, S. (2009). Svms modeling for highly imbalanced classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions*, *39*(1), 281–288.

Taylan, O., & Darrab, I. (2012). Fuzzy control charts for process quality improvement and product assessment in tip shear carpet industry. *Journal of Manufacturing Technology Management*, *23*(3), 402–420.

Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*(5500), 2319–2323.

Trafalis, T. B., Adrianto, I., Richman, M. B., & Lakshmivarahan, S. (2013). Machine-learning classifiers for imbalanced tornado data. *Computational Management Science*, (pp. 1–16).

Tu, Y., Yang, Z., & Benslimane, Y. (2011). Towards an optimal classification model against imbalanced data for customer relationship management. In *Natural Computation (ICNC), 2011 Seventh International Conference on*, vol. 4, (pp. 2401–2405). IEEE.

Van Cutsem, B., & Gath, I. (1993). Detection of outliers and robust estimation using fuzzy clustering. *Computational statistics & data analysis*, *15*(1), 47–61.

Vapnik, V. (2000). *The nature of statistical learning theory*. Springer-Verlag New York Inc.

Vasu, M., & Ravi, V. (2011). A hybrid under-sampling approach for mining unbalanced datasets: applications to banking and insurance. *International Journal of Data Mining, Modelling and Management*, *3*(1), 75–105.

Veropoulos, K., Campbell, C., & Cristianini, N. (1999). Controlling the sensitivity of support vector machines. In *Proceedings of the international joint conference on artificial intelligence*, vol. 1999, (pp. 55–60). Citeseer.

Walters, D., & Wilkinson, W. (1994). Wireless fraud, now and in the future: A view of the problem and some solutions. *Mobile Phone News*, *24*, 4–7.

Wang, C., Guo, R., Chiang, M., & Wong, J. (2008). Decision tree based control chart pattern recognition. *International Journal of Production Research*, *46*(17), 4889–4901.

Wang, L., Ni, H., Yang, R., Pappu, V., Fenn, M. B., & Pardalos, P. M. (2013). Feature selection based on meta-heuristics for biomedicine. *Optimization Methods and Software*, (ahead-of-print), 1–17.

Wang, S., Li, Z., Chao, W., & Cao, Q. (2012). Applying adaptive over-sampling technique based on data density and cost-sensitive svm to imbalanced learning. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, (pp. 1–8). IEEE.

Warren Liao, T. (2011). Diagnosis of bladder cancers with small sample size via feature selection. *Expert Systems With Applications*, *38*(4), 4649–4654.

Wei, W., Li, J., Cao, L., Ou, Y., & Chen, J. (2012). Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, (pp. 1–27).

Weiss, G. (2004). Mining with rarity: a unifying framework. *Sigkdd Explorations*, *6*(1), 7–19.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, *5*(2), 241–259.

Wu, G., & Chang, E. (2005). Kba: Kernel boundary alignment considering imbalanced data distribution. *Knowledge and Data Engineering, IEEE Transactions on*, *17*(6), 786–795.

Wu, G., Wu, Y., Jiao, L., Wang, Y., & Chang, E. (2003). Multi-camera spatio-temporal fusion and biased sequence-data learning for security surveillance. In *Proceedings of the eleventh ACM international conference on Multimedia*, (pp. 528–538). ACM.

Xanthopoulos, P., Pardalos, P. P. M., & Trafalis, T. B. (2013). *Robust data mining*. Springer.

Xi, J. (2008). Outlier detection algorithms in data mining. In *Intelligent Information Technology Application, 2008. IITA'08. Second International Symposium on*, vol. 1, (pp. 94–97). IEEE.

Xie, Y., Li, X., Ngai, E., & Ying, W. (2009). Customer churn prediction using improved balanced random forests. *Expert Systems with Applications*, *36*(3), 5445–5449.

Yang, J., & Yang, M. (2005). A control chart pattern recognition system using a statistical correlation coefficient method. *Computers & Industrial Engineering*, *48*(2), 205–221.

Yang, S. (2010). Process control using vsi cause selecting control charts. *Journal of Intelligent Manufacturing*, *21*(6), 853–867.

Yoon, K., Kwon, O., & Bae, D. (2007). An approach to outlier detection of software measurement data using the k-means clustering method. In *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*, (pp. 443–445). IEEE.

Yue, D., Wu, X., Wang, Y., Li, Y., & Chu, C. (2007). A review of data mining-based financial fraud detection research. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, (pp. 5519–5522). IEEE.

Zarandi, M., & Alaeddini, A. (2010). A general fuzzy-statistical clustering approach for estimating the time of change in variable sampling control charts. *Information Sciences*, *180*(16), 3033–3044.

Zavaljevski, N., Stevens, F., & Reifman, J. (2002). Support vector machines with selective kernel scaling for protein classification and identification of key amino acid positions. *Bioinformatics*, *18*(5), 689–696.

Zhang, Q., Liu, D., Fan, Z., Lee, Y., & Li, Z. (2011). Feature and sample weighted support vector machine. In *Knowledge Engineering and Management*, (pp. 365–371). Springer.

Zhang, Q., Liu, D., Fan, Z., Lee, Y., & Li, Z. (2012). Feature and sample weighted support vector machine. *Knowledge Engineering and Management*, (pp. 365–371).

Zhang, Y., Meratnia, N., & Havinga, P. (2010). Outlier detection techniques for wireless sensor networks: A survey. *Communications Surveys & Tutorials, IEEE*, *12*(2), 159–170.

Zhao, J., Li, X., & Dong, Z. (2007). Online rare events detection. *Advances in Knowledge Discovery and Data Mining*, (pp. 1114–1121).

Zhao, S., Zhang, H., & Li, L. (2012). A new algorithm for imbalanced datasets in presence of outliers and noise. In *Natural Computation (ICNC), 2012 Eighth International Conference on*, (pp. 30–34). IEEE.

Zhong, S., Tang, W., & Khoshgoftaar, T. M. (2005). Boosted noise filters for identifying mislabeled data. Tech. rep., Technical report, Department of computer science and engineering, Florida Atlantic University.

Zhu, X. (2006). Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, *2*, 3.

Zimek, A., Schubert, E., & Kriegel, H. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*.

Zorriassatine, F., Al-Habaibeh, A., Parkin, R., Jackson, M., & Coy, J. (2005). Novelty detection for practical pattern recognition in condition monitoring of multivariate processes: a case study. *The International Journal of Advanced Manufacturing Technology*, *25*(9), 954–963.