Rails Quality Data Modelling via Machine Learning-Based Paradigms

Ali Zughrat

Thesis submitted as a requirement for the degree of

Doctor of Philosophy

in the Department of Automatic Control and Systems Engineering

University of Sheffield, UK

May 2015

I would like to dedicate this thesis to

my parents

and

my family

ACNOWLEDGEMENT

All praises and glory are due to almighty Allah, the sole lord of the universe.

I am grateful to many people who have played a crucial role through their help and support during the course of my PhD studies. I would like to express my deep felt gratitude to my supervisors, Professor Mahdi Mahfouf and Dr. George Panoutsos for giving me the opportunity to pursue my PhD degree under their supervision with whom I consider it an honour to work with. This dissertation would not have been possible without their insights, encouragement, mentoring, wisdom, inspiration and confidence. I really appreciate and value all their constructive criticism, feedback and guidance.

I am immensely thankful to all the staff and my fellow students at Intelligent Systems Lab in Amy Johnson Building for their friendship, in particular, Osman Ishague, Alicia Adriana Rodrigues, Mohamed Ehtiawesh, Raymond Muscat, Ali Baraka, Olusayo Obajemu, Amir Khondabi, and Drs Musa Abdulkareem, Yong Yang, Sid-Ahmed Gaffour and Mouloud Denai.

I would also like to thank the academic, technical and support staff of the department of Automatic Control and Systems Engineering at the University of Sheffield for their guidance and unlimited assistance during my studies.

I greatly acknowledge my wife and my daughter for their prayers, patience, encouragement, support and understanding during the period of conducting this research. Also, I want to thank my relatives and friends in my homeland for their encouragement and support. I would like to take this opportunity to express my gratitude to my mother in law, her sons and daughters for their support and help.

I am deeply indebted to my family; mother, father, brothers and sisters who have been a great source of cooperation throughout my life. I thank them for their prayers, understanding, patience that has made it possible for me to focus on my studies hoping that this achievement is well-worth their sacrifices throughout the period I have been away from Libya. Indeed, this is a very small acknowledgement of their trustworthy love and affection.

At the end as at the beginning, all praises and thanks are to almighty Allah.

ABSTRACT

Machine-learning has emerged as a paradigm for real industrial processes modelling and classification; Industrial manufacturing processes are increasingly dependent on data-driven modelling and machine learning support for large process structures so as to recognise complex patterns, reduce cost management, enhance quality control measures and make intelligent decisions based on the measured data. Complex industrial processes, e.g., the rails manufacturing process, are known to include hundreds of sub-processes. Hence the overarching purpose of this research is to develop a novel and efficient rails data classification framework to address several challenges associated with complex rails manufacturing process operated by Tata Steel Europe. The modelling problem is not trivial as the data are highly imbalanced with the number of good rails being much higher than the rejected rails. Another major challenge in data-driven modelling is associated with the volume and properties of the data.

This thesis contains four key contributions. Firstly, data pre-processing and feature selection environment based on RapidMiner and Matlab packages is presented to deal with complex and large scale data. An imbalanced problem is dealt with via two different approaches i.e., bootstrapping-based over-sampling and under-sampling. Such approach succeeded in selecting the most important variables to rails production process and providing more balanced rails dataset. Secondly, an iterative support vector machines with bootstrapping-based over-sampling and under-sampling classification approach is presented. The novelty and value of the integrated strategy lies in iteratively addressing the volume and complexity scenarios of rails data while maintaining good generalization capabilities. The next contribution lies in producing a new approach to rails data classification via iterative fuzzy support vector machine (IFSVM)-based data sampling. The proposed method incorporates the class distribution advantages of efficient data sampling and the unique learning mechanism of IFSVM. This technique delivers an optimal trade-off between the execution time and the overall classification performance.

In the final contribution, a new integration strategy combining IFSVM with fuzzy c-means clustering (FCMs) is proposed. FCMs enabled the proposed IFSVM to be applied to a small scale dataset thus reducing the number of support vectors. The integration concept potentially inhibits the computational complexity of the proposed

IFSVM and thus improves the classification performance since the complexity of computations is proportional to the number of support vectors.

TABLE OF CONTENTS

Chapter 6 - ITERATIVE FUZZY SUPPORT VECTOR MACHINES: RAILS DATA CLASSIFICATION FRAMEWORK ………………...…..……..…...… 120

LIST OF FIGURES

LIST OF TABLES

GENERAL ABBREVIATIONS

ANNs	Artificial Neural Networks
ANFIS	Adaptive Neural Fuzzy Inference System
EDA	Exploratory Data Analyses
FP	False Positive
FN	False Negative
FSVMs	Fuzzy Support Vector Machines
FSVMs-CIL	Fuzzy Support Vector Machines- Class Imbalance Learning
FCMs	Fuzzy C-Means Clustering
GRBF	Gaussian Radial Basis Function
Gm	Geometric-mean
GA	Genetic Algorithm
ISVM	Iterative Support Vector Machine
IFSVM	Iterative Fuzzy Support Vector Machine
KDD	knowledge Discovery in Databases and Data mining
KKT	Karush-Kuhn-Tucker Condition
LSVM	Lagrangian Support Vector Machines
MLP	Multilayer Perceptron
NBC	Naïve Bayesian Classifier
NDT	Non- Destructive Testing
NN	Neural Network
Y0T	Output of Training Dataset
PCA	Principle Component Analysis
PSVM	Proximal Support Vector Machine
QP	Quadratic Programming
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
RBF	Radial Basis Kernel Function

SVMs	Support Vector Machines
SMO	Sequential Minimal Optimization
SMOTE	Synthetic Minority Oversampling Technique
SVs	Support Vectors
TP	True Positive
TN	True Negative
X0T	Training Dataset
WLSVM	Weighted Lagrangian Support Vector Machines
WSVMs	Weighted Support Vector Machines

GENERAL SYMBOLES

$\boldsymbol{\varphi}(.)$	Activation Function
$\boldsymbol{b}$	Bias
$\boldsymbol{s_i}$	fuzzy membership function
$\boldsymbol{K}$	Kernel
$\boldsymbol{\alpha_i}$	Lagrangian Multiplier
C	Misclassification Penalty or the Regularization Parameter
$\boldsymbol{\varphi}$	Mapping Function
$\boldsymbol{\mu}$	Membership Function
$\boldsymbol{w}$	Norm of the Hyper-plane
$\boldsymbol{Y}$	Observed Output
$\boldsymbol{\widehat{Y}}$	Predicted Output
$\boldsymbol{p_{ij}}$	Pearson Correlation Coefficient
$\boldsymbol{R_{mm}}$	Ratio of Majority to Minority Class
$\boldsymbol{R}$	Symmetric Matrix
$\boldsymbol{H1, H2}$	Separating Hyper-planes
$\boldsymbol{\xi}$	Slack Variable
$\boldsymbol{\sigma}$	Variance Parameter or Width of the Gaussian RBF

Chapter 1

Introduction

1.1 Background and Motivations

Today's rails transport offers one of the safest cargo and passenger transport systems around the globe. Nowadays, a large capital investment and innovations are required in steel rails and locomotive power since the cost of laying down the rails infrastructure can be very high where safety measures, inspections and maintenance must be guaranteed for the machinery and the overall infrastructure. However, high quality materials can reduce maintenance costs and improve reliability. Tata Steel Europe is one of the leading steel manufacturing companies and supplier of high quality rails products. Process enhancement, investment, research and development allow the company to improve products quality and reduce manufacturing costs required by the rails production. This research deals with investigating rails manufacturing process operated by Tata Steel Europe, assessing the quality of final products of rails production route and design classification frameworks based on machine learning paradigms.

The future competitiveness of the rails industry depends significantly on its ability to tailor the produced rails to meet desired specifications with respect to the mechanical properties and quality assurance. Industrial process plants (e.g., rails manufacturing route) are usually heavily equipped with a large number of instruments (e.g., sensors) to deliver data for process production quality analysis and control. The collected manufacturing data contains useful information and knowledge that would be further integrated with the main manufacturing process to enhance products, reduce costs and improve decision making. This volume of data could not practically be visualized and analysed by hand within a reasonable timeframe. Accordingly, data-driven modelling approaches play a crucial role in dealing with large scale-datasets, nonlinear input-output mapping, learning to recognise complex useful patterns and making intelligent decisions based on the data.

The quality of rails can be either good or bad based on non-destructive testing (NDT) and human-based knowledge. This research focuses on the small number of rejected rails verified via an automatic and manual ultrasonic testing for the presence of internal irregularities such as cracks and flaws, to find root causes as well as identifying bottlenecks in the production route and thus applying appropriate control measures to improve process yields (reduce defects). The data accumulated from the rail manufacturing process is the culmination of more than two years of production period. The rail production line consists of three key production stages, steel making, continuous casting, rolling and finishing as will be shown explicitly in chapter 3.

1.2 Problem Statement

The original rails data collected from the rails production route is very large, with over 200 variables and over 65000 data records covering a two-year production period. Large scale data sets mean that data manipulation is not straight forward and therefore data classification is not possible. A Tata Steel process expert will normally be closely involved in the data pre-processing stage to clarify variable correlations and redundancy among data items and therefore construct a better set of rails data. Due to non-uniform rails data formats and huge volumes of rails data, it is a true challenge to optimise the rails manufacturing process of knowledge acquisition from data with machine learning-based techniques. Accordingly, data pre-processing environment including a solid variable selection framework for the rails manufacturing process will be designed for inputs and variables selection. It is worth mentioning that all potential inputs were measured during rails production process and where consequently an enormous increment of rails data inputs occurs. The influence of these inputs to the final product varies and thus should be assessed for the following reasons:

- Computationally expensive training procedure due to the ‘curse of dimensionality’ is more likely to weaken the overall performance of the models and classifiers as the number of input variables increases. Thus, an input reduction procedure and/or a powerful hardware equipment may be required.
- Redundant and insignificant variables may disturb the true input-output relationships.

Many traditional approaches to data-driven modelling and machine-learning classification problems assume that the target classes share similar prior probabilities. However, this assumption is grossly violated in many industrial problems. More often than not, it is the case that the ratios of prior probabilities between classes are extremely skewed. An initial inspection for rails data has shown that there are only few rejected rails. This situation is known as the imbalanced dataset problem. The amount of imbalance varies depending on the problem. A data set is imbalanced if the samples belonging to the majority class outnumber the samples belonging to the minority class. Since standard machine learning techniques and other modelling algorithms yield better classification performance with balance data sets, quality classification is not reachable with the current rails data set structure. The classification would always be biased in favour of the dominating class (majority), while the data belonging to the minority class tend to be misclassified. Therefore, direct data resampling approaches are to be applied to change the class distribution of rails data. Most studies designed to address the imbalanced dataset problem have dealt with low dimensional data. However, rails dataset gathered from rails manufacturing route involve extremely high dimensionality. A common solution to the imbalance problem is data resampling. The most widely applied data sampling strategies are external methods, such as over-sampling and under-sampling or internal methods such as cost sensitive learning. Other algorithms combine the aforementioned methods to minimize their drawbacks and enhance classifiers’ generalization performance. Both data oversampling and under-sampling strategies will be applied in this research to rails dataset.

Another problem towards a successful rails data classification is the ability of machine learning-based paradigms to effectively tackle the noise and outliers that exist in rails dataset. Consequently, a fuzzy-based machine learning paradigm titled iterative fuzzy support vector machine (IFSVM) is presented to address such concern. Fuzzy support vector machine (FSVM) works similarly to support vector machine (SVM), except that a specific membership degree is given to each data point so that various data points can make different contributions to the decision surface learning. The FSVM model prevents noise and outliers from creating narrower margins. However, in the SVM model case, equally training each data point may cause over-fitting. The calculation of membership values is based on the sparse distribution of the training points, with outliers and noise being assigned proportionally smaller membership values than other points.

The final challenge in rails data classification is the curse of dimensionality. A Sequential Minimal Optimization (SMO) technique will be employed with machine learning iterative-based scheme (i.e., ISVM and IFSVM) to break the quadratic optimization (QP) problem into a series of small QP problems. Fuzzy C-means (FCM) clustering scheme will also integrated with the proposed machine learning-based paradigms not only to solve the above concern but also to reduce the number of support vectors and enhance classification performance.

1.3 Aims and Objectives

The aim of this research is to design classification architectures to deal with imbalanced rails data provided by Tata Steel Europe via machine learning-based paradigms. The rails production system is multidimensional with more than 200 input variables. The output that represents the quality of rails is either good or bad based on non-destructive testing (NDT).

Therefore, the objectives of this project are as follows:

1. To develop a new framework for identifying the most relevant data variables in rails production route via correlation analysis and neural-fuzzy based model input selection approach to eliminate redundant information and construct a parsimonious dataset that represent the original rails production data.

2. To develop a modelling technique based on an adaptive neuro-fuzzy inference system (ANFIS) classification with fuzzy C-means clustering.

3. To address the class imbalance problem of rails data via applying data resampling schemes i.e., bootstrapping-based over-sampling and under-sampling.
4. To develop a machine learning-based paradigm titled iterative support vector machine (ISVM) with bootstrapping-based over-sampling and under-sampling for rails data classification.

5. To develop an iterative fuzzy support vector machine classification algorithm with bootstrapping-based over-sampling and under-sampling and extend the developed ISVM modelling technique to include fuzzy set theory.

6. To optimise the proposed IFSVM via designing an integration strategy with FCMs clustering to reduce the number of support vectors and achieve better generalisation ability.

1.4 Thesis Outline and Contributions

The overarching aim of this research is to develop generic machine learning-based paradigms with applicability in various engineering and scientific disciplines. In this particular work, the proposed techniques aim to cope with various challenges that exist in rails quality data provided by Tata Steel Europe as described in Section 1.2. The sub-sections below outline the major contributions of the thesis and provide a brief description of each chapter.

The objective of Chapter 2 is to provide a literature review that covers a wide perspective of knowledge discovery in data bases and data mining, including machine learning-based paradigms. Available machine learning and data mining techniques from a variety of applications and research disciplines are reviewed providing the methodology of technique selection. Most recent and novel machine learning approaches are classified and discussed, providing the motivations for choosing data-driven machine learning approaches.

Chapter 3, titled ‘rails through manufacturing process’ presents an overview of rails manufacturing route operated by Tata Steel Europe. A clear explanation is given in this chapter of the rails-through process data infrastructure and key production stages of rails production route i.e., steel making, continuous casting, rolling and finishing. This chapter also emphasises briefly the current research on rails data modelling. Finally, the challenges and opportunities related to rails manufacturing data are addressed.

Chapter 4, titled ‘rails data pre-processing, neural-fuzzy model input selection and artificial neural network modelling’ presents the new rails data processing framework based on exploratory data analysis and input variables selection scheme to find the most important inputs and a parsimonious model data set. A well-structured environment based on RapidMiner software, excel and Matlab is designed to handle the curse of dimensionality of rails data due to the fact that complexity of models and classifiers strongly depends on the number of input dimensions and the size of data samples. Applying the above mentioned techniques, will identify data outliers, redundant variables missing value, correct wrong data entries and most importantly select the most relevant inputs to the rails manufacturing process. This chapter also introduces an adaptive neural-fuzzy inference system classification technique based on fuzzy C-means Clustering. The integration strategy of ANFIS with FCMs scheme aims at yielding a good generalization ability in terms of performance accuracy and consistency.

Chapter 5 introduces a new machine-learning classification technique titled ‘iterative support vector machines with bootstrapping-based over-sampling and under-sampling’. The proposed algorithm tackles the problem of learning from severely imbalanced Rails dataset via a new iterative support vector machine algorithm with bootstrapping-based over-sampling and under-sampling, combining the good generalisation ability of SVMs with the class distribution advantages of resampling techniques. The novelty and value of the presented classification formulation lies in simplifying the complexity (curse of dimensionality) of rails dataset by integrating data sampling schemes with iterative support vector machines. This contribution was presented in the 19th World Congress of the International Federation of Automatic Control IFAC, Cape Town, South Africa, August, 2014.

Chapter 6 is inspired by the idea of margin maximization to promote the generalisation capacity of the ISVM classifier presented in Chapter 5 via utilizing fuzzy logic concept. In this chapter a fuzzy margin is proposed and optimised to boost the generalisation capacity of ISVM. The idea of SVMs is reformulated into a new FSVM by assigning a fuzzy membership function to each data point. Therefore, a new formulation titled ‘Iterative fuzzy support vector machines classification with bootstrapping-based over-sampling and under-sampling is presented. This integration concept is applied to two rails datasets and reveals a significant improvement in classification performance of the proposed IFSVM in contrast to ISVM. Another contribution is also presented in this chapter defined as iterative fuzzy support vector machines-based fuzzy C-means (IFSVM-FCMs) clustering. Fuzzy C-means clustering FCMs enables the proposed IFSVM learning algorithm to be applied on small scale datasets without high computational costs. Instead of using large dataset, the integration concept with FCMs that is used to provide the classifier with a smaller dataset proves to be promising and demonstrate several advantages not only in providing better generalization and accuracy but also in drastically reducing the number of support vectors. Part of this work will be presented in the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2015), August, Istanbul, Turkey, 2015. It is worth mentioning that the radial basis function (RBF) Kernel-type applied in Chapters 5 and 6 with the proposed machine learning-based paradigms could be seen as a sensible alternative to the use of complex polynomials. Radial basis function offers superior generalization performance.

Chapter 7, titled 'conclusions and future work', summarises the main findings of this thesis and also outlines recommendations for future work.

Chapter 2

A Review of Applications of Machine-Learning for Large Scale Complex Systems

2.1 Introduction

Today's manufacturing consists of different types of processes. These processes are complex and caught between the growing need for cost minimization, product quality improvement, process safety and short manufacturing time. For these requirements to be achieved, the industrial process parameters have to be chosen very carefully. The selection of optimum process parameters has a significant influence on reducing manufacturing costs, enhancing productivity of the process and ensuring a better quality of products Venkata Rao, 2011. However, for manufacturing process optimization, the major challenge one may encounter is the fact that such processes are complex and are highly non-linear. Due to the high number of influential parameters and massive complexity of many industrial processes, conventional modelling and optimization methods are no longer adequate. Advanced modelling and data classification techniques have become increasingly a crucial target for complex manufacturing process modelling, classification and optimization.

The primary objectives of this chapter are to provide a broad overview of the theory and concepts relevant to large complex systems modelling and the existing research pertinent to the problem of rail manufacturing process data modelling and classification. It is also to deliver a methodology: how the most appropriate approaches to rails manufacturing process modelling and classification are selected?

Available machine learning and knowledge discovery in databases techniques are explored from a variety of application disciplines and research. The following review of the literature is organized based on these subject areas where the aims are to address the key research issues of extracting knowledge and interesting patterns from large complex systems and the limitations and capabilities of existing approaches. It also identifies the current trends and common difficulties, provides some examples of successful applications and outlines the challenges with the problem of rail manufacturing process modelling and classification. It is vital to discuss the definitions and methodologies employed in the process of classifying and modelling large complex systems to benefit those who are not familiar with this domain.

The last section of this chapter will outline some challenges in applying data mining to a real complex manufacturing process i.e., the curse of dimensionality and the class imbalance problem.

2.2 Data Driven Modelling Approaches

As an engineering discipline, data-driven approaches comprise a technology that describes various complex behaviours, and have recently been an attractive alternative for modelling many complex industrial systems. Data driven modelling consists of analysing the system's behaviour straight forwardly form historical data and finding correlations and relationship between the input and output variables without any explicit knowledge of the physical behaviour of the system or/and human intervention (Michell 1997; Solomatune *et al.* 2008).

A distinct feature of data-driven modelling techniques is that no prior information about the process under study is necessary. Such techniques play a crucial role in data modelling, optimization and control of large complex industrial processes for which other mathematical based methods are expected to fail in identifying the bottlenecks, because of the difficulty in describing unexpected disturbances and in handling the complexity of such systems as they often consist of several sub-processes.

Industrial process plants are usually heavily equipped with a large number of instruments. The primary objective of such instruments is to deliver data for process production quality analysis and control. The collected manufacturing data contains knowledge and useful information that would be further integrated with the main manufacturing process to enhance products, reduce costs and improve decision making.

It was approximately two decades ago when researchers from academia and industry began to explore and benefit from big size data being measured and stored in online servers by constructing predictive models based on this data. The concept of 'big data' (data from multiple sources) has since emerged as an important vehicle for exploiting quantitative information to extract meaningful knowledge.

In the last two decades data-driven modelling approaches established themselves as a remarkable alternative to the standard means for classification and modelling of large complex processes. The term industrial complex systems here refers to a class of problems for which the volume of data variables is either significantly large and/or each of the data variables has a behaviour which is entirely unknown or individually unpredictable, but in spite of this, the system as a whole can possess an analysable properties. Industrial manufacturing processes are increasingly demanding data-driven modelling and machine learning support for large process structures containing hundreds of sub-processes. Technically, such process structures are categorized by the solid relation to the assembly of products (e.g., Rails production Process).

Data-driven methods for data analysis and systems learning have to a great extent been probabilistic, but recently there has been a potential increase in the development of other types of data driven systems, especially with the machine learning community. Recently, and within the machine learning community, data-driven methods have been significantly developed, achieving the same level and even a better overall performance in contrast to other modelling techniques. A simple Schematic of the general approach to data driven modelling is presented in Figure 2.1 (Solomatune *et al.* 2008).

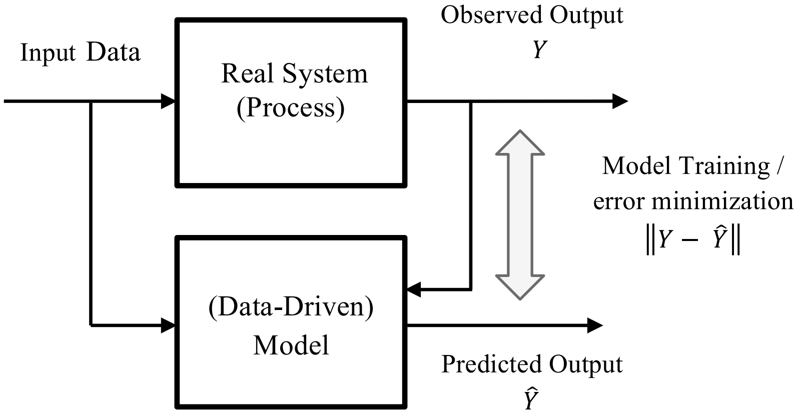

Input Data
Real System (Process)
Observed Output Y
Model Training / error minimization $\|Y - \hat{Y}\|$
(Data-Driven) Model
Predicted Output $\hat{Y}$

Figure 2.1: Schematic of general approach to data-driven modelling (Solomatune *et al.* 2008)

The General principle that the date-driven modelling process follows consists of:

- Study the problem.
- Data collection.
- Data preparation.
- Feature selection.
- Build and test the model.

These core steps generally follow the so-called Occam's razor principle (Mitchell 1999; Solomatune *et al.* 2008). In data-driven modelling, it is not only the model parameters that are subject to optimization but also model's structure. There is a considerably increasing attention given to data-driven models particularly for the purpose of production quality enhancement, process control and system performance monitoring. However, great challenges arise due to the complexity of the industrial processes in both structure and automation degrees (Yin *et al.* 2014). In the Engineering domain, large complex systems development (e.g., Rail manufacturing process) necessitates the coordination of hundreds of sub-processes. One of the biggest challenges of learning from such complex systems is computational efficiency (Han *et al.* 1993). Another major challenge in data-driven modelling is associated with the volume and properties of the data. Practically, the data accumulated from industrial processes are strongly co-linear (Kadlec *et al.* 2009).

As described in Chapter 1, the rail manufacturing process is a complex process that includes hundreds of data variables and thousands of data records are difficult to be modelled either via knowledge based approaches (linguistics) or via physical based approaches.

Most data-driven methods which represent large advances on most standard experimental modelling consist of the following overlapping fields (Solomatine and Ostfeld 2008):

- Soft Computing: It includes evolutionary computing, fuzzy logic and neural networks; it also includes other areas within machine learning.
- Artificial Intelligence: it is referred to as the study of how expert intelligence (human based knowledge) can be integrated into computers.
- Computational Intelligence: it is known to be very close to computational Intelligence field but with special emphasis given to fuzzy rule-based systems, artificial neural networks and evolutionary computation.
- Machine Learning: they are considered as a sub-area of artificial intelligence that focuses on computational Intelligence's theoretical foundations.
- Data mining in databases and Knowledge discovery (KDD) focuses on the development of algorithms and techniques for making sense of data. Data mining is considered as a part of a wider KDD.

According to Zain *et al*. (2012), Garg *et al*. (2013), Khashei and Bijari (2010) and Sadoyan *et al.* (2005), one of the commonly used data-driven approaches is artificial neural networks. They are self-adaptive tools that employ universal function approximation to estimate any function with arbitrary accuracy. Artificial neural networks are computational networks that attempt to simulate the networks of nerve cell (neurons) of the biological (animal or human) central nervous systems (Graupe 2006). Neural networks-based approaches have gained a significant reputation in solving data mining manufacturing problems. Garg *et al.* (2014) introduced an empirical data driven modelling of rapid prototyping (RP) process i.e., fused deposition modelling. A comparative analysis between the modelling techniques such as regression analysis and artificial neural networks (ANNs) was discussed. Another study was proposed by Asiltürk and Çunkaş (2011) to develop surface roughness model based on artificial neural network and multiple regression to predict surface roughness in steel. An experiment was designed to measure cutting speed, feed and cutting of depth. A high accuracy of surface roughness was estimated by ANN in contrast to multiple regression models. Other examples of data-driven approaches include Bayesian approaches (Kumar and Pal 2011), hidden Markov Model (Tobon-Mejia *et al.* 2012), fuzzy rule based systems (Zadeh 1965).

Artificial neural networks ANNs have also been described as a fault diagnosis paradigm (Chen and Liao 2002; Hoskins *et al.* 1991). Liao *et al.* (2001) presented a multi-layer perceptron neural network type on modelling welding data. Liao and Wen (2007) published a survey about the application of ANN as a powerful classification and clustering tool from 1995 to 2005. ANN has been also introduced as a data-driven approach for quality improvement in industrial process. The neural network model presented by Oh *et al.* (2001) was used to establish the relationship between quality variables and process. The proposed framework succeeded in identifying the main cause of malfunctions and also provide parameter enhancement.

The availability of large amounts of data has identified a serious problem as to how to extract a useful knowledge. Despite the useful analytical properties of artificial neural network approaches, they are well known for not being able to handle large size databases and their practical capabilities are limited by the curse of dimensionality (Bishop 2006). They also require a large computational time for big data analyses (Sadoyan *et al*. 2005). Their solution to the optimization problem relates to local minima whereas other learning techniques such as Support vector machine point at global maxima (Trotter *et al*. 2001). Hence, artificial neural networks cannot necessarily be the appropriate choice for modelling large complex systems.

2.3 Knowledge Discovery in Databases and Data Mining (KDD)

Across wide application areas, data are being accumulated and stored at a rapid rate and large scale. The availability of such large volumes of data has led to the problem of how to extract a meaningful oriented knowledge. Consequently, new advanced computational methods and tools have been developed to help experts and researchers extract knowledge and useful patterns from large databases (Fayyad *et al.* 1996). These tools and paradigms are the subject of knowledge discovery and data mining fields. Knowledge discovery and data mining (KDD) is the overall process of finding useful knowledge, regularities and relations among the observed data (Fayyad. *et al*. 1996; Giudici 2003; Klosgen and Zytkow 2002). Fayyad *et al.* (1996) define the process of KDD as the nontrivial process of finding novel, ultimately understandable and potentially effective patterns in data. Such a process usually consists of the following steps: (i) data preparation, (ii) data pre-processing, (iii) data mining, (iv) data evaluation (v) implementation (Köksal *et al.* 2011). Data mining is usually seen as a single step of KDD. The inclusion of data mining and the extraction of useful patterns in databases are also known by different names by various communities (e.g., information discovery, knowledge extraction, pattern processing and information harvesting) (Fayyad. *et al.* 1996). The overall structure of KDD process is illustrated in Figure 2.2.

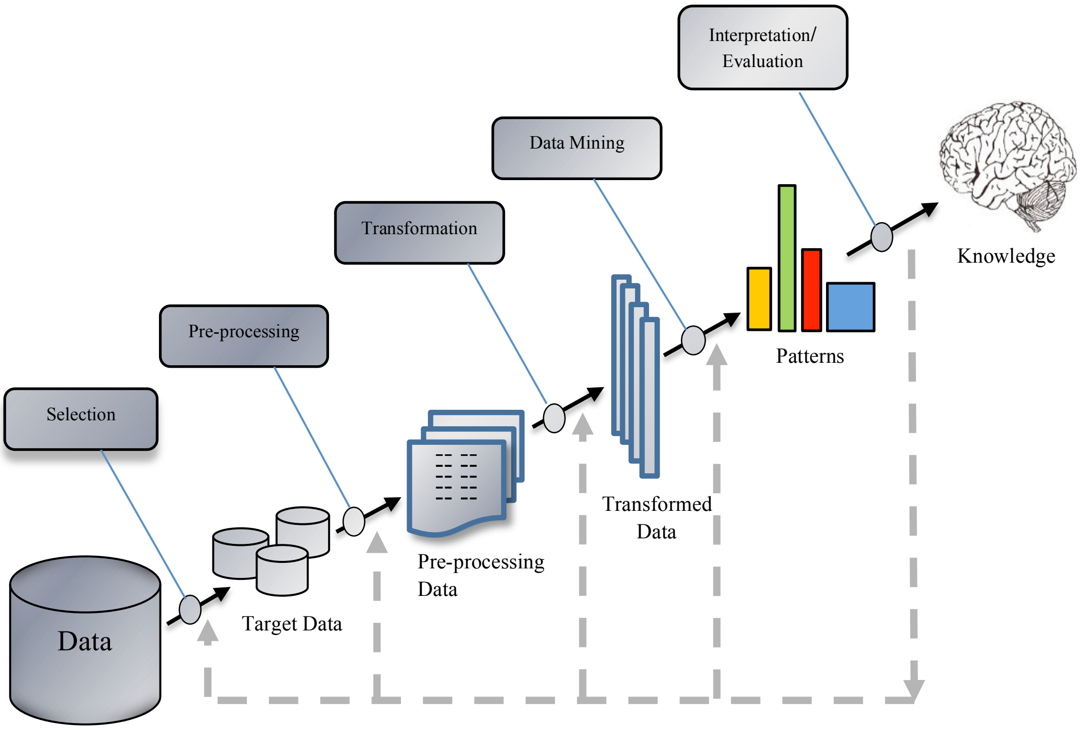


Figure 2.2: The architecture of knowledge discovery process (Fayyad. *et al.* 1996)

In practice, knowledge discovery and data mining techniques have broadly and successfully been applied to provide an effective vehicle for quality improvement and a better control of industrial products and processes. The rail manufacturing process is complex and faces major challenges related to product quality, process monitoring and fault diagnosis (Harding *et al*. 2006). Han and Kamber (2001) and Harding *et al.* (2006) stated that the type of knowledge to be explored defines the data mining algorithms to be executed. Several KDD and data mining techniques have been developed to overcome these problems including data classification, prediction, concept description regression and clustering techniques (Harding *et al.* 2006). A set of tasks can be accomplished via practical data mining in many problems of many application areas (e.g., classification, estimation, clustering and prediction).

Wu *et al.* (2008) has presented and discussed the top 10 commonly used data mining techniques in the last two decades, some of which are general and applicable to many problems in any field (Berkhin 2002), whereas others are specifically tailored to tackle a certain class of problems. Köksal *et al.* (2011) has pointed out that data mining techniques have recently applied and successfully solved quality and control problems of large complex systems.

Pham and Afify (2005) and Choudhary *et al.* (2008) have also reviewed data mining applications in industrial domains. They discussed several paradigms and evaluated their advantages and drawbacks in different real applications where they have been effectively employed. Harding *et al.* (2006), Feng and Kusiak (2006), and Choudhary *et al.* (2008) have carried-out a survey on the applications of data mining in engineering design and manufacturing and clearly show the potential scope of data mining in these fields in accomplishing competitive advantages in contrast to other techniques. The main advantage of data mining in contrast to other conventional techniques is that the collection of data required to be analysed is performed during normal runs of the industrial process under study. Consequently, there is no need to dedicate a special machine for data collection.

Practically, the area of systems modelling in manufacturing environments is a fertile research ground. Different techniques and tools for exploiting data have recently been proposed. In fact, data mining techniques can be mainly divided in two categories: classification and clustering techniques (Mucherino *et al.* 2009). In practice, the main targets of data mining concept (prediction) can be achieved using the primary data mining methods i.e., classification and clustering. Figure 2.3 shows data mining major strategies.

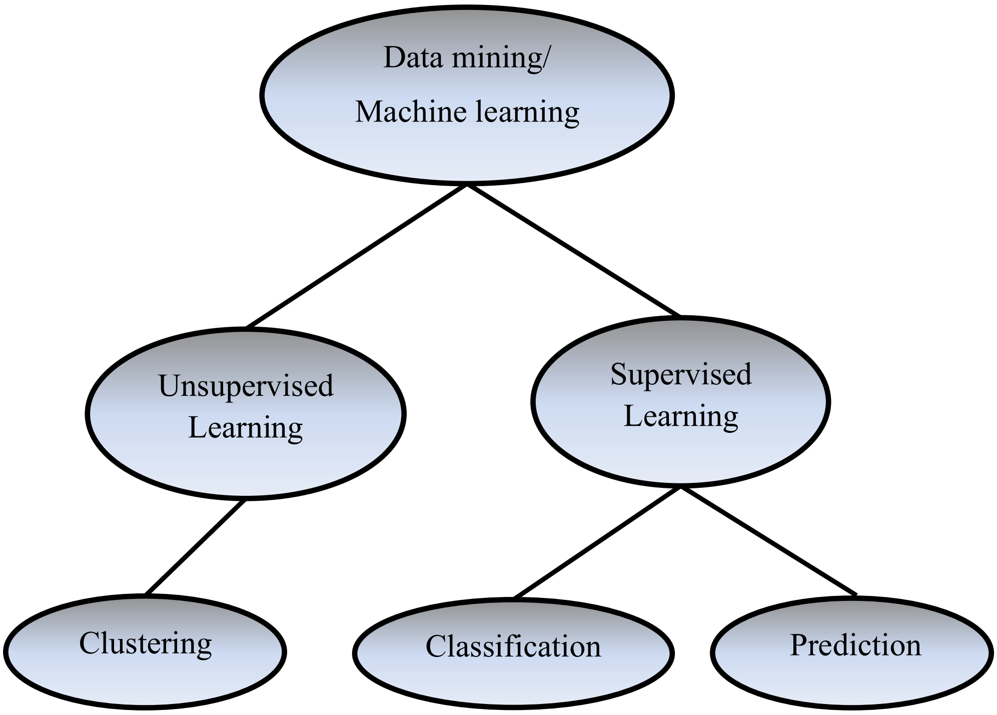


Figure 2.3: Data mining and machine learning core strategies

The data mining and machine learning strategy is divided into two categories: supervised learning and unsupervised learning (Bishop 2006). Data classification and prediction are the core areas of supervised learning. Within the category of unsupervised learning, one of the fundamental tools is clustering which seeks to extract knowledge and information from unlabelled data. The idea of supervised and unsupervised learning will be briefly demonstrated in the following section.

2.4 Supervised and Unsupervised Learning

The two fundamental classes of learning methods are supervised and unsupervised learning. In supervised learning, the cases i.e., data points and their labels are both known and provided to the algorithm (Fung 2001). The aim is then, to learn the concept in the way that when a new case appears to be classified, the algorithm should predict a label for this case (predict the correct output value for any valid input objects) (Fung 2001); (Bishop 2006). In other words, supervised learning is related to "learning by example" techniques, which means that examples of problem are presented to a learning agent e.g., Neural Networks, Support vector Machines as well as the solution. The goal is then, for the learning agent to provide good solutions to the similar unseen problems (Oliver 2004). The learned model can be represented in different forms either mathematically or as classification rules. As described in Figure 2.3, one of the common supervised learning techniques is data classification.

Unlike supervised learning, unsupervised learning refers to the problem of finding a hidden structure (clusters, patterns) in unlabelled data (Duda *et al*. 2001). In other words, unsupervised learning is defined as given a training data that is not hand-labelled and attempts to find patterns in the data that can be used to determine the correct output value for new data instances (Duda *et al.* 2001).

Unsupervised learning can be considered as "trial-and-error" learning which means that there is no error or optimal signal to evaluate desirable solutions (Oliver 2004). In machine learning, procedures that use unlabelled data are said to be unsupervised. In practice, if the experimental data is not hand-labelled by their category membership it is not possible to learn and explore its internal features. To cover this dilemma, some assumptions are to be considered (Duda *et al*. 2001):

1) Prior probabilities for each observer are known;
2) The class-conditional probabilities are also known;
3) The values of parameter vector are known;
4) Date category labels are unknown.

Unsupervised Learning is a closely related task to the problem of density estimation and includes many techniques that aim to extract and demonstrate the key specification of the data (Duda *et al.* 2001). Most algorithms employed in unsupervised learning are based on data classification and data mining methods for the purpose of pre-processing the data. One fundamental approach to unsupervised learning is data clustering (Fung 2001). The next section will provide more thorough review of classification methods and how these techniques relate to specific data mining applications (manufacturing processes).

2.4.1 Classification

In practice, the nature of the data effectively governs whether the model is suitable for classification, estimation, or prediction (Chen *et al*. 1996); (Roger and Girolami 2011). Classification is the learning technology that classifies the data into one of the pre-defined classes (Fayyad *et al.* 1996). The input space is divided into decision regions whose boundaries are called decision surfaces (Bishop 2006). Classification is a functional learning tool in many manufacturing areas, for example, in rail manufacturing process, defected rails are classified to find patterns and derive the rules to enhance productivity.

There exist a wide variety of classification techniques such as Radial basis function, support vector machine (SVM), neural networks, nearest neighbour, Bayesian classification and decision trees. Classification techniques are applied in many application areas including fraud detection, speech recognition and medical diagnosis (Alpaydim 2010); (Choudhary *et al.* 2008). Other techniques are also utilized for classification purposes such as genetic algorithm (GA), rough set theory, fuzzy logic and various hybrid methods (Han and Kamber 2006). It is important to highlight that each technique provides better performance than others when applied to a certain problem. Therefore, there is no universal machine learning and data mining tool that suits every problem (Fayyad *et al.* 1996).

A major challenge for data classification algorithms is the structure of the data. Typically, the data accumulated for industrial manufacturing processes are strongly co-linear with high level of noise and redundancy. Apart from the above stated challenges, these data are most likely to include hundreds or/and thousands of data records (Kadlec *et al*. 2009).

Knowledge acquisition and machine learning are well known problem in building advanced expert systems. Irani *et al.* (1993) developed one of the earliest expert systems for modelling and diagnosis for the semiconductor manufacturing process based on knowledge extraction. The designed paradigm is reliable with the data and meets the expectations of process experts. A broad discussion about the applicability of data mining techniques on in-line control, automatic default classification and inspection is given by (McDonald 1999). A proposal has also been presented to extend these algorithms to enhance the products reliability and apply final testing procedures phases. Braha and Shmilovici (2002) presented three data classification techniques (neural network, decision tree and composite classifier) for wafer cleaning process. The objective of the new classifier named, the advanced wafer cleaning algorithm, was to make the cleaning process more understandable by classifying the experimental data into pre-defined classes and predict to which the new data belong.

Artificial neural network (ANN) and Naïve Bayesian Classifier (NBC) were employed by Perzyk *et al.* (2005) in steel casting process as classification tools. The study was carried out on two identical data sets with binary outputs. It was found that, based on comparative analysis; the prediction errors of Naïve Bayesian Classifier were lower than those of the ANNs.

An extensive stream of work is dedicated to encounter class imbalance learning problems. The performance of traditional classification algorithms is limited when applied to highly imbalanced datasets. The classification problems of imbalanced datasets have extensively been tackled via SVMs and FSVMs algorithms. In real life systems, as the size of datasets increase, knowledge extraction, exploration, analysis and control gets more complicated and resource consuming (Choudhary *et al.* 2008). These techniques have proven to be potentially effective, fast and achieve good generalization capabilities (Vapnik 1995); (Bishop 2006); (Roger and Girolami 2011). Another perhaps a more distinct feature of these tools is their ability to handle the curse of dimensionality in large scale manufacturing systems with large size of datasets. However, these tools are relatively new to researchers from academia and industry alike whose area of expertise is not related to artificial intelligence and computer science (Choudhary *et al*. 2008).

Rojas and Nandi (2006) proposed an SVM based framework to classify and detect bearing-faults of rolling elements. The classifier was fast with a successful classification performance of 95%.

Fung and Mangasarian (2005) proposed proximal support vector machine algorithm (PSVM) for generating linear and nonlinear classifier by assigning the data points to either a positive or a negative class as well as ensuring margins maximization. PSVMs comprise equality instead of in inequality constraints by allocating data points to either positive or negative classes. In contrast to the standard SVMs that require a costly computational time to find a solution to a quadratic or a linear program, PSVM requires nothing more than solving a non-singular system of linear equations. The proposed algorithm is fast and can easily handle large scale data sets, as shown by the classification of around two million points, ten-attributes in a round 20.8 seconds. However, PSVMs have their potential disadvantage as they have a sensitivity to outliers and noise because of over-fitting (Jing 2005).

Jing (2005) presented a robust proximal support vector machine algorithm (RPSVM) to deal with the task of over-fitting (noise and outliers dominance minimization) in comparison to PSVMs algorithms for two class classification. The main concept of the proposed algorithm RPSVMs is to change the training set to a fuzzy set by assigning each data point to a membership value based on its relative importance in the data set. The experimental results of the proposed algorithm show that it can reduce over-fitting. However, the generalization ability may be enhanced further by selecting a good methodology to computing the membership values.

Nguyen and Ho (2005) proposed a new method to minimize the complexity of support vectors obtained in their solution. The key feature of the proposed algorithm is the selection of the closest support vectors corresponding to the same class and exchanging them by new constructed vectors. The bottom-up technique simplifies the support vector solutions in which the newly constructive vectors only need to find a single maximum point of a one variable function on the interval (0, 1). The experimental results proved that the approach is computationally simple compared to other reduced set methods. However, it cannot handle dimensionality.

Dong *et al.* (2005) proposed an algorithm to solve the problem of training support vector machine on huge size databases with thousands of classes. In this approach, a two-step procedure (parallel optimization, sequential optimization) is tailored to train the support vector machines. In the parallel optimization step, the key idea is to quickly eliminate most of the non-support vectors by employing block diagonal matrices to approximate the original matrix. This step will divide the original problem into hundreds of sub-problems which can be solved more efficiently. Consequently, the training time for the sequential optimization can be reduced. In the sequential optimization, the idea is to speed up the training process by integrating some functional strategies such as kernel cashing and efficient calculation of kernel matrix. Experiments on huge size databases showed that the algorithm has a high training speed and a good generalization performance. However, the proposed algorithms have not taken in to account imbalanced and/or corrupted data with noise or outliers.

Geebelen *et al.* (2012) proposed a new algorithm called Smoothed Separable Case Approximation (SSCA) to reduce the number of SVM. The proposed approach upper bounds the weight vector of the pruned solution which reduces the number of support vectors. Results showed that the upper bounding the weight vector is crucial which leads to numerical stability and avoids over-fitting during the approximation phase.

Hwang *et al.* (2011) proposed a new weighted approach based on Lagrangian Support Vector Machines (LSVM) to tackle the problem of imbalanced data classification. The main concept of the proposed algorithm is to assign an appropriate weight to data classes. The majority class has to receive lower weight than the minority class. Most importantly, the weight should satisfy $w_i \in [0,1]$ so that a convergence can be achieved when training the weighted lagrangian support vector machines (WLSVMs). WLSVM has some better advantages in contrast to standard support vector machines as it can be learned iteratively which can make the training phase faster than using quadratic programming for training. However, WLSVM is slower than SVM when using the sequential minimal optimization (SMO) based scheme proposed by (Platt 1998).

Lin and Wang (2002) propose a type of fuzzy support vector machines algorithm (FSVMs). The main idea of the algorithm is to apply a fuzzy membership to each data input of support vector machine (SVM) and reformulate SVM in to FSVM. An appropriate fuzzy membership is chosen in a way such that the lower bound of fuzzy memberships must be defined; then, the property of the main data is to be selected and finally make connections between fuzzy memberships and this property. However, this method is likely to yield good results if the distributions of the given training data of each class are well spread around the central means. The formulation of the algorithm is not complete where the linear separable cases cannot be discussed. However, comparative experimental results against standard SVM on real data set are not provided (Tao and Wang 2004).

Tao and Wang (2004) proposed a new fuzzy support vector machine algorithm based on the weighted margin. The basic idea is to employ the fuzzy membership function to weight the margin. This approach incorporates the idea from SVM and fuzzy neural networks for a better classifier performance. The influence of data inputs can be either reduced or avoided by applying the fuzzy membership for each training vector to weigh the margin. The advantage of modifying SVM by the idea of fuzzy neural system is to apply some fuzzy membership functions.

Consequently, experiments on real data sets illustrate that NFSVM can yield robust outcomes in contrast to standard SVM.

Xiong *et al.* (2005) proposed a new algorithm of fuzzy support vector machines based on fuzzy c-means clustering. The algorithm is based on the idea of applying fuzzy c-means clustering scheme to each class of data set. The key feature is that during clustering with a suitable fuzziness parameter, the algorithm will get rid of the data that are less important and choose the important samples such as support vectors to represent the other similar samples that are close to the cluster centres. Experimental results of the proposed fuzzy support vector machines showed that less quadratic programing time is needed compared with conventional SVMs.

Batuwita and Palade (2010) proposed an approach to improve FSVMs for class imbalance learning CIL (called FSVMs-CIL). This method is used to handle the class imbalance dilemma for the task at hand in the presence of noise and outliers. The basic idea is to assign fuzzy-membership values for the training examples based on their importance in order to reduce the effect of the above concerns. This approach is evaluated on ten real world data sets, containing around ten thousand records. Experiments show that the proposed algorithm is effective and outperform other existing internal and external imbalanced learning methods. However, experiments were limited to small data sets and the authors have not proven the robustness of their algorithm on large scale data set where a much larger optimization problem is required.

Based on the previous literature review sections, there are many application areas in the manufacturing industries where machine learning and data mining tools are used for classification include quality improvement, control, fault diagnosis and condition monitoring. Support vector machines, fuzzy support vector machines, decision tree, hybrid neural networks and other hybrid methods are well known to accomplish classification tasks. Fuzzy logic is usually incorporated within these techniques to deal with the uncertainty and noise that exist in the data. Clustering which is the primary unsupervised learning tool will be discussed in the next section.

2.4.2 Clustering

Clustering analysis is a fundamental task of data mining and is defined as assigning a set of units or observers into a certain number of groups or subsets called clusters so that the observers in one cluster are similar to each other than they are to the observers in other clusters (Hair *et al.* 1987). Clustering is an unsupervised learning technique which is tailored to explore and extract hidden structures and similarities in datasets based on probability density models or similarity metrics (Xu and Wunsch 2005); (Buhmann 1995). Clustering analysis does not use category labels that tag observers with prior identifiers, the absence of information distinguishes cluster analysis (unsupervised learning) from discriminant analysis (supervised learning) (Jain 2010).

Cluster analysis has played an important role in wide range of fields. All clustering methods have their own advantages and drawbacks, and are applicable to various data structure (Suen 2000). Almost all clustering techniques involve a process of measurements, either the magnitude of the distance between two observers (i.e. sum of squared distance criterion) or the magnitude of their similarity to each other (Helmuth 1980).

Practically, clustering algorithms are divided into two groups: hierarchal and partitional (Jain 2010). Hierarchical clustering algorithms iteratively seek to produce nested clusters either in agglomerative way by producing each data point in its own cluster and gathering the most similar pairs or in divisive mode by starting with all given data points in one cluster recursively and dividing each cluster into sub clusters (Jain 2010).

In contrast to hierarchical clustering, partitioned clustering classifies the data into K groups, which together satisfy the requirements of a partition (Kaufman and Rousseeuw 1990):

- Each group must contain at least 1 observer.
- Each observer must belong to exactly one group.

The most widely used algorithm for partitioning is the K-means algorithm. Since partitioned algorithms are remarkably more applicable in pattern recognition and clustering analysis (Jain 2010), two K-means algorithms are to be presented in the next section; K-means and Fuzzy K-means algorithms.

2.4.2.1 K-means Clustering

K-means is said to be a proto-type clustering technique. It is a method that is commonly designed to automatically partition N-dimensional data set into K groups or clusters (Wagstaff *et al.* 2001). Although k-means algorithm was first proposed over 5 decades ago, it is still considered as one of the effective tools for clustering analysis as it is efficient, simple and easy to implement (Jain 2010). K-means algorithm is based on the principle of maximizing the similarity between observers within a cluster and minimizing the similarity between observers in different clusters (Dehariya *et al.* 2010).

Practically, iterative k-means procedure for a given N-dimensional data set is summarized as follows (Jain 2010); (Wan *et al.* 1988):

1) Initialize the centre of the clusters;
2) Assign each data point to its closest cluster centre;
3) Set the position of each cluster to the mean of all data points related to that cluster;
4) Compute a new cluster centre;
5) Repeat steps 2, 3and 4 until convergence.

The most commonly used measure of similarity at this stage between two observers is the Euclidean distance which is basically the measurement of a straight line drown between these two observers (Hair *et al.* 1987). The above steps will be repeated until convergence when there is no further change in assignment of instances to clusters and the new cluster centre is the same as the previous cluster centre (Wan *et al.* 1988).

When dealing with K-means algorithm, three parameters are to be taken into account by the researcher: the number of clusters K, the cluster initialisation step, and the distance metric where the most critical choice is K (Jain 2010).

2.4.2.2 Fuzzy C-means Clustering

In fuzzy cluster analysis many algorithms have been developed. Generally, the most widely used is the fuzzy C-means algorithm conceived by Dunn and generalized by Bezdek (Zahid *et al.* 2001). The fuzzy C-means (FCMs) algorithm has been applied to a wide variety of engineering and scientific disciplines such as pattern recognition, data mining; this algorithm is an objective function optimization approach which uses the squared-norm to measure similarity between prototype and data points (Zhang and Chen 2003).

The fuzzy C-means method offer an important insight into the data by producing a degree of membership to individual data vectors within clusters. Fundamentally, this algorithm seeks a minimum of heuristic global of cost function (Duda *et al*. 2001), where each point has a degree of belonging to clusters rather than belonging completely to just one cluster (Dehariya *et al*. 2010). Thus, points on the edge of a cluster, maybe in the cluster to a lesser degree than points in the centre of a cluster (Dehariya *et al*. 2010). The fuzzy C means algorithm is summarized as follows: (Dehariya *et al*. 2010):

1) Choose a number of clusters;
2) Randomly assign to each point coefficients for being in the cluster;
3) Repeat until the algorithm has converged (that is, the coefficients' change between two iterations is no more than the given sensitivity threshold).

Wang *et al.* (2006) proposed a machine learning based framework for classification and fault detection. An integrated strategy based on hieratical clustering and K means partitioning was employed to classify the defected patterns. Sebzalli and Wang (2001) utilized the fuzzy C-Means clustering (FCMs) approach and Principle component analysis (PCA) to a refinery catalytic process. The aim was to identify bottlenecks and develop operational strategies for desirable products specifications and to minimize product lose during system changeover.

According to Alpaydim (2010), clustering can be used as a dimensionality reduction method. The aim was to perform data exploration and understanding overall data structure and find correlations between variables. It has been reported by Alpaydim (2010) that clustering can be used as a pre-processing stage similar to dimensionality reduction approaches. New research has been presented by Ordieres Meré *et al.* (2004) in designing an integrated strategy between clustering and neural network. The model has been able to successfully predict the structure properties of galvanised steel by implement a clustering scheme in the first instance and then apply networks.

Fuzzy clustering based approaches have also succeeded in detecting welding flows from radiographic images. Liao *et al.* (1999) developed a methodology to process each welding image based on two clustering paradigms i.e., fuzzy C means clustering FCMs and fuzzy K nearest neighbour. A performance comparison was performed between these two clustering methods.

An outstanding characteristic of Clustering is that it can also be incorporated with supervised learning techniques, i.e., an optimization process. Xia *et al.* (2005) proposed an approach which exploits the advantage of K-means clustering algorithm to minimize the number of support vectors (SVs) for the training of (SVM). The basic idea is to apply K-means clustering to select a set that represent the structures of whole data set at the same time with fewer data points based on the observation that the large amount of original dataset is redundant for training SVM. The advantage of the algorithm lies in the fact that the smaller the input dataset is, the fewer SVs will be produced which will lead to less computational time and memory.

Xiong *et al.* (2005) presented a new algorithm of fuzzy support vector machines based on fuzzy c-means clustering. The algorithm is based on the idea of applying fuzzy c-means clustering scheme to each class of data set. The key feature is that during clustering with a suitable fuzziness parameter, the algorithm will eliminate the data that are less important and choose the important samples such as the support vectors to represent the other similar samples that are close to the cluster centres. Experimental results of the proposed fuzzy support vector machines show that less quadratic programing time is needed compared with conventional SVMs.

Another commonly used approach to the problem of unsupervised learning is the bayes classifier. This classifier is different to the maximum likelihood methods as they view the parameter vector as quantities whose values are fixed but unknown where, as in the bayes classifier, one assumes that the parameter vector is random variables with unknown prior distribution (Duda *et al.* 2001). Practically, such a classifier will lead to poor result when dealing with unlabelled samples unless some explicit statements about the experimental samples are to be assumed (Duda *et al.* 2001), these statements are as follows:

1) Prior probabilities for each observer are known.
2) The class-conditional probabilities are also known.

3) The values of parameter vector are known.
4) Date category labels are unknown.

Various efforts from an application view point are closely related to the cause of defects and faults of different types of manufacturing systems or processes and therefore data mining and knowledge discovery inevitably will lead to better operations of the manufacturing enterprise (Harding *et al.* 2006). An increasing awareness has been indicated by recent trends as more and more people are utilizing data mining and machine learning to overcome problems in manufacturing. The subjects reviewed in this chapter have mainly emphasized on providing a general concept of machine learning and data mining algorithms and their applications in various industrial areas. A more broadened literature review relating to the novel approaches we have tailored to tackle the problem of applying machine learning and data mining techniques on rails manufacturing process will be emphasized in the core Chapters 5 and 6 respectively.

Exploring and organizing data into appropriate grouping should significantly prevail in many engineering and scientific fields. Although many algorithms have been proposed for clustering analysis, the fuzzy C means algorithm has a rich and diverse history as it was independently discovered in many various field and is still one of the well-known algorithms for clustering because it is easy to implement, simple, efficient, and has empirical success (Jain 2010). In contrast to K means algorithm, fuzzy C means algorithm has one drawback is that the probability of membership of a point in a cluster depends explicitly on the number of clusters and when this number is specified incorrectly, serious problems may arise (Duda *et al.* 2001).

One of the most important problems in data analysis is cluster validation where the researcher needs to test solutions of data mining problems for robustness (Buhmann 1995). It is important to note that the data representation issue predetermines what kind of cluster structures can be discovered in the data (Buhmann 1995). An additional issue relates to the adequate selecting of the algorithm and correctly choosing the initial set of clusters (Fung 2001). The size of the data set is also an important factor, because most of the clustering algorithms require multiple data scans to achieve convergence (Fung 2001).

Data preparation tasks and data quality, particularly for rails manufacturing data, have not been discussed yet. Potential efforts relating to data preparation process are needed for a better overall modelling and classification performance. Data cleaning also requires a more generic process for complex databases to enable a successful accuracy of data mining in manufacturing industry. Data cleaning and data pre-processing, including a solid variable selection framework for the rail manufacturing process, will be discussed in details in Chapter 4.

Mining from imbalanced domains is indeed a very challenging problem from both performance and algorithmic prospectives. In the development of a classification model, choosing the objective function and the class distribution incorrectly can hinder the performance of standard classifiers and modelling algorithms. The classification would always be biased in favour of the dominating class (majority), while the data related to the minority class tend to be misclassified. Such a concern can be overcome via data resampling techniques (Akbani *et al.* 2004; Batuwita and Palade 2010 ; Estabrooks *et al.* 2004) to lead to balanced data. The problem of class imbalance learning and the most influential methods to overcome this concern will be presented next,

2.5 Class Imbalance Learning

Nowadays, data are being collected at a dramatic pace across a wide variety of manufacturing processes and in many application areas. Such data are complex and high dimensional where one has to deal with significantly large number of attributes. The curse of dimensionality creates problems to machine learning community. Another, perhaps more well-known and potential challenge which has come into light recently is the class imbalance problem. Imbalanced data correspond to data sets where there are many more examples of one class than the other class.

As described in Chapter 1, the rails through process data provided by Tata Steel Europe include a large amount of data records and data variables. Such data are often very difficult to model due to class imbalance problem and to its high dimensional nature. Class imbalance learning has a significant impact on the overall performance of models and classifiers.

Data mining and machine learning techniques are known to have weaknesses when applied to imbalanced data sets where they tend to be overwhelmed by the majority class and lead to a poor classification and modelling performance on the minority class. The majority of concept learning tools are beforehand designed with the assumption that the training sets are well-balanced. For many domains, particularly complex manufacturing systems, this is not usually the case in which one class is represented by a few numbers of examples and the other class is represented by a relatively larger number.

Very few examples in a minority class cannot provide sufficient information and result in great performance degradation (Wang and Yao 2009). There has been a great attention from machine learning and data mining community given to overcome the class distribution problem and thus various solutions were proposed at the data level and algorithmic level (Chawla *et al.* 2002; Chawla *et al.* 2004; Estabrooks *et al.* 2004; Liu *et al.* 2009; Wang and Yao 2009). These approaches can be categorized into two sets: internal techniques and external techniques (Estabrooks *et al.* 2004).

The internal techniques, algorithmic level strategy, are designed to modify the structure of the algorithms or build new ones in a way that the algorithm pays more attention to the minority class. The second general approach to solving the class imbalance problems is an external approach, data level strategy, which resamples only the data under study without modifying the algorithm (Estabrooks *et al.* 2004).

However, it is unclear whether the internal approaches are more effective than the external approaches (Estabrooks *et al.* 2004). The internal approaches have the disadvantage of being algorithm specific. In machine learning and data mining, this is a critical issue because data sets presenting various characteristics are better classified by different algorithms and it is quite difficult for the class imbalance problem to transport the modification from one model to another (Estabrooks *et al.* 2004). It is worth noting that external approaches are independent and can straightforwardly re-balance the training data before training the model. External approaches also have the property that allows the user to set up the desired ratio between the majority and minority class. The external sampling strategies are the best choice of changing the class distribution of the rail manufacturing process data. These types and their way of implementation together with the data classifiers will be explained in detail in Chapters 5 and 6.

For dealing with the class imbalance in a classification problems, *(*Estabrooks *et al.* 2004) combined the over-sampling with under-sampling techniques. The proposed technique proved to be effective. The bias applied was regulated carefully via an elimination strategy so to prevent unreliable classifier to contribute in the text classification process.

Japkowicz (2000) presented a study on the effect of imbalance problem in dataset. Three strategies have been evaluated: Random under-sampling, focused resampling and a recognition-based induction scheme. A simple artificial 1D data was examined. Both under-sampling and the focused resampling were effective in contrast to the recognition based induction scheme.

Chawla *et al.* (2002) proposed a synthetic minority oversampling technique (SMOTE). SMOTE is able to generate a new synthetic minority examples via combining minority examples that lie together. The oversampling technique presented is sophisticated (Estabrooks *et al*. 2004), but the authors did not consider different class distribution ratios. Several sampling methodologies with different class distributions were evaluated by (Batista *et al.* 2004). Different data over-resampling and under-sampling techniques included SMOTE, TOMEK and SMOTE+ENN were examined. SMOTE resulted in a good performance for databases with a small number of majority class examples.

Wang and Yao (2009) extended SMOTE presented by (Chawla *et al.* 2002) into a novel paradigm called (SMOTEBagging) for solving multi class dataset in ensemble model. This approach is internal where only the algorithm has been modified to solve the class imbalance problem. A comparison framework has been also designed to compare two models OverBagging and SMOTEBagging. SMOTE proved diversity in systems resampling and improve its overall performance. Li (2007) discussed the bagging ensemble variation (BEV) system for imbalanced data classification. The algorithm effectively proved to utilize the minority class data without generating synthetic data and make any amendments to classification system.

Although many resampling strategies were exist, oversampling and under-sampling methodologies have received remarkable attention as external methods to counter the class imbalance problem (Batista *et al.* 2004), (Chawla *et al*. 2004), (Chawla *et al.* 2002). A brief overview on the most applicable external sampling strategies in addition to the cost sensitive learning will be presented in the next sections.

2.5.1 Under-Sampling

The mechanism for under-sampling seeks to reduce the number of majority class examples in the training set. Under-sampling is an external independent sampling method that can straightforwardly re-balance the training data before training the classifier. Therefore, it can be employed with any classification and modelling algorithm. In random under-sampling, the training dataset is rebalanced by randomly removing majority class examples until a desired class ratio between the majority and minority class is achieved.

Despite its simplicity, the under-sampling approach proved to be significantly efficient and can be easily implemented. However, under-sampling approach has been reported in Hwang *et al.* (2011), Akbani *et al.* (2004), Liu *et al.* (2006) and Chawla *et al.* (2002) to throw away potentially useful data and some crucial information might be lost. It thus, could dramatically disturb the decision boundary of the classifier in the classification problem.

2.5.2 Over-Sampling

Oversampling is another external (data level) sampling technique that can also be applied to change the class distribution of databases. In random over-sampling, the majority class examples in the training dataset are randomly duplicated until a desired ratio between the majority and minority class is achieved. The oversampling technique has gained extra attention. The advantage of a such a technique is that it is external and therefore, easily transportable as well as very simple to implement (Estabrooks *et al*. 2004; Garcia and He 2009; Yang et al. 2011 a).

Another advantage of this technique is that no information is lost since all instances are employed (Yang *et al.* 2011 a). However, by creating additional examples, oversampling leads to a high computational cost. Thus, a sufficient amount of memory is required to hold the whole training set. Moreover, randomly replicating the data might contain erroneous values which could negatively impact learning performance (Chawla *et al.* 2002).

2.5.3 Cost Sensitive Learning

Highly imbalanced problems have generally highly non-uniform error costs where the influence of the majority class is dominant and the minority class is only represented by a few examples with less influence to the modelling and classification problem. Internal approaches deal with the modification of learning model in order to make it less sensitive the imbalancing problem. In cost sensitive learning, the main objective is to minimize the total cost of majority class examples rather than modifying the class distribution of the data the way external resampling techniques do (Chawla *et al*. 2004; Garcia and He 2009).

It has been reported that cost sensitive learning leads to better results than random resampling techniques (Japkowicz 2000). However, the main drawback of this technique is its restriction to just few modelling applications and necessity to predefine the misclassification costs which is not usually available in datasets. In data classification problems, various algorithmic approaches have been designed for different modelling and classification algorithms, such as fuzzy systems (Fernández *et al.* 2009), neural networks (Zhou and Liu 2006) and support vector machines (Akbani *et al.* 2004).

Due to the imbalance nature of the rail manufacturing process data, a well-designed data resampling schemes and their influence on the overall rail data classification process will be discussed in details in Chapters 5 and 6. The foundations for most of the current sampling strategies to real world industrial learning problems will also be presented in a broad overview.

2.6 Evaluation Metrics for Class Imbalance Learning

Performance metrics of class imbalance problems provide a simple way of examining a classifier's performance on a given dataset. In the machine learning community, there are several performance evaluation metrics which were proposed for the tackling class imbalance learning problem such as area under the curve (AUC) (Bradley 1997), receiver operating characteristic (ROC), F-measure, Geometric-mean (Gm) and confusion matrix. However, these metrics are inherently sensitive to any changes in imbalanced datasets and thus, in certain situations, they can be deceiving (Garcia and He 2009). G-mean and F-measure are functions of confusion matrix (Liu *et al.* 2009).

In the presence of unequal error costs in imbalanced datasets, ROC curve (Garcia and He 2009); (Estabrooks *et al.* 2004); (Liu *et al*. 2013) represents an appropriate method for performance measures. Whereas, In the case of learning from imbalanced datasets where equal error costs apply, error rate is a functional performance measure criteria (Chawla *et al.* 2002); (Liu *et al.* 2009). AUC is also a reliable performance metric for cost sensitive learning problems (Liu *et al*. 2009).

In data classification domains, the confusion matrix is a powerful metric by which the performance of a classifier can be assessed (Prajapati and Patle 2010); (Tang *et al.* 2009); (Hwang *et al.* 2011). However, a better understanding of classifier's performance can be achieved via the performance of misclassification especially in the case of imbalanced data (Veropoulos *et al.* 1999). The structure of the given data and the modelling framework type are the major factors on which performance metric is to be chosen. A detailed overview on the performance metrics used in rail data classifications and their success rate are to be discussed in Chapters 5 and 6.

2.7 Summary

This chapter has presented a literature review and discussed briefly various aspects relevant to large complex systems modelling. First, the description of data driven modelling approaches is provided to understand the process of knowledge discovery in databases and data mining in general. The most influential machine learning techniques i.e., data classification and data clustering are explored from a variety of application disciplines which is the main focus of research. The advantages and disadvantages of these techniques on real world applications are discussed. An overview highlighting the similarities between several disciplines and common connections to data classification are also identified.

Subsequently, class imbalance learning problem is investigated both in class distribution and costs. Existing contributions and research pertinence to the problem of rail data modelling and classification are also presented. The most influential data level and algorithm level sampling strategies i.e., over-sampling, under-sampling and cost sensitive learning techniques are briefly illustrated.

In the next chapter, a detailed description of the rail manufacturing process as provided by Tata Steel Europe and Key Production Stages will be presented. The chapter will also include the recent developments and contributions in research for a successful modelling and classification of the rail process. Consequently, the next chapter will also outline the challenges of rail production data i.e., data complexity and curse of dimensionality.

Chapter 3

Rails Though Manufacturing Process

3.1 Introduction

Real world industrial processes are complex entities whose performance in terms of expected quality of delivered products and cost management are of a strategic importance. The rails manufacturing process is one of these complex systems which includes many wide-ranging characteristics and integrated sub-processes. For systems modelling, the reliability and performance issues of these complicated manufacturing processes have recently attracted considerable attention. In this chapter, a general overview of the rails manufacturing process operated by Tata Steel Europe is presented. In addition, the key stages of steel production line are also briefly described. Next, rails data key challenges and opportunities that have significant impact towards a successful data modelling and classification implementation are also addressed. The aim of this chapter is also to present and assess the rails production line as well as the rails data accumulated from this process. In practice, rails quality can either be ‘‘good’’ or ‘‘bad’’ based on Non-Destructive Testing (NDT) in addition to the knowledge which may be provided by human expert.

3.2 An Overview of Rails Manufacturing Line

The future competitiveness of the rails industry depends significantly on its ability to tailor the produced rails to meet desired specifications with respect to the mechanical properties and quality assurance. Recently, there has been worldwide increased effort in the area of rails production towards the enhancement of rails quality with the aid of data modelling techniques which has become a subject of major interest. An initial inspection of any rails production data can indeed reveal that there maybe a few rejected rails. Advanced analytical approaches are needed for such rejections to find the root causes as well as identifying bottlenecks in the production route and thus applying appropriate control measures to improve process yields (reduce defects). Production costs of rails production line may undoubtedly be reduced significantly via improving process yields (reducing defects) and consequently meeting the quality requirements for the customers.

3.2.1 Rails Manufacturing Process

In this section, a brief overview is provided of the process by which the rails data were collected. The rails manufacturing data utilized in this research were gathered from an integrated steel production process which pertains to Tata Steel Europe. The main features of the rails manufacturing process are illustrated in Figure 3.1 (Yang *et al.* 2011 a).

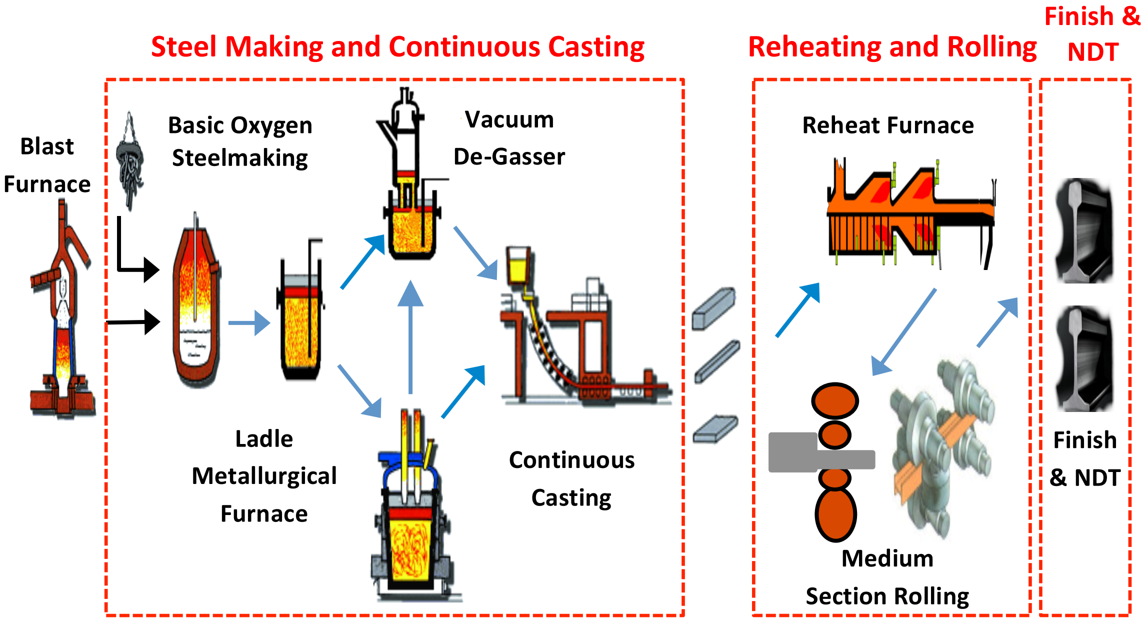


Figure 3.1: Tata Steel Rails manufacturing process Route (Yang *et al.* 2011a)

3.2.2 Rails Production Route: Key Production Stages

The word ‘process’ used in this thesis refers to several interacting sub-processes, each of these parts fulfils a certain role towards the production of quality rails. The rails manufacturing process can be separated into three sub-systems including: steel making, continuous casting, rolling and finishing. These stages are the three key sub-processes of Tata Steel Europe production line.

3.2.2.1 Steel Making

At the steel making stage, the iron ore as well as additional scrapes are continuously loaded into the top of a big blast furnace where iron oxides are converted into liquid iron denoted as hot metal. The liquid products (molten iron) are drained from the blast furnace to a basic oxygen steelmaking furnace in which carbon-rich molten iron is refined into steel. The basic oxygen steelmaking process or as also called, oxygen converter process, blows oxygen through the molten iron, the key to the process is to reduce carbon content of the alloys and changes it into low-carbon-steel at the same time as separating as many of the other chemical impurities as possible. For further secondary steel making, the liquid steel is passed through a ladle metallurgical furnace in order to adjust the chemical structures of the steel via adding extra alloys. The de-gasser unit will then improve steel cleanness by removing harmful hydrogen and other gases. Figure 3.2 illustrates the structure of steel making stage

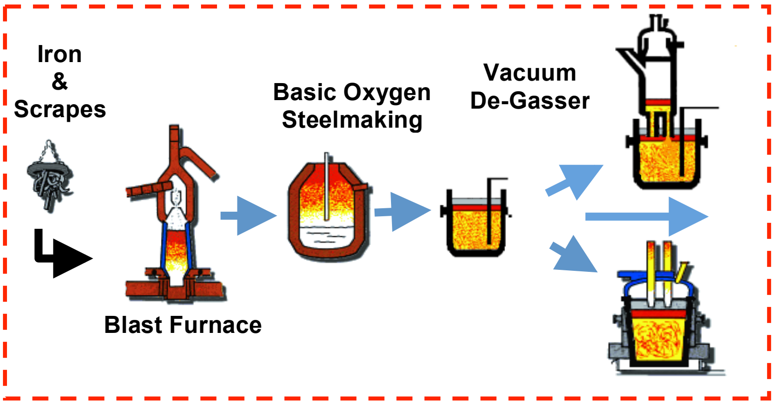


Figure 3.2: Rails Production Steel Making Stage

3.2.2.2 Continuous Casting

In the stage of continuous casting, the rails liquid steel is conveyed via a multi-strand continuous casting machine where 8-tonne steel blooms (Yang *et al.* 2011a). Continuous bloom casting is incorporated with tight control via alloy addition. The blooms are heated in a reheating furnace and then fed directly to straightening operations and multi-pass rolling mills at a proper temperature in order to yield rails with a maximum length of 120 meters. The internal integrity of the produced blooms is preserved by eliminating oxide inclusion. Recently, this stage has received an extra attention throughout the investments in dynamic spray control and mould electromagnetic stirring to ensure cast bloom consistency with minimal segregation and to prevent the internal cracks and flows. A simple continuous casting stage is shown in Figure 3.3.

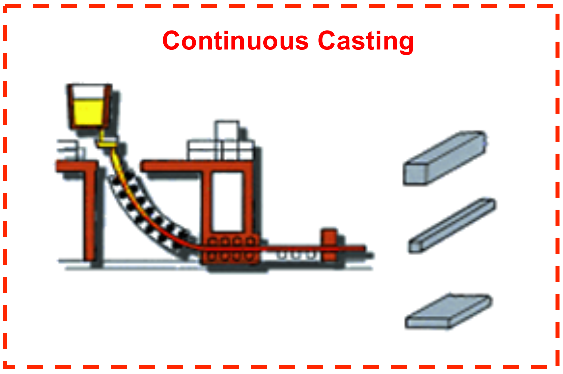


Figure 3.3: Rails Production Continuous Casting stage

3.2.2.3 Rolling and Finishing

One of the main rails forming processes is hot rolling stage. At this stage, the blooms pass through the rolling stands after being reheated in the furnace as shown in Figure 3.4. The blooms cross-section is reduced and reshaped in each pass until the desired rails shape is achieved. The final stage of the rails manufacturing process involves a preventative measure against cracks and flaws and evaluating the properties of every rails via a complete NDT. The aim of NDT is to ensure that rails meet applicable standards and quality control specifications, as well as for dimensional accuracy, before dispatch to clients (Yang *et al.* 2011 a).

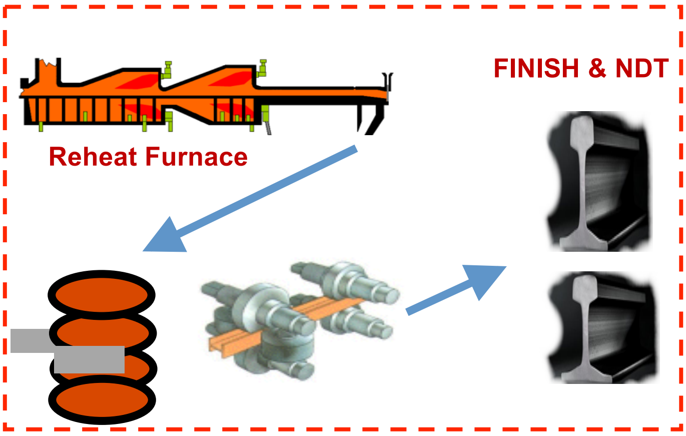


Figure 3.4: Rails Production Rolling and finishing stage

Rails data accumulated from Tata Steel Rails production line have only a single output consisting of integer values of (0, 1, 2 and 3) where 0 represents good rails and values (1, 2 and 3) represent defected rails with progressive degrees. This research focuses on the small number of rejected rails, which are distinguished from the good rails as they have cracks and/or internal flows, by building analytical approaches to discover the root causes of such defects and therefore tailor an adequate optimization and control measures to enhance product quality and reduce manufacturing costs.

3.2.3 A Review of the Rails Through-Process Data

Industrial processing plants are usually heavily equipped with a large number of instruments such as sensors and transmitters. The main purpose of these instruments is to deliver meaningful data for process monitoring and quality control. Researchers from academia and industry started to devote increasing efforts to the volume of data being measured and stored in the process industry by constructing predictive models based on this data. The data collected from rails manufacturing process are the culmination of more than two years of a production period. Online data servers are utilized to collect real-time variables, process parameters, quality inspection data and management information from the rails production route via extensive instrumentations allocated for online monitoring and process control. An overview of the rails through-process data gathered from Tata Steel online server is presented in Figure 3.5.

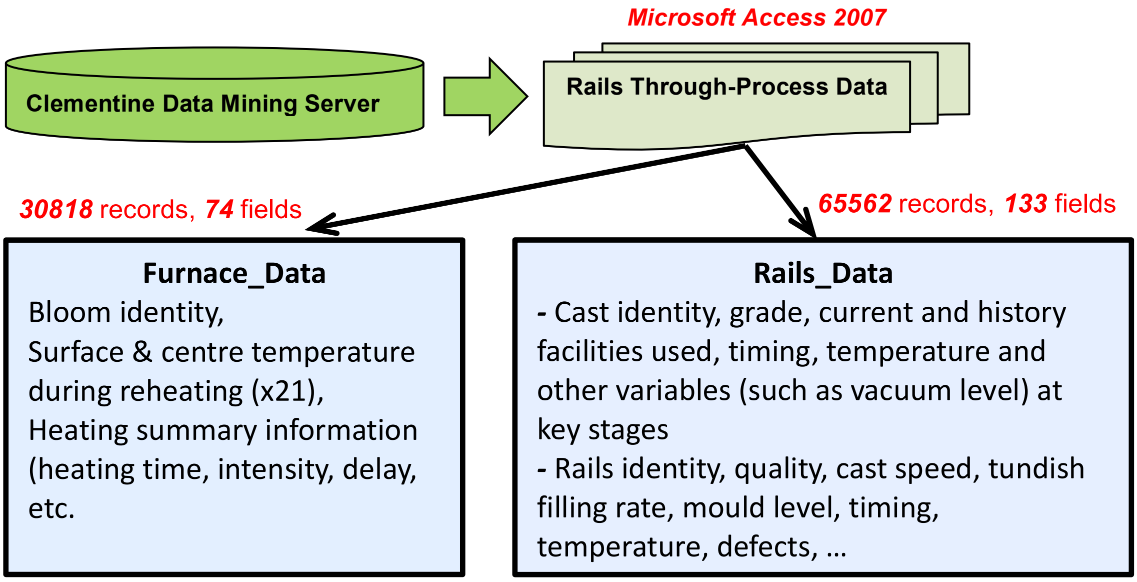

Microsoft Access 2007
Clementine Data Mining Server
Rails Through-Process Data
30818 records, 74 fields
65562 records, 133 fields
Furnace_Data
Bloom identity,
Surface & centre temperature during reheating (x21),
Heating summary information (heating time, intensity, delay, etc.
Rails_Data
- Cast identity, grade, current and history facilities used, timing, temperature and other variables (such as vacuum level) at key stages
- Rails identity, quality, cast speed, tundish filling rate, mould level, timing, temperature, defects, …

Figure 3.5: An Overview of Rails Through-Process Data

For a successful data collection, as shown in Figure 3.6, a solid data infrastructure has been designed to collect the rails data from rails production line. The data accumulated include real time variables, management information, quality inspection data, key unit production time histories and process parameters. A master server is utilised to store the overall data where the completion of advanced level of KDD was performed. The original rails quality data gathered from the key production stages are very large, with over 200 variables and around 83000 data records. The main concern of this study is the small number of the rejected rails, verified via an automatic and manual ultrasonic testing for the presence of internal irregularities such as cracks and flaws, to find root causes as well as identifying bottlenecks in the production route and thus applying an appropriate control measures to improve process yields (reducing defects).

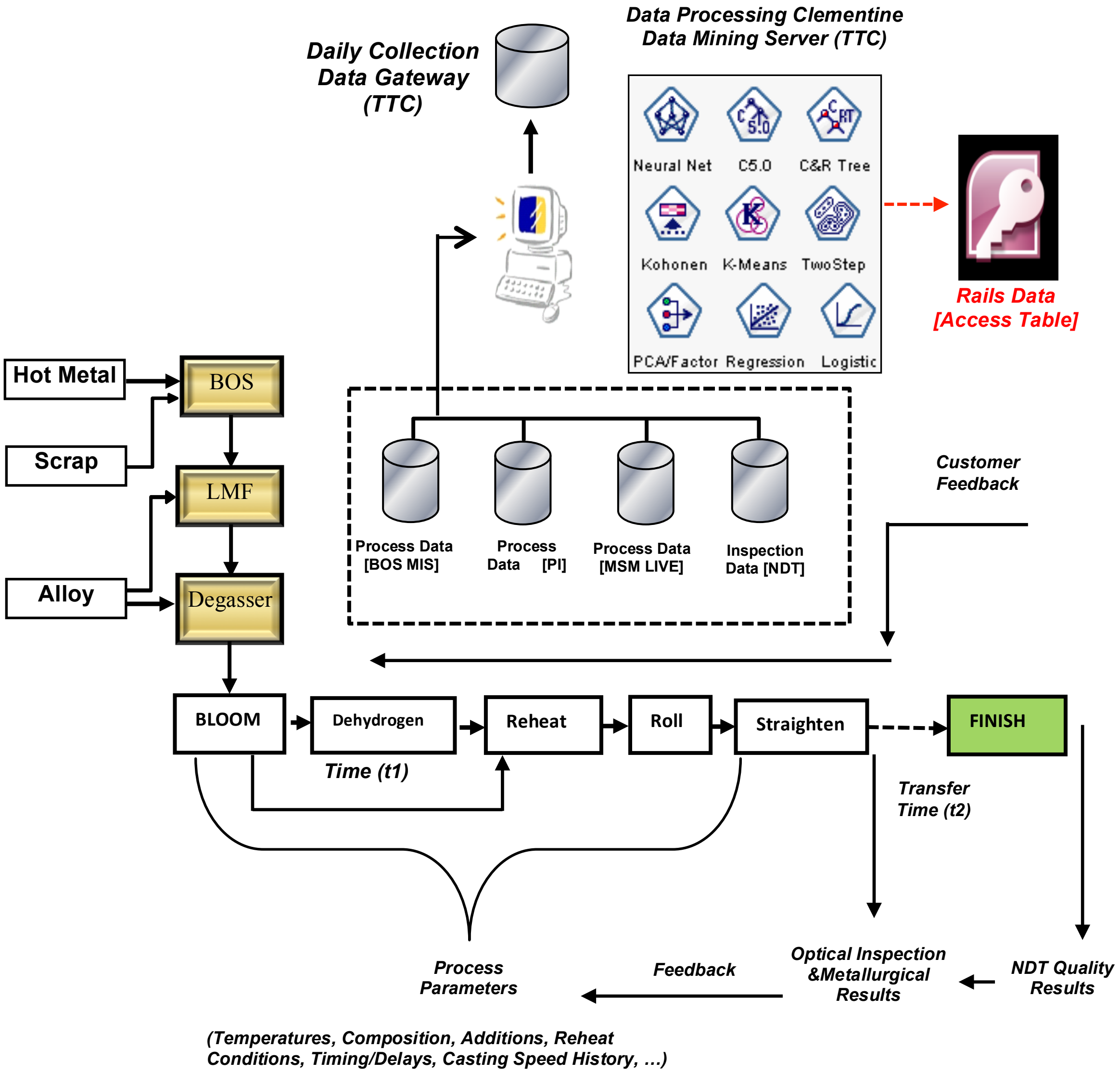


Figure 3.6: Rails Project Data Infrastructure

Despite the advances in rails-making technology, changes in material properties, equipment faults and variations in operating conditions, the product quality could be completely different from the desired specifications. An initial inspection of the Rails through Process Data has revealed some fundamental challenges i.e.,

- Presence of significant noise, conflicting information, lack of understanding among many data variables;
- Redundant information, and outputs (defects) cannot be quantified accurately;
- Large amount of data records and data variables, difficulty of manipulation and visualisation;
- Both the number of data fields and data file size almost violate the limit of Microsoft Access 2007, therefore, cause computational difficulties;

3.3 An Overview of Rails Data Modelling

The overall data collected from the rails manufacturing route presented above is the culmination of more than two-years of manufacturing period. The data collected include more than 65000 data records and over 200 data variables. For rails data modelling and analysis, many potential challenges arise based on preliminary investigations. Basic pre-processing tasks can prove to be computationally expensive and resource-intensive due to the volume and complexity of rails data. Consequently, modelling and optimisation procedures and algorithms need to be tailored with particular care (Yang *et al.* 2011 a).

Simple tasks, copying, pasting and plotting, cannot be executed straightforwardly because they often violate the specifications and norms of most commonly used software. Based on these challenges, the choice of designing an adequate data analysis tool becomes crucial due to the fact that a single computing algorithm cannot successfully provide potential requirements in the rails data modelling cycle. Hence, an integrated software environment was carefully designed to perform rails data modelling as illustrated in Figure 3.7.

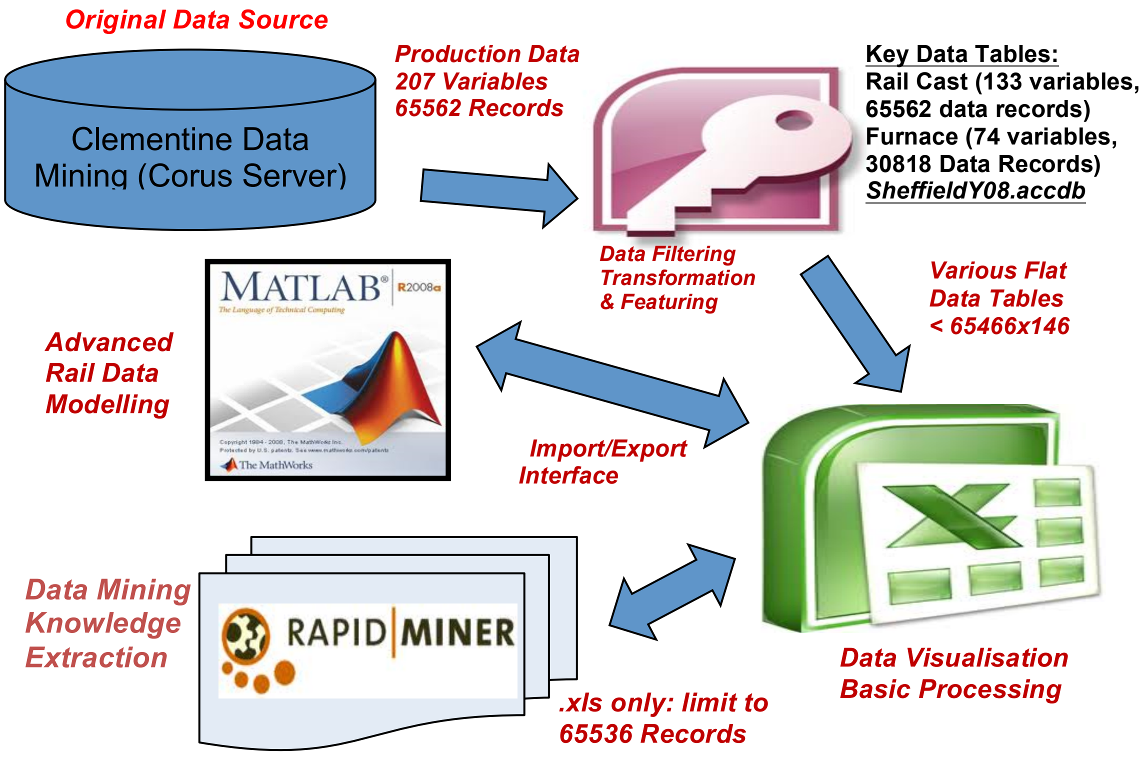


Figure 3.7: An Integrated analysis environment for Rails Data Mining

3.3.1 Current Research On Rails Data Modelling

Industrial process plants are usually equipped with a number of instruments (sensors). The primary objective of these instruments is to deliver data for process analysis and control. It was approximately two decades ago when researchers from academia and industry began to make use of the large amounts of data being measured and stored in online servers by constructing predictive models based on this data. Furthermore, the concept of 'Big Data' began to emerge whereby it refers to data points that transcend disciplines (multi-disciplinary).

With the increasing demands on economic requirements, cost management, systems performance and production quality improvement, industrial manufacturing plants have become more complex (integrated) in terms of both automaton and structure degrees. These issues become the most critical aspects for systems modelling and are receiving increased attention. With the various operational constraints and requirements for Tata Steel rails production line, the design of an appropriate process-based model is of strategic important for product quality improvement.

Due to the fact that most of industrial plants are complex with unforeseen disturbances such as equipment and instrument faults, variations in material properties and changes in plant operating conditions, the final product quality is likely to be far different from specifications. In order to enhance the final product quality, an appropriate modelling strategy which relates batch process conditions and the final quality is inevitably required where the success and performance of real-time quality modelling and control depend on the accuracy and availability of the designed model.

Several data-driven modelling approaches have been developed where online measurements are applied for final product quality prediction. Most of commonly used approaches utilise historical training data for identifying inferential model for quality prediction, this data were collected from previously completed unsuccessful or successful runs.

ANNs have recently been well-known to hot rolling process modelling and control tool (Yang *et al*. 2011 a; Öznergiz *et al*. 2009; Lee and Choi 2004). This is due their quick interpolation ability and the flexible self-learning mechanism. Another ANN application for modelling the rails hot rolling process is presented by Altınkaya *et al.* (2014). In this study, ANN was used to model the production parameters of deferent types of rails in the rails rolling process where the aim was to achieve optimum parameter values of various types of rails. The ANN model presented is reliable and effective and provides useful way for precise decision making.

ANN and Naïve Bayesian classifier (NBC) are employed by Perzyk *et al.* (2005) in steel casting process as classification tools. The study was carried out on two identical data sets with binary outputs. It was found, based on comparative analysis, that the prediction errors of Naïve Bayesian Classifier are lower than that of the ANNs.

A real time visual inspection system (VIS) has been proposed by Li and Ren (2012) for the detection of rails discrete surface defects. An image acquisition system is applied to acquire a rails image, and then, a track extraction algorithm is used to cut a sub-image of rails track. Having enhanced the contrast of the rails image via a normalization method, VIS approach detects the defects using the defect localization methodology. Even though this technique is robust to noise, it still involves human image inspection and thus, it is unreliable and time consuming.

A robust SVM classification framework was designed by Rojas and Nandi (2006) to classify and detect bearing-faults of rolling element. The classifier has proved to be fast and a successful classification performance of 95% has been achieved. Ph Papaelias *et al.* (2008) presented a comprehensive study reviewing non-destructive evaluation (NDE) techniques that are in use in North America and Europe for rails defect detection. This study outlined the background theory about the most applicable techniques used to facilitate condition data into productive maintenance procedures.

Although many studies focusing on the identification of possible failures in rails exist, all of which focusing on either rails manufacturing sub-process or a certain part in this sub-process, no attempts of modelling or/and classifying the whole rails manufacturing process including: steel making, Casting , rolling and finishing stages have so far been reported.

3.3.2 Challenges and Opportunities of Rails Manufacturing Data

Making on-specification products is an important endeavour, and also a challenge for the rails manufacturing process. Such a complex process may not produce the desired quality products because of the instability of operating conditions and uncertainty of row materials. Products quality prediction of rails production process would be helpful so that one can make adjustments to process conditions.

Nowadays, the growth of data, both structured and unstructured, has presented a remarkable challenges as well as opportunities for organisations in academia and industry. With growing data volumes, it is essential for such organizations to make use and extract real-time information form the measured data. Meanwhile, there have been competitive efforts that use data to deliver better insights and extract knowledge to decision-makers for fulfilling a better production quality and improve systems performance. In practice, a major challenge for data modelling and classification algorithms is the structure of the data. Typically, the data accumulated for industrial manufacturing process are strongly co-linear with high level of noise and redundancy. Apart from the above stated challenges, these data are most likely to contain hundreds or/and thousands of data records (Kadlec *et al.* 2009).

Learning from rails manufacturing process data is an emerging area with several challenges are to be addressed. The volume of the collected data is significantly large, making the modelling process very challenging. Consequently, only relevant inputs should be used in the rails modelling process. This, however, led to another challenge which is the identification of relevant data variables. Technical difficulties also arise when learning from large scale data sets; as such data are more likely to increase the search space for modelling and classification algorithms in a computationally explosive manner. Moreover, Large Data Modelling algorithms are Hungry for Resources. Therefore, a sufficient amount of memory is required to hold the training set.

Analysing high dimensional data is a crucial task demanding much caution and care. The curse of dimensionality does not revolve only around the inclusion of large number of records in databases, but there can also be a very large number of attributes (variables). Most techniques that aim to overcome the high dimensionality focus on feature selection methods, i.e. choosing an optimum subset of features based on their importance to the overall process. Such variable selection can be achieved manually using human-based knowledge to identify the relevant variables or algorithms-based feature selection. Despite the popularity of feature selection paradigms, several drawbacks on the overall accuracy have been reported in overcoming the course of dimensionality (Rokach and Maimon 2006).

Another, perhaps, more significant challenge towards a successful rails data modelling is to design a rails data pre-processing framework capable of identifying data outliers, redundant variables, correcting wrong data entries, missing value decision and checking data conflicts. Rails process problems are similar to most real world plant problems that can be considered as having randomness, disturbances and complex nonlinear dynamics. This is due to the potential variations in operating conditions and changes in material properties. It is widely known that rails process manufacturing plants are usually high dimensional, large scale, highly uncertain, nonlinear and involve human interactions via qualified engineers to apply empirical knowledge to improve final process products.

Another well-known potential challenge which has come into light recently is the class imbalance problem. Nowadays, Data are being accumulated at a dramatic pace across a wide variety of manufacturing processes in many application areas. Imbalanced data correspond to data sets where there are many more examples of one class than the other class.

Mining from imbalanced domains is indeed a very significant problem from both performance and algorithmic prospective. In the development of a classification model, choosing the objective function and the class distribution incorrectly can hinder the performance of standard classifiers and modelling algorithms. The classification would always be biased in favour of the dominating class (majority), while the data related to the minority class tend to be misclassified. Such a concern can be overcome via data resampling techniques (Batuwita and Palade 2010); (Akbani *et al.* 2004); (Estabrooks *et al.* 2004) to lead to balanced data. Class imbalance learning poses serious impacts on the overall performance of models and classifiers.

Data mining and machine learning techniques are known to have weaknesses when applied to imbalanced data sets where they tend to be overwhelmed by the majority class and lead to a poor classification and modelling performance on the minority class. The majority of concept learning tools are beforehand designed with the assumption that the training sets are well-balanced. For many domains, particularly complex manufacturing systems, this is not usually the case in which one class is represented by few numbers of examples and the other class is represented by a large number.

For rails manufacturing data, data preparation tasks and data quality enhancement have not been discussed yet. Potential efforts relating to data exploratory analysis are needed for a better overall modelling and classification performance. Data cleaning and consolidation also demand a more generic process for the complex rails data to enable a successful performance accuracy. The application of exploratory data analysis techniques will significantly determine the types of other approaches that data analyst can utilize to examine a given dataset. Exploratory data analyses are suitable for both qualitative and quantitative data that include; identifying data outliers, redundant variables, correcting wrong data entries, missing value decision and checking data conflicts. Figure 3.8 outlines the general concept of the exploratory data analysis that will take place in the next chapter to encounter the above mentioned concerns related to Rails manufacturing data.

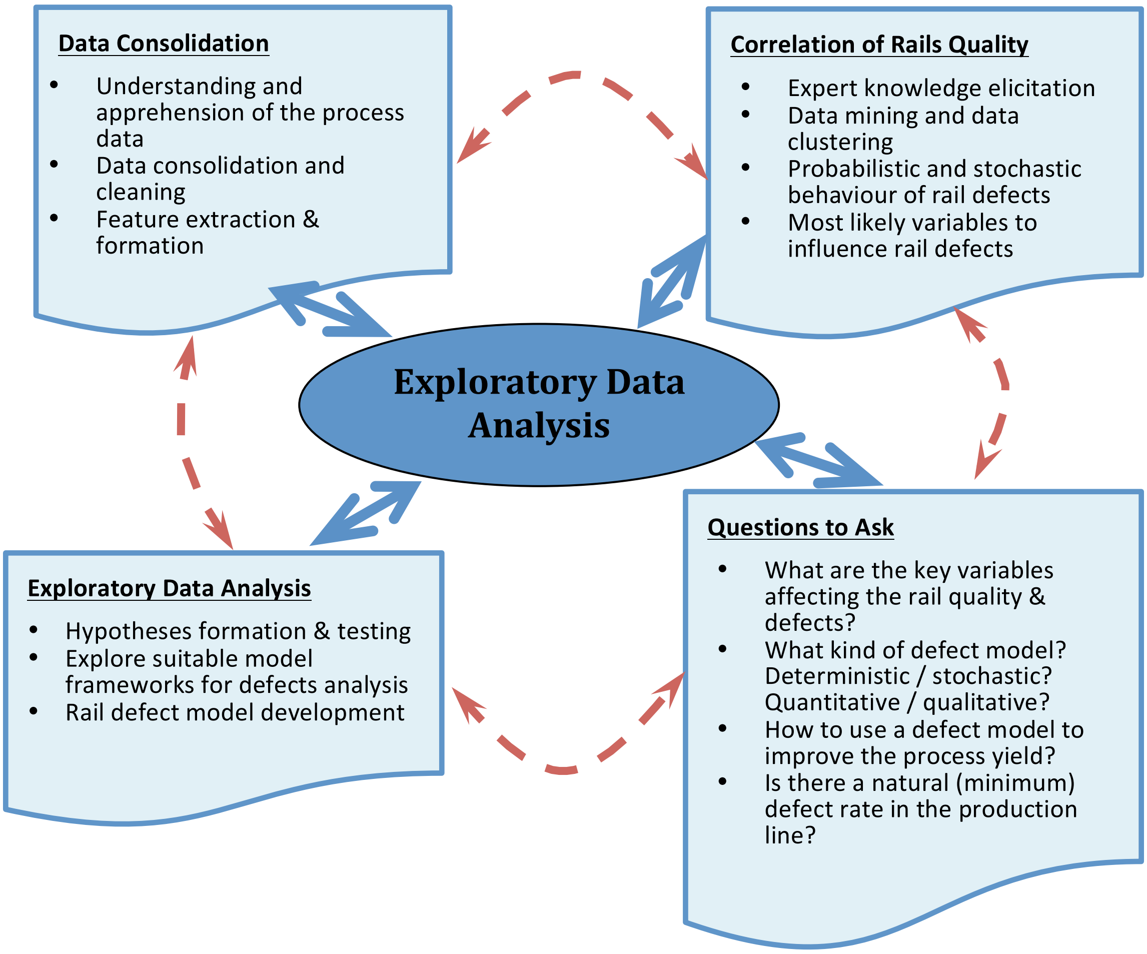

Data Consolidation
Understanding and apprehension of the process data
Data consolidation and cleaning
Feature extraction & formation
Correlation of Rails Quality
Expert knowledge elicitation
Data mining and data clustering
Probabilistic and stochastic behaviour of rail defects
Most likely variables to influence rail defects
Exploratory Data Analysis
Questions to Ask
What are the key variables affecting the rail quality & defects?
What kind of defect model? Deterministic / stochastic? Quantitative / qualitative?
How to use a defect model to improve the process yield?
Is there a natural (minimum) defect rate in the production line?
Exploratory Data Analysis
Hypotheses formation & testing
Explore suitable model frameworks for defects analysis
Rail defect model development

Figure 3.8: Exploratory Data Analysis via Data Mining

3.4 Summary

Mining from large scale-imbalanced datasets is indeed a very important problem from both the algorithmic and performance perspective. Therefore, Special attention has to be paid to the nature and properties of the data in Data modelling and classification implementation. The complexity of models and classifiers depends on the number of inputs dimensions and the size of data samples. These issues determine both the space complexity and the time to train such models.

This chapter presented an overview of rails manufacturing line and key production stages understudy. The rails production project data infrastructure and a brief view of the integrated software environment tailored for Rails Data modelling is also asserted. Rails data key challenges and opportunities that have significant impact towards a successful data modelling and classification implementation are also addressed.

It is clear that the generalization capability of most modelling techniques is highly dependent on the structure and properties of the given data. In the next chapter, particular attention will be given to encounter the challenges in rails production data structure using firm data exploratory analysis and data pre-processing techniques. Data reduction and feature selection framework that form fewer, most relevant inputs will be designed which are capable of choosing an optimum subset of features based on their importance to the overall process. The main characteristics of exploratory data analysis, feature selection and input reductions algorithms will also be discussed. Moreover, most recent research work in these areas will also be presented. Finally, the modelling of the new formed rails manufacturing data using the adaptive neuro fuzzy inference system (ANFIS) approach will also be presented.

Chapter 4

Rails Data Pre-processing, Neural-Fuzzy Based Model Input Selection and Artificial Neural Network Modelling

4.1 Introduction

Exploratory data analyses (EDA) are capable of formation, consolidation and analysis of large amounts of data. Exploratory data analyses have proven to be useful in a variety of fields where large samples are produced and need to be analysed; this volume of data could not be practically visualized and analysed by hand within a reasonable timeframe.

In order to make the data more suitable for data mining, exploratory data analyses are extremely powerful techniques when applied correctly. Practically, they are far different to statistical approaches aiming to examine specific hypotheses. Exploratory data analysis and data pre-processing techniques are quantitative traditions that seek to help researchers clearly understand the data when little or no statistical hypotheses exist.

Based on an initial inspection of rails data, several challenges have been revealed. These challenges are related to process complexity and poor quality data. The Rails production process operated by Tata Steel Europe contains a large amount of data records and data variables. Such data are often very difficult to model due to class imbalance problem and to its high dimensional nature. Class imbalance learning has serious implications on the overall performance of models and classifiers. Another challenging issue for rails data, apart from those stated above, relates to the structure of the data. Typically, the data pertaining to the industrial processes are strongly non-linear. Generally, there are two ways to deal with the data complexity problem. One way is by transforming the input variables into a newly reduced space with less non-linearity, the other way is to select a subset of the data input variables.

Most techniques that are designed to overcome the high dimensionality problem and data complexity focus on variable and feature selection methods, i.e. extracting an optimum subset of variables based on their influence and importance to the process. Such a variable selection can be achieved either manually via human-based knowledge extraction to identify the relevant variables or through an algorithm-based feature selection approach (Rokach and Maimon 2006).

4.2 Exploratory Data Analysis: A Pre-Processing Framework

Data pre-processing or data preparation is a crucial phase for both the machine learning and the knowledge discovery process, as most industrial databases tend to be incomplete and noisy. This is because the data gathering methods are often loosely controlled due to some hardware and/or practical problems (instruments malfunction) which would lead to out-of-range and missing values. For instance, some process variables are not being measured for a few instances, or their real values may be influenced significantly by uncertainty (outliers). Moreover, the data gathered are typically of a large size (usually numerous gigabytes or more) and they are likely collected from heterogeneous sources. In real-life Industrial processes, the true relationship between input variables and process output is hardly understood as process units are more likely to be different because of their nature and therefore they should be normalized or scaled prior to the application of any learning or modelling techniques.

Analysing data that have not been carefully screened for such concerns will not reveal the relationship between variables and the output, and also can yield misleading interpretations and results. Consequently, the representation and quality of the data are important before executing any analysis which will help improve the accuracy and effectiveness of all the subsequent data modelling and mining processes.

The core methods of data pre-processing are data cleaning, data integration, data transformation and feature selection. Data being gathered from complex industrial processes tend to be noisy, containing outlier values that deviate from the expected ones (errors), inconsistent (containing discrepancies between different data items) and incomplete (lacking a certain attribute of interest or containing only aggregate records). If there exist inappropriate and redundant information present or noisy data, then data analysis during training phase is difficult due to the fact that quality decisions must be based on quality data set (Han and Kamber 2006). Many Real world databases are highly vulnerable to the aforementioned commonplace properties.

The rails data at hand were collected from a rails production process covering two years of production period, consisting of 65000 data records and around 200 data variables (Yang *et al.* 2011 a). A preliminary inspection of the rails data revealed that the data contain conflicting information and some redundant variables. Some records also include significant amount of missing values. Missing data, particularly for very large tuples that have missing records for some attributes, could not be used for modelling as their inclusion will distort analysis of the rails data. Many of data variables are related to production management, as a result, they should be removed from the original data set (Yang *et al.* 2011 a).

It is not only the above-mentioned concerns that cause difficulties to the pre-processing, modelling and data analysis stages, but also data dimensionality tends to govern the true input-output results. Moreover, it causes difficulties in data manipulation and visualisation. The aim of data pre-processing stage is generally based on detecting outliers, modifying wrong data entries, identifying missing values and irrelevant variables. Owning to the fact that a careful data preparation is an essential part of exploratory data analysis, a human inspection can sort through the entire database to identify unrelated patterns. A process proficient from Tata Steel Europe was closely involved in pre-processing stage to advise and provide the necessary knowledge about rails process data collection, demonstrate the most important and correlated data items and detect irrelevant variables (Yang *et al.* 2011 a).

After the ‘human inspection’ stage, the original rails datasets were reduced not only in dimension but also in length. Non-relevant data variables were eliminated and data records with many missing values were deleted. The resulting rails data set, after human inspection stage, consists of 39687 records and 140 data variables (Yang *et al.* 2011 a). The data has only one output consisting of integer values of (0, 1, 2 and 3) where 0 represents “good” rails and values (1, 2 and 3) represents defected rails with one or several flaws.

Having pre-processed the data set as above, it still remained very large and the data space dimensionality was very high. In order to achieve any meaningful feature classification the dimensionality of the input data space needed to be reduced even further. As a result, data dimensionality reduction and the selection of the most important variables will be outlined in the next section.

4.3 Model Input Selection and Rails Data Reduction

Variable and feature selection is a crucial step in complex system modelling that has become the focus of research in areas of application where data tend to a relatively large number of variables and the input-output relationship is often not clearly understood. In the modelling process, the selected data have to satisfy certain selection criteria where it is highly important that all process dynamics must be included in the consolidated dataset.

As already stated, the output of rails manufacturing process contains class labels of final products where the input-output relationship is ambiguous and needlessly large numbers of inputs exist. The presences of irrelevant and redundant inputs as well as the high dimensionality space tend to confuse and distort input-output results and lead to poor modelling outcomes. As a result, feature selection should play an important role and be useful as part of the rails data analysis process, due to the fact that it showed which features are important for model elicitation, and how these features maybe related.

Variable and feature selection includes many advantages i.e., enabling data visualization and understanding, training time reduction, tackling effectively curse of dimensionality to improve the prediction performance of the predictors, enhancing generalization by reducing over-fitting (Guyon and Elisseeff 2003). However, in practice, some methods have supplementary emphasis on one aspect over another.

There are many different methods tailored for the purpose of input variable selection, including statistical test or correlation analysis (Miller 2002);(Yu and Liu 2003), principal component analysis (Luo et al. 2008), direct objective optimisation based approaches (Perkins *et al.* 2003). Due to the fact that some methods put more emphasis on one approach than another, input variable selection methods in this research are designed mainly for finding or for ranking all relevant variables to rails manufacturing process that are useful to build a good predictor. The nature of the data, however, has a significant influence on what type of input selection algorithm should be used. Accordingly, there are several questions one should answer prior to solving variable and feature selection problems (Guyon and Elisseeff 2003):

1) Is there any domain knowledge (Guyon and Elisseeff 2003)?
   Tata Steel process expert has been closely involved in data pre-processing stage to clarify variable correlations and redundancy among data items and therefore contrast a better set of rails data.

2) Are data features having same units (Guyon and Elisseeff 2003)?
   Rails data contain tens of inputs that have different units and consequently they need to be normalized beforehand.

3) Is there any interdependency among features (Guyon and Elisseeff 2003)?
   Correlation analysis is a sufficient tool to observe interdependency of features. This step will be discussed in details in the next sections.

4) Is it important to prune the input variables? Human expert knowledge that is applied in the data pre-processing section helps eliminating irrelevant variables and deleting missing records.

5) Is it important to assess features individually (Guyon and Elisseeff 2003)?
   Understanding the influence of each input variable on the rails process is a crucial step because each data input has a different contribution on the system than the other inputs and therefore an input ranking method is to be considered.

6) Is there any suspicion that the data at hand is dirty?
   Data pre-processing is an important step to overcome such concern via finding outliers, removing missing values and identifying relevant inputs.

7) Is any predictor needed (Guyon and Elisseeff 2003)?
   The answer to this question is 'yes' so to assess the influence of each input variable.

8) Is there any idea what is to be tried first (Guyon and Elisseeff 2003)?
   If no, a linear predictor is the best solution that can be applied.

The correlation coefficient analysis, according to the above questions and their answers, is to be carried-out first for a better visualization of the data to be modelled and to reveal any relationships between inputs and outputs as well as detecting any redundancies or non-significant inputs.

4.3.1 Rails Data Correlation Analysis

The transformation of the variable into a solid and understandable form before the training process is common practice in data modelling and machine-learning methods. The process is carried-out to reduce the dimensionality of data inputs and to optimise the generalization performance (Bishop 2006). This transformation is usually implemented before the data is presented to the learning and modelling network and is known as data pre-processing. Once the data are pre-processed, a set of inputs and the desired set of outputs are presented to the learning agent.

As mentioned in the previous section, correlation coefficient analysis can significantly help identify linear dependencies between inputs and outputs. Although the rails production line is characterised by non-linear behaviour, a better visualization of the data to be modelled as well as identifying the input-output redundancies can be achieved via correlation analysis. The Pearson correlation coefficient is the most familiar measure of dependence between two quantities, which can be calculated via the following equation:

$$p_{ij} = \frac{\sum_{K=1}^{N}(x_i(k) - \bar{x}_i)(x_j(k) - \bar{x}_j\,)}{\sqrt{\sum_{K=1}^{N}(x_i(k) - \bar{x}_i)^2 \quad \sum_{K=1}^{N}(x_j(k) - \bar{x}_j)^2}} \tag{4.1}$$

For a series of n measurements of X and Y represented as x_i and y_j where i , j =1, 2, …, n, the correlation matrix is a matrix containing all the correlation coefficients where p_{ij} ϵ [-1, +1] is the Pearson correlation coefficient between variables x_i, x_j , it indicates the linear dependency between output and given input if one of the variables is the output, otherwise it represents co-linearity or a measure of the degree to which two input variables are correlated. $x_i(k)$ is the kth sample of x_i , $\bar{x}_i$ is the sample mean and N refers to the total number of samples (Y. Y. Yang *et al.* 2011). In the case of positive linear relationship (correlation), p_{ij} is +1 and −1 in the case of a perfect negative linear relationship and the values between −1 and +1 showing the degree of linear dependency between the variables, there is a weak relationship (closer to uncorrelated) as the variables approach zero.

Data sets of high dimensionality are not comprehensively correlated in all the features because of the inherent sparsity of the data. Finding linear correlations among such dataset leads to very large number of correlation coefficients which will be summarized in a compact form of symmetric matrix R, defined as follows:

$$\boldsymbol{R} = [p_{ij}]_{n \times n} = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{pmatrix} \tag{4.2}$$

where p_{ij} is the correlation coefficient defined in equation (4.1), n is the total number of data variables in the given data set and $\boldsymbol{R}$ is symmetric with $p_{ij} = 1 , i = \{1, 2, \dots, n\}$. In this research, only the data variables that are related to steel-making and continuous casting sub-processes have been chosen (called as rails-cast data, with process variables $x_1 \sim x_{70}$), as the two mentioned sub-processes have significant influence on the internal rails quality. Before applying correlation analysis on the rails data, it is worthwhile defining rails data variables in terms of their weights. Data weights play a role in describing the main features of a collection of data, where instead of each of the data variables contribute equally to the final effect or result, some data variables contribute more than others. Figure 4.1 demonstrates the contribution of rails data points based on their weights.

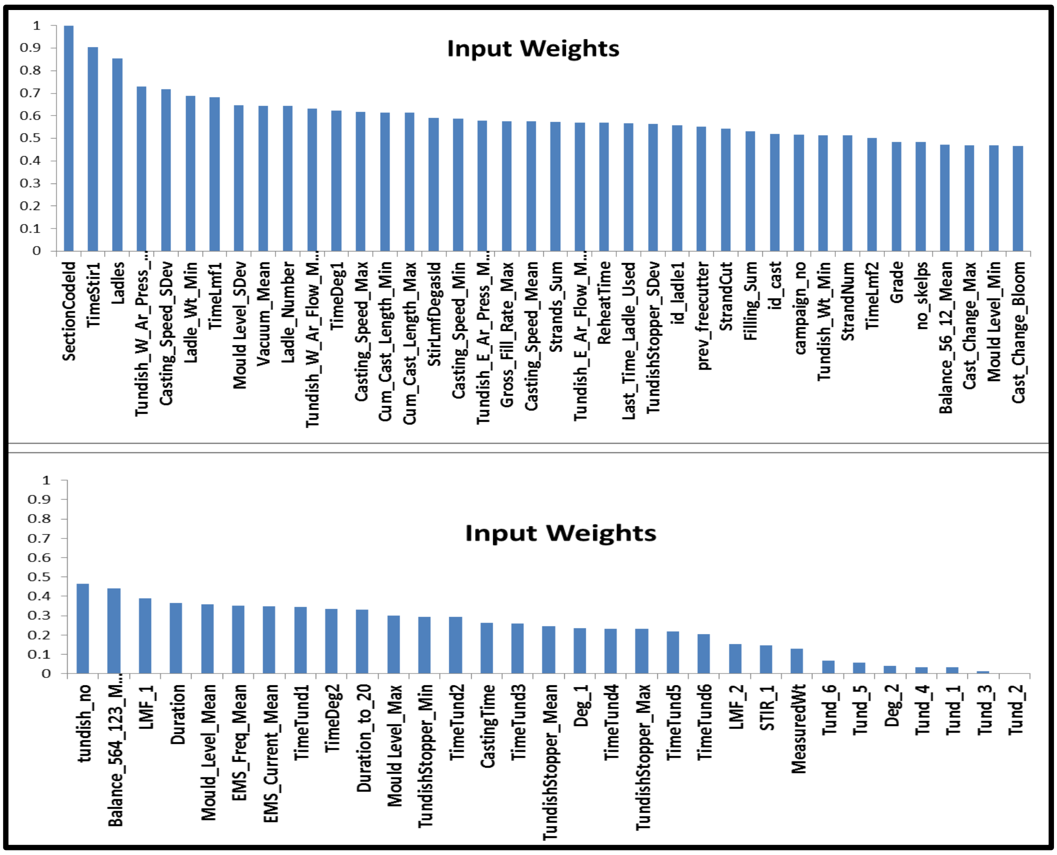
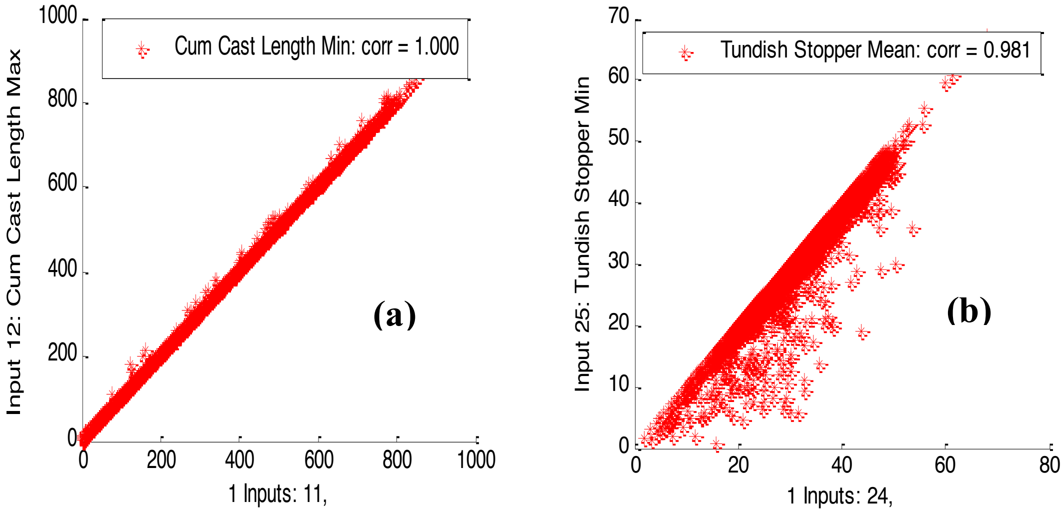
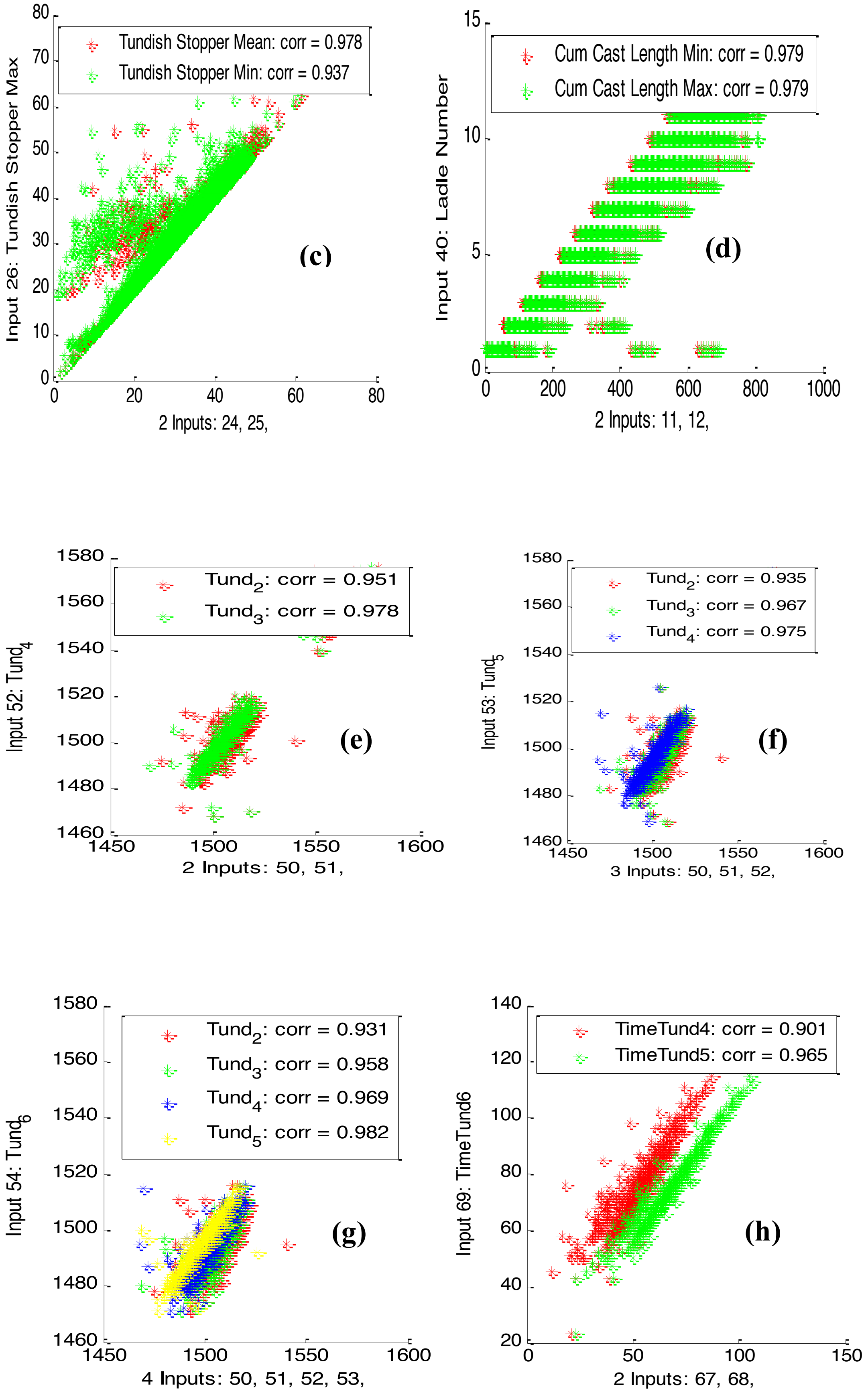


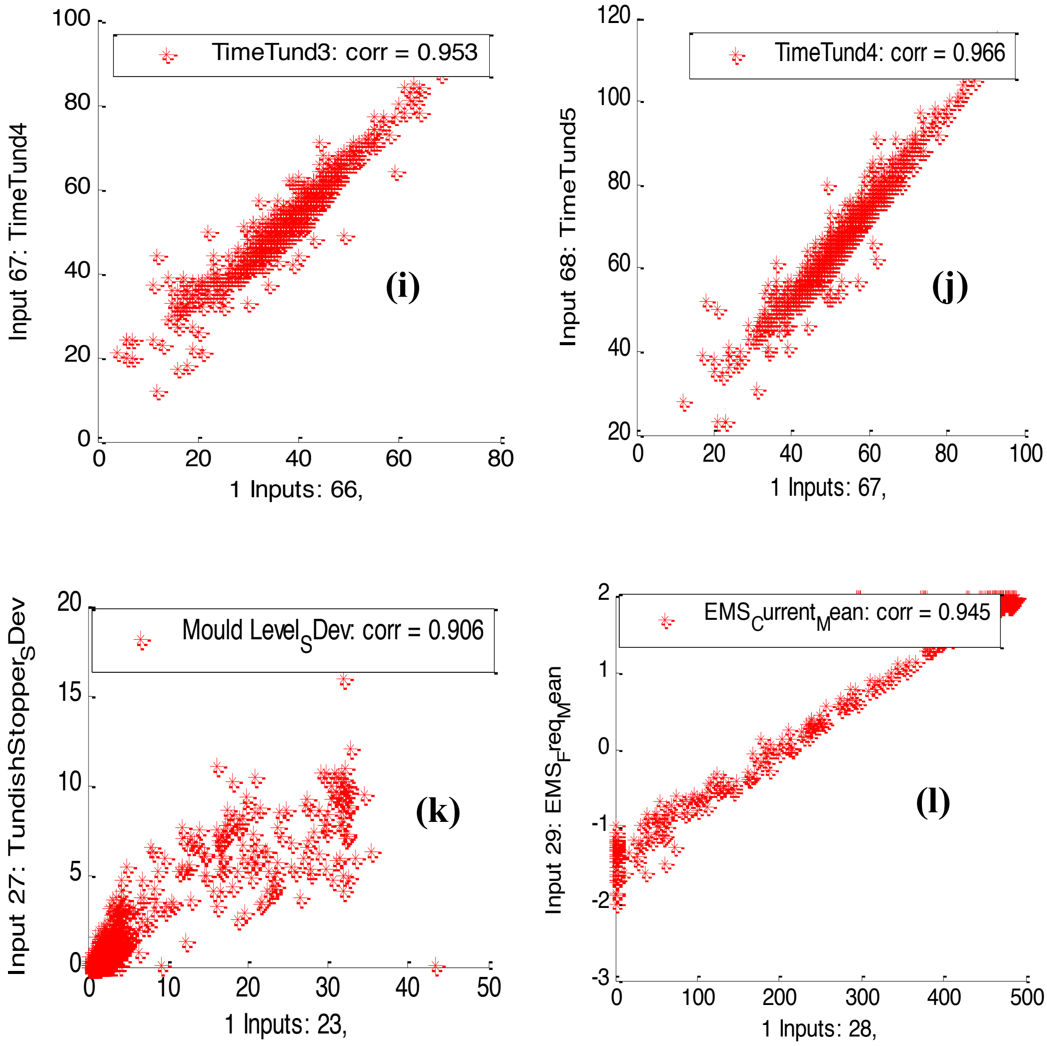
Figure 4.1: Rails Data Input Weights

Figures 4.2 and 4.3 show the correlations among some data inputs, and between data inputs and output respectively. The graphs also show that there is a significant correlation among some different inputs as presented in Figure 4.2, where many correlation coefficients exceed 0.95 ($p > 0.95$). In contrast, there is a slight linear relationship between data inputs and the number of rejected rails (output) with the maximum correlation coefficient being 0.144 related to the 9th input as shown in Figure 4.3:





Tundish Stopper Mean: corr = 0.978
Tundish Stopper Min: corr = 0.937
Input 26: Tundish Stopper Max
2 Inputs: 24, 25,
0
20
40
60
80
0
10
20
30
40
50
60
70
80
(c)
Cum Cast Length Min: corr = 0.979
Cum Cast Length Max: corr = 0.979
Input 40: Ladle Number
2 Inputs: 11, 12,
0
200
400
600
800
1000
0
5
10
15
(d)
Tund$_2$: corr = 0.951
Tund$_3$: corr = 0.978
Input 52: Tund$_4$
2 Inputs: 50, 51,
1450
1500
1550
1600
1460
1480
1500
1520
1540
1560
1580
(e)
Tund$_2$: corr = 0.935
Tund$_3$: corr = 0.967
Tund$_4$: corr = 0.975
Input 53: Tund$_5$
3 Inputs: 50, 51, 52,
1450
1500
1550
1600
1460
1480
1500
1520
1540
1560
1580
(f)
Tund$_2$: corr = 0.931
Tund$_3$: corr = 0.958
Tund$_4$: corr = 0.969
Tund$_5$: corr = 0.982
Input 54: Tund$_6$
4 Inputs: 50, 51, 52, 53,
1450
1500
1550
1600
1460
1480
1500
1520
1540
1560
1580
(g)
TimeTund4: corr = 0.901
TimeTund5: corr = 0.965
Input 69: TimeTund6
2 Inputs: 67, 68,
0
50
100
150
20
40
60
80
100
120
140
(h)



Figures 4.2: Rails data input variables correlation analysis where (a) represents the correlation between input 11 and input 12, (b) represents the correlation between input 24 and input 25, (c) represents the correlation between inputs 24,25 and input 26, (d) represents the correlations between inputs 11, 12 and input 40, (e) represents the correlation between inputs 50,51 and input 52, (f) represents the correlation between inputs 50, 51, 52 and input 53, (g) represents the correlation between inputs 50, 51, 52, 53 and input 54, (h) represents the correlation between inputs 67, 68 and input 69, (i) represents the correlation between input 67 and input 68, (j) represents the correlation between input 66 and input 67.

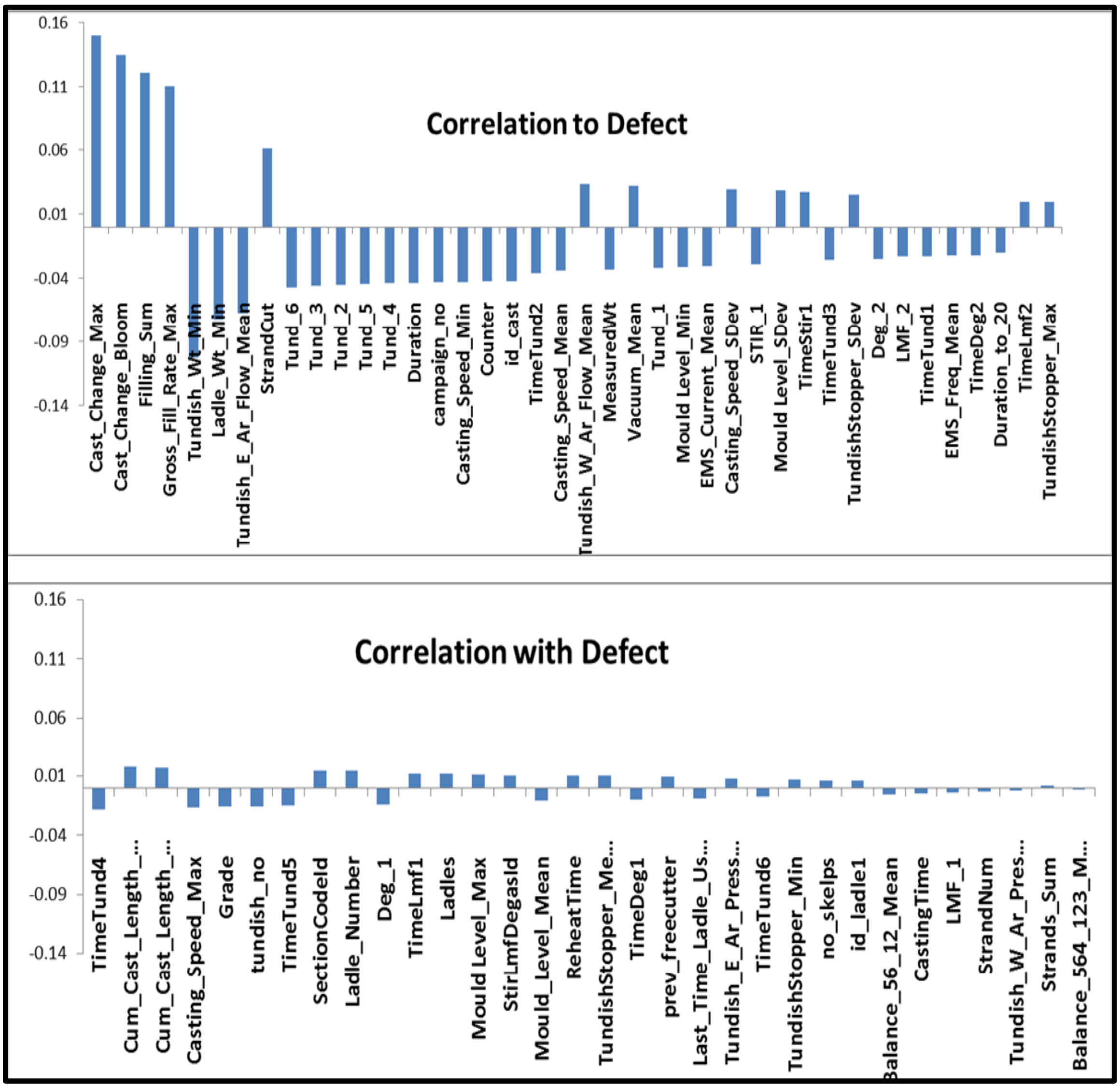


Figure 4.3: Rails quality data variables, Input/output correlation analysis

Highly correlated input variables reflect significant redundant information contained in the corresponding variables, in the sense that no additional information is gained by adding them, which need to be eliminated. Table 4.1 includes the top-10 correlation coefficients.

Table 4.1: Top 10 significant correlation coefficients

Rank	Output Correlation	Variables	Input Correlation	Input Variables
1	0.144	x_9	0.982	x_{53}, x_{56}
2	0.133	x_{10}	0.981	x_{24}, x_{25}
3	0.111	x_8	0.979	x_{11}, x_{40}
4	0.106	x_{13}	0.979	x_{12}, x_{40}

5	-0.097	x_7	0.978	x_{24}, x_{26}
6	-0.074	x_6	0.975	x_{52}, x_{53}
7	-0.071	x_{30}	0.967	x_{51}, x_{52}
8	0.060	x_3	0.966	x_{67}, x_{68}
9	-0.050	x_{37}	0.965	x_{68}, x_{69}
10	-0.047	x_{54}	0.953	x_{66}, x_{67}

According to Table 4.1, a highly non-linear relationship between inputs and output has been detected; such non-linearity is caused by having too many variables trying to have the same function. Table 4.1 also shows a solid linear relationship among data input variables. Such results are not powerful enough to identify inputs that are most relevant to describe the output. Consequently, an input variable selection-based non-linear predictor becomes desirable to retain the most relevant inputs that will be used for the actual modelling.

Many algorithms have been tailored for the purpose of identifying outliers and influential points. Embedded methods, which perform input selection in the training process and are usually specific to given learning models, improve predictor performance in contrast to simpler variable ranking methods such as correlation methods; however such improvements are not significant due to the fact that domains with large scale data suffer from the curse of dimensionality (Guyon and Elisseeff 2003). Other methods are proposed to deal with imbalanced data sets where the distributions of class labels are not same. A model input selection using neural network based feed-forward selection will be discussed in the next section.

4.3.2 ANN-Based Feed-Forward Model Input Selection Algorithm

In many real world applications, a possible model bias (approximation error) is reduced by collecting a high number of variables. Unfortunately, many variables are sometimes highly correlated, and a complex model may comprise many irrelevant variables. As a result, these models may be difficult to interpret and may have less predictive power. In such cases, a more *parsimonious* model becomes desirable.

Practically, as a separate data pre-processing step, there are several reasons behind the interest in reducing rails data dimensionality and discarding irrelevant features (Alpaydim 2010):

1) The number of input dimensions and the size of the data govern the complexity of most learning algorithms. Consequently, technical difficulties also arise when learning from large scale data sets, as the training time may be increased. Therefore, a sufficient amount of memory is required to hold the training set (Alpaydim 2010).

2) Data can be plotted and analysed visually for outliers and structure when it is represented in a few dimensions (Alpaydim 2010).

3) Simpler models have proven to be very robust on small datasets (Alpaydim 2010).

4) A better view of the process that underlies the data can be achieved if the Data is represented in fewer features (Alpaydim 2010).

5) Human-based knowledge extraction can be easily applied on small scale dataset (Alpaydim 2010).

Approaches such as neural network are proposed to distinguish between the most significant input variables based on their importance and the other irrelevant and redundant inputs are omitted accordingly. A simple 3-layer feed forward neural network has proven to be very effective for the purpose of input-output mapping as well as approximating any continuous non-linear function with arbitrary accuracy provided that sufficient hidden neurons are employed (Hornik 1991). The iterative feedforward input selection algorithm is presented in Figure 4.4:

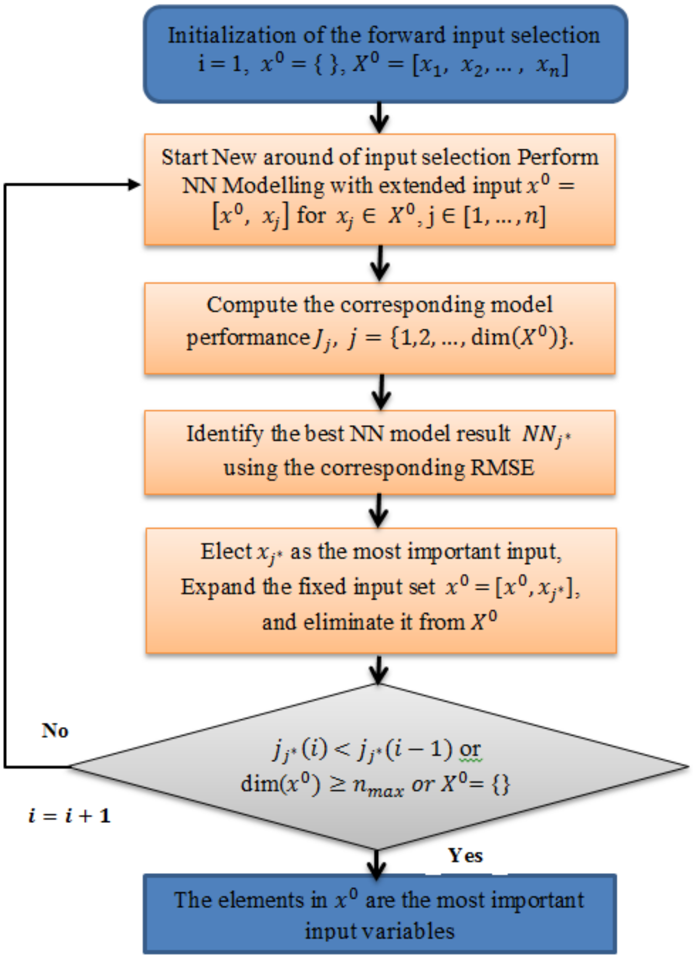

Initialization of the forward input selection
$i = 1,\ x^0 = \{\ \}, X^0 = [x_1,\ x_2, \dots,\ x_n]$
Start New around of input selection Perform NN Modelling with extended input $x^0 = [x^0,\ x_j]$ for $x_j \in X^0, j \in [1, \dots, n]$
Compute the corresponding model performance $J_j,\ j = \{1,2, \dots, \dim(X^0)\}$.
Identify the best NN model result NN_{j^*} using the corresponding RMSE
Elect x_{j^*} as the most important input, Expand the fixed input set $x^0 = [x^0, x_{j^*}]$, and eliminate it from X^0
$j_{j^*}(i) < j_{j^*}(i-1)$ or $\dim(x^0) \geq n_{max}$ or $X^0 = \{\}$
No
$i = i + 1$
Yes
The elements in x^0 are the most important input variables

Figure 4.4: Summarized Neural Network Forward Input Selection Procedure

As it has been summarized in Figure 4.4, the input selection is performed via an iterative scheme by choosing a single input for the feed forward neural network modeling in the first iteration. The algorithm is designed so that all the input variables are trained as the single input candidate $x_j \in X^0$ provided that, at each of the iterations, the NN modeling is performed with an extended input variable from the identified input set X^0.

Having each input variable sampled, a performance evaluator is used to categorize the corresponding inputs. Root mean square error (RMSE) is a good measure of accuracy and is applied here to quantify the difference between values implied by predictor and the real values of the quantity being predicted. The input variables with the minimum Root mean square error are ranked as highly important inputs; consequently, they are removed from the input data set X^0, and added to a new set x^0. In each round of the NN input selection, the corresponding model performance is computed based on RMSE; the input with the least RMSE value is eliminated from input data set X^0 and stored in the set x^0 accordingly.

The resulted RMSE values of irrelevant and redundant Inputs would always stay high, hence they would not be chosen as an important input. The above NN input selection scheme is repeated until a sufficient number of inputs identified or the model performance is not improved by training more inputs (Yang *et al.* 2011a). The number of hidden units is to be chosen carefully due to the fact that with a few hidden units, the net does not have enough free parameters to fit the training data well (Duda *et al.* 2001). Empirical rules are used to choose the number of hidden units based on the number of NN model inputs.

Having performed the NN forward-based input selection, 40 inputs have been selected as the most important input variables where the rest of inputs are omitted. All the subsequent analysis will be based only on these inputs. The results achieved by applying the aforementioned NN feedforward input selection procedure is summarize in Table 4.2.

Table 4.2: Input Ranking Using NN Feed-Forward Selection

No.	Variables	No.	Variables	No.	Variables	No.	Variables
1	x_8	**11**	x_{24}	**21**	x_{51}	**31**	x_{23}
2	x_2	**12**	x_{46}	**22**	x_{63}	**32**	x_{20}
3	x_9	**13**	x_{13}	**23**	x_{54}	**33**	x_{48}
4	x_{53}	**14**	x_{34}	**24**	x_{58}	**34**	x_{59}
5	x_{16}	**15**	x_{67}	**25**	x_{12}	**35**	x_{61}
6	x_{47}	**16**	x_{22}	**26**	x_{25}	**36**	x_{50}
7	x_3	**17**	x_{28}	**27**	x_{37}	**37**	x_{15}
8	x_{68}	**18**	x_{19}	**28**	x_{62}	**38**	x_{55}
9	x_{11}	**19**	x_{41}	**29**	x_1	**39**	x_7
10	x_{65}	**20**	x_6	**30**	x_{57}	**40**	x_{35}

As mentioned previously, applying feed-forward neural network model has proved to be very effective in selecting the most significant input variables. Such variables will lead to a good generalization performance of modelling and learning algorithms. The next section will present the application of the neural network model to the rails manufacturing process dataset.

4.4 RapidMiner Rails Data Neural Network Model

4.4.1 RapidMiner

Rapid Miner is a free access software that was originally designed in 2001 to perform artificial intelligence based tasks. Recently, the open-source software has been developed as standalone data-mining engine, data evaluation and visualization. An interesting feature of the software is that it is a comfortable user-interface containing over 500 operators tailored to generate different models and execute various tasks such as classification, clustering, regression and data transformation. Rapid Miner also comprises powerful high-dimensional plotting facilities.

4.4.2 Neural Network Model

The neural network operator provided by RapidMiner learns a model by means of a feed-forward neural network which is trained by a back-propagation algorithm-based multi-layer perceptron. Figure 4.5 illustrates the neural network framework designed for rails data modelling using the RapidMiner software.

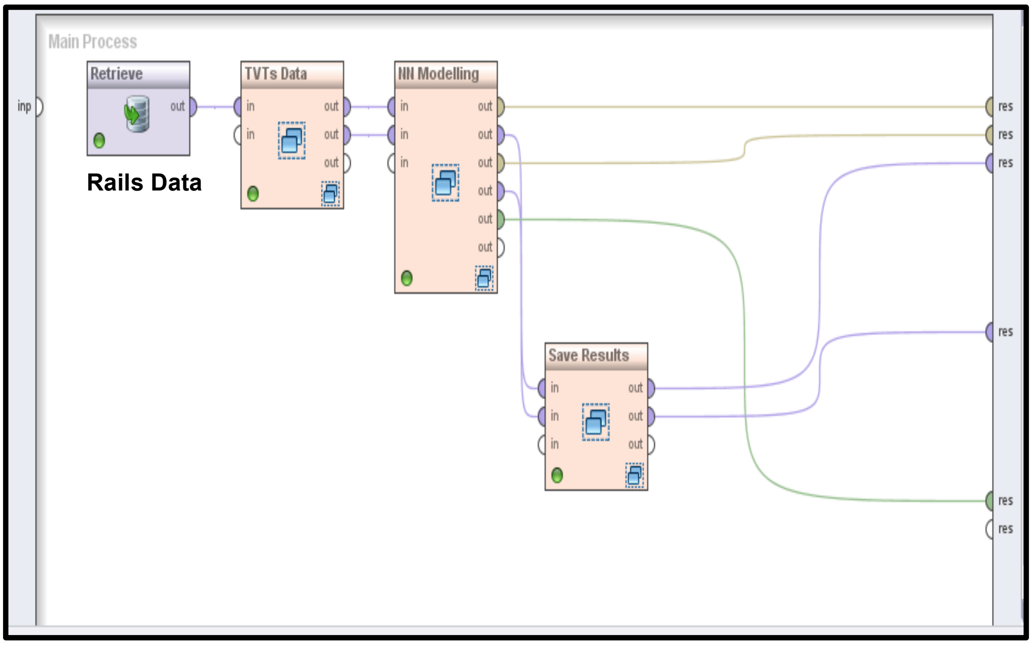


Figure 4.5: Rails data Neural Network Modelling Framework

ANN is usually referred to as neural network (NN), is a computational model that is motivated by the functional aspect and sophisticated structure of human biological neural networks. As a promising artificial intelligence technique, neural networks have been widely applied in many application areas. The neural network paradigm has the ability to organize its structural constituents which are known as perceptrons (neurons) so as to execute certain computations better than today's digital computers do (Haykin 1999). Advanced neural networks are more likely used for complex relationships modelling between inputs and outputs or to find patterns in data. Neural networks offers useful properties and capabilities in contrast to other modelling techniques such as non-linearity, input-output mapping, adaptivity and simplicity (Haykin 1999).

The neural network processing unit (neurons) is a fundamental feature of neural networks. The perceptron computes a weighted sum of the input signals and then passes the result through a nonlinear activation function (Yoo *et al.* 2006). The nonlinear model of the perceptron (neurons) is shown in Figure 4.6. However, mathematically, the perceptron's output can be represented as follows (Yoo *et al.* 2006), (Haykin 1999):

$$u_k = \sum_{j=1}^{n} w_{kj}\, x_j \tag{4.3}$$

$$y_k = \varphi(v_k) = \sum_{j=1}^{n} w_{kj}\, x_j + b_k \tag{4.4}$$

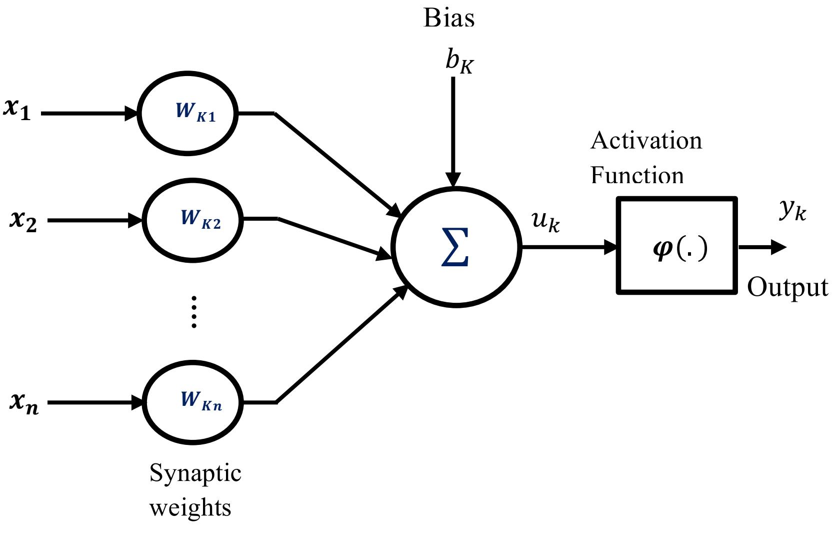


Figure 4.6: Nonlinear Model of Neural Network's Neuron (Haykin 1999)

where x_1, x_2, …, x_n are the input signals; w_{k1}, w_{k2}, …, w_{kn} are the synaptic weights of the perceptron k, u_k is the linear combiner output, b_k is the bias or a constant threshold value, $\varphi(.)$ is the activation function, $v_k$is the induced local field of the perceptron and y_k is the output of the neuron. The activation signal, commonly referred to as the squashing function, limits the perceptron's output to a closed interval, e.g. [0,1] or alternatively [-1,1] (Yoo *et al.* 2006).

The activation function maps the inputs from linear space to a nonlinear space. It is worth mentioning that the nonlinear characteristic of artificial neural networks comes from the nonlinearity of the activation function. Although there are many types of activation functions such as threshold function and piecewise-linear function, sigmoid function, whose graph is s-shaped as shown in Figure 4.7, is utilised in the design of artificial neural networks and therefore, the values of the attributes are scaled to -1 and +1. The input values are converted into output signals based on the activation functions (Che *et al.* 2011). The sigmoid function satisfies the following relationship (Bishop 2006):

$$\varphi(u) = \frac{1}{1 + e^{-au}} \tag{4.5}$$

where a is a slope parameter. The above equation also called the log-sigmoid function because the sigmoid can also be formulated using hyperbolic tangent function instead of this function where it will be called a tan-sigmoid. The sigmoid represented above has the property of being similar to the step function but with the addition of uncertainty region.

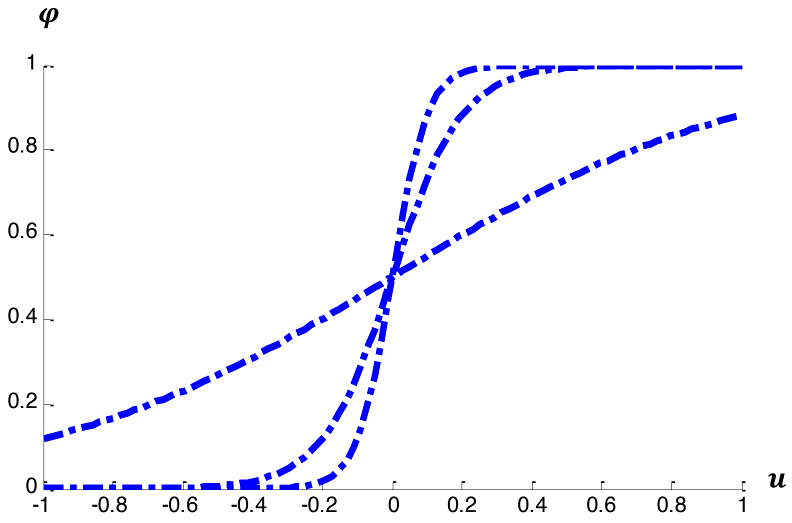


Figure 4.7: Sigmoid Function for Varying Slop Parameter

A feedforward neural network-based back-propagation algorithm is employed to learn the model. The rails input data are mapped on to a set of appropriate output based on the multilayer perceptron (MLP) which consists of multiple layers of nodes in a directed graph where every layer is connected to the following one. Each node is a processing unit (neuron) with a non-linear activation function. The structure of a sequential connections multi-layer neural network is shown in Figure 4.8 (Haykin 1999), (Yoo *et al.* 2006):

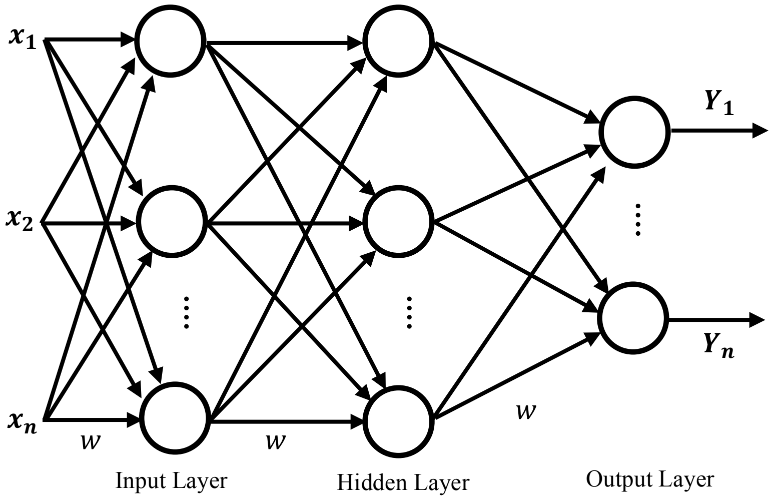


Figure 4.8: Architecture of Multi-layer Neural Network

The Neural network architecture consists of an input layer of nodes (neurons), hidden layers of neurons and an output layer. Each connection in the neural network is associated with a numeric number called weight. The back-propagation algorithm used for training neural network is popularised by Rumelhart *et al.* (1986). The algorithm has the ability to iteratively adjust the weights between the neurons in order to minimize the error function by some smaller amount (Örkcü and Bal 2011). This process will be repeated for sufficiently large number of training cycles until convergence where the error value is small. Based on the previous correlation analysis, human expert knowledge and input weights analysis, neural network model is examined on five rails data sets. Figures 4.9- 4.13 present the result of NN model on these datasets.

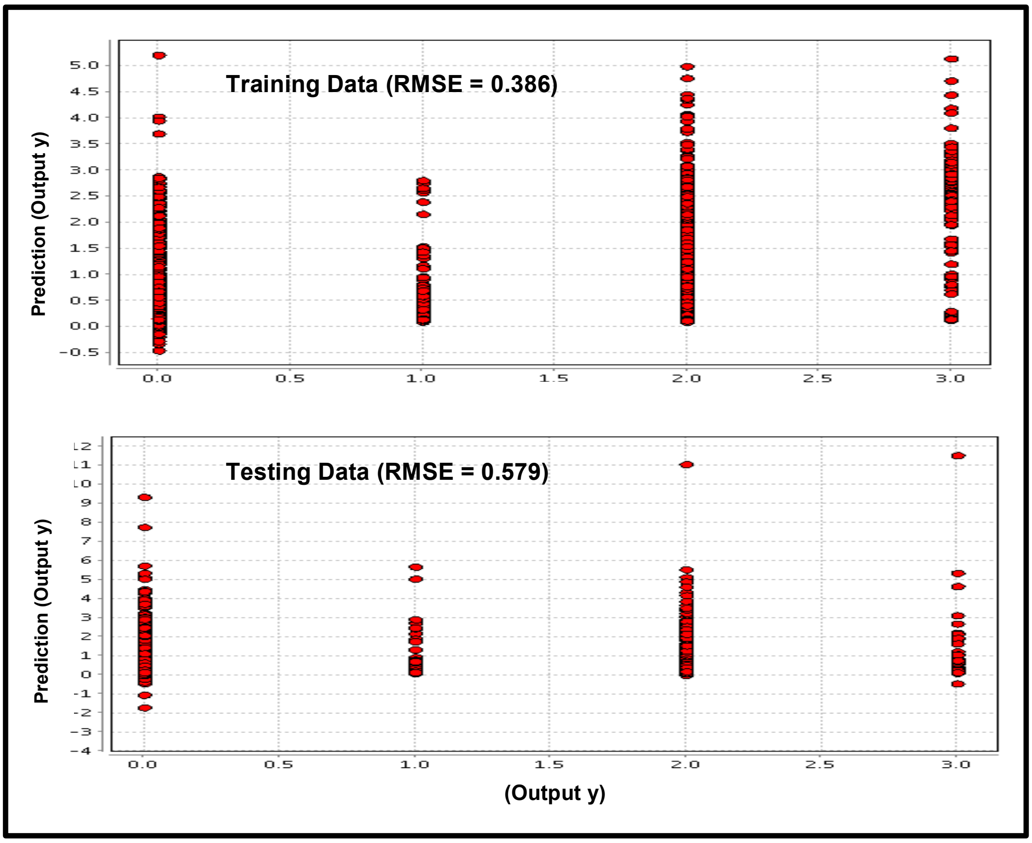


Figure 4.9: Neural Network model of full rails data set (143 inputs and 65466 data records)

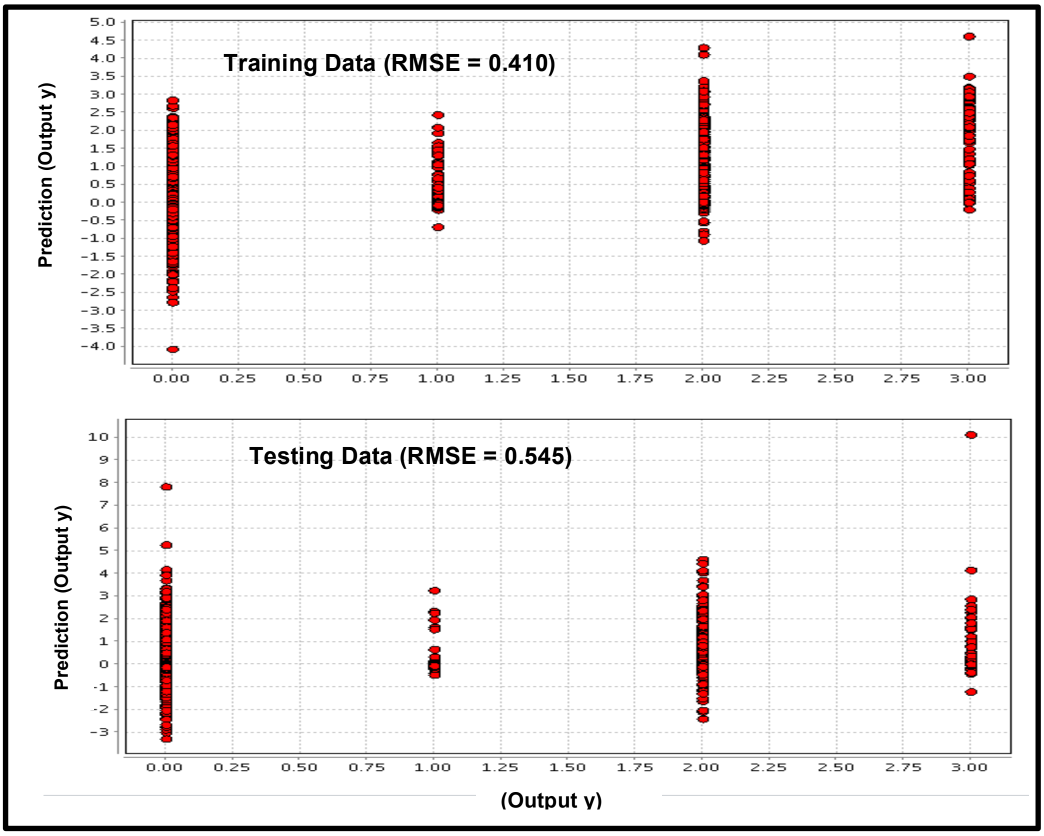


Figure 4.10: Neural Network model of full rails data set (70 inputs and 39327 data records)

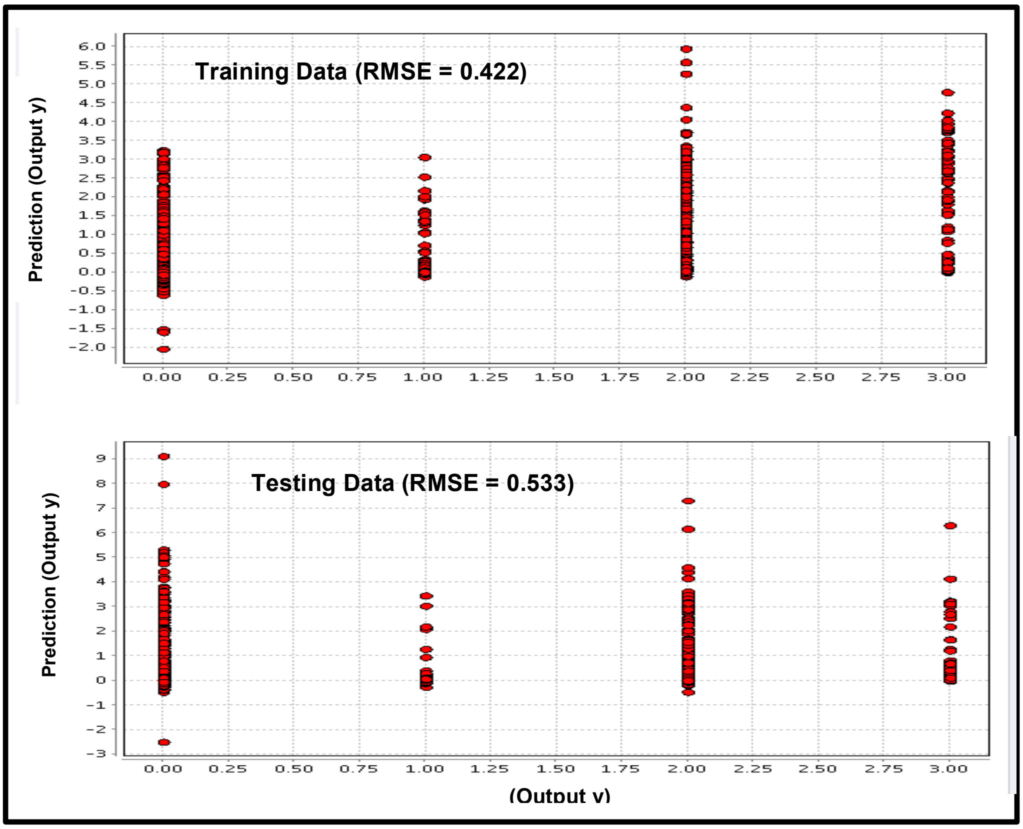


Figure 4.11: Neural Network model of full rails data set (40 inputs and 39327 data records based on input weights analysis)

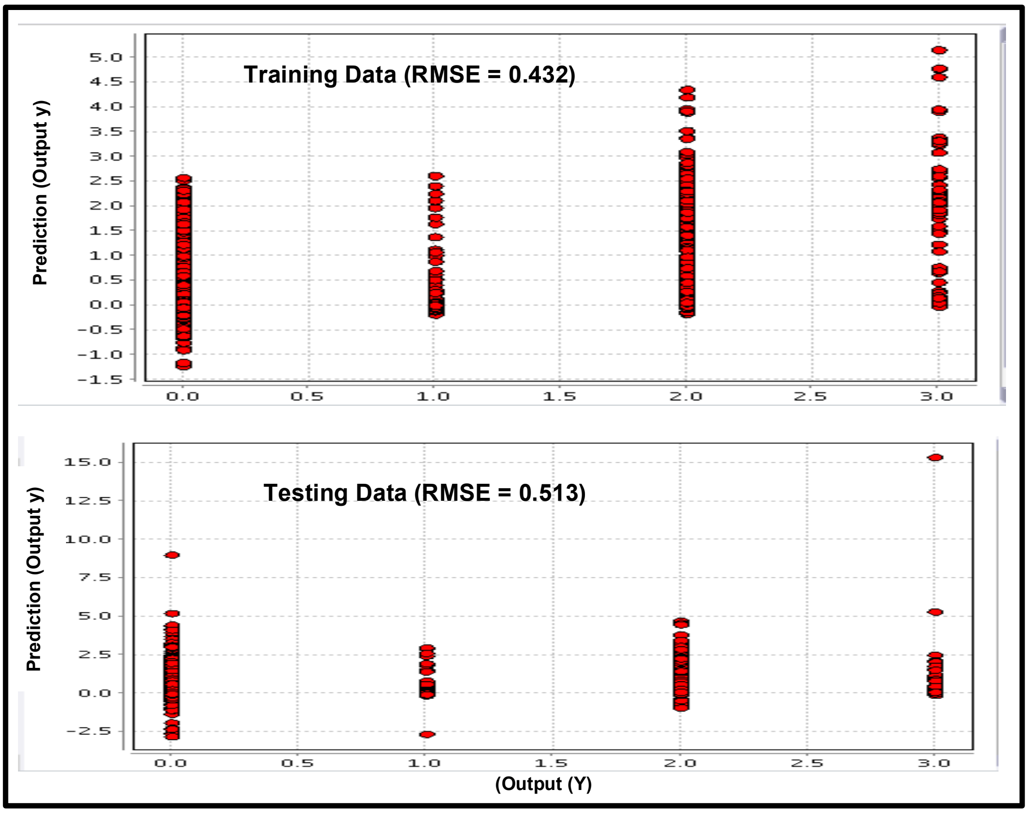


Figure 4.12: Neural Network model of full rails data set (40 inputs and 39327 data records based on Input correlation analysis)

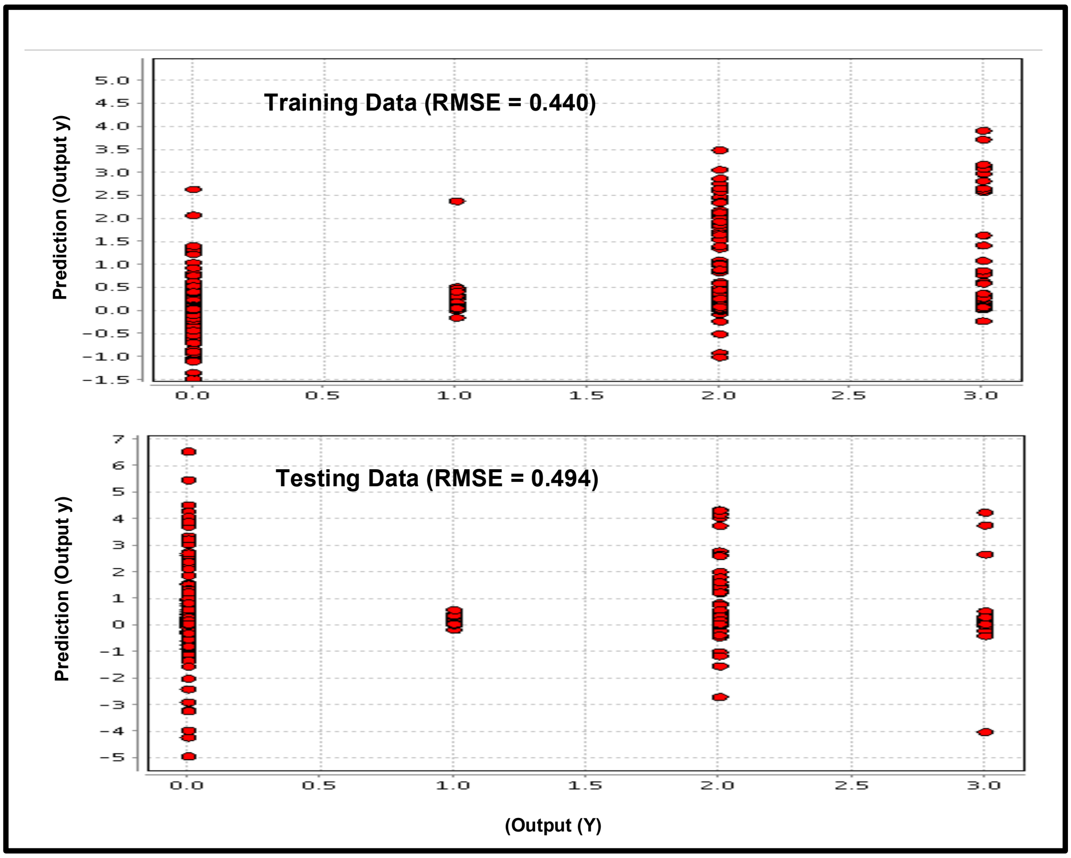


Figure 4.13: Neural Network model of full rails data set (27 inputs and 39327 data records based on Expert Knowledge manual selection)

The results of neural network model are as shown in Table 4.4, where four rails datasets are compared. The optimal RMSE values are obtained for all datasets. However, the comparison shows poor modelling results with a performance of less than 10% were obtained for all rails datasets based on RabidMiner neural network model. Such a poor generalization performance is due to the fact that rails data are highly imbalanced where the good rails far outnumber the rejected (bad) rails. A significant classification improvement is achievied when applying data resampling techniques as will be shown in chapters 5 and 6. Class imbalance problem significantly hinders the performance of standard classifiers and modelling algorithms. The performance would always be biased in favour of the dominating class (majority), while the data related to the minority class tend to be misclassified.

Table 4.3: Rapid Miner Neural Network Performance Summary

Rails Data Size	Data Selection method	Specificity Training/ Testing	Sensitivity Training/ Testing	Accuracy Training/ Testing	RMSE (Training)	RMSE (Testing)
70 inputs and 39327 data records	ANFIS model Input Selection	0.994/0.984	0.232/0.110	0.949/0.930	0.410	0.545

40 inputs and 39327 data records	Input Weights selection (> 0.4)	0.995/0.987	0.165/0.095	0.949/0.930	0.422	0.533
40 inputs and 39327 data records	Input Correlations selection	0.991/0.987	0.163/0.110	0.942/0.933	0.432	0.513
27 inputs and 39327 data records	Expert Knowledge (Manual selection)	1.000/0.997	0.041/0.024	0.943/0.937	0.440	0.494

The performance measures of neural network model are assessed as follows:

Specificity = TN / (TN + FP) (4.6)

Sensitivity = TP / (TP + FN) (4.7)

Accuracy = (TN + TP) / (TP + TN + FP + FN) (4.8)

Specificity is the ability of the algorithm to accurately classify the majority class whereas Sensitivity is the ability of the algorithm to accurately classify the minority class. Accuracy refers to the overall percentage that both classes are correctly classified. The optimal classification metrics are the specificity (4.6) and sensitivity (4.7) since the rejected rails, minority class, are more important to be correctly classified. Sensitivity, specificity and accuracy (4.8) performances can be employed as performance metrics throughout the confusion matrix. The confusion matrix for the neural network modelling problem is illustrated in Table 4.4.

Table 4.4: Confusion Matrix

	Predicted positive	Predicted Negative
Real Positive	TP (True Positive)	FN (False Negative)
Real Negative	FP (False Positive)	TN (True Negative)

As mentioned previously, by applying the feedforward neural network model, correlation analysis and input weights selection scheme have proved to be very effective in selecting the most significant input variables. Although redundant as well as irrelevant input variables were omitted, the rails dataset is significantly imbalanced where the number of the majority classes outweighs the number of minority classes. Figures 4.14, 4.15 and 4.16 illustrate the class distributions of rails data, Heating Time for individual bloom and the influence of blooms changeover to rails defect.

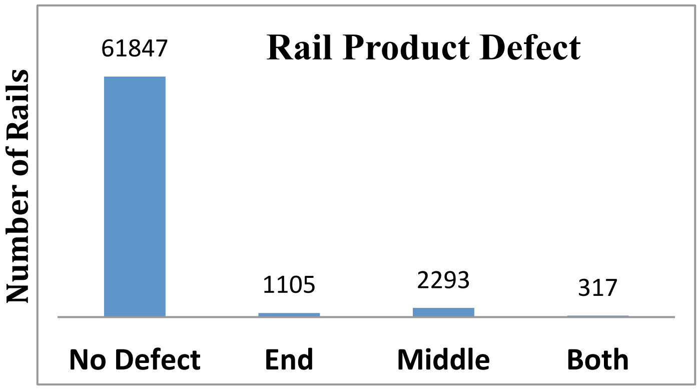


Figure 4.14: Rails Data Class distribution

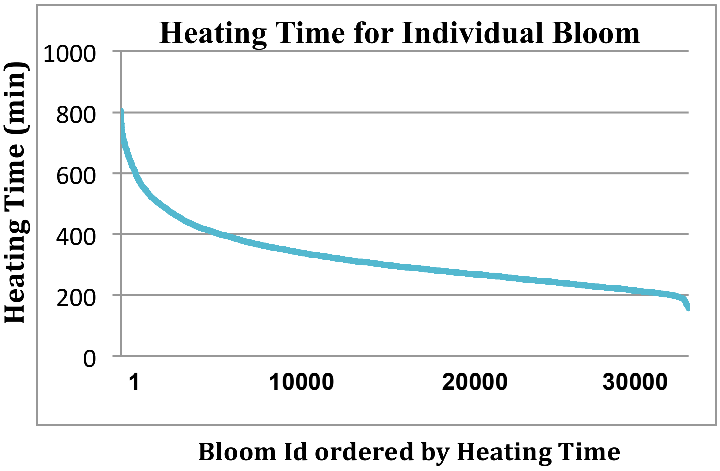


Figure 4.15: Heating Time for Rails individual bloom

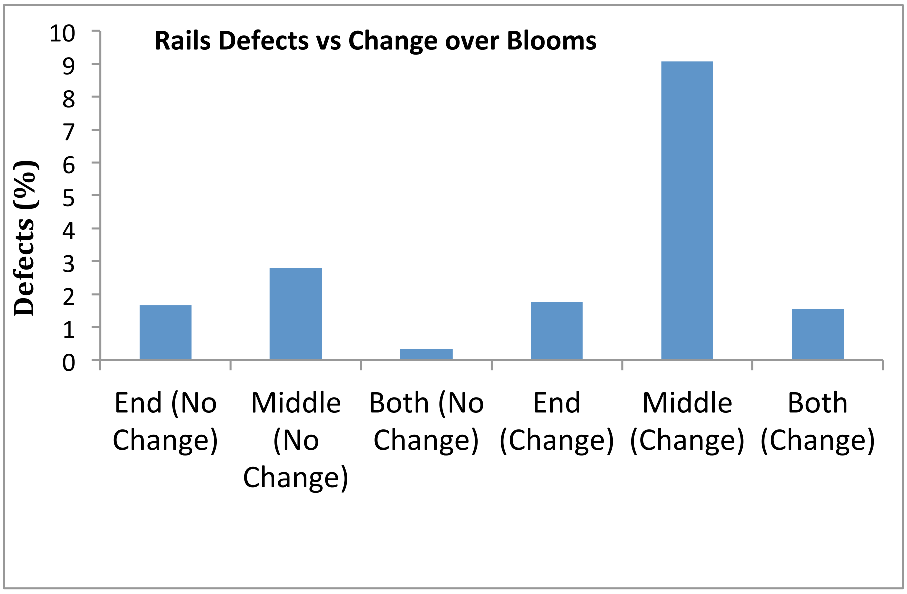


Figure 4.16: Rails Defects position *vs* Change over Blooms

Due to the fact that standard modelling algorithms and machine learning techniques yield better modelling performance with balance data sets, quality modelling results are not reachable with the current rails data set structure. Therefore, a direct data resampling approach is to be applied to improve the class distribution of rails data.

4.5 ANFIS Model for Rails Quality Data Classification with Bootstrapping-Based Over-sampling

As has been discussed in section 4.4, 40 input variables were selected using the NN forward input ranking scheme. The original output of rails production line is varies between o and 3, $y \in \{0, 1, 2, 3\}$ based on the NDT ultrasonic testing. $y = 0$ refer to a good rails and $y = \{1, 2, 3\}$ representing rejected rails as per flow position (end, middle, both) respectively. An adaptive neural-fuzzy modelling approach was developed for the purpose of data classification; only 10 % of the data were successfully classified. Such a low success rate, of classifying the rejected rails, was found due to the class imbalance in the training data set. Class imbalances hinder the performance of standard classifiers and modelling algorithms, the classification would always be biased in favour of the dominating class (majority) while the data belonging to the minority class tend to be misclassified (Estabrooks et al. 2004). In order to overcome such a concern, Bootstrapping-Based Over-sampling scheme will be adopted in the next section.

4.5.1 Bootstrapping for Rails Data Resampling

A data set is imbalanced if the samples belonging to the majority class outnumber the samples belonging to the minority class. Since standard machine-learning techniques and other modelling algorithms yield better classification performance with balance data sets, quality classification is not reachable with the current rail data set structure. Therefore, a direct data resampling approach is to be applied here to change the class distribution of rails data. Figure 4.17 displays the class distribution of rails data set.

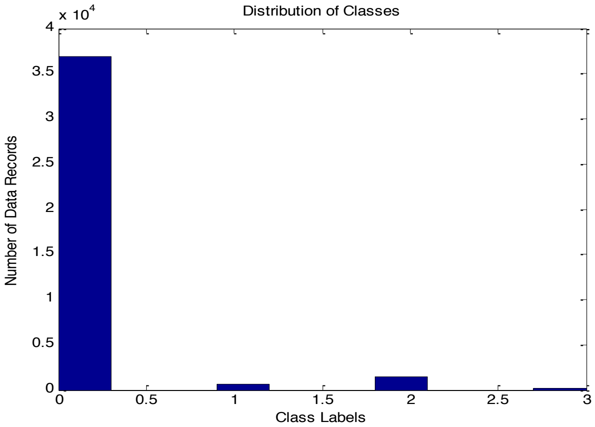


Figure 4.17: Rails data Class Distributions (where 0 represents good rails and values (1, 2 and 3) represent defected rails)

Changing the class distribution can be conducted via different resampling strategies: over-sampling, under-sampling or combination of both (Estabrooks et al. 2004). The advantage of such techniques is that they are external and therefore, easy transportable as well as very simple to implement (Estabrooks et al. 2004). The resampling scheme used here is over-sampling the minority class data as it avoids unnecessary information loss (Yang *et al.* 2011 a).

For the over-sampling to be carried out, the original rails-cast data (40 input variables) are separated in two sub-sets. One set is for the dominating class and the other is for the minor class. Subsequently, the minority class data is fed in to the bootstrapping resampling algorithm. The bootstrapping resampling algorithm yields a multiple randomly resampled subsets that have the same size as the size of the original minority subset (Yang *et al.* 2011 a).

The resampled subsets are combined with the majority class data to shape the resampled training data that is ready for the subsequent training procedures. The design parameter R_{mm} which defines as the ratio of the number of samples belonging to the majority class to that belonging to the minority class plays a crucial role in the bootstrapping over-sampling algorithm as it controls the level of imbalance for the resampled training data set. All of the existed resampling techniques are tailored to resample until the desired ratio between the majority and minority classes is reached.

Oversampling has proved to be effective as there is no information from the original training set is lost since all instances from majority and minority classes are kept. However, the drawback is that the training data size is significantly increased.

Therefore, the training time is also increased and sufficient amount of memory is required to hold the training set. Since the dimensionality of rails data set is very high, it is importantly to take in to account the resampling time in order to keep time as well as memory complexity under reasonable constraints. Figure 4.18 shows the influence of over-sampling strategy on the rails data.

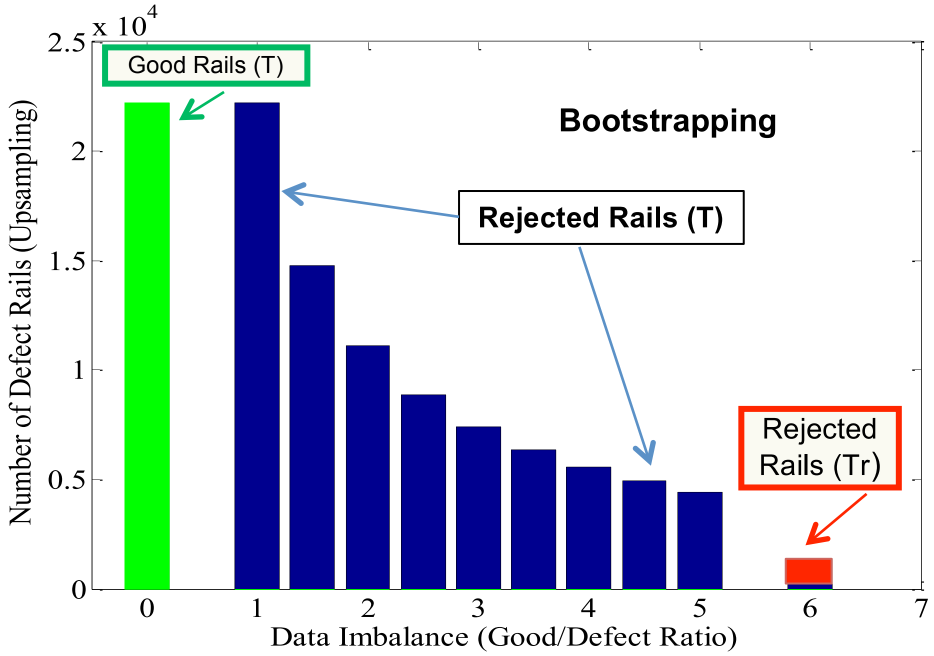


Figure 4.18: The Influence of R_{mm} on the Resampled Training Data.

4.5.2 An Adaptive Neural-Fuzzy Inference System

4.5.2.1 The Neural-Fuzzy Approach

Neural network and fuzzy logic are both complementary frameworks rather than competitive for many applications. Therefore, it is advantageous to employ these techniques as a combination rather than exclusively. Practically, such a combination is called a hybrid intelligent system. A neuro fuzzy hybrid system is one of the popular combinations that have been applied extensively in various domains. The essential part of the neuro fuzzy approach is connected with the attempt to unite the advantageous of fuzzy and neural techniques in standalone hybrid structure referred to as adaptive network.

A unique feature of the adaptive network is its ability to identify patterns and adapt themselves to cope with varying environment. Neural-fuzzy modelling is concerned with model extraction from numerical data that represents a system's dynamic behaviour. As per the above mentioned methodology, system modelling can serve two purposes:

- The system's behaviour can be directly predicted from the derived model;
- The derived model can be utilized to design a controller.

The crucial steps of neuro fuzzy modelling are:

- The fuzzification of input variables;
- Computation of the degree of satisfaction for linguistic terms;
- Fuzzy inferred parameters and premise conjunction;
- Output defuzzification.

4.5.2.2 An Adaptive Neural-Fuzzy Inference System Classification Based Fuzzy C-means Clustering.

As per their design principles, most of classification algorithms tend to enhance the overall classification accuracy without taking into account the class distribution of the data at hand. Therefore, the minority classes, when the data is extremely skewed as shown in Figure 4.17, are likely to be misclassified. This problem was addressed in the previous section using the Bootstrapping-based Over-sampling framework.

Having resampled the rails data set, ANFIS is chosen in this study as a classifier since the data labels which can be used as output set are determined.

ANFIS is presented in this research as a neural network that generates the rules of a fuzzy logic system. The fuzzy inference model is expressed as a collection of fuzzy rules as follows (Yang *et al.* 2011 a):

$$R_i\text{: } \textbf{If } x_1 \text{ is } A_{i1} \textbf{ and } x_2 \text{ is } A_{i2} \text{ .... } \textbf{and } x_n \text{ is } A_{in} \textbf{ then } y = z_i\ (\boldsymbol{x}) \qquad (4.9)$$

where $x_1, x_2, \ldots, x_n$ are data samples ($x_i \in U$) and y ∈ V are linguistic variables, A_{ij} are fuzzy sets defined on the universe U, z_i (x) is the calculated output. A simple ANFIS framework with two rules is presented in Figure 4.19. Layer 1 is the fuzzy membership mapping of the input, the "and" operation is performed by Layers 2 and 3 is the fire strength normalization of each fuzzy rule, the "then" part is performed by Layer 4, and the overall output of the fuzzy system is calculated in Layer 5.

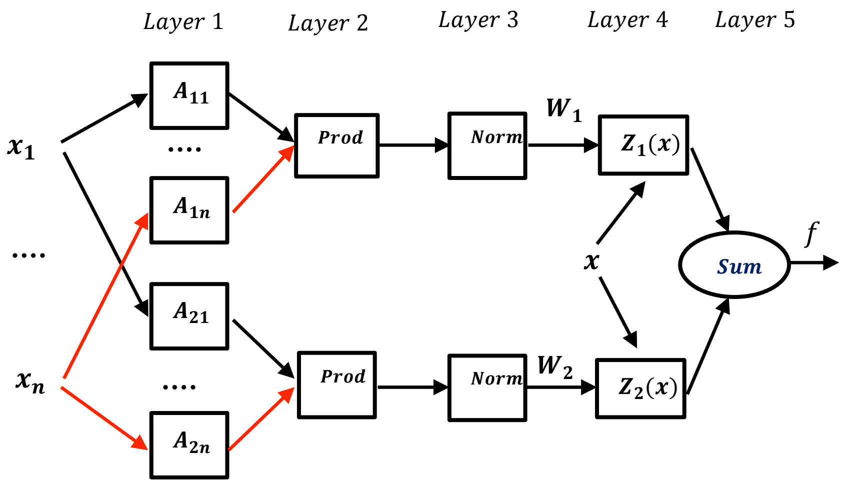


Figure 4.19: A Simplified ANFIS Structure (Yang *et al.* 2011 a)

Mathematically, the five layers that form ANFIS architecture can be represented as follows (Jang 1993):

Layer 1: Every node i in this layer is an adaptive square node with a node function:

$$O_{1,i} = \mu_{Ai}(x) \tag{4.10}$$

where x is the input to node i, $A_i$is the linguistic label (small or large) associated with this node function and μ_{Ai} is the membership function of A_i. μ_{Ai} is of the form (Jang *et al.* 1997):

$$\mu_A(x) = \frac{1}{1 + \left|\frac{x - c_i}{a_i}\right|^{2b}} \tag{4.11}$$

where $\{a_i, b_{i,} \, c_i\}$ is the premise parameter set and x is the input.

Layer 2: Each node in this layer is a fixed node whose output is the product of all the incoming signals. The fixed node calculates the firing strength w_i (Jang *et al.* 1997):

$$O_{2,i} = w_i = \mu_{Ai}(x).\mu_{Bi}(y), i = 1, 2. \tag{4.12}$$

Layer 3: Each node in this layer is a fixed node where the i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of all rules' firing strengths (Jang *et al.* 1997):

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1+w_2}, i = 1, 2. \tag{4.13}$$

Layer 4: Every node i in this layer is an adaptive node with a node function (Jang *et al.* 1997):

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_{i)} \quad (4.14)$$

where $\bar{w}_i$ is a normalized firing strength (output of layer 3) and $(p_i, \ q_i, \ r_{i)}$ is the consequent parameter set.

Layer 5: this layer contains a circle node (sum node) that computes the overall output as the summation of all incoming signals i.e. (Jang *et al.* 1997):

$$\text{The overall output } O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (4.15)$$

In this study, the fuzzy C-mean clustering (FCM) has been chosen to create fuzzy models due to its simplicity and efficiency (Yang *et al.* 2011 a). Fuzzy C-Mean clustering is a commonly used technique which subdivides the data set in to P clusters where each data element has a degree of belonging to exactly one cluster. Having chosen the number of clusters P which is equal to the number of fuzzy rules, the FCM partitions the data samples in to P fuzzy clusters so that the minimization of the following objective function is achieved as follows:

$$J_m = \sum_{k=1}^{N} \sum_{i=1}^{p} \mu_{ik}^{\alpha} ||\boldsymbol{x}(k) - \boldsymbol{c}_i \, ||^2 \quad (4.16)$$

Where μ_{ik} is the degree of membership of the kth data sample in the ith cluster, c_i is the centre of the ith cluster, $x(k)$ is the ith of measured data and α is the exponent value of μ_{ik} (Babushka *et al.* 1983). FCM converges to a local minimum of J_m so that the fuzzy membership degree μ_{ik} satisfies the following constraints (Babushka *et al.* 1983):

$$\sum_{i=1}^{p} \mu_{ik} = 1, \ \ \forall k \in \{1, N\}; \ \ 0 \le \mu_{ik} \le 1, \forall k, i \in \{1, p\} \quad (4.17)$$

The parameters of the fuzzy inference system are optimised using a hybrid learning algorithm. a combination of error back-propagation gradient descent method and 'Least-squares' method has been applied for training the membership function which has Gaussian form and other ANFIS parameters. The final output f can be expressed as a linear combination of the consequent parameters. The output f can be of the form:

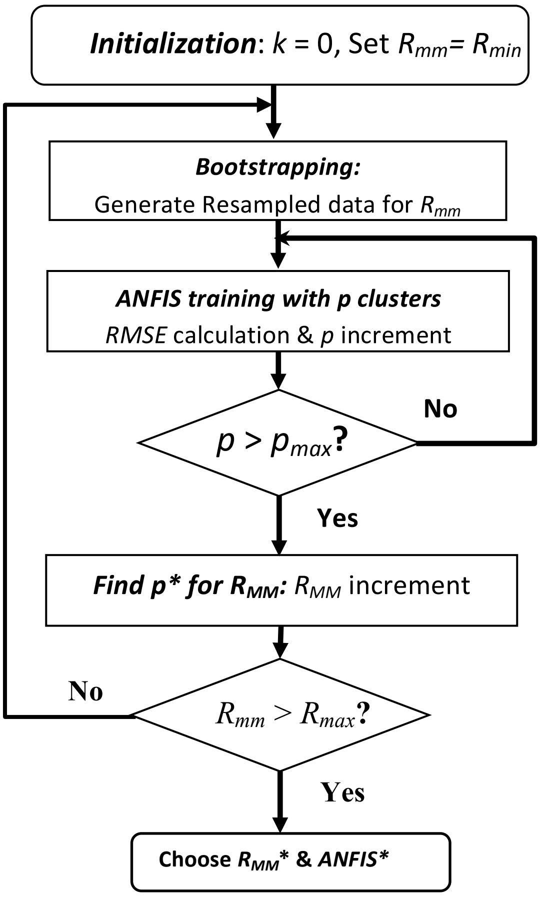
$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \tag{4.18}$$

$$= \bar{w}_1 f_1 + \bar{w}_2 f_2 \tag{4.19}$$

$$= (\bar{w}_1 x)p_1 + (\bar{w}_1 y)q_1 + (\bar{w}_1)r_1 + (\bar{w}_2 x)p_2 + (\bar{w}_2 y)q_2 + (\bar{w}_2)r_2 \tag{4.20}$$

Where f is linear in the consequent parameters $(p_1,\ q_1,\ r_1,\ p_2,\ q_2,\ r_2)$.

The majority to minority ratio R_{mm} and the number of fuzzy rules P are two crucial parameters that extremely influence the modelling performance. An exhaustive search via bootstrapping over-sampling and adaptive neural-fuzzy classification framework is carried out In order to identify these optimal design parameters; Figure 4.20 illustrates the scheme of iterative optimization algorithm (Yang *et al.* 2011 a).



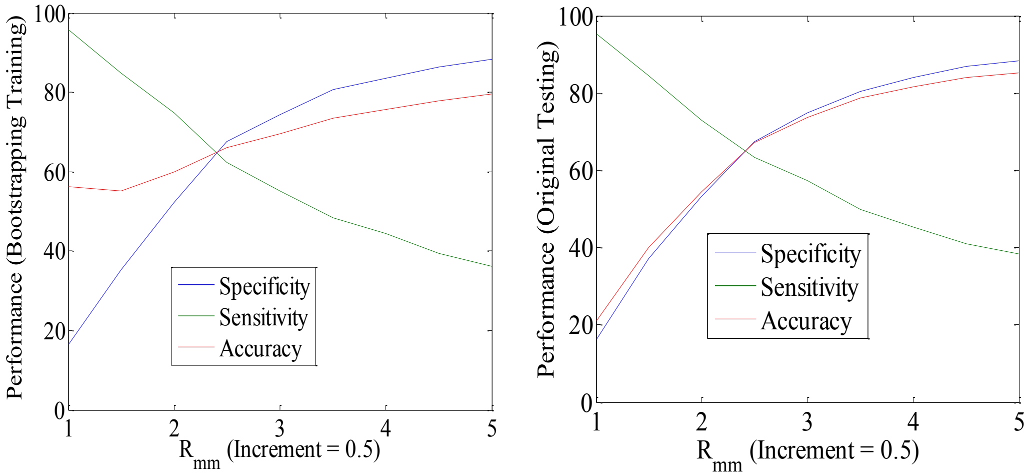
Initialization: k = 0, Set R_{mm}= R_{min}
Bootstrapping:
Generate Resampled data for R_{mm}
ANFIS training with p clusters
RMSE calculation & *p* increment
$p > p_{max}$?
No
Yes
Find p for R$_{MM}$:* R_{MM} increment
No
$R_{mm} > R_{max}$?
Yes
Choose R_{MM}* & *ANFIS**

Figure 4.20: Iterative bootstrapping-ANFIS optimization scheme for P & R_{mm}

4.6 ANFIS Model Interpretation and Performance Evaluation

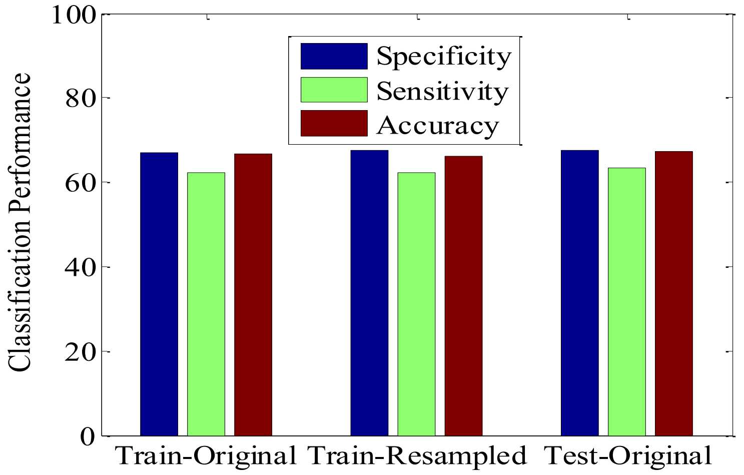
The pre-processed rails-cast data (40 inputs) need to be divided into two data sets before applying the iterative bootstrapping-ANFIS optimization scheme shown in Figure 4.20. 70 % of data samples are used as training set and 30 % are used as testing set. For a better classification performance, the two fundamental parameters mentioned earlier are specified as follow: the majority to minority case data ratio R_{mm} is chosen to be within the range of $[R_{min}{\sim}R_{max}]$ = [1, 5] with an increment of 0.5 at every step length, and the number of fuzzy rules P, known also as prototype, is selected as $[P_{min}, P_{max}]$ = [2, 10] with an increment step of 1. In addition, the number of epochs is 100 (Yang *et al*. 2011 a).

The ultimate classification performance of the bootstrapping-ANFIS scheme is influenced by the variation of data ratio R_{mm}. Because only using the training data set is used for bootstrapping resampling, the performance will be biased towards the majority class which has more dominance if a large R_{mm} arises, i.e., specificity. In contrast, the performance will be biased towards the minority class as it has a reduced dominance if R_{mm} is small, i.e., sensitivity (Yang *et al*. 2011 a). Figure (4.21) shows the classification performance of bootstrapping-ANFIS algorithm where the optimal R_{mm} value is approximately 2.5.



(a) R_{mm} for oversampled training data

(b) R_{mm} for original testing data



(c) Classification performance for R_{mm} = 2.5

Figure 4.21: classification performance of bootstrapping-ANFIS algorithm

4.7 Algorithm Discussion

Learning from the misclassified points (minority class examples) can reveal a better understanding of classifier's performance on imbalanced dataset. As a result, the confusion matrix presented in Table 4.5 is used as a measure of performance evaluation of estimators. It allows performance visualization of ANFIS algorithm by means of assessing the success rate of majority and minority classes. The output of the optimised ANFIS model $y = \{0, 1, 2, 3\}$ is converted to binary numbers $\{0, 1\}$, where the minority class (Rejected rails) is represented by 1 and the majority class (good rails) is represented by 0.

Specificity, Sensitivity and accuracy are adopted as binary classification performance evaluators, where specificity is the proportion of the negative classes that are correctly classified, sensitivity is the proportion of the positive samples that are correctly classified and accuracy is the overall percentage of predictions that were correctly classified (Yang *et al.* 2011 a). Figure 4.21 reveals a good generalization performance of ANFIS model when using a balanced data set. On the other hand, the success rate of classifying the rejected rails (sensitivity) is very poor with only 10 % of the testing data and 16 % of the training data when applying the ANFIS algorithm on the original rails data set (Imbalanced data). Figure 4.22 illustrates the classification performance of ANFIS model using the original rails data without balancing.

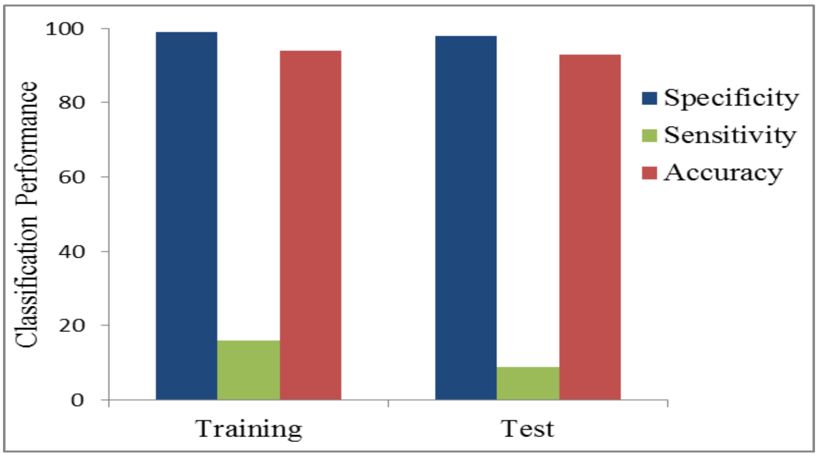


Figure 4.22: ANFIS classification performance using the original rails data (imbalanced)

In conclusion, Figures 4.21 and 4.22 illustrate clearly the impact of the class imbalance problem on the prediction performance of classification algorithms. Class imbalances hinder the learning performance of standard classifiers and modelling algorithms. Learning from the original rails data without balancing has led to a poor classification performance at around than 10 % of the testing data and 16 % of the training data when applying the ANFIS algorithm Figure 4.22. Such a low success rate, of classifying the rejected rails, was found due to the class imbalance of training data. However, the bootstrapping-ANFIS algorithm yields better prediction performance for both the majority and minority classes when using balanced data set with 65 %. Figure 4.21c illustrates the sensitivity, specificity and accuracy performances for the ANFIS model with the re-sampled rails data set corresponding to R_{mm} = 2.5.

4.8 Summary

This chapter has mainly focused on two concepts as follows:

1. Handling the complexity of real industrial rails data provided by Tata Steel Europe via applying data exploratory data analysis (data pre-processing) and input selection algorithm.
2. The second concept related to utilizing data mining and knowledge discovery in databases towards the construction of robust modelling frameworks i.e., RapidMiner neural networks and neuro fuzzy-based ANFIS approaches.

The next chapter will encounter another data re-sampling techniques. Moreover, different data classification approach based on SVM will also be applied for rails data classification. Finally, a comparison analysis of the proposed support vector machines for data classification will be discussed in contrast to the neural network modelling results presented in this chapter.

Chapter 5

Iterative Support Vector Machines: Rails Data Classification Framework

5.1 Introduction

Over the last few years, there has been a growing demand for analysing the properties of complex systems and reducing manufacturing costs while enhancing process yields based on the data being provided. Support vector machines have emerged as a powerful machine learning paradigm for solving classification problems in many fields, most particularly those of a complex nature (Yu 2013), (Ghaedi *et al*. 2014), (Janakiraman *et al.* 2014); this being due to the fact that support vector machines are able to reach a high generalization performance which can lead to an accurate classification, and to their ability to learn relatively quickly from large scale datasets using solid mathematical optimization methods and most significantly SVM's possibility of allowing a theoretical analysis using computational learning theory. Although there exist various approaches utilizing the "kernel trick", support vector machines-based classification algorithms are probably the most known "kernel algorithms".

SVMs, being computationally powerful techniques for binary classification, have gained much popularity in understanding the interaction and influence of input features on overall process yield (Widodo and Yang 2007). SVMs are supervised learning tools that embody the structural risk minimization principle presented by Vapnik and colleagues (Vapnik 1995), (Burges 1998).

From an algorithmic perspective, SVMs have some advantages over other classification methods as they have a solid mathematical structure, a significant overall classification precision and the ability to terminate to a global classification solutions as the hyper-planes are determined by support vectors (Batuwita and Palade 2010b). Another attractive feature of support vector machine algorithm is that they do not require much manual parameter manipulation which makes their use much easier. Despite these computationally attractive features, SVMs are known to have weaknesses when applied to imbalanced datasets where they tend to be overwhelmed by the majority class and lead to a poor classification performance on the minority class (Shao *et al.* 2014), (Batuwita and Palade 2010a). Imbalanced data correspond to data sets where there are many more examples of one class than the other class.

This chapter introduces a new iterative SVM formulation for solving the rails data binary classification problem that leads to an efficient successive classification. This framework referred to as iterative support vector machine is proposed for severely imbalanced rails data classification and is based on the incorporation of data re-sampling techniques with the SVM algorithm. Data resampling techniques, oversampling and under-sampling, are the best choice for overcoming the class imbalance problem. In this work, a successful inclusion of a unique the learning mechanism of ISVM and the class distribution advantages of resampling techniques has been achieved. Experimental results on rail data set are effective not only on machine's overall generalization performance, but also on drastically reducing the algorithm's time complexity and the number of support vectors.

The class imbalance problem could seriously detriment to the prediction performance of most classification techniques. As per their design principles, most standard learning models tend to ignore the minority class as they are overwhelmed by the majority class (Chawla *et al.* 2004), (Giles 2007), (Liu *et al.* 2006), (Garcia and He 2009). However, the most important task in any classification problem is to also correctly classify the minority class examples. The adverse implication of class imbalance learning problem is the capability of imbalanced data to remarkably compromise the overall performance of standard classifiers and modelling algorithms (Garcia and He 2009), (Liu *et al.* 2009). With the special attention devoted from academia and industry to the class imbalance problem, there are many methods tailored to overcome such a concern.

This chapter will start by demonstrating the formulation of standard SVM for pattern classification with a brief review of the past and on-going researches related to SVMs classification and the statistical learning theory, upon which the SVM is based. Secondly, the iterative SVM formulation will be presented as a maximum margin classifier that implements the principle of structural risk minimization. The structural risk minimisation principle seeks to minimise the upper bound of the generalization error. Thirdly, the development of SVMs from being a linear classifier to a non-linear maximum margin classifier will be discussed. This chapter also presents the key solutions to the class imbalance problem of rails data. Finally, the application of the proposed ISVM on rails data with bootstrapping-based over-sampling and under-sampling will be discussed.

5.2 Support Vector Machines

Support vector machines are discriminators that utilize structural risk minimization to find decision hyper-plane with a maximum margin between groupings of feature vectors. SVMs are often used to classify binary and multi-class datasets. Since their introduction by Vapnik (1995), SVMs have proven to be very effective and efficient in solving many classical problems in various application areas, mostly those of supervised learning.

SVMs are a relatively new in the field of machine learning that can lead to a number of advantages over other approaches, some of which are as follows:

1) The solution of the problem is unique (the solution is the global minimum of the corresponding classification problem).
2) The sparseness of the solution (the solution is formed by a few training examples referred to as support vectors);
3) SVMs possess a rigid theoretical structure based on the statistical learning theory (optimisation theory and structural risk minimisation);
4) The solution of SVMs has good generalisation properties;

5) SVMs are applicable to linear and non-linear problems through the use of 'Kernel trick';

6) Generally, the Radial Basis Function and the polynomial are known to be a special forms of SVMs.

SVMs are particularly suitable for binary classification, although they can also be extended to multiclass applications. The basic idea behind SVMs is to find a hyper-plane ***H*** which separates the high-dimensional data into its two classes as shown in Figure 5.1. The training set is separable if a hyper-plane can divide the given dataset into two half-spaces corresponding to the positive and negative classes. The hyper-plane that maximizes the margin (minimal distance between the positive and negative examples) is then selected as the unique SVM hypothesis. Since the given data may often not be linearly separable, the notion of a "kernel feature space" is introduced which casts the data into a higher dimensional space where the data are separable. However, SVMs are considerably slower in contrast to other learning techniques such as neural networks. Overall, SVM's are intuitive, theoretically well-founded, and successful.

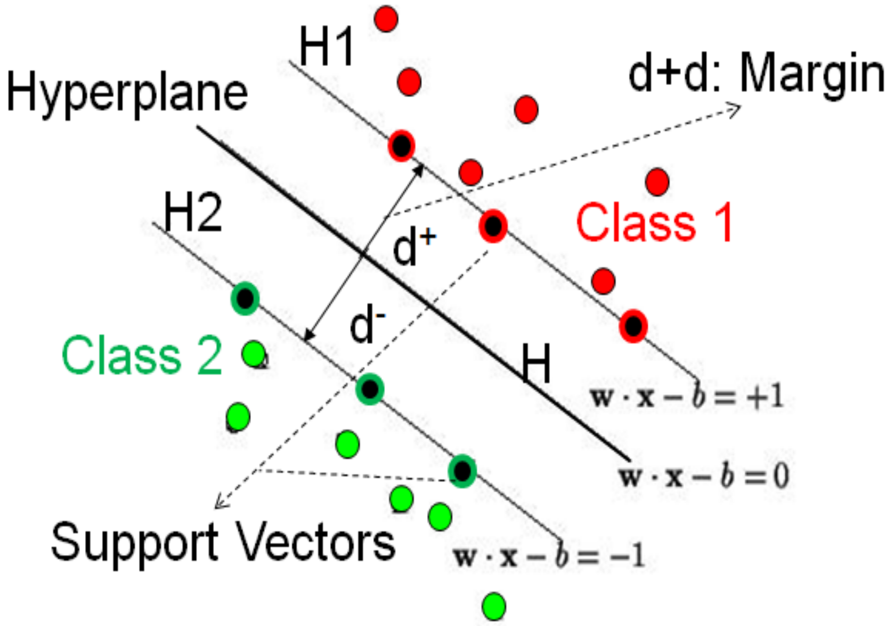


Figure 5.1: SVM separating Hyper-planes for binary classification

For a given data set S of training points that are labelled as follows:

$$(y_1, x_1), \dots, (y_i, x_i) \tag{5.1}$$

where $x_i \in R^n$ represents an n-dimensional data points, y_i represents the classes of which these data points are belonging to $y_i = \{-1, 1\}$ and $i = 1, \dots, n$. The goal of SVM is to find a hyper-plane that separates the data into two classes with as a big margin as possible. To find the hyper-planes $H1, H2$ that better separate the classes, the data records are mapped into a higher dimensional space via a mapping function φ, then the separating hyper-plane can be represented as $w \,.\, \varphi(x) + b = 0$ (Batuwita and Palade 2010b).

Previous studies from researchers on data classification focused on applying kernel techniques (support vector machines, fuzzy support vector machines) on real-world dataset such as (Vapnik 1995), (Boser *et al.* 1992), (Burges 1996) (Duan and Keerthi 2005), (Cortes C. and Vapnik V. 1995). Due to the ongoing rapid growth of data in a wide variety of real world applications, researchers have broadened the idea of SVM into various applications such as Fuzzy SVM (Lin and Wang 2002), (Lin and Wang 2004) and Lagrangian support vector machines (Mangasarian and Musicant 2001).

Reviewing past and on-going research that focuses on applying SVMs as a knowledge extraction technique is a crucial task for any researcher. Such a review can highlight the advantages and disadvantages of specific tasks as well as identifying other research paradigms to solve the problem at hand.

A new algorithm defined as Sequential Minimal Optimization (SMO) for training support vector machines is proposed by Platt (1998). The SMO approach is unlike standard Support vector machine training algorithms that require numerical solution to an extensive large quadratic programming (QP) optimization dilemma. The key insight of SMO is that it breaks the QP problem at hand in to a series of small QP problems. The obtained small QP problems are then solved using less time of QP optimization as an inner loop. In addition, no extra matrix storage is required by SMO due to the fact that this approach only solves tow Lagrange multipliers analytically. Consequently, SMO is able to handle a large training dataset.

Standard algorithms for training support vector machines require expensive computations and as result, a great number of support vectors will be produced. Downs *et al.* (2002) presented an algorithm that distinguishes unnecessary support vectors and eliminates them while maintaining the solution otherwise unaffected.

Researchers normally employ the SMO proposed by Platt (1998) to produce an initial support vectors set and then eliminate support vectors that are linearly independent in feature space. Linear, polynomial and RBF kernels functions are used for this purpose. The Results of this method reduces the number of support vectors. Nevertheless, severe reductions in generalization performance can occur when discarding even small number of support vectors (Syed *et al.* 1999).

Due to the fact that the support vector machine algorithm produces more support vectors in a large dimensional space, other researchers have been interested in designing a reduced set methods that can tackle this problem (Burges 1996), (Burges *et al.* 1997), (Schoelkopf *et al.* 1999), (Scholkopf and Smols 2001). Some researchers (DeCoste and Mazzoni 2003) applied an approximation method together with the reduced set methods to speed up the query time (optimization phase) of Support Vector Machines (Nguyen and Ho 2005).

According to Platt (1998), SVMs are considerably slower in the quadratic programming (QP) optimization phase in contrast to other methods with similar generalization performance. A new reduced set method was presented by Burges (1996) to address this problem. This method is to decrease the complexity of decision rule using SVM. It computes an approximation to the decision rule by means of a reduced set of vectors.

Lee *et al.* (2011) introduced a distance measurement technique that employs the Euclidean distance-based function to replace the optimal hyper-plane as classification decision making function in the SVMs. In this approach, when a new data sample is casted into another vector space, the average distances between the new data samples and the support vectors are measured using Euclidean distance function. In contrast to the Standard SVM where finding the optimal hyper-plane is dependent on the type of the kernel function and the value of the soft margin parameter C, the Euclidian-SVM have low impact on the implementation of kernel function and soft margin parameter C. Therefore, the issue of choosing an appropriate Kernel function and soft margin parameter C can be avoided. However, the classification phase of the proposed approach is computationally expensive when applied to a large scale dataset.

For a better classification performance, kernel selection is the main task in applying SVMs. Prajapati and Patle (2010) compare the classification performance of SVM with different types of kernel functions. Linear kernel function, polynomial and radial basis kernel function (RBF) were utilized for performing classification. For selected feature, the comparison concluded that the RBF is suitable for large scale datasets and the kernel selection has a significant impact on the overall performance accuracy.

Twin support vector machine (TWSVM) is a kernel-based paradigm presented by Jayadeva *et al.* (2007). TWSVM is a binary classifier that uses labelled data to perform its training procedure. It finds two non-parallel separating hyper-planes by solving two SVM problems, each of which is smaller than in standard SVM. However, TWSVM lead to a low performance with small number of labelled data. To tackle such a weakness, a nonparallel-planes semi-supervised classifier termed as 'Laplacian smooth twin support vector machines Lap-TSVM' is developed by Chen *et al.* (2014). The formulation of Lap-STSVM converts the optimal constrained quadratic programming problems (QPPs) of Lap-TSVM into unconstrained minimization problems (UMPs). Secondly, to make the (UMPs) twice differentiable, a smooth technique is produced. Finally, a newton-Armijo algorithm is designed to solve the UMPs. A comparison results based on real-world datasets proved that Lap-STSVM provides good generalization capability than TSVM, Lap-TSVM and Lap-SVM.

An efficient weighted lagrangian twin support vector machine (WLTSVM) is proposed by Shao *et al.* (2014) for class imbalance data classification. The concept of the proposed algorithm is based on using different training points for the construction of two separating hyper-planes. The (WLTSVM) algorithm introduces a graph-based under-sampling strategy to keep the proximity information which is robust to outliers. It also introduces a quadratic cost functions that speeds up the algorithm's computations. The algorithm has proven its efficiency and feasibility for class imbalance learning in contrast to some other twin weighted support vector machines.

An important challenge for machine learning techniques, especially SVM, is the class imbalance problem. A novel framework, referred to as second-order cone programming support vector machine (SOCP), has been developed by Maldonado and López (2014) for overcoming this concern where the assumption errors cost equality is made and each data point is treated independently. The formulation of the algorithm is based on applying a cost-sensitive learning for classifying imbalanced data via direct margin maximization. The proposed second-order cone programming support vector machine algorithm achieves better results compared to other SOCP-SVM formulations.

Tian *et al.* (2013) proposed a novel classifier that is completely different from the existing non-parallel classifier such as generalized eigenvalue proximal support vector machine (GEPSVM) and the twin support vector machines (TSVM) (Jayadeva *et al.* 2007). The non-parallel support vector machine (NPSVM) classifier possesses several advantages over other classifiers such as (GEPSVM) and (TSVM). (NPSVM) implements the structural risk minimization principle and two primal problems are created. The formulation of the dual optimization problem is as same as that of SVM which can be solved efficiently by Sequential minimization problem. Finally, (NPSVM) has it's inherit sparseness as standard SVM. Despite the above features, the proposed algorithm has five parameters need be carefully tunes and thus is computationally expensive. Moreover, the proposed algorithm cannot be extended in a straight forward manner to multi-class classification problem.

5.2.1 Statistical Learning Theory

5.2.1.1 Binary Classification Problem

The SVMs paradigm seeks to solve a binary classification problem. However, in practice, it can be extended to cover multi-class problems. Simple two-class classification problem can be represented as follow:

For a given data set S of labelled training points $(y_1, x_1), \dots, (y_l, x_l)$, each sample of the training set x_i is belonging to $y_i \in \{-1, +1\}$. The aim is to find the classifier with decision function $f(x)$. The overall classification performance is measured based on a classification error represented as follows:

$$error(f(x), y) = f(x) = \begin{cases} 0, & \text{if } f(x) = y \\ 1, & \text{otherwise} \end{cases} \tag{5.2}$$

5.2.1.2 Empirical Risk Minimization and Structural Risk Minimization

For a binary classification problem, the SVM learning algorithm with a set of adjustable parameter $\boldsymbol{\alpha}$ seeks to find $\boldsymbol{\alpha}$ so it learns the mapping $x \mapsto y$. This will yield a possible mapping $x \mapsto f(x, \boldsymbol{\alpha})$ that defines the algorithm. The algorithm's performance obtained via the empirical risk error as follows:

$$R_{emp}(\boldsymbol{\alpha}) = \frac{1}{N}\sum_{i=1}^{N} error(f(x_i , \boldsymbol{\alpha}), y_i) \qquad (5.3)$$

where $\boldsymbol{\alpha}$ is the set of adjustable parameter and N is the training set size. The risk minimisation principle is referred to as the empirical risk minimisation (ERM) which is implemented by various machine learning techniques. However, over-fitting may occur if the complexity of the algorithm is high. The principle of the minimisation does not take in to account the algorithm complexity. The empirical risk minimization principle guarantees the existence of a solution assuming the lose function is continuous. However, this condition is not always satisfied.

Empirical risk minimization (ERM) principle does not however guarantee the stability and uniqueness of the solution. Practically, it is advised to utilize prior information to decide which solution from within the correspondence class of functions is well-matched for minimal empirical risk. Another known minimisation principle that considers the complexity of the learning algorithm is the structural risk minimisation (SRM), the expected risk can be minimised as follows:

$$R_{exp}(\boldsymbol{\alpha}) = \int error(f(x_i , \boldsymbol{\alpha}), y_i) dP(x, y) \qquad (5.4)$$

The term $P(x, y)$ is the prior probability. In some cases, the risk cannot be explicitly obtained since the prior probability is unknown.

5.2.2 Multi-Class Classification Problem

As it has already been mentioned, SVMs have been designed to seek binary classification and they have been extended to perform multi-class classifications. However, a multi-class classification is not straight forward. Accordingly, extending the binary classification problem to handle a multi-class classification problem is performed by various techniques such as one-against-one (OAO), one-against-all (OAA), One-against-higher order and Directed Acyclic Graph SVM (DAGSVM). Due to the fact that many real-world classification problems involve multi-classes, these techniques have shown remarkable success in mapping the generalisation abilities of binary classification to multi-class domain.

Numerous schemes proposed by Evgeniou *et al.* (2000); Guermeur (2002); Guyon *et al.* (1993); Hsu and Lin (2002) to solve the K-class pattern recognition problems. One-against-one introduced by Knerr *et al.* (1990) is the most popular and successful multi-class SVM technique. The first application of this method is introduced by Friedman (1996). One-against-one creates a binary SVM for each combination of target label and every unseen data record are classified to its side of the decision boundary. As each binary classification assigns one example to one of the two classes, this method is also known as a 'voting scheme'. Splitting the multi-class problem to a multiple binary sub-problem has the advantages that different decision boundaries are created for each class pair. However, this possibly leads to a very complex decision boundary.

Another popular multi-classification scheme is one-against-all in which K binary SVM is built (Hsu and Lin 2002). It separates one class c_i from the rest by building a decision boundary in each attempt of building the model. The model is created by assigning label (+1) to c_i and the target label (-1) to the rest of remaining classes. The advantage of this method is that it only needs to contrast the K models. However, the involvement of all classes in every SVM can be computationally expensive. Moreover, in high dimensional datasets, it is sometimes challenging to separate one class from the rest.

The third known method of SVMs multi-classification is Directed Acyclic Graph SVM (DAGSVM) proposed by Platt *et al.* (2000). The training phase of this technique is the same as one-against-one by creating $K(K-1)/2$ binary classifier. However, in the training phase, it distinguishes itself as it employs a rooted binary tree graph with $K(K-1)/2$ internal nodes and K leaves. Every node in the tree structure is binary SVM of i th and j th classes. The advantage of (DAGSVM) is that the unseen examples can be classified using $K-1$ evaluations (Hsu and Lin 2002).

5.2.3 Linear Support Vector Machines Classifier

SVMs inherently related to the family of Linear Machine learning because their aim is to search the optimal (maximum margins) hyper-plane. Two cases of a linear classifier can be considered i.e., the separable case where a perfect mapping can be achieved and the non-separable case where a perfect mapping is unattainable.

5.2.3.1 Separable Case

For a binary classification problem, consider data set S of labelled training points $(y_1, x_1), \ldots, (y_l, x_l)$, each sample of the training set x_i is belonging to $y_i \in \{-1, +1\}$. SVM aims to find the separating hyper-plane that separates the positive class examples (+1 labels or red points) from the negative class examples (-1 labels or blue circles) with margin maximisation as shown in Figure 5.2.

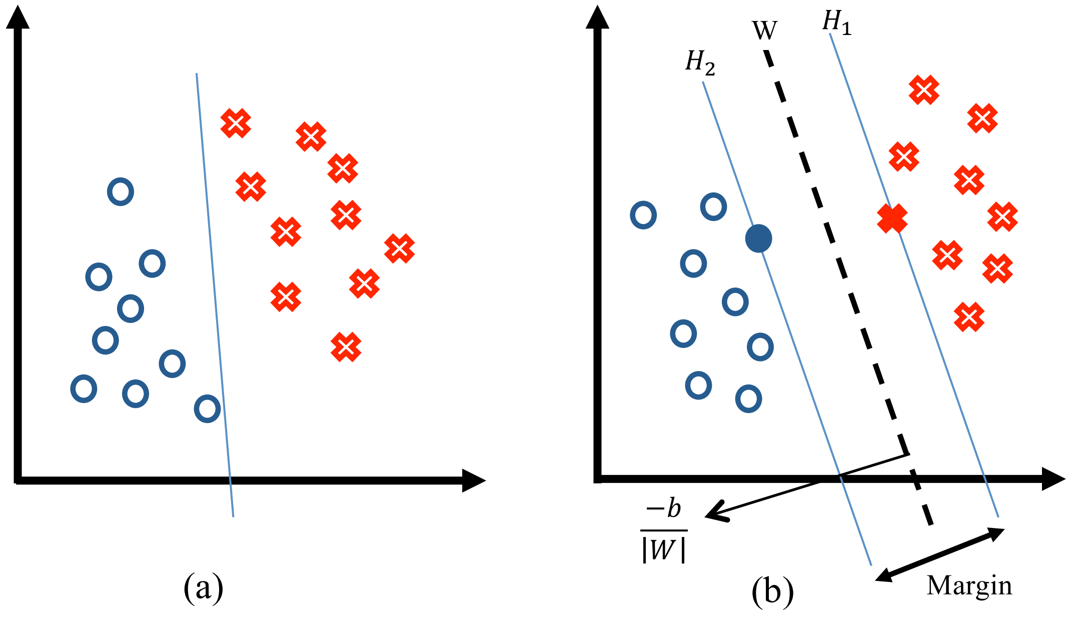


Figure 5.2: SVM Margin Maximisation for binary classification

Left figure (a) shows a separating hyper-plane (Solid Line) where there is no margin between the two classes in left. Where in figure (b), the maximum margin (the space between two solid lines) is resulted and the best generalization is reached via support vectors i.e., the two red and blue points lay on the separating hyper-planes (H_1, H_2).

The key aspect behind searching a separating hyper-plane with margin maximisation is that the hyper-plane with largest margins is more robust, resistant to noise and provide good generalization abilities to that with smaller margins. Mapping that separates the positive classes (+1 labels or red points) from the negative classes (-1 labels or blue circles) is achieved as follows:

$$f(\mathrm{x},\mathrm{y}) = sign\ (\mathrm{w}.\mathrm{x} + b) \tag{5.5}$$

where w is a weight vector and b is the bias value (offset from origin)

Having achieved mapping, the hyper-plane is of the form:

$$\mathrm{w}.\mathrm{x} + b = 0 \tag{5.6}$$

A successful linearly separating hyper-plane between two datasets is achieved if the pair {w, b} is chosen such that the mapping in equation (5.5) is optimal. Figure 5.2 shows a clear separation between two classes.

5.2.3.2 Separating Hyper-Plane with a Maximum Margin

The creation of separating hyper-plane significantly depends on the value of {w, b} given in equation (5.6). SVMs classifier seek out the separating hyper-plane that best maximise the margin between two classes as shown in Figure 5.2 (b). The binary classification problem is said to be optimally separated when such a boundary is reached. In the linear separation, the data satisfy the following constraints:

$$\mathrm{w}.\mathrm{x}_i + b \geq +1 \qquad y_i = +1 \tag{5.7}$$

$$\mathrm{w}.\mathrm{x}_i + b \leq -1 \qquad y_i = -1 \tag{5.8}$$

A convenient compact representation of the above constraints is as follows:

$$y_i(\mathrm{w}.\mathrm{x}_i + b) \geq +1 \qquad \forall_i \tag{5.9}$$

Equation 5.9 holds for the training examples that lie on the canonical hyper-planes (H_1 ,H_2) shown in Figure 5.2. The margin p can be calculated as the distance between H_1 and H_2:

$$p = \frac{|1-b|}{||w||} - \frac{|-1-b|}{||w||} = \frac{2}{||w||} \tag{5.10}$$

H_1 and H_2 are parallel and share the same normal w where doted data points fall on the two hyper-planes. The optimal maximum margin hyper-plane that separates that data is the one that minimises $||w||^2$ subject to the constraints in following optimization problem:

$$Min \ \frac{1}{2} \ ||w||^2 \quad \text{s.t} \ \ y_i \ (w \, . x_i + b) \geq 1, \ \ \forall_i \qquad (5.11)$$

The solution to equation (5.11) is independent of the bias value b. The optimal hyper-plane will be moved to the direction of w for any change in the value of b and the maximum margin.

5.2.3.3 Lagrangian Formulation

With a set of inequality constraints, solving the minimisation problem of $||w||^2$ is performed using Langrangian multipliers. Langrangian multipliers are effectively well known solutions to such problem for two reasons:

- handling the constraints will be easier;
- The training data will only appears in a dot product form between vectors. The Langrangian multipliers α_i for each constraint are produced and the structure of the minimisation problem given in equation (5.11) becomes:

$$Min_{w,b} \ L(w,b,\alpha) \equiv \frac{1}{2} \ ||w||^2 - \textstyle\sum_{i=1}^{N} \alpha_i y_i (x_i \, . w - b) + \sum_{i=1}^{N} \alpha_i \qquad (5.12)$$

Subject to the constraints:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \qquad (5.13)$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \qquad (5.14)$$

The above formulation is convex quadratic programming problem since the objective function L is convex. Since the constraints are equal, the dual formulation is the result of substituting the inequality constraints into the objective function. The dual formulation becomes:

$$\text{Max} \ L_D = \textstyle\sum_{i=1}^{\alpha_i} - \frac{1}{2} \sum_{ij}^{N} \alpha_i \alpha_j y_i y_j x_i x_j \qquad (5.15)$$

Subject to:

$$\sum_{i=1}^{N} \alpha_i y_i = 0\,, \qquad \alpha_i \geq 0 \tag{5.16}$$

The training of SVMs is a problem of maximising equation (5.15) with respect to α_i, subject to constraint in equation (5.16) with a positive Lagrangian multiplier $\alpha_i \geq 0$. The optimal hyper-plane and the bias are given as:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{5.17}$$

$$b = \frac{1}{2}\, w(x_+ + x_-) \tag{5.18}$$

The optimal solution is in the form of a linear combination of $x_i s$, where the Lagrangian multiplier $\alpha_i = 0$ for every x_i excluding the once that lie on the hyper-planes H_1 and H_2 where $\alpha_i \geq 0$. The points that lie on the hyper-planes in the classification problem are called Support vectors. In data classification problems, the number of support vectors is normally much less than the number of the training data. However, in the classification problems of large scale datasets, a large number of support vectors are more likely to be produced. An interesting feature of SVMs is that the quadratic programming is a convex problem where there are no local minima. SVMs always find global minima and the solution is optimal.

5.2.3.4 The Karush-Kuhn-Tucker Condition

The Karush-Kuhn-Tucker (Kuhn and Tucker 1951) condition establishes the requirements needed to achieve an optimal solution to the SVM optimization problem. Based on KKT condition, the solutions to w, b and α for the primal problem in equation (5.12) should satisfy the condition:

$$\frac{\partial L(w^*,b^*,\alpha^*)}{\partial w} = w_v - \sum \alpha_i y_i x_{iv} = 0 \qquad v = 1, \ldots, d \tag{5.19}$$

$$\frac{\partial L(w^*,b^*,\alpha^*)}{\partial w} = \sum \alpha_i y_i = 0 \tag{5.20}$$

$$y_i(x_i\,.w + b) - 1 \geq 0, \quad , \ \forall_i \tag{5.21}$$

$$\alpha_i \geq 0 \quad \forall_i \tag{5.22}$$

$$\alpha_i(y_i(x_i\,.w+b)-1\ \geq 0,\quad ,\ \forall_i \tag{5.23}$$

The SVM problem is a convex optimization problem and the above KKT conditions are important and appropriate for w^*, b^*, α^* to be a solution. Consequently, the solution of SVM problem is equivalent to finding a solution to KKT conditions. Practically, the first KKT expresses the optimal separating hyper-plane as a linear combination of the vectors in the given training dataset where:

$$w^* = \sum_{i=1}^{N} \alpha_i{}^* y_i x_i \tag{5.24}$$

Whereas the second KKT condition requires that α_i coefficients of the training examples should satisfy:

$$\sum_{i=1}^{N} \alpha_i{}^* y_i = 0\,, \qquad \alpha_i{}^* \geq 0 \tag{5.25}$$

5.2.3.5 The Linearly Non-Separable Case

In the previous sections, the idea of linearly separating the training examples via support vector machines SVMs has been discussed. This formulation is restricted to the problem where the data is linearly separable. However, in many real world classification problems, large scale datasets do not satisfy this condition where it is inherently nonlinear. The above formulation must be extended to tackle the problem of non-linearity and thus the separating hyper-planes can be found.

Data non-linearity can be handled via the creation of the objective function that trades off misclassification against minimizing $||w||^2$. Misclassifications are tackled by introducing the slack variable $\xi \geq 0$ for every data point. The constraints with the applied slack variable become as follows:

$$\mathrm{w.x}_i + b \geq +1 - \xi \qquad for \quad y_i = +1 \tag{5.26}$$

$$\mathrm{w.x}_i + b \leq -1 + \xi \qquad for \quad y_i = -1 \tag{5.27}$$

The old constraints in equation (5.7) and equation (5.8) can be violated via introducing the slack variable, such violation causes a penalty referred to as C. The new problem is the summation of misclassification errors and minimizing $||w||^2$:

$$\|w\|^2 + C\left(\sum_i \xi_i\right) \tag{5.28}$$

C is the misclassification penalty or the regularization parameter applied to adjust the relation between $\|w\|^2$ and the slack variable. The idea of non-separable case is shown in Figure 5.3:

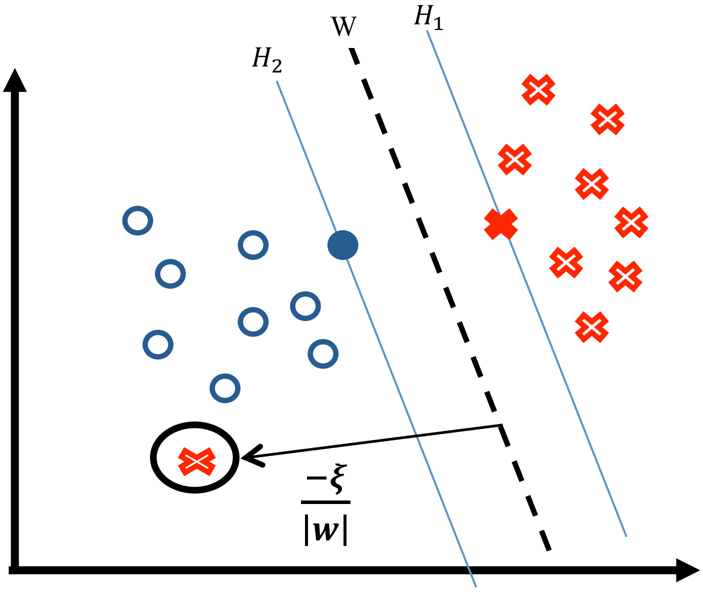


Figure 5.3: SVM non-separable case. The encircled data point is misclassified and thus has appositive ξ.

As it is shown in the separable case, the solution has the following expansion:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{5.29}$$

where the training data examples that satisfy $\alpha_i \geq 0$ are referred to as Support vectors. The penalty function belongs to the slack variables is linear, which will not be exist in the dual Lagrangian problem. The objective problem of the dual formulation is as follows:

$$\text{Max } L_D = \sum_{i=1}^{\alpha_i} - \frac{1}{2}\sum_{ij}^{N} \alpha_i \alpha_j y_i y_j x_i x_j \tag{5.30}$$

Subject to the following:

$$\sum_{i=1}^{N} \alpha_i y_i = 0\,, \qquad C \geq \alpha_i \geq 0 \tag{5.31}$$

The dual formulation for non-separable case is much similar to the linear separable case, the only difference is the upper bound of C with respect to the coefficient α_i. The objective problem converges to the linearly separable case when the misclassification penalty $C \longrightarrow \infty$.

5.2.4 Non-Linear Support Vector Machines Classifier

In the previous section, the concept of linear support vector machine has been discussed, and how such technique could handle the misclassified data points. The idea of SVM can be extended to effectively solve real-world problems i.e., the classification of non-linear decision boundaries. The linearity restrictions can be handled in a way that SVM maps the input space to a higher-dimensional feature space (perhaps infinite-dimensional) via some mapping function $\varphi(x)$ where the data is separable. An attractive feature of SVMs is that mapping the data from the input space to the new feature space is not necessary to be known, but only the kernel mapping, that maps the inner products of the input space to inner products of the feature space. This method was proposed by (Guyon *et al.* 1993).

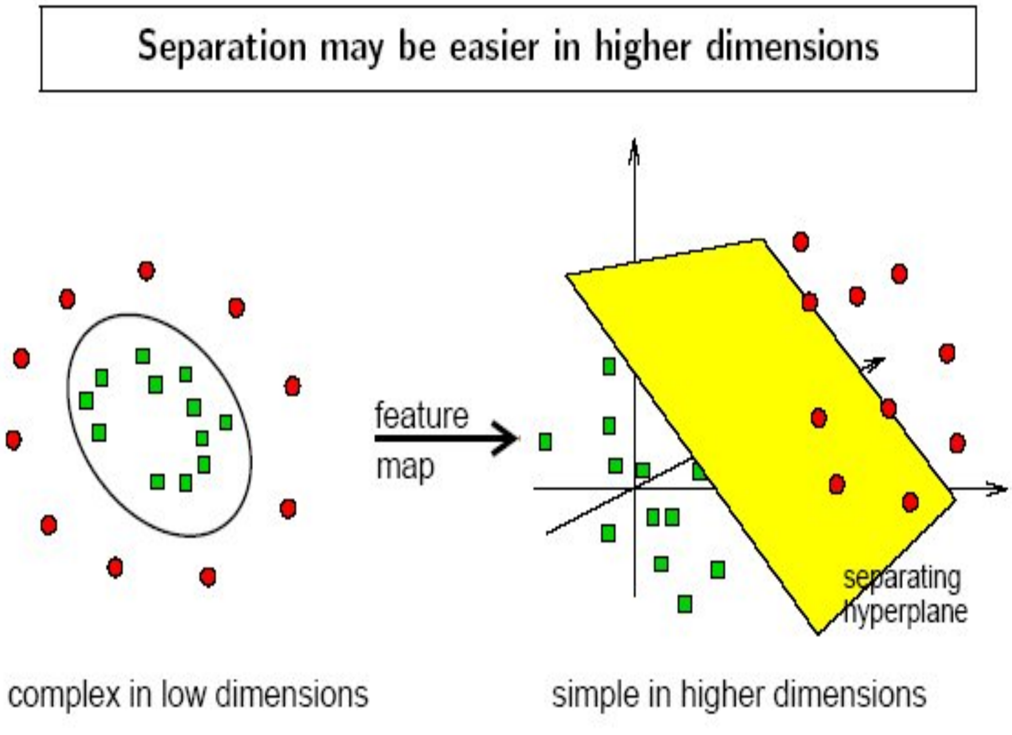

Separation may be easier in higher dimensions
feature
map
$\varphi(x)$
separating
hyperplane
complex in low dimensions
simple in higher dimensions

Figure 5.4: The data is mapped from input-space to a higher feature-space by the mapping function φ. Complex overlapped data in low dimension space (left figure) is mapped via the kernels to a simple higher dimensional space (right figure) where the data is separable (Prajapati and Patle 2010).

As shown in Figure 5.4 (Prajapati and Patle 2010), with the use of the kernel function $K(x, y) = (\varphi(x), \varphi(y))$ presented by Guyon *et al.* (1993) , the separating hyper-plane can be easily obtained. The kernel function can better correspond to new feature space in the case when it is symmetric as follow: $K(x,y) = (\varphi(x), \varphi(y)) = (\varphi(y), \varphi(x) = K(y, x)$. In practice, the choice of kernel needs to satisfy Mercer's theory where the kernel matrix $K = (K(x_i, x_j))_{ij=1}^{n}$ should be positive semi-definite (it has non-negative eigenvalues).

Although many kernels exist, the kernel choice only applies to the ones that satisfy these conditions:

- The Kernel must satisfy Mercer' theorem (Burges 1998).
- The kernel must be symmetric.

These conditions are described in details in Appendix A.

The commonly used kernel functions include:

1) The polynomial kernel:

$$K(x, y) = ((x, y))^d \tag{5.32}$$

2) The Gaussian kernel (Radial Basis Function):

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2/2\sigma^2} \tag{5.33}$$

3) The linear Kernel:

$$K(x, y) = x_i . x_j \tag{5.34}$$

4) Sigmoid Kernel Function:

$$K(x, y) = \tanh(\gamma(x, y) - \theta) \tag{5.35}$$

The complexity of a given training set can significantly affect the performance of learning algorithms that make use of it. Practically, certain types of learning algorithms might not be able to learn a suitable prediction function for the training dataset. In such case, one has no choice but to manipulate the data where the learning become possible. In other cases, the learning algorithm is violated because of the structure of the data and as result mapping becomes essential.

5.3 Class Imbalance Learning Key Solutions (Rails Data Resampling)

Early work investigating the rail data via an iterative SVM classification strategy with various options and key parameters built in the iterative strategy have led to a successful classification performance of only 25% of rails data. This is to be expected, since the original rail data is severely imbalanced. Imbalanced data refers to any data that exhibits unequal class distribution between its classes (Garcia and He 2009). Figure 5.5 illustrates clearly the imbalance nature of the rails dataset.

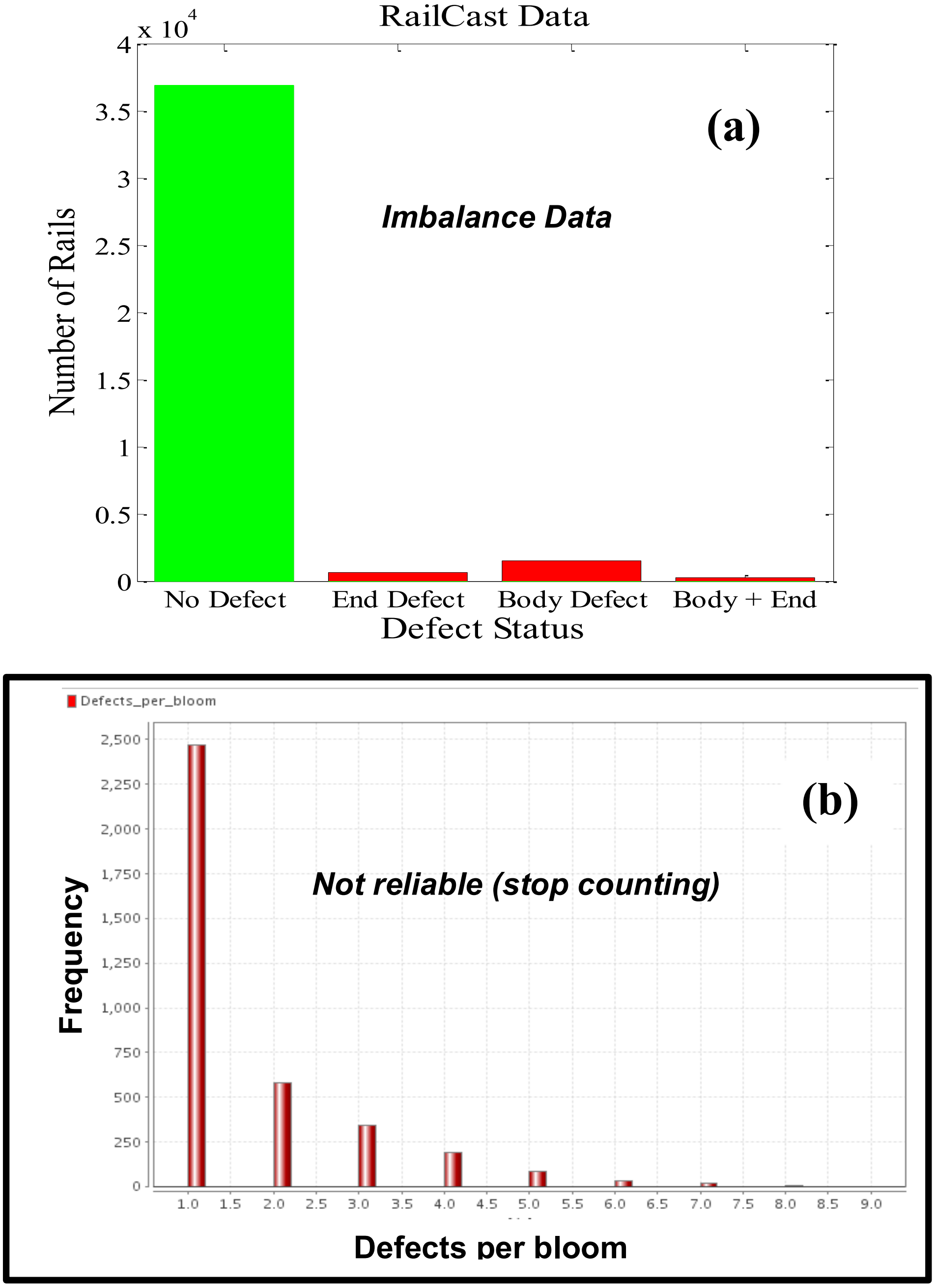


Figure 5.5: Class distributions of rails dataset

As per their design principles, most standard learning models tend to ignore the minority class as they are overwhelmed by the majority class (Chawla *et al.* 2004); (Giles 2007); (Liu *et al.* 2006); (Garcia and He 2009). However, the most important task in any classification problem is to correctly classify the minority class examples. The adverse implication of class imbalance learning problem is the capability of imbalanced data to remarkably compromise the overall performance of standard classifiers and modelling algorithms (Garcia and He 2009); (Liu *et al.* 2009).

With the great attention devoted from academia and industry to the class imbalance problem, there are many methods tailored to overcome such concern. These methods can be categorized into three categories as follows: data level techniques (external methods) in which the data is pre-processed and rebalanced before applying the classifier (Chawla *et al.* 2002); (Estabrooks *et al*. 2004); (Liu *et al*. 2009). The internal or algorithm level approaches only modify the structure of the algorithm so to pay extra attention minority class (Tang *et al.* 2009); (Garcia and He 2009); (Hwang *et al.* 2011). Cost sensitive methods incorporate various classification costs for each class by combing both algorithm and data level methods (Liu *et al.* 2006); (Freitas *et al.* 2007); (Chawla *et al.* 2008).

Breiman (1996) proposed the concept of bootstrapping aggregation for ensembles construction. This method generates multiple types of predictors and employing it to obtain an aggregated predictor (Li 2007); (Galar *et al.* 2012). These multiple types are formed by replicating the training set using bootstrap and using these as new training set.

Chawla *et al.* (2002) proposed a synthetic minority oversampling technique (SMOTE). SMOTE is able to generate a new synthetic minority examples via combining minority examples that lie together. The oversampling technique presented is sophisticated (Estabrooks *et al.* 2004), but the authors did not consider different class distribution ratios. Several sampling methodologies with different class distributions were evaluated by (Batista *et al*. 2004). Different data over-resampling and under-sampling techniques included SMOTE, TOMEK and SMOTE+ENN were examined. SMOTE result in a good performance for databases with a small number of majority class examples.

A variation of SMOTE, named generative over-sampling, was introduced by Liu *et al.* (2007). The generative over-sampling creates new data examples by learning form the existed training points. The new approach selects the probability distribution to model the minority class examples and then, from this model, the new data points are generated. Generative over-sampling is effective only with large number of minority class examples. Whereas, the probability distribution estimates may not be accurate when applying to small minority examples.

Others claim that wrong synthetic minority examples may be produced using most of the existing over-sampling approaches (Barua *et al.* 2014). Consequently, a new approach termed as majority weighted minority oversampling technique (MWMOTE) is proposed by Barua *et al.* (2014) to effectively handling the class imbalance problem. MWMOTE detects minority class examples and assigns them weights based on their euclidean distance from the nearest majority class examples. A clustering method will then be applied to generate the synthetic examples for the weighted minority class examples. The proposed method shows its effectiveness in in terms of performance metrics.

In this research, only the prominent sampling techniques which can be effectively used to train SVM on large imbalanced data are considered. Under-sampling and over-sampling are common sampling techniques that have proven their usefulness in achieving an optimal sampling rate and therefor aiding classifiers yielding a better generalization capability (Batuwita and Palade 2010b), (Garcia and He 2009), (Akbani *et al.* 2004), (Hwang *et al.* 2011). Data under-sampling and bootstrapping-based over-sampling techniques will briefly be discussed in next section.

5.3.1 Under-Sampling

Under-sampling is an external independent sampling method that can straightforwardly re-balance the training data before training the classifier. In random under-sampling, the training dataset is rebalanced by randomly removing majority class examples until a desired class ratio R_{mm} between the majority and minority class is achieved. The design parameter R_{mm} controls the imbalance level for the resampled training dataset. R_{mm} is the ratio of the number of examples related to the majority class to that related to the minority class. Despite its simplicity, Under-sampling approach has proved to be significantly efficient since the rails data are highly dimensional data. However, under-sampling approach has been reported (Hwang *et al.* 2011), (Akbani *et al.* 2004), (Liu *et al.* 2006), (Chawla *et al.* 2002) to discard potentially useful data and some crucial information might be lost. It thus could dramatically disturb the decision boundary of the classifier. The significant R_{mm} achieved via under-sampling the rails data is a value of 1. Figure 5.6 illustrates the influence of under-sampling on rails dataset.

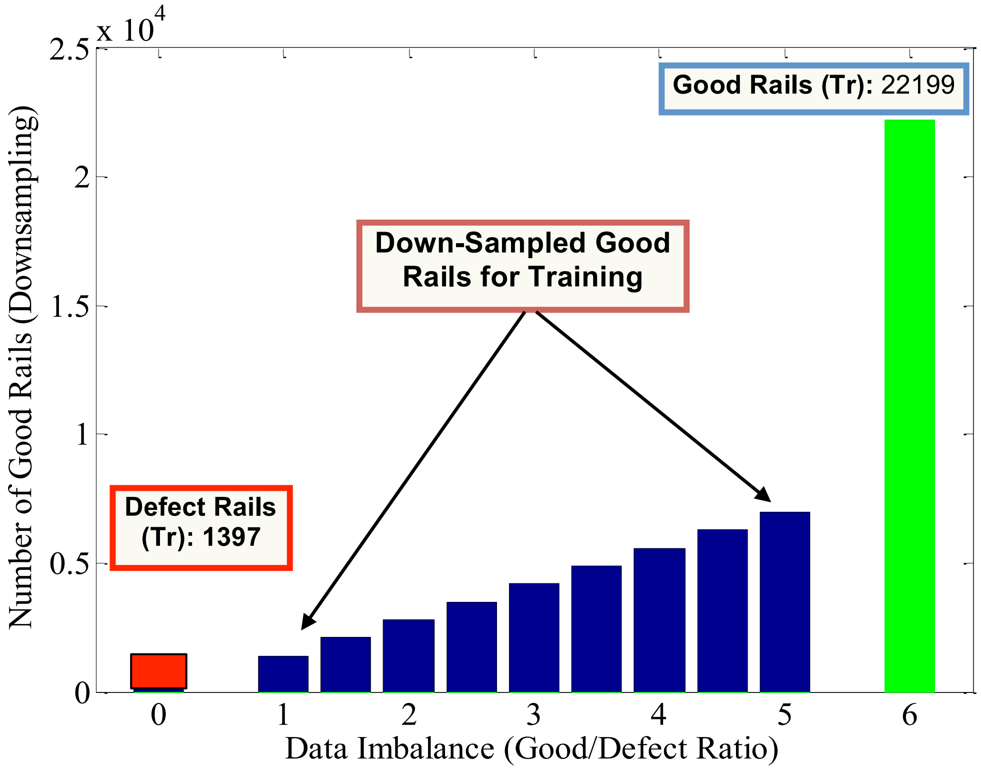


Figure 5.6: the influence of undersampling on rail dataset

5.3.2 Bootstrapping-Based Over-Sampling

Bootstrapping based-oversampling is another sampling technique that is also applied to change the class distribution of rails data. The original rails data are firstly segregated into two subsets, one is for the dominating majority class and the other is for the minority class. The minority class data is subsequently fed to the bootstrapping resampling algorithm which yields a multiple randomly resampled datasets each with the same size as the size of the original minority subset. The resampled subsets are combined with the oversampled minority class data and then mixed with the majority class data to form the final resampled data that is ready for subsequent training measures. Due to the dimensionality nature of the rail data, the optimal sampling rate R_{mm} achieved via bootstrapping based oversampling approach is 5. The effect of bootstrapping-based over-sampling on the Training Data is shown in Figure 5.7.

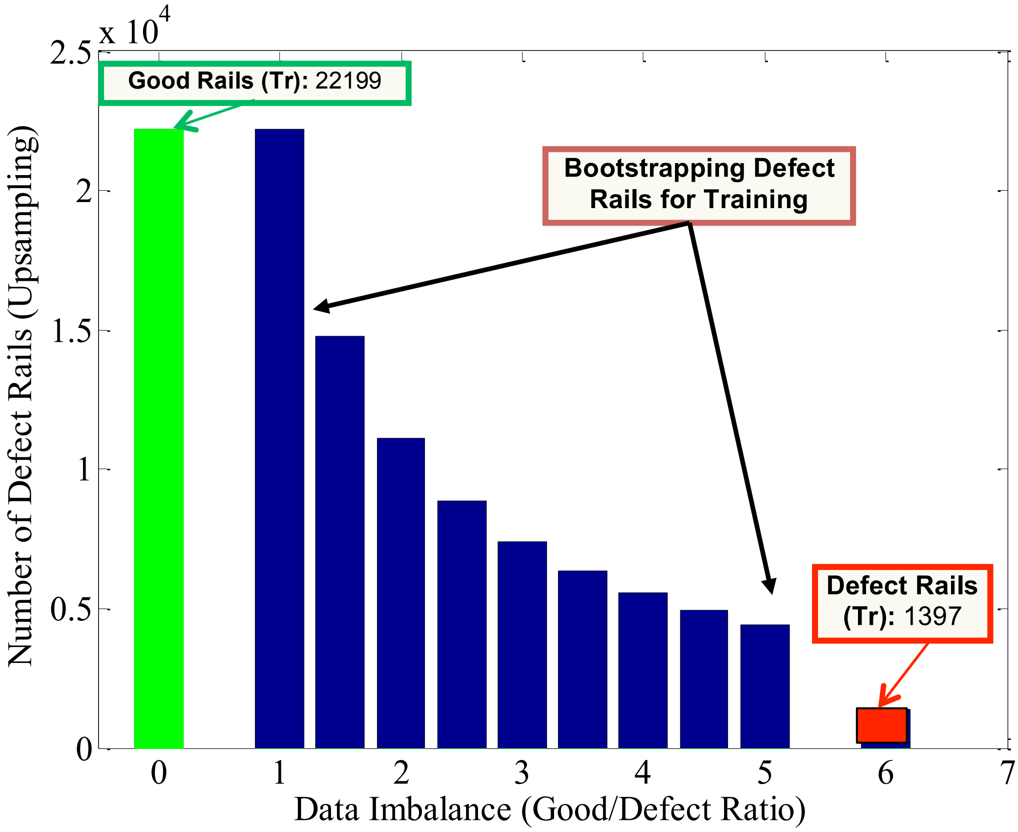


Figure 5.7: The effect of bootstrapping-based over-sampling on the Training Data

The advantage of this technique is that no information is lost since all instances are employed. However, by creating additional examples, oversampling leads to a high computational cost. Thus, a sufficient amount of memory is required to hold the whole training set. Moreover, randomly replicating the data might contain erroneous values which could negatively impact learning performance (Chawla *et al.* 2002).

Practically, Applying only bootstrapping is not enough and oversampling has to be done. The reason behind such integration scheme is that bootstrapping utilizes the same examples as the original minority class data which could lead to over fitting. Moreover, the classifier might lose its generality. Consequently, an integrated bootstrapping-based over-sampling framework is designed to increase the overall classification performance. Figure 5.8 shows the influence of under-sampling and oversampling strategies on rails data.

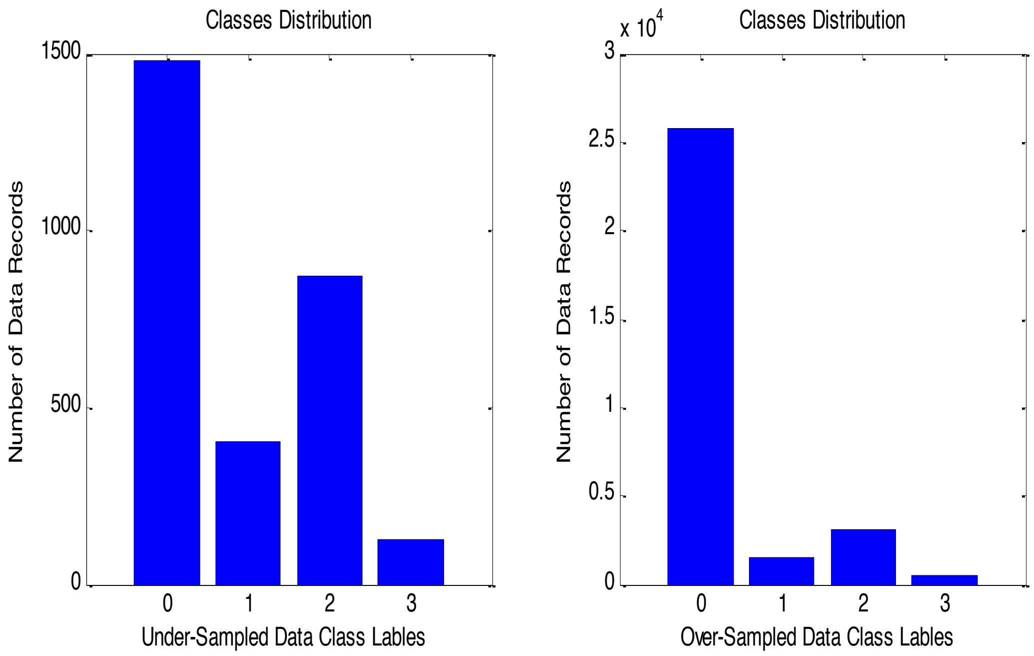


Figure 5.8: R_{mm} Influence on overall size of rails training data

5.4 Iterative Support Vector Machines Algorithm (ISVMs)

Due to the complexity of the Rails manufacturing process and unforeseen disturbances, such as variations in process operating conditions, changes in raw material properties and equipment faults, Rails product quality could be extremely different from specification. In order to enhance the final product quality, modelling and classifying the final quality of Rails is inevitably needed to provide process operators with information on the product quality and direction of intervention for quality assurance. In this section, a new iterative Support vector machine algorithm is proposed for rails data classification.

Based on the statistical learning theory (Vapnik 1995), SVM application to pattern classification is a powerful supervised machine learning technique that finds the optimal hyper-plane by separating the high dimensional data into its two classes. In this section, the theory of SVMs in classification problems is briefly reviewed. Given a set of labeled instances $\{(x_1, y_1), (x_2, y_2) \ldots., (x_l, y_l)\}$, each x_i has a class label $y_i \in \{-1,1\}$ which denotes two classes separately. For binary classification problems, SVM model builds an optimum hyper-plane that better separates the classes by margin maximization. Such a hyper-plane can be found by minimizing the following objective function (Giles 2007):

$$\min\left(\frac{1}{2}w^{T}.w+C\sum_{i=1}^{l}\mathcal{E}_i\right) \tag{5.36}$$

$$S.t.\ y_i(w^T.\Phi(x_i)+b)\geq 1-\mathcal{E}_i,\mathcal{E}_i\geq 0\quad i=1,\dots,l \tag{5.37}$$

where $'w'$ and $'b'$ are the norm of the hyper-plane and the bias respectively. Φ_i is the mapping function from an input space to a higher dimensional feature space and C is the regularization parameter that defines constraint violation cost. $\mathcal{E}_i$ is the slack variable where $\mathcal{E}_i > 0$ hold for misclassified points (Akbani *et al.* 2004). Practically, the convex quadratic programming problem (QP) can be solved by introducing the nonnegative Lagrangian multiplier α_i and transform it to a dual problem:

$$\text{Max}\,W(\alpha)=\sum_{i=1}^{\alpha_i}-\frac{1}{2}\sum_{ij}^{N}\alpha_i\alpha_j y_i y_j x_i x_j \tag{5.38}$$

In a higher feature space, finding the optimal hyper-plane is computationally expensive and complicated. It was not until Guyon *et al.* (1993) have showed that the so called ‘kernel trick’ can be significantly applied to overcome the above concerns. As described in equation 5.38, the training phase only includes the training data as scalar inner products form $x_i.x_j$. Thus, mapping the data to a higher feature space can be achieved via the kernel trick as follows:

$$\max W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j\Phi^T(x_i)\,.\,\Phi(x_j) \tag{5.39}$$

$$s.t.\ \textstyle\sum_{i=1}^{l}y_i\alpha_i=0,\ 0\leq\alpha_i\leq C,\ i=1,\dots,l$$

An important advantage of the SVM is that the transformation from input space to higher dimensional feature space can be done implicitly using the ‘kernel trick’ where $K(x_i,x_j)=\Phi^T(x_i).\Phi(x_j)$ is the Kernel matrix.

The choice of kernel plays a crucial rule in the proposed algorithm and the performance is largely depends on the kernel. However, there is no general role available as to which kernel should be used. In the field of machine learning, Gaussian, polynomial, and sigmoid functions are three commonly used kernel functions. The kernel function utilized in the proposed algorithm is Gaussian Radial Basis Function (GRBF) given as follows:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \quad i = 1, 2, \dots, N \tag{5.40}$$

Where σ is the variance parameter, after solving the QP in equation (5.1) and finding the optimal values of α_i, the values fall in the margins which have nonzero α_i are referred as the support vectors by which the position of the hyper-plane is defined. In the ISVM algorithm, the iterative strategy is based on the grid search optimization scheme of the regularization parameter (C) and the width of (GRBF). The structure of the proposed ISVM algorithm is illustrated as shown in Figure 5.9.

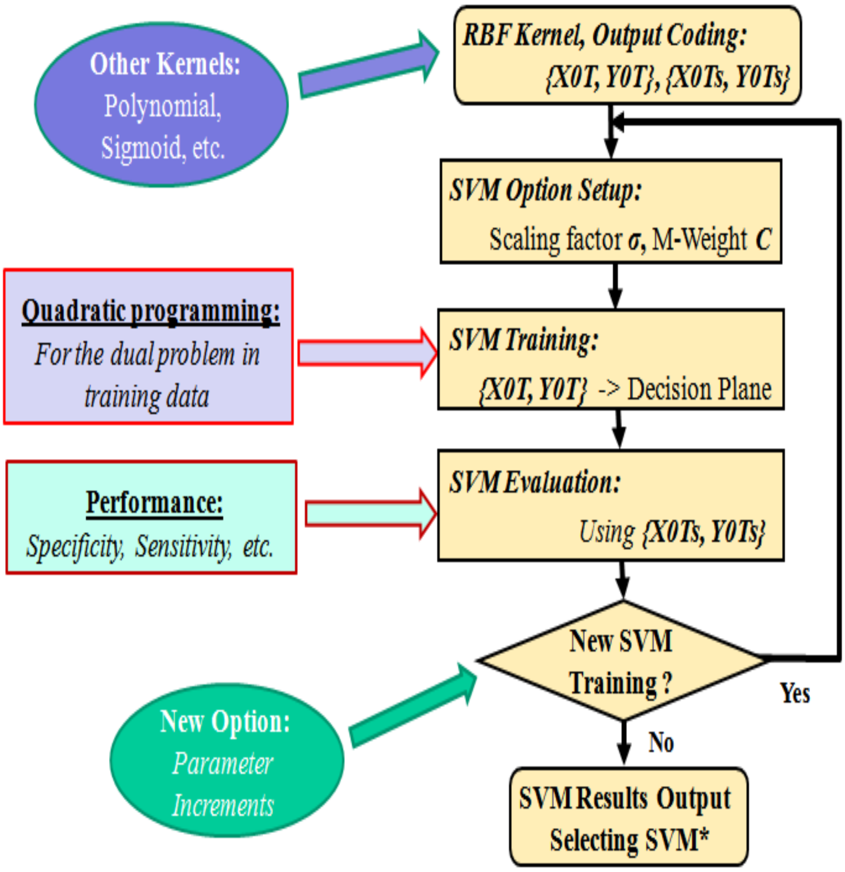


Figure 5.9: Architecture of iterative support vector machine algorithm

In the classification problem, there are obvious questions that arise when applying support vector machines i.e.,

- Which kernel is the best?
- How can one find the optimal parameters of the kernels?
- How one can select the penalty parameter C?

In practice, one of the formal and reliable metrics for parameter selection is to decide on parameter ranges and the optimal performance assessed using an exhaustive grid search over the parameter space. A crucial step for a successful SVM classification is data normalization. Fail to normalize the data will increase the running time and in some cases, no convergence will occur.

In the optimization procedure of the ISVMs, a quadratic programming algorithm was utilized when SVMs were invented in 1995 (Cortes and Vapnik 1995). The quadratic programming algorithm was slow and only small scale dataset could be run successfully. It is until 1998, where another technique referred to as sequential minimal optimization (SMO) was proposed by Platt (1998). The SMO procedure is decomposed into only a few tasks for optimizing the Lagrangian multipliers (alpha). SMO procedure iterates through the given dataset updating Lagrangian multipliers (only two of α_i at a time as shown in equation (5.39). In such a way, the SMO simplifies the optimization problem in to smaller steps providing a remarkable increment in the training time.

Applying the aforementioned framework on the imbalanced rails data set (39 variables) has led to a poor generalization performance as well as large number of support vectors (4949 support vectors). The model's performance is skewed towards the negative (majority) class where the positive (minority) class is poorly classified. Figure 5.10 illustrates the classification performance of the iterative SVM algorithm when learning from severely imbalanced rails data (Original dataset).

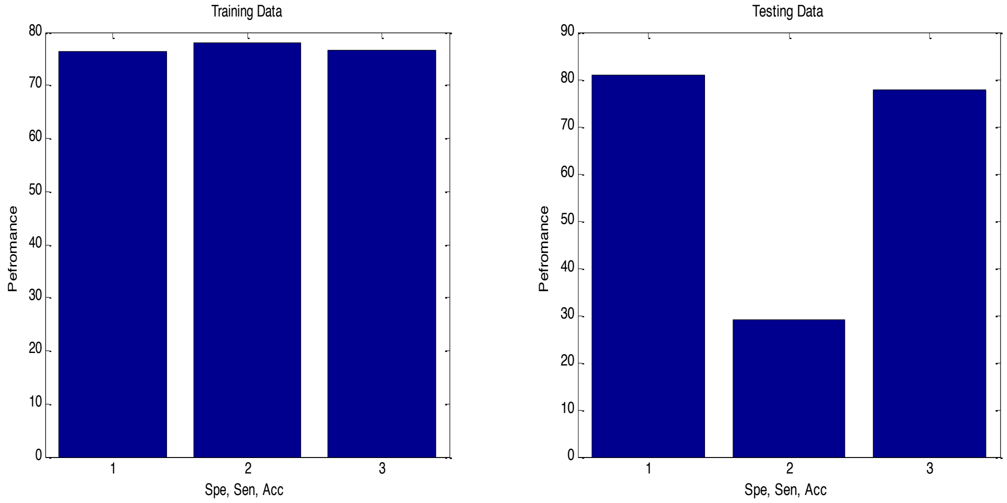


Figure 5.10: Preliminary SVM classification result

In a classification task, the most important task is to correctly classify the minority class instances. Deep investigations were carried out on what are the root causes of such skewness of the model's performance towards the majority classes?

Due to the fact that classifiers and modelling algorithms are designed to learn from sampled data, SVM algorithm tends to be sensitive to class imbalance (Batuwita and Palade 2010b),. Subsequently, the majority classes are likely to be classified correctly whereas the minority classes are approximately ignored. In other words, the classification performance is skewed to the majority class. The class imbalance has a dominant influence on the classification performance.

To overcome this concern and to improve the performance of the iterative SVM algorithm, a bootstrapping-Based over-sampling and under-sampling schemes (previously discussed in section 5.3) are applied to change the class distribution of the data. It is not only the model' performance has been significantly improved but also the number of the support vectors has been also reduced. It is worth mentioning that Large Data Modelling is Hungry for Resources and no convergence occurs when using 4GB memory due to computationally expensive optimization phase. As a result, computer memory is extended to 16 GB instead of 4GB. The next sections will discuss the application of the proposed iterative support vector machine (ISVM) algorithm on Rails dataset with bootstrapping-based over-sampling and under-sampling.

5.5 Iterative Support Vector Machines Based Rails Data Under-Sampling

Data resampling techniques, oversampling and under-sampling, are the best choices for overcoming the class imbalance problem. A separate optimization problem is to be performed in the binary SVM classifier. It is of great importance to achieve an optimal generalization performance. Training the proposed ISVM model require tuning various user-defined parameters. For a better classification performance, these parameters need to be learned with great care. Table 5.1 summarises ISVM parameters employed in rails data classification. Additionally, the resulted SVM model parameters are listed in Table 5.2:

Table 5.1: User-defined parameters of ISVM binary classification model

Kernel	The choice of kernel plays a crucial rule in the proposed algorithm and the performance is largely depends on the kernel. The choices may include linear and nonlinear kernel types as described in section 5.2.4. Practically, the kernels contain on or more parameters that need to be carefully optimized to a good generalization.
misclassification penalty or regularization parameter (C)	The regularization parameter C is applied to adjust the relation between $\|\|w\|\|^2$ and the slack variable. It trades off margin size and training error.
Training data Size	Practically, training data size significantly effect SVM performance, performance increase with more training data. However, large scale data might produce large number of support vectors and thus expensive computations.

Table 5.2: The important factors of the resulting ISVM model

Number of support vectors (SVs)	The number of support vectors in the resulting SVM model is influenced by the size of the training data, Support vector machine produce large number of support vectors when applying to large datasets
SVM execution time	SVM parameters and training data size can significantly influence the training time of SVM model
Performance %	The classification performance is the most important issue in applying SVMs and is achieved if one successfully selects and optimizes the parameters described in Table 5.1.

The data set was randomly split into training and testing set in the ratio of 7 to 3. For the under-sampling, the rails data is under-sampled as described in Section 6.3.1 until both classes were equal and R_{mm} value of 1 is achieved. Figure 5.11 shows the classification performance of ISVM algorithm on the under-sampled rails dataset.

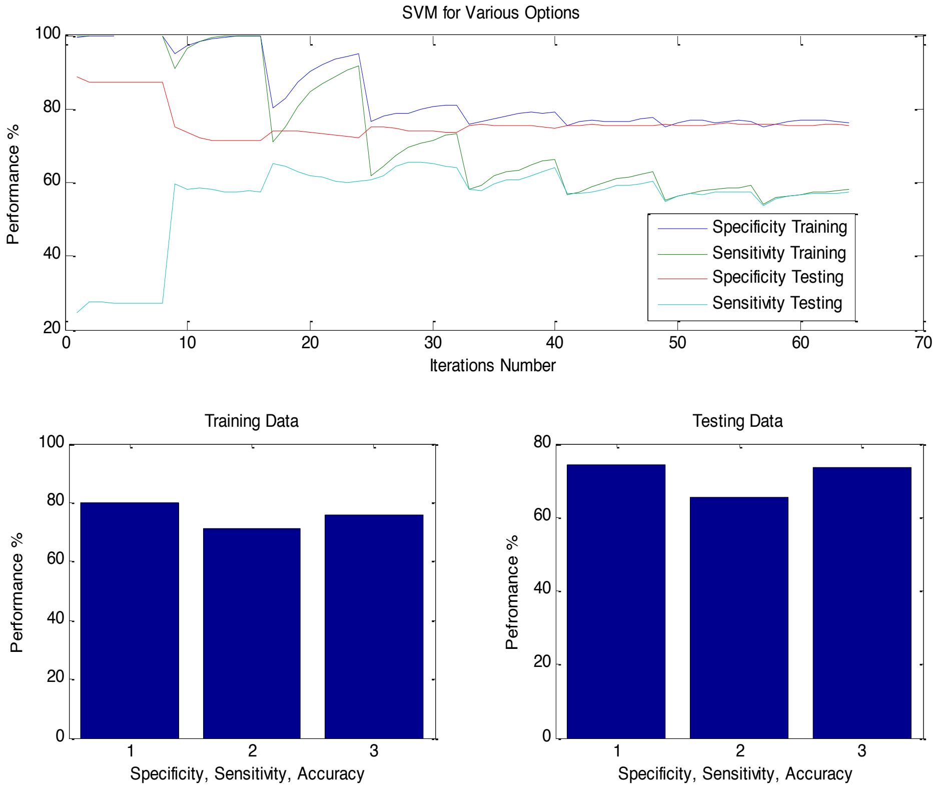


Figure 5.11: ISVM performance for Under-sampled rails data

It can be clearly seen from Figure 5.11 that sensitivity performance improved significantly through data resampling, from about 29% to around 65.5 %. The classification accuracy rate of ISVM classifier is not only influenced by the two parameters i.e., regularization parameter (C) and the width of the Gaussian RBF (σ), but also other factors, including the quality and dimensionality of datasets.

In practice, the most reliable method to parameter selection is to decide on parameter ranges and the optimal performance assessed using an exhaustive grid search over the parameter space. The maximum number of iterations of the proposed algorithm is controlled via the number of parameters employed in the grid search. Parameters are selected based on the following ranges: C = {1, 2,…, 15} and σ = {4, 7, 10, 13, 17, 20}.

Gaussian radial basis function (GRBF) is the best choice for providing superior generalization capability as it has less parameter than other nonlinear kernels. Moreover, it is capable of handling the nonlinearity between classes and features. The training time of ISVM with under-sampling is 3.67 minutes. It can clearly be stated that algorithms-based Under-sampling technique is less time consuming and thus superior. The integration strategy with the under-sampling scheme succeeded in drastically reducing the number of support vectors to 2171. The speed of ISVM classification algorithm depends on the number of support vectors (Manikandan and Venkataramani 2009). SVM classification results obtained with under-sampling is presented in Table 5.3.

Table 5.3: Performance result of ISVM classifier with under-sampling

Classifier	Sampling Type	Data Size	# SVs	Execution Time	sensitivity %
ISVM	Under-Sampling	2877	2171	3.67 minute	65.3 %

Based on the above obtained results, three observations can be made as follows:

1) A good generalization performance is achieved via The Gaussian kernel. The kernel function potentially maps rails data into infinite dimension space where as other kernels such as the linear or polynomial uses feature space with fixed number of dimensions.
2) The classification accuracy rate of SVM classifier is not only influenced by the two parameters i.e., regularization parameter (C) and the width of the Gaussian RBF (σ), but also other factors, including the quality and dimensionality of datasets.
3) Under-sampling produces a datasets with low dimensionality. The class distribution advantages of under-sampling enable SVM algorithm to yield small number of support vectors. Consequently, the classification time will be decreased.

Although the ISVM algorithm provides a good classification performance with under-sampling, it is important to examine the same algorithm with other sampling techniques. The next section will discuss the application of ISVM with bootstrapping-based oversampling scheme.

5.6 Iterative Support Vector Machine with Bootstrapping-Based Over-Sampling

Bootstrapping based-oversampling is another sampling technique that is also applied to change the class distribution of rails data. The oversampling technique has gained extra attention. The advantage of such a technique is that it is external and therefore, easily transportable as well as very simple to implement (Estabrooks *et al.* 2004). Moreover, over-sampling the minority class data avoids unnecessary information loss (Yang *et al.* 2011a).

Over-sampling has its drawbacks and results in datasets with high dimensionality. Consequently, the computational cost associated with SVM is increased. The classification performance of SVM algorithm on the over-sampled rails dataset is presented in Figure 5.12.

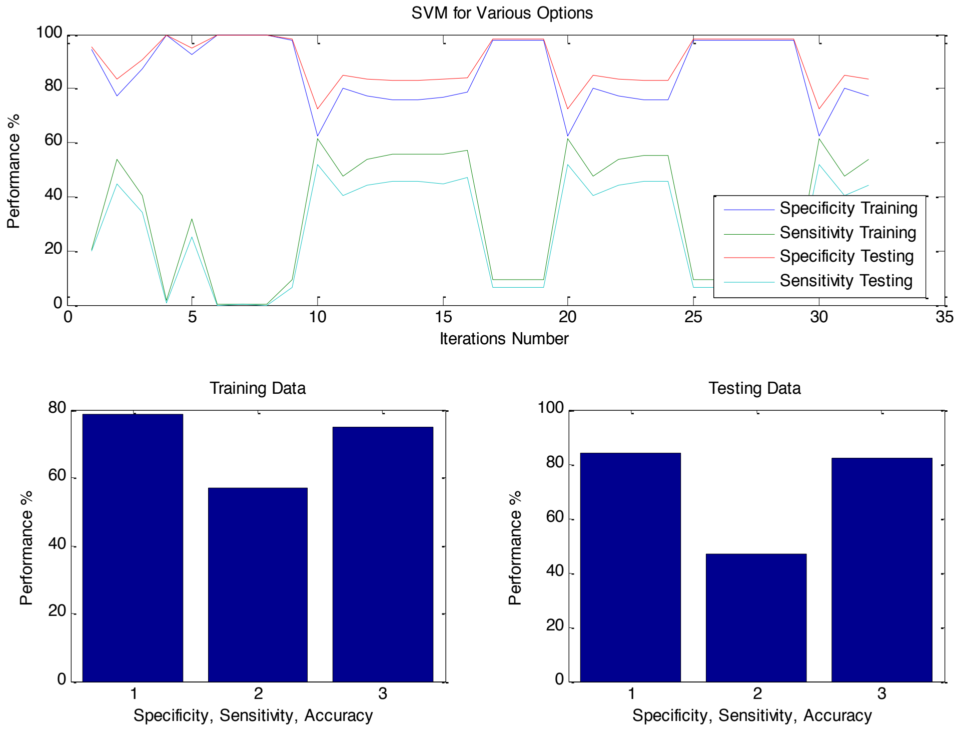


Figure 5.12: ISVM with bootstrapping-based over-sampling classification performance

Over-sampling is a resources hungry technique since it creates a multiple randomly resampled datasets. Therefore, a significant increment of the overall size of the data is occurred. Due to the dimensionality nature of the data being produced and computer memory restrictions, the optimal sampling rate achieved via bootstrapping based oversampling approach is 5.

The training time of ISVM with bootstrapping-based over-sampling is 77.86 hours. Bootstrapping-based over-sampling yields a large number of support vectors i.e. 23452. As shown in Table 5.4, ISVM is sensitive to the bootstrapping-based over-sampling, which is the only shortcoming of the presented algorithm. Bootstrapping-based Over-sampling causes performance degradation for ISVM classifier to 47.1% and therefore leads to a poor generalization capability.

Table 5.4: SVM performance with bootstrapping-based over-sampling

Classifier	Sampling Type	Data Size	# SVs	Execution Time	sensitivity %
ISVM	Bootstrapping-Based Over-Sampling	30992	23452	77.86 hour	47.11 %

5.7 Performance Metrics

The evaluation criterion is the best guidance of modelling and classification performance assessment. This section describes the performance measures for class imbalance learning that is employed for evaluating SVM results. For binary classification problem, Confusion matrix as shown in Table 5.5 is the most effective referenced source for performance evaluation. The classification performance is assessed using the accuracy, sensitivity and specificity as follows:

$$Accuracy\ (\%) = \frac{TP + TN}{TP + FP + FN + TN} \tag{5.41}$$

$$Sensitivity\ (\%) = \frac{TP}{(TP + FN)} \tag{5.42}$$

$$Specificity\ (\%) = \frac{TN}{(TN + FP)} \tag{5.43}$$

However, with a highly skewed data distribution, the overall accuracy in equation (5.41) is not a promising metric since classifiers inherently learn from the majority class and therefore ignore the minority class (Tang *et al.* 2009):

Rails data are highly imbalanced data where the good rails far outnumber the rejected rails. The optimal classification metrics that gain an extra emphasis in this work are based on the sensitivity at equation (5.42) and specificity at equation (5.43) since the rejected rails, minority class, are more important to be correctly classified. In other words, it is of strategic importance in the rails process to distinguish if a combination of input parameters will yield rails with cracks and flaws (Muscat *et al*. 2014).

Sensitivity and specificity refer to the ability of the classifier to correctly classify the minority and the majority classes respectively, whereas the accuracy is defined as the overall percentage that both classes are correctly classified. Entries of confusion matrix along the main diagonal represent the total number of correctly classified examples. Whereas, other entries than those on the main diagonal represent classification errors.

Table 5.5: Binary Classification Confusion Matrix

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

where:

- True Positive (TP): Rejected rails accurately classified as rejected rails.
- True Negative (TN): Good rails accurately classified as good rails.
- False Positive (FP): Good rails inaccurately classified as rejected rails.
- False Negative (FN): Rejected rails inaccurately classified as good rails.

5.8 Comparative Analysis and Model Evaluation

As stated earlier, the support vector machine algorithm is sensitive to the class imbalance learning (Batuwita and Palade 2010b); (Lin and Wang 2002). In data classification, the choice of a Kernel function is challenging and becomes a central problem (Micchelli and Pontil 2005); (Prajapati and Patle 2010). Mapping the non-linear input space to a higher feature space (linear) via a kernel function depends significantly on the nature of the data. Therefore, The RBF as a kernel is employed due to its clear implementation and the potential effectiveness on overall performance (Sahoo *et al.* 2013); (Prajapati and Patle 2010).

Applying the proposed ISVM framework on the imbalanced (original) rails dataset has led to a poor generalization as well as a large number of support vectors. Support vector machine parameters i.e., regularization parameter (C) and the width of RBF, are optimized based on the grid search approach. The model's performance is skewed towards the majority class where the minority class is poorly classified at less than 25%.

We study the performance of the proposed algorithm with the class distribution advantages of sampling techniques. Bootstrapping based over-sampling and under-sampling schemes with different sampling rates are tailored to overcome the class imbalance phenomena. Consequently, it is not only the model's performance that has been significantly improved but the number of the support vectors has also been reduced. Table 5.6 illustrates a performance comparison of ISVM algorithm with under-sampling and bootstrapping-based oversampling.

Table 5.6: Performance comparison of SVM algorithm

	Number of Support vectors	**Execution Time**	**Sensitivity % of testing set**
Under-sampling	2171	3.67 minute	65.3 %
Bootstrapping-Based Oversampling	23452	77.86 hour	47.1%

With under-sampling, the ISVM algorithm has shown a good generalization with significant classification performance increment of 65.3%. Moreover, under-sampling technique succeeded in drastically reducing the number of support vectors of SVM classifier to 2171. The advantages of fewer support vectors will mostly mean short computational time and small memory requirements (Zheng *et al.* 2013). Theoretically, under-sampling has mostly the best trade-off between algorithm generalization capability and the number of support vectors. The maximum number of iterations is controlled via the number of parameters utilized in the grid search scheme. The results agree with the hypothesis that under-sampling the majority class reduces the total number of training examples, speeding up the training time and accordingly ensure promising classification performance.

The employment of the Bootstrapping-based over-sampling technique causes performance degradation to 47.1% and thus weak generalization capability, because time complexity grows dramatically as the size of the data increase. The ISVM model built from an over-sampled data yields a large number of support vectors. The experimental results show that Bootstrapping-based over-sampling increases the computational cost associated with the ISVM training algorithm. It is worth mentioning that Large Data Modelling is Hungry for Resources and no convergence occurs when using 4GB memory due to the computationally expensive optimization phase. As a result, the computer memory has been extended to 16 GB. Generally, the SVM technique has drawbacks as it can be computationally expensive when dealing with large scale data and it tends to produce a large number of support vectors.

5.9 Summary

The main purpose of this chapter is to produce a new approach to rails data classification via iterative SVMs with bootstrapping-based oversampling and under-sampling. The effectiveness of the proposed classification formulation has been applied on two datasets. As can be seen in Table 5.6, the results show that SVM is a promising algorithm for the resampled rail data classification problem. Resampling techniques adopted in our experiment play crucial role for effective data classification, however, under-sampling can suppress the number of support vectors and result in a SVM with a significant performance gain. It also shows a significant reduction of the complexity of memory and training time.

Class imbalance is not the only problem which tends to govern the performance of the learning algorithms, but there are other elements which potentially hinder the classification performance such as the overall size of the data set. The Gaussian radial basis function kernel guarantees superior generalization due to its flexibility.

In the next chapter, a new iterative fuzzy support vector machine (IFSVM) classification-based paradigm is introduced. The IFSVM is proposed for severely imbalanced rail data classification with bootstrapping-based oversampling and under-sampling.

Chapter 6

Iterative Fuzzy Support Vector Machines: Rails Data Classification Framework

6.1 Introduction

Real world industrial processes are complex environments whose performance in terms of expected quality of delivered products and cost management are of a strategic importance. In industrial applications, there is a remarkably increasing need to reduce manufacturing costs while enhancing process yields. Knowledge discovery and data mining techniques have therefore become a significant target for industrial and research initiatives to develop frameworks and guidelines to identify bottlenecks in the production routes and key areas for improvements. The idea of knowledge discovery in data bases and machine-learning revolves around extracting information and unknown interesting patterns such as groups, unusual records and dependencies from given data set via constructive learning algorithms such as support vector machines and fuzzy support vector machines.

Due to the ongoing rapid growth of data in a wide variety of real world applications, the researchers have broadened the idea of SVM into various applications such as fuzzy SVM (Lin and Wang 2002), (Lin and Wang 2004), Lagrangian support vector machines (Mangasarian and Musicant 2001). FSVMs, being computationally powerful techniques for binary classification problem, have gained much popularity in understanding the interaction and influence of input features on overall process yield. FSVM works similarly to SVM, except that a specific membership degree is given to each data point so that various data points can make different contributions to the decision surface learning. FSVM model prevents noise and outliers from creating narrower margin by assigning a membership degree to each data point. However, in the SVM model case, equally training each data point may cause over-fitting. The calculation of membership values is based on the sparse distribution of the training points, with outliers and noise being assigned proportionally smaller membership values than other points (Shilton and Lai 2007). Despite the computationally attractive features of SVMs and FSVMs, the class imbalance problem significantly hinders their prediction performance in classification problems.

In this chapter, a new iterative fuzzy support vector machine classification framework for the same data used in studies of chapters 4 and 5. To generalize the proposed IFSVM so that it overcomes both the problems of learning from highly imbalanced data and the influence of data dimensionality on the overall generalization performance, a new strategy was designed by incorporating the unique learning mechanism of FSVM and the class distribution advantages of resampling techniques. Data resampling techniques, oversampling and under-sampling, are the best choice for overcoming the class imbalance problem. The classification results offer a better understanding of the effect of resampling techniques on imbalanced datasets.

A hybrid learning framework using iterative fuzzy support vector machines based fuzzy c-means clustering (IFSVM-FCMs) is also proposed in this chapter. Fuzzy c-means clustering is an effective clustering scheme which selects the informative examples from large scale datasets and prevents the classification model to make a full search in the entire training set. This strategy renders classification model to be applicable to very large scale datasets which otherwise would be computationally very expensive. Combined with the fuzzy c-means clustering, IFSVM achieves a fast and scalable solution without prediction performance degradation.

The results show that the proposed active learning strategy (IFSVM-FCMs) can be used to address the class imbalance problem and provide a remarkable classification performance. It has been proven that as the class imbalance ratio decrease, the proposed IFSVM-FCMs can achieve better prediction performance than with highly imbalanced training set. With the implementation of data resampling techniques i.e., bootstrapping-based over-sampling and under-sampling, more balanced class distributions can be provided to the learner in the earlier steps of the learning.

6.2 Fuzzy Systems

Fuzzy logic systems FLSs are rule-based systems that utilize the theory of fuzzy sets and fuzzy logic proposed by Zadeh (1965). Fuzzy logic systems are applied in various application areas to represent knowledge in a similar way to human brains allowing decisions to be made based on vague information. Fuzzy logic is a branch of various-valued logic employs the fuzzy set theory. Fuzzy logic has the ability to represent variables and relationships in linguistic terms such as small, medium and large and build a model based on fuzzy if-then rules.

Since their introduction by Zadeh in his seminar paper ''fuzzy sets'' (Zadeh 1965), fuzzy logic systems have been known to be one of the most important areas of fuzzy set theory, which includes approximating the membership of objects to a set (Zadeh 1965). However, the degree of truth is described by fuzzy logic using truth values between 0 and 1. The fuzzy set is characterized by the corresponding membership functions. The concept of fuzzy set is that any element is given a degree of membership of this set which is different to the ordinary crisp set where its membership is defined by either a value of 0 or 1.

Recently, the theory of fuzzy set has played a crucial role in dealing with uncertainty. Fuzzy logic and fuzzy set theory have evolved into powerful tools for handling uncertainties inherent in complex systems. In real world engineering problems, the uncertainty existed in datasets could not be avoided due to instruments malfunction, measurement errors and human inference. A three types of uncertainty were presented by Mendel (2003) such as fuzziness, strife and non-specificity.

A number of different types of membership function (MF) have been proposed for fuzzy logic system. The most used forms of membership functions are those which are convex and normal. However, various operations on fuzzy sets lead to fuzzy sets with subnormal and nonconvex forms (Ross 2010). Membership functions can be symmetrical and typically expressed in one dimensional universe. Nonetheless, they certainly can be defined on multi-dimensional universe.

A cording to membership function shapes, there are eleven shapes of fuzzy membership functions. However, Gaussian, triangular, trapezoidal, piecewise linear and bell-shaped are by far the most commonly encountered functions in real world applications (Mendel 2001). A brief explanation on some of the above mentioned membership functions will be discussed below.

6.2.1 Triangular Membership Function

The construction of triangular membership function is based on three scalar parameters i.e., left vertex, centroid and right vertex. The triplet (left vertex, centroid and right vertex) is referred as (a, d, c). The calculation of (a, d, c) of the triangular function is given as follow:

$$\mu_A(x) = \begin{cases} \dfrac{x-a}{b-a}, a \leq x \leq b, \\ \dfrac{c-x}{c-b}, b \leq x \leq c \\ 0, \quad otherwise \end{cases} \tag{6.1}$$

Where $a, b\ and\ c$ are the left vertex, centroid and right vertex of the triangular membership function. The term $\mu_A(x)$ represents the membership function of x. The shape of triangular membership function can be either symmetric or asymmetric type. The shape of triangular membership function is shown in Figure 6.1.

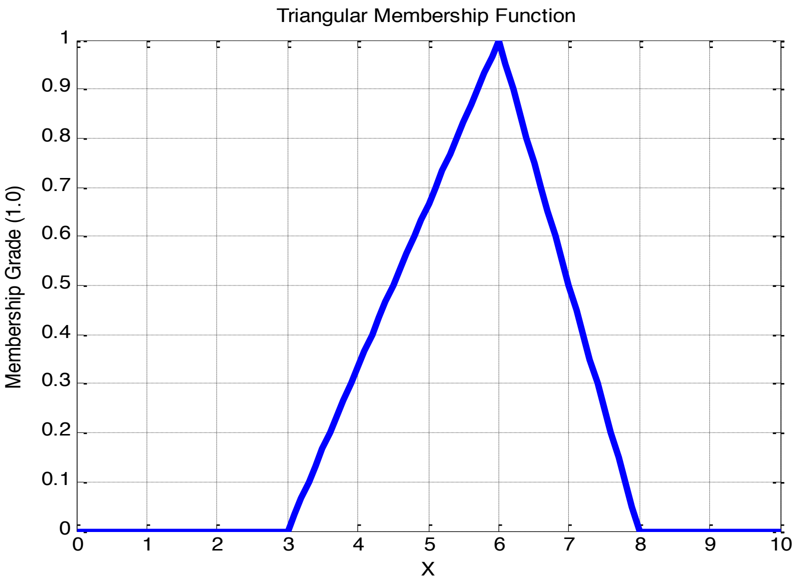


Figure 6.1: Triangular Membership Function

6.2.2 Gaussian Membership Function

In the Gaussian membership function, which is commonly used to represent linguistic terms, there are only two parameters i.e., the width and the centre of the Gaussian membership function (v, σ) respectively. The calculation of a symmetric Gaussian membership function can be given as:

$$\mu_A(x) = \exp\left(\frac{-(x-v)^2}{2\sigma^2}\right) \tag{6.2}$$

The shape of Gaussian membership function is shown in Figure 6.2.

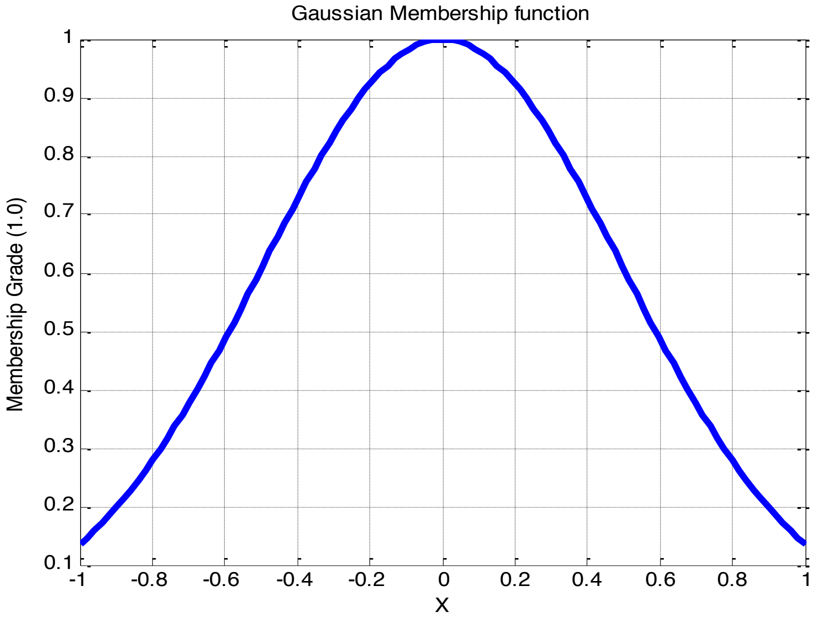


Figure 6.2: Gaussian Membership Function

In data classification problems, Gaussian (Radial Basis Function) is the best choice for providing superior generalization capability as it has less parameter than other nonlinear kernels. Moreover, it is capable of handling the nonlinearity between classes and features.

6.2.3 Bell-Shaped Function

The bell-shaped membership function has symmetrical shape and its calculation given as follows:

$$f(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \tag{6.3}$$

Where $''a''$ is the width of the curve, $''b''$ is the positive parameter and $''c''$ is curve centre. A bell-shaped membership function is shown in Figure 6.3.

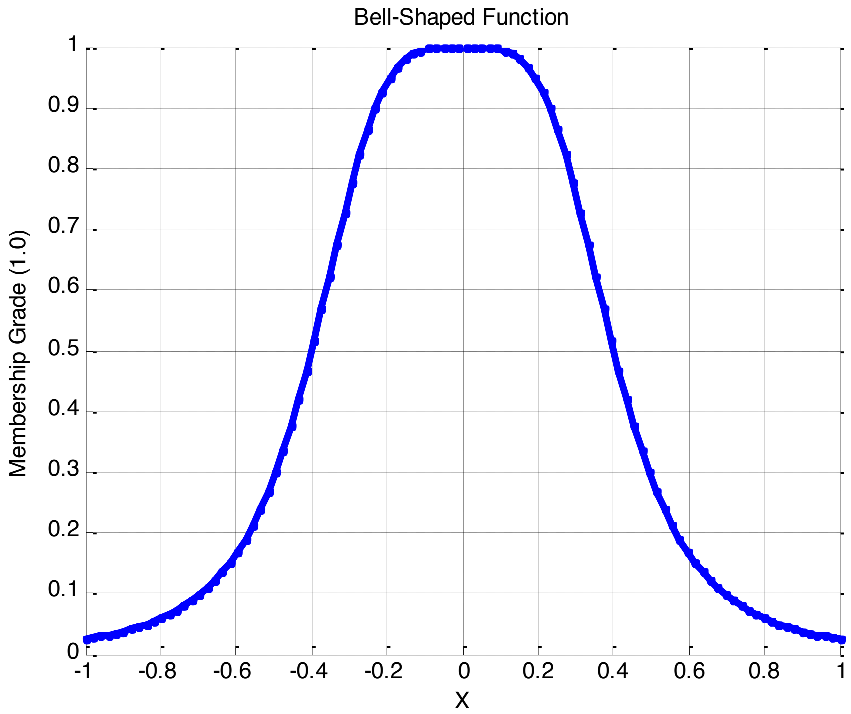


Figure 6.3: Bell-shaped Membership Function

6.2.4 Trapezoid Membership Function

The construction of trapezoid membership function requires four parameters (a, b, c, d) which are defined as left convex, upper length starting point, upper length terminal point and right convex. The calculation formula of Trapezoid Membership function is as follows:

$$\mu_A(x) = \begin{cases} \dfrac{x-a}{b-a}, & a \leq x \leq b, \\ 1, & b \leq x \leq c, \\ \dfrac{d-x}{d-c}, & c \leq x \leq d, \\ 0 & otherwise \end{cases} \tag{6.4}$$

Where $a, b, c \; and \; d$ are the left vertex, upper length starting point, upper length terminal point and right convex of trapezoid function respectively. The term $\mu_A(x)$ represents the membership function of x. A simple trapezoid membership function is shown in Figure 6.4:

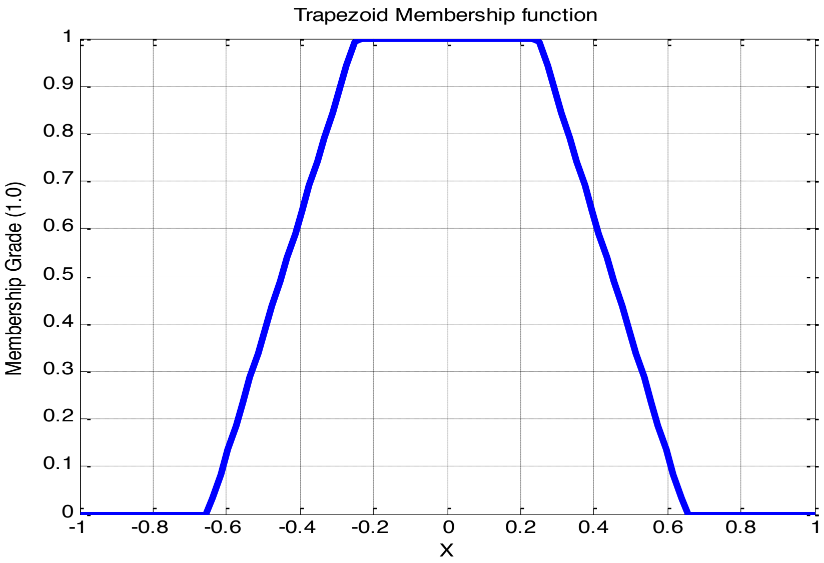


Figure 6.4: Trapezoid Membership Function

In practice, there is no explicit mothed by which one can choose the membership function, it can be chosen either arbitrarily or based on the researcher's experience (Mendel 2001).

6.3 Fuzzy C-Means Clustering

Exploring and organizing data into appropriate grouping increase dramatically in many engineering and scientific fields. Although many algorithms have been proposed for clustering analysis, Fuzzy C means algorithm has a rich and diverse history as it was independently discovered in different field and it is still one of the most widely used algorithms for clustering as it is easy to be implemented, simple, efficient, and has empirical success (Jain 2010). In contrast to K means algorithm, fuzzy C means algorithm has one drawback is that the probability of membership of a data-point in a cluster depends explicitly on the number of clusters and when this number is specified incorrectly, serious problems will ensue (Duda *et al.* 2001).

As discussed in the previous chapter, growth in both the dimensionality of rail data via applying bootstrapping based-oversampling scheme and the large number of support vectors yielded via SVMs as well as FSVMs has raised some shortcomings in the overall generalization capability. The problems outlined above can be eliminated via FCMs clustering scheme. FCM clustering attempt to partition the data based on the fuzzy partition criteria by which each data point of fuzzy partition relates to different clusters with different membership degrees (Yang *et al.* 2011 b).

FCMs became a useful data mining tool since it was proposed by Dunn (Dunn 1973) and developed by Bezdek (Bezdek 1981), for identifying interesting patterns and discovering clusters in the underlying data. The concept of FCMs is that each data point belongs to a cluster based on a membership degree. The mean location of each cluster is marked using a cluster centre and keeps updating along with membership degrees for each data point in the cluster until an appropriate location is reached. The iterative adjustment is based on the following minimization of the following objective function which represents the distance between the cluster centre and each given data point:

$$j(X;U,V)=\sum_{k=1}^{N}\sum_{i=1}^{c}\mu_{ik}{}^{m}d^{2}\|X_k-V_i\|^2 \tag{6.5}$$

Where $V=(V_1,V_2,\ldots,V_C)$ is center vectors, C is the number of fuzzy clusters, X refers to data samples and U is Fuzzy partition matrix whose elements $\mu_{ji}\in[0,1]$ represent the membership degree of X in cluster j. ${d_{ij}}^2=\|X_k-V_i\|$ is distance metric between X_i and V_j. Fuzzy clustering is carried out according to the following solutions of 6.5:

$$\mu_{ij}=\frac{1}{\sum_{j=1}^{N}(d_{ik}/d_{jk})^{\frac{2}{m-1}}}\;;1\le i\le c\,,1\le k\le N \tag{6.6}$$

$$v_i=\sum_{k=1}^{N}(\mu_{ik})^m x_k\bigg/\sum_{k=1}^{N}(\mu_{ik})^m\,;\;1\le i\le c \tag{6.7}$$

FCM algorithm iteratively runs through (6.6) and (6.7) to optimize cluster centers and fuzzy partition matrix. The fuzzy C means algorithm is summarized as follows (Dehariya *et al.* 2010):

4) Choose a number of clusters;
5) Randomly assign to each point coefficients for being in the cluster;
6) Repeat until the algorithm has converged (that is, the coefficients' change between two iterations is no more than the given sensitivity threshold).

The fuzzy C means method offers an important insight into the data by producing a degree of membership to individual data vectors within clusters.

Fundamentally, this algorithm seeks a minimum of heuristic global of cost function (Duda *et al.* 2001), where each point has a degree of belonging to clusters rather than belonging completely to just one cluster (Dehariya *et al.* 2010). Thus, points on the edge of a cluster, maybe in the cluster to a lesser degree than points in the centre of a cluster (Dehariya *et al.* 2010).

One of the most important problems in data analysis is cluster validation where the researcher needs to test solutions of data mining problems for robustness (Buhmann 1995). It is important to note that the data representation issue predetermines what type of cluster structures can be discovered in the data (Buhmann 1995). An additional issue is correctly selecting proper algorithm and correctly choosing the initial set of clusters (Fung 2001). The size of the data set is also a crucial issue, because most of the clustering algorithms require multiple data scans to achieve convergence (Fung 2001).

For an effective performance of fuzzy support vector machines on large datasets and tackling the sensitivity of SVMs to outliers and noises, Wu *et al.* (2014a) incorporated the fuzzy support vector machines with a novel partition index minimization (PIM) clustering scheme. The novel PIM clustering scheme is presented to discriminate between outliers and noise involved in training data set and generate fuzzy membership to the fuzzy support vector machine algorithm. With the aid of PIM clustering scheme, the FSVM shows robust results than other algorithms.

A conjunctive use of weighted support vector machines (WSVMs) and fuzzy C-mean clustering approach is presented by Hu *et al.* (2007) to predict short term load. The training data points are clustered according to their similarity degree and the data points that share homogenous features are selected and employed as an input to the model. The selected data points are viewed as more important as the older once. As result, a new learning machine named as weighted support vector machines WSVMs is constructed where each new data is assigned a weight vector.

For an effective acquiring of patterns from data, a fuzzy C-means clustering technique based on kernel methods is proposed by Wu and Xie (2003). The new algorithm termed as fuzzy kernel c-means (FKCM) is based on an integration strategy between FCMs clustering and mercer kernel function that is capable to deal with some issues in fuzzy clustering. The advantages of this technique are shown that the FKCM is not only effective on spherical shape clusters, but also annular ring shapes. However, the challenge in applying FKCM is in choosing a proper kernel function and the way to optimize the associated parameters.

A comparative analysis between two important clustering technique namely fuzzy C-means clustering and centroid based fuzzy K-means clustering is presented by Ghosh and Dubey (2013). The performance of the two algorithms is evaluated based on the efficiency of clustering output. The behaviour patterns of the two techniques are analysed based on the number of clusters and data size. Similar results are obtained from both algorithms; however, FCMs are more suitable for handling the issues related to noisy and incomplete data.

6.4 Iterative Fuzzy Support Vector Machine: Rails data Classification approach

Fuzzy support vector machine is defined as an extension of the idea of SVM with fuzzy membership to yield FSVM. In SVM, misclassified observations are not a concern and one would not care about some training data-points such as noise and outliers as to whether they are misclassified or not if the meaningful points are classified correctly (Lin and Wang 2002). FSVM provides an effective approach to deal with the above concern. The main idea is to allocate a membership function to every data point in the data set at hand; this fuzzy membership can be considered as the attitude of the corresponding training point towards either the majority or minority class (Lin and Wang 2002). Many real world applications calculate the membership values based on the sparse distribution of the training points, where smaller membership values are given to the noise and outliers than other training points (Shilton and Lai 2007).

Reviewing the past and on-going researches that focus on extracting knowledge from data set by applying SVM and FSVM is a crucial task for the researcher. The review can highlight the advantages and disadvantages of specific task as well as identifying other research paradigms to solve the problem at hand. The Rail through process data contains a large amount of data records and data variables. Such data is often very difficult to model due to class imbalance problem and to its high dimensional nature. Consequently, data quality and data dimensionality will dominate model performance. Furthermore, the choice of modelling approach will also influence the performance of the model (Gunn and Kandola 2002).

Previous studies from researchers on data classification focused on applying kernel techniques (support vector machines, Fuzzy support vector machines) on real-world dataset such as (Vapnik 1995), (Boser *et al.* 1992), (Burges 1996), (Duan and Keerthi 2005), (Cortes C. and Vapnik V. 1995).

Due to the ongoing rapid growth of data in a wide variety of real world applications, the researchers have broadened the idea of SVM into various applications such as Fuzzy SVM (Lin and Wang 2002), (Lin and Wang 2004) and Lagrangian support vector machines (Mangasarian and Musicant 2001). FSVM works similarly to SVM, except that a specific membership degree is given to each data point so that various data points can make different contributions to the decision surface learning. The FSVM model prevents noise and outliers from creating narrower margin. However, in the SVM model case, equally training each data point may cause over-fitting. The calculation of membership values is based on the sparse distribution of the training points, with outliers and noise being assigned proportionally smaller membership values than other points (Shilton and Lai 2007).

Lin and Wang (2002) proposed one type of FSVMs. The main idea of the associated algorithm is to apply a fuzzy membership to each data input of SVM and reformulate SVM in to FSVM. An appropriate fuzzy membership is chosen in such a way that the lower pound of fuzzy memberships must be defined; then, the property of the main data is to be selected and finally make connections between fuzzy memberships and this property. However, this method is likely to yield good results only if the distributions of the given training data of each class happen to be around the central means. The formulation of algorithm is not complete where the linear separable cases cannot be discussed. However, comparative experimental results against standard SVM on real data set are not provided (Tao and Wang 2004).

Further work was carried-out by the same authors to design a noise model which can employs two factors in training data vectors, the trashy factor and confidence factor, and spontaneously generates fuzzy memberships of the given training data vectors from a heuristic strategy by employing the obtained factors and a mapping function (Lin and Wang 2004). The obtained model is utilized to estimate the probability of the noisy information and outliers and then, employ this corresponding probability to tune the fuzzy membership in FSVMs. The experiments ensure low generalization error; however, the use of FSVMs with kernel functions leads to more complicity since there exist many parameters. Also, it is computationally expensive to search the optimal parameter in the training process.

Tao and Wang (2004) proposed a new fuzzy support vector machine algorithm based on the weighted margin. The basic idea is to employ the fuzzy membership function to weight the margin. This approach incorporates the idea from SVM and fuzzy neural networks for a better classifier performance. The influence of data inputs can be either reduced or avoided by applying the fuzzy membership for each training vector to weigh the margin. The advantage of modifying SVM via the idea of neural fuzzy system is to apply some fuzzy membership functions. Consequently, experiments on real data sets illustrate that NFSVM can yield robust results in contrast to standard SVM.

Xiong *et al.* (2005) presented a new algorithm using fuzzy support vector machines based on fuzzy c-means clustering. The algorithm is based on the idea of applying fuzzy c-means clustering scheme to each class of data set. The key feature is that during clustering with a suitable fuzziness parameter, the algorithm will get rid of the data that are less important and will choose the important samples such as the support vectors to represent the other similar samples that are close to the cluster centres. Experimental results of the proposed fuzzy support vector machines showed that less quadratic programing time is needed compared with conventional SVMs.

(Shilton and Lai 2007) introduced a new algorithm for the calculation of membership values using an iterative fuzzy support vector machine. In contrast to (Lin and Wang 2002), this approach does not take in to account the form of the distribution of the training vectors. It iteratively makes use of the result obtained from SVM training process and information about misclassified training vectors (error vectors) to adjust and generate membership values, with outliers being given smaller values of membership than other training points. The FSVM process will be repeated with these new membership values for a certain number of cycles until convergence.

(Tang and Qu 2008) proposed a new fuzzy membership function to solving classification problem for FSVM. The algorithm defines the membership function not only based on the distance between each data point and the means of class, but also similarity between data points which is defined by K-nearest neighbour distances. Results show that such algorithm can achieve a good performance on decreasing the effect of outliers.

(Batuwita and Palade 2010) proposed an approach to improve FSVMs for class imbalance learning CIL (called FSVMs-CIL). This method is presented to handle the class imbalance dilemma for the task at hand in the presence of noise and outliers. The basic idea is to assign fuzzy-membership values for the training examples based on their importance in order to reduce the effect of the above concerns. This approach is evaluated on ten real world data sets, containing around ten thousand records. Experiments show that the proposed algorithm is affective and outperform other existing internal and external imbalanced learning methods. However, experiments are limited to small data sets and the authors have not proven the robustness of their algorithm on large scale data set where much larger optimization problem is required.

Fuzzy rough set based support vector machine (FRSVM) was proposed by Chen *et al*. (2010). In this algorithm, a kernel based fuzzy rough set is firstly proposed and then, the membership function for every data point is computed using the lower approximation operator in the kernel based fuzzy rough set. The hard margin standard SVM is reformulated in to FRSVM by transforming constraints in normal SVM. Comparison analyses show that the proposed algorithm is efficient in a way that improved the generalization of hard margin SVMs.

Wang *et al.* (2003) outlined the advantages of connecting kernel machines with fuzzy systems, produced a link between kernels and fuzzy rules and presented a learning paradigm for fuzzy classifier named as positive defined fuzzy classifier (PDFC). The proposed PDFC is built from the given data points based on standard SVM with the IF-part fuzzy rules given by Support vectors where the fuzzy inference on IF-part of fuzzy rules is the evaluation of the kernel function. In this work, PDFCs with various reference functions ensure good performance since the upper bound of the expected risk is minimized by the learning process.

A novel fuzzy support vector machines approach for multi-class classification problem is proposed by Schwenker *et al.* (2014). The algorithm is capable of benefiting from fuzzy labelled training data and determines fuzzy memberships for each data input. The algorithm can be viewed as an extension of the fuzzy support vector machine approach for fuzzy labelled data into two class classification structure. Based on three benchmark datasets, the inclusion of fuzzy labelled data points in the training set demonstrates effective results of the proposed algorithm.

In classification problem, it is known that the choice of the fuzzy membership function can effectively reduce the influence of outliers. Jiang *et al.* (2006) proposed a new Fuzzy SVM with a new fuzzy membership function. The membership function which is represented by kernels is calculated in the feature space for nonlinear classification. The proposed method shows an improvement in the classification performance and generalization in contrast to that presented by Zhang (1999).

Due to the fact that support vector machines are more likely to produce large number of support vectors in the classification problem, Muscat *et al.* (2014) presented a hierarchical fuzzy support vector machine-based fuzzy C-means clustering algorithm for severely class imbalance rails data modelling. An internal (biased) fuzzy support vector machine is integrated with external data resampling techniques. For a better compromise between model performance and training time, the algorithm was integrated with FCMs clustering. Promising results were obtained in terms of reducing the number of support vectors while maintaining a good generalization performance.

Yang *et al.* (2011b) proposed a hybrid classification framework based on fuzzy support vector machines with kernel fuzzy C-means clustering for binary classification problems of noises and outliers. The FCM is firstly employed to cluster each class in the high dimensional feature space. The data points located far from the centre are selected to form a new training data set with membership degrees. Finally, the FSVM algorithm is applied to classify the new dataset. Experimental results show that the kernel fuzzy C-means based fuzzy support vector machine algorithm present reasonable membership degrees and in more efficient the standard FCM-FSVM algorithm.

Despite of all the theoretical and practical advantages of SVMs, they can be limited in their performance to outliers or noises in the training dataset due to over-fitting (Wu *et al.* 2014b). These types of uncertainties result in some examples being more important than others for decision making (Yang *et al.* 2011 b). FSVMs have been widely applied to address such uncertain problems. FSVMs introduce a Fuzzy membership values $0 < s_i \leq 1$ to each training point x_i; i.e. the less important data points are being assigned a lower fuzzy membership. In the classification problem, s_i is considered as the attitude of the corresponding training point towards one class (Lin and Wang 2002). The IFSVM model can be described by reformulating the quadratic programming problem as follows:

$$\min\left(\frac{1}{2}w^T.w + C\sum_{i=1}^{l} s_i\mathcal{E}_i\right) \tag{6.8}$$

$$s.t.\ y_i(w^T\,.\,\Phi(x_i) + b) \geq 1 - \mathcal{E}_i\,, \mathcal{E}_i \geq 0, \qquad i = 1, \dots, l$$

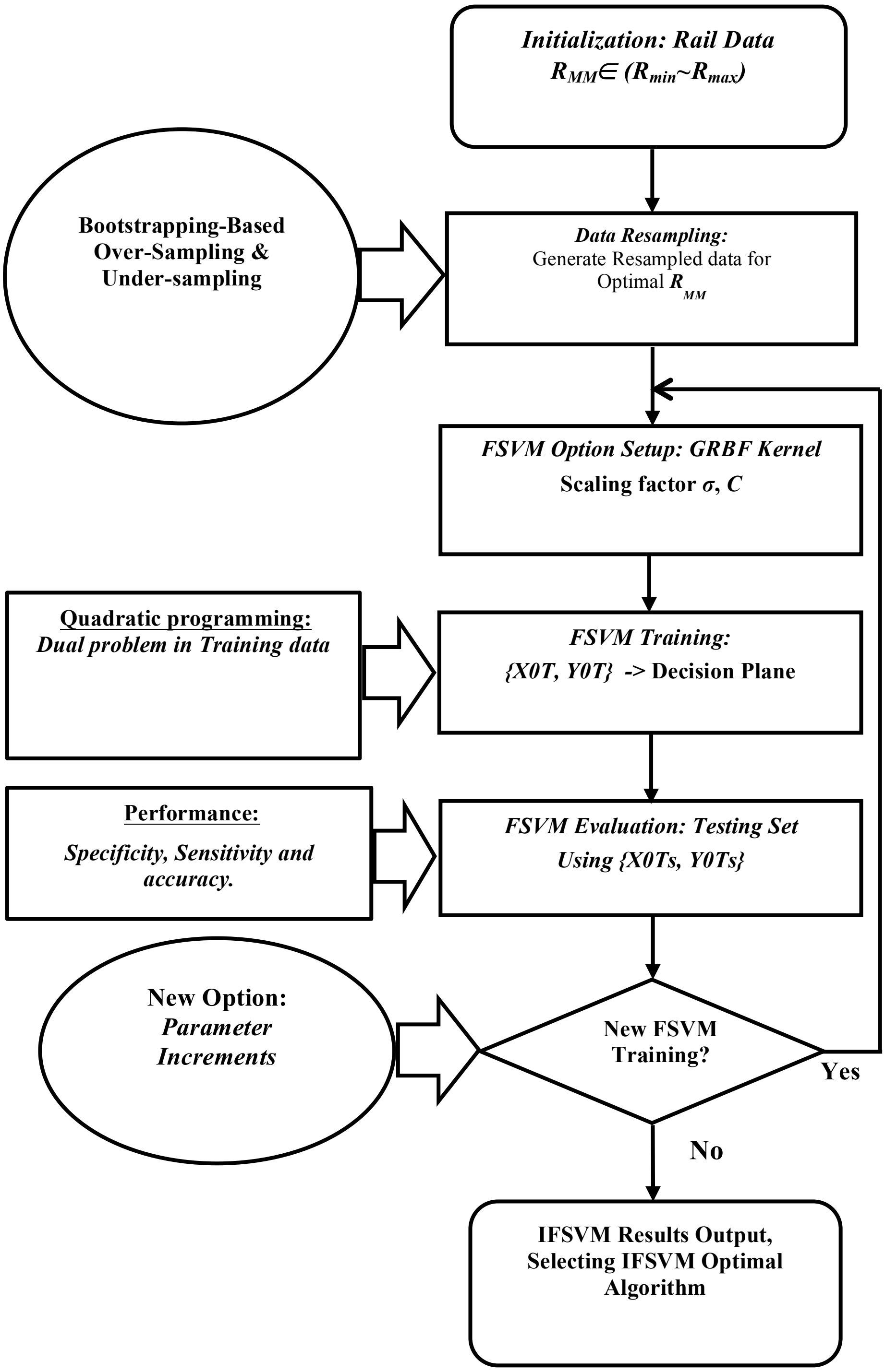
where $s_i\mathcal{E}_i$ is the error measurement with different weighting. The term $\sum_{i=1}^{l} s_i\mathcal{E}_i$ is also referred as a weighted sum of empirical errors to be minimized in when applying fuzzy SVMs. If a misclassified point x_i is not in a mixed cluster, its fuzzy membership s_i is small and hence its error $\mathcal{E}_i$ can be large. However, if x_i is in a mixed cluster, its fuzzy membership is 1 and hence its error $\mathcal{E}_i$ must be small such that $s_i\mathcal{E}_i$ remains minimized. This means that the decision boundary tends to move to overlapping regions to reduce empirical errors in this region. The IFSVM optimization problem can be solved by constructing the dual Lagrangian subject to these constraints:

$$s.t. \textstyle\sum_{i=1}^{l} y_i\alpha_i = 0\,,\ 0 \leq \alpha_i \leq s_iC,\ i = 1, \dots, l \tag{6.9}$$

In the IFSVM algorithm, the iterative strategy is based on the grid search optimization scheme of the regularization parameter (C) and the width of (GRBF). The fuzzy membership function s_i can be a function of a distance between each point and its class center (Lin and Wang 2002) where:

$$s_i = \begin{cases} 1 - \|x_+ - x_i\|/(r_+ + \delta) & if y_i = +1 \\ 1 - \|x_- - x_i\|/(r_- + \delta), & if y_i = -1 \end{cases} \tag{6.10}$$

x_+ , x_- are the mean of both classes and r_+, r_- are the centers of these classes. $\delta > 0$ is a constant to avoid $s_i = 0$. The flowchart of the proposed IFSVM algorithm is depicted in Figure 6.5.



Initialization: Rail Data
$R_{MM} \in (R_{min} \sim R_{max})$
Bootstrapping-Based Over-Sampling & Under-sampling
Data Resampling:
Generate Resampled data for Optimal R_{MM}
FSVM Option Setup: GRBF Kernel
Scaling factor σ, C
Quadratic programming:
Dual problem in Training data
FSVM Training:
{X0T, Y0T} -> Decision Plane
Performance:
Specificity, Sensitivity and accuracy.
FSVM Evaluation: Testing Set
Using {X0Ts, Y0Ts}
New Option:
Parameter Increments
New FSVM Training?
Yes
No
IFSVM Results Output, Selecting IFSVM Optimal Algorithm

Figure 6.5: Overall procedure of the IFSVM model where R_{MM} is the ratio of majority to minority classes, R_{min} and R_{max} are minimum and maximum ratios respectively, σ is the variance parameter of Gaussian function, X0T and X0Ts are training and testing data sets respectively, Y0T and Y0Ts are the output of training and testing sets respectively and C is the regularization parameter that defines constraint violation cost.

For each data point, the membership value influences the misclassification cost where the more important data points are assigned a higher cost and thus, an optimal hyper-plane can be achieved. By solving the dual-optimization problem in (6.8) which is the upper bound of the values of α_i, w and b can be recovered the same way as standard SVM algorithm. The main difference between SVM and FSVM is that the regularization parameter (error penalty) C of FSVM is multiplied by fuzzy membership S_i. In FSVM model, the concept behind is to set a fuzzy membership to each input point and to reformulate SVM so that different input points can make different contributions to the learning of the decision surface. FSVM is also based on the maximization of the margin similar to the classical SVM. However, it uses fuzzy membership function instead of fixed weights to prevent noisy data points from making narrower margins (Wang and Chiang 2007).

There exist several methods that combine the standard support vector machines with fuzzy theories. However, most of these techniques are based on assigning a different misclassification costs that makes different contributions for each data point when finding the separating hyper-planes. The misclassification costs can be defined via applying different fuzzy membership functions. It is important to mention that fuzzy classification methods usually include fuzzification, defuzzification and fuzzy reasoning. However, fuzzy support vector machines employ only fuzzy membership functions and the aforementioned tasks are not involved (Batuwita and Palade 2010).

6.5 Iterative Fuzzy Support Vector Machines-Based Under-Sampling

As stated in the previous section, in IFSVM, different membership values (or weights) are assigned to each data point to reflect their importance. For overcoming the class imbalance problem, a hybrid framework based on the proposed iterative fuzzy support vector machine IFSVM with under-sampling scheme is applied in this section for rails data classification.

A separate optimization problem with the inclusion of membership function s_i is to be performed in the binary IFSVM classifier. It is of great importance to achieve an optimal generalization performance. Training the proposed IFSVM model require tuning the same pre-defined parameters in the standard SVMs i.e., regularization parameter (C) and the width of the Gaussian RBF (σ) in addition to the parameters of the membership function. For a better classification performance, these parameters need to be learned with great care. The fuzzy membership defines the importance of each data point to the overall classification problem where the more important points are assigned a bigger membership value.

The data set was randomly split into training and testing set in the ratio of 7 to 3. For the under-sampling, the rails data is under-sampled as described in chapter 5 until both classes were equal and R_{mm} value of 1 is achieved. Figure 6.6 shows the classification performance of IFSVM algorithm on the under-sampled rails dataset.

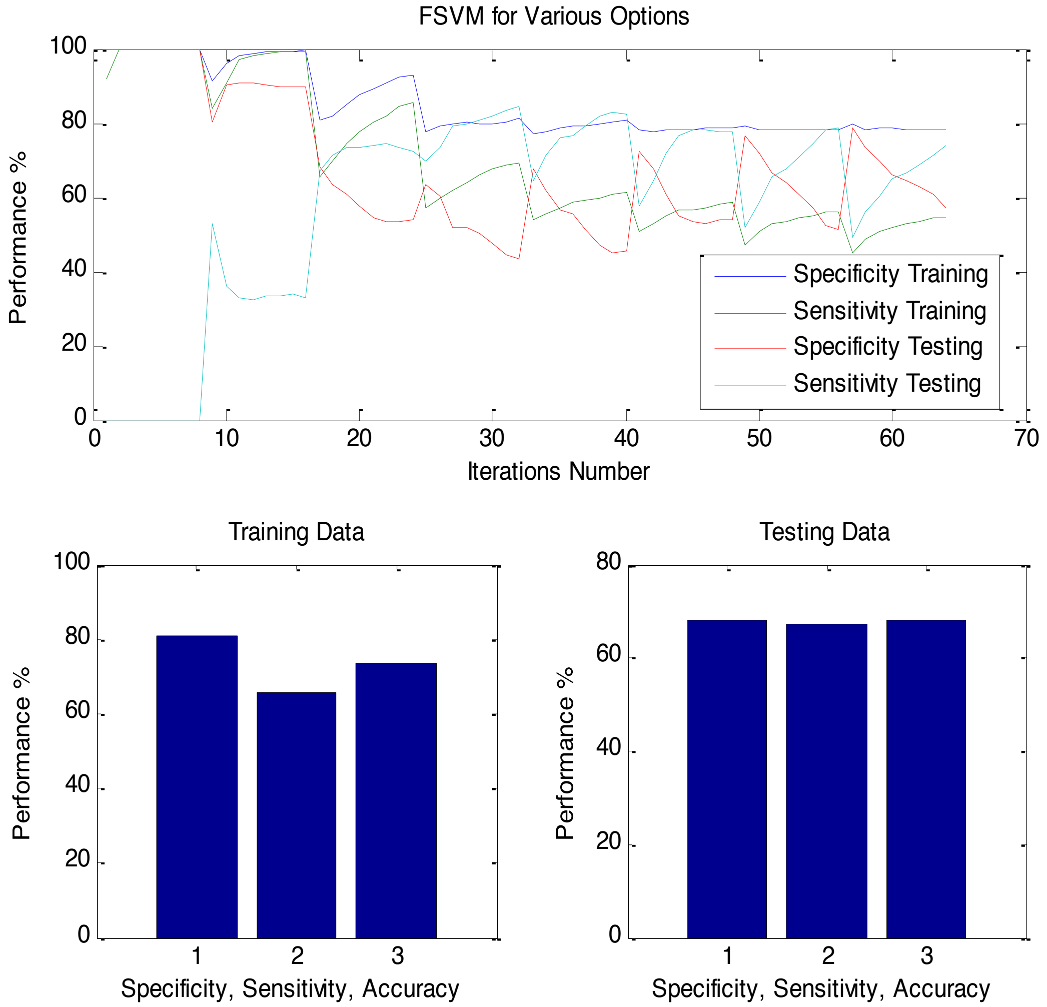


Figure 6.6: Classification Performance of IFSVM with under-sampled data

It can be clearly seen from Figure 6.6 that sensitivity performance improved significantly through data resampling, from about 29% (on original imbalanced dataset) to around 67.2 %. This work has revealed that the IFSVM algorithm with under-sampling runs quicker i.e. 3.48 minutes. The classification accuracy rate of

IFSVM classifier is not only influenced by the two parameters i.e., regularization parameter (C) and the width of the Gaussian RBF (σ), but also other factors, including the quality and dimensionality of datasets.

The parameter selection criteria tailored for the IFSVM algorithm is based on an exhaustive grid search over the parameter space. The maximum number of iterations of the proposed algorithm is controlled via the number of parameters employed in the grid search. Parameters are selected based on the following ranges: $C = \{1, 2,\dots, 15\}$ and $\sigma = \{4, 7, 10, 13, 17, 20\}$. As C increase, then the margin is wide and the tuning parameter is large. Therefore, many observations violate the margin and so there are many support vectors.

Gaussian radial basis function (GRBF) is the best choice for providing superior generalization capability as it has less parameter to be optimized than other nonlinear kernels. Moreover, it is capable of handling the nonlinearity between classes and features. The training time of IFSVM with under-sampling is 3.48 minutes. It can clearly be stated that algorithms-based Under-sampling technique is less time consuming and thus superior. The hybrid architecture of IFSVM with the under-sampling scheme succeeded in drastically reducing the number of support vectors to 2206. The speed of IFSVM classification algorithm depends on the number of support vectors (Manikandan and Venkataramani 2009). Results also support the efficiency of our training algorithm not only on the overall classification performance but also in producing small number of support vectors. IFSVM classification results obtained with under-sampling is presented in Table 6.1.

Table 6.1: Performance result of IFSVM classifier with under-sampling

Classifier	**Sampling Type**	**Data Size**	**# SVs**	**Execution Time**	**sensitivity %**
IFSVM	Under-Sampling	2877	2206	3.48 minute	67.14 %

The above-stated sensitivity analysis ratifies the good results achieved via under-sampling approach, finding a good trade-off between the execution time of the algorithms and the classification performance. The results agree with the hypothesis that under-sampling the majority class reduces the total number of training examples, speeding up the training time and accordingly ensure promising classification performance.

Based on the above obtained results, three observations can be made as follows:

4) A good generalization performance is achieved via The Gaussian kernel. The kernel function potentially maps rails data into a higher dimensional space where as other kernels such as the linear or polynomial uses feature space with fixed number of dimensions.

5) The classification accuracy rate of IFSVM classifier is not only influenced by the two parameters i.e., regularization parameter (C) and the width of the Gaussian RBF (σ), but also other factors, including the quality and dimensionality of datasets.

6) Under-sampling produces datasets with low dimensionality. The class distribution advantages of under-sampling enable IFSVM algorithm to yield small number of support vectors. Consequently, the classification time will be decreased.

Although IFSVM algorithm provides good classification performance with under-sampling, it is important to examine the same algorithm with other sampling technique. Next section will discuss the application of IFSVM with bootstrapping-based oversampling scheme.

6.6 Iterative Fuzzy Support Vector Machines with Bootstrapping-Based Over-sampling

Bootstrapping based-oversampling is another sampling technique that is also applied to change the class distribution of rails data. The advantage of such a technique is that it is external and therefore, easily transportable as well as very simple to implement (Estabrooks *et al.* 2004). Moreover, over-sampling the minority class data avoids unnecessary information loss (Yang *et al.* 2011 a).

Over-sampling has its drawbacks and results in datasets with high dimensionality. Consequently, the computational cost associated with SVM is increased. The classification performance of IFSVM algorithm on the over-sampled rails dataset is presented in Figure 6.7.

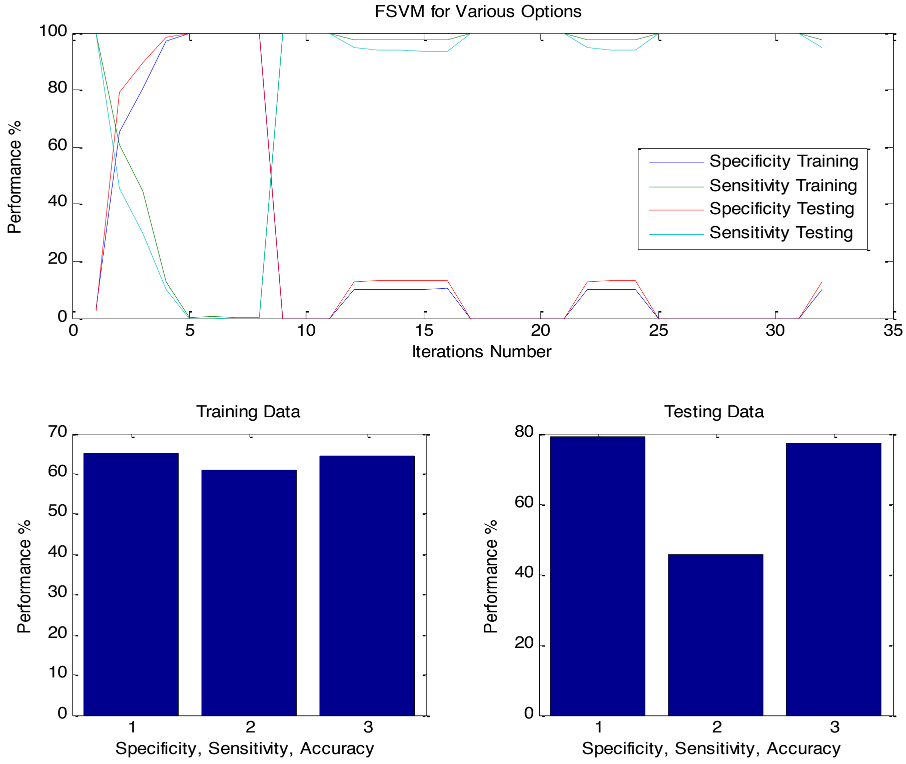


Figure.6.7: IFSVM classification performance with bootstrapping-based over-sampling

The integration strategy of IFSVM with over-sampling is computationally expensive since Over-sampling creates a multiple randomly resampled datasets. Therefore, a significant increment of the overall size of the data is occurred. Due to the dimensionality nature of the data being produced and computer memory restrictions, the optimal sampling rate achieved via bootstrapping based oversampling approach is 5.

The training time of IFSVM with bootstrapping-based over-sampling is 73.28 hours. Bootstrapping-based over-sampling yields a large number of support vectors i.e. 23452. Although it is a solid mathematical structure, it is worth mentioning that the IFSVM technique has drawbacks as it can be computationally expensive when dealing with large scale data and it tends to produce a large number of support vectors as shown in Table 6.2. To overcome this concern, IFSVM-FCMs is presented in the next section.

As shown in Table 6.2, IFSVM is sensitive to the bootstrapping-based over-sampling, which is the only shortcoming of the presented algorithm. Bootstrapping-based Over-sampling causes performance degradation for IFSVM classifier to

47.1% with large number of support vectors being produced and therefore leads to a poor generalization capability.

Table 6.2: IFSVM performance with bootstrapping-based over-sampling

Classifier	Sampling Type	Data Size	# SVs	Execution Time	sensitivity %
IFSVM	Bootstrapping-Based Over-Sampling	30992	27675	73.28 hour	45.6 %

6.7 Iterative Fuzzy Support Vector Machines-Based Fuzzy C-Means Clustering

Class imbalance is not the only problem which tends to govern the performance of the learning algorithms, but there are other elements which potentially hinder the classification performance such as the overall size of the data set. For a better generalization capability of IFSVM and in order to speed up its optimization problem, a hybrid IFSVM-FCMs is presented in this section.

FCMs Clustering seeks to reduce the size of rails dataset. The computational cost of the proposed IFSVM_FCMs is composed of the cost of FCMs clustering, training data size and the cost for searching IFSVM optimal parameters and finding IFSVM model. The speed of IFSVM classification depends on the number of support vectors (Manikandan and Venkataramani 2009), (Kang and Cho 2014).

A distinct feature of SVM and FSVM is their ability to present the solution by means of a small scale dataset of training examples which leads to massive computational advantages. With the aid of fuzzy c-means clustering, the IFSM seeks to reduce the size of dataset and enhance the classification performance. Using the hybrid fuzzy support vector machines based fuzzy C-means clustering algorithm, the existence of global minimum solution with less computations is therefore guaranteed. The architecture of the proposed IFSVM-based FCM clustering algorithm is presented in Figure 6.8.

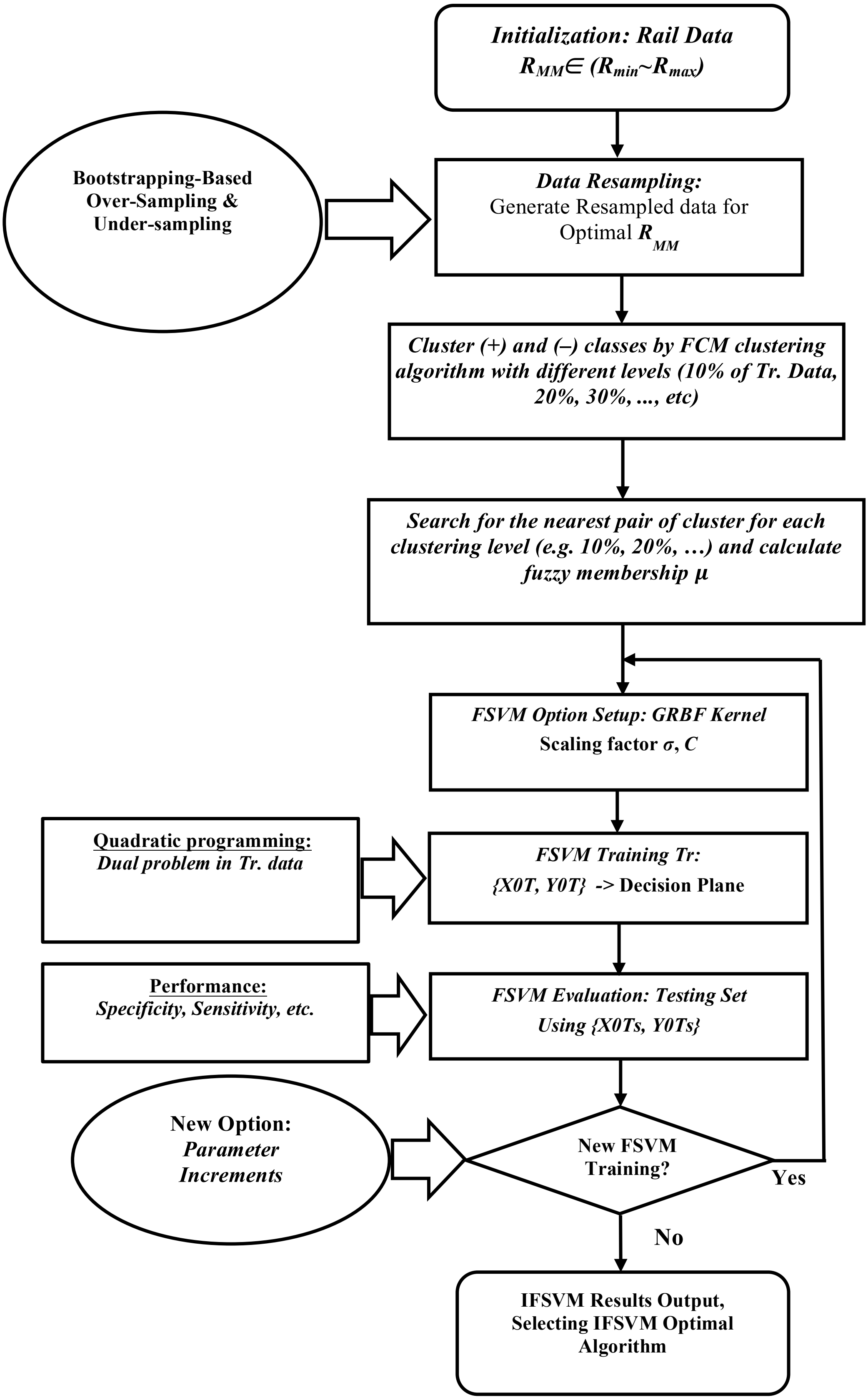

Initialization: Rail Data
$R_{MM} \in (R_{min} \sim R_{max})$
Bootstrapping-Based Over-Sampling & Under-sampling
Data Resampling:
Generate Resampled data for Optimal R_{MM}
Cluster (+) and (–) classes by FCM clustering algorithm with different levels (10% of Tr. Data, 20%, 30%, ..., etc)
Search for the nearest pair of cluster for each clustering level (e.g. 10%, 20%, …) and calculate fuzzy membership μ
FSVM Option Setup: GRBF Kernel
Scaling factor σ, C
Quadratic programming:
Dual problem in Tr. data
FSVM Training Tr:
{X0T, Y0T} -> Decision Plane
Performance:
Specificity, Sensitivity, etc.
FSVM Evaluation: Testing Set
Using {X0Ts, Y0Ts}
New Option:
Parameter Increments
New FSVM Training?
Yes
No
IFSVM Results Output, Selecting IFSVM Optimal Algorithm

Figure 6.8: Architecture of IFSVM-based FCM clustering algorithm where where R_{MM} is the ratio of majority to minority classes, R_{min} and R_{max} are minimum and maximum ratios respectively, $\boldsymbol{\sigma}$ is the variance parameter of Gaussian function, X0T and Y0T are training and testing data sets respectively and C is the regularization parameter that defines constraint violation cost.

FCM clustering is performed on the down-sampled and bootstrapping-based over-sampled dataset which has 2877 and 30992 data points respectively. Weighting exponents, m, of 2 with random initial number of clusters are used for the FCM approach. Parameters are selected with the same ranges as the standard SVM presented in chapter 5 where $C = \{1, 2,\ldots, 15\}$ and $\sigma = \{4, 7, 10, 13, 17, 20\}$. Due to memory restrictions, only two clustering levels i.e., 10 % and 20 % are considered with 10 % represents the least number of cluster centers and results in the minimum number of training points. Figures 6.9 and 6.10 show the performance of the proposed IFSVM algorithm on the over-sampled dataset with different FCM clustering levels (10% and 20 % respectively).

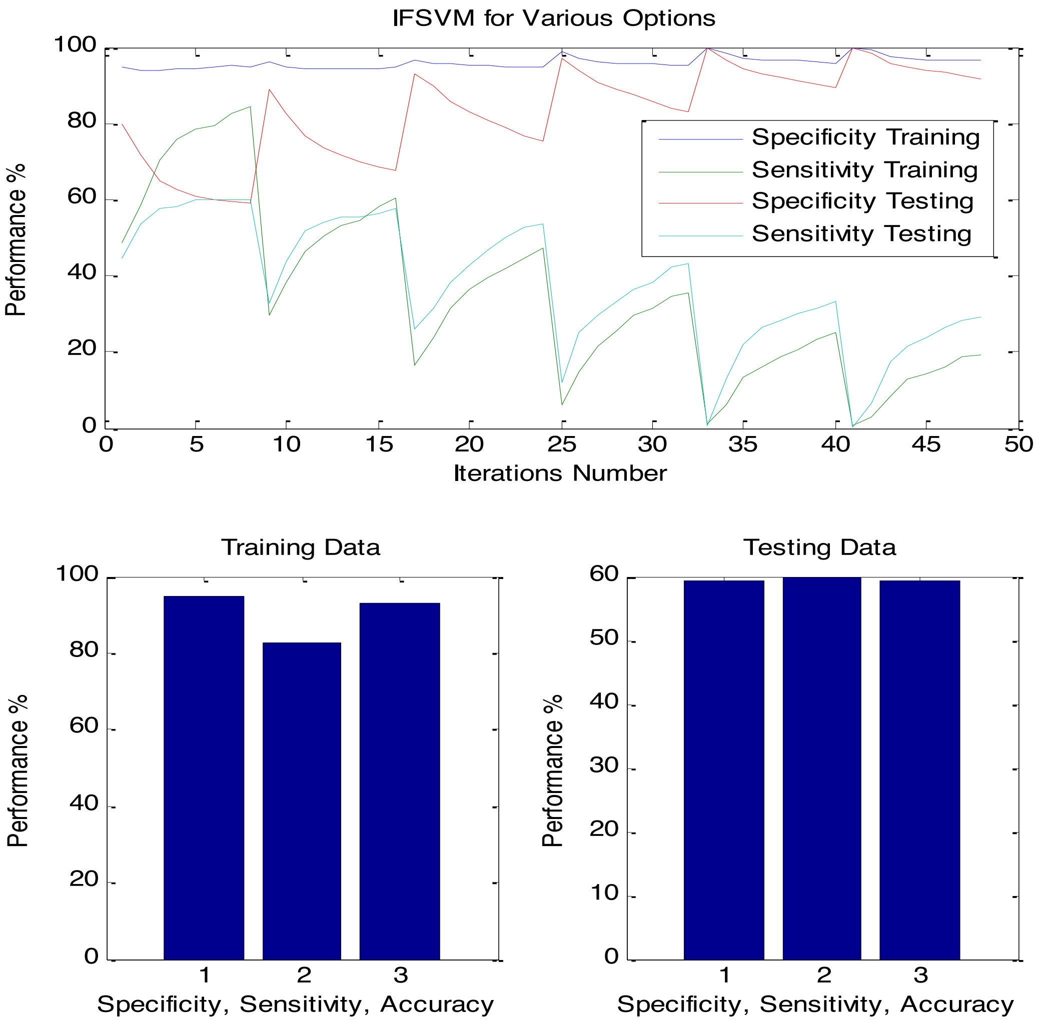


Figure 6.9: Performance of IFSVM-FCM algorithm with 10% of data points

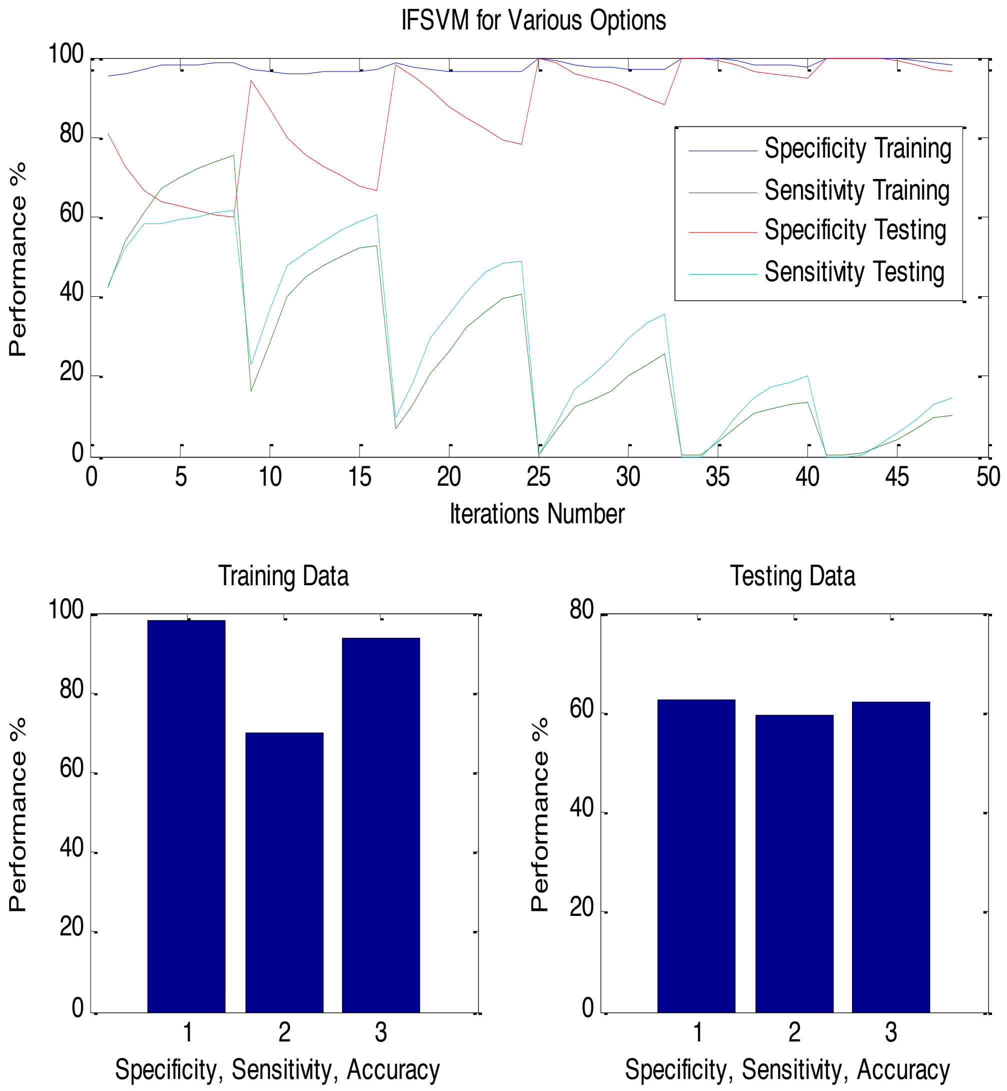


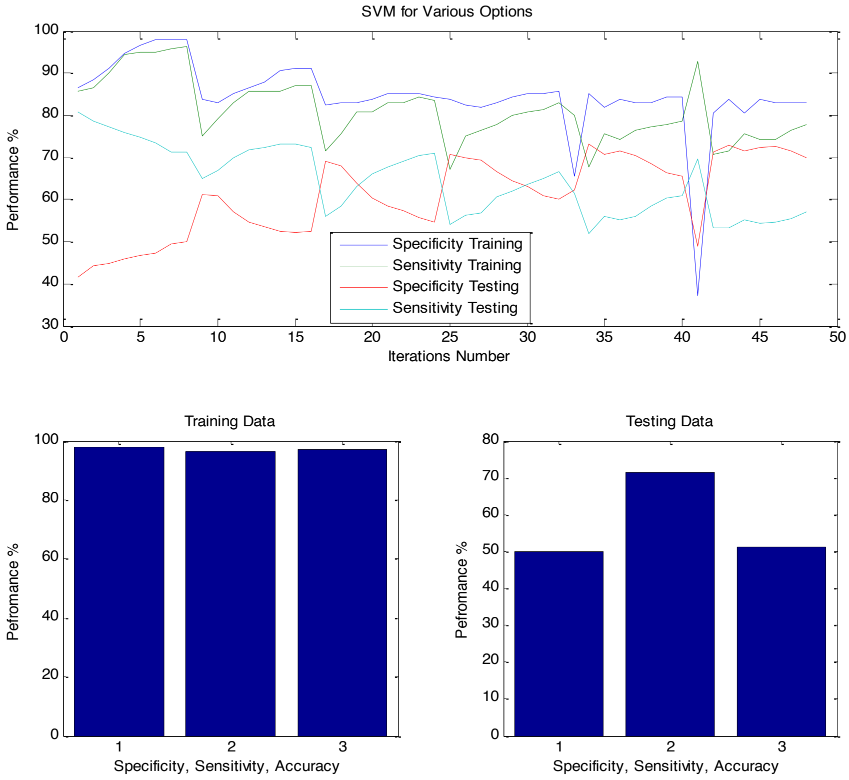
Figure 6.10: Performance of IFSVM-FCM algorithm with 20% of data points

The integration strategy of IFSVM and FCM clustering led to promising results as shown in Table 6.3. Having the rails data clustered and fuzzified, the IFSVM is able to build the model using only 10% and 20 % of the training examples and therefore reducing the number of support vectors. Such reduction in the training points has also reduced the model's training time. In table 6.3, it is clearly shown that the IFSVM based FCM clustering performs better than the IFSVM presented in Section 6.6. The results illuminate the proposed IFSVM-FCM have a better generalization capability with a sensitivity of 60 % in contrast to the IFSVM which led to only 45.6 % as shown in Table 6.2.

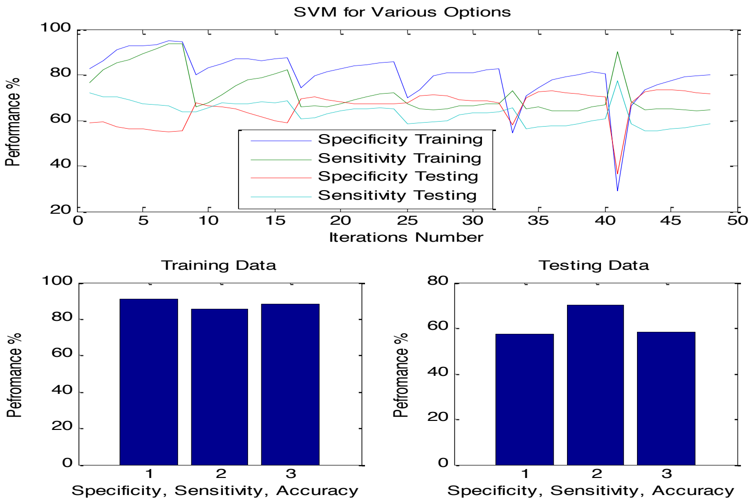
Table 6.3: IFSVM-FCM algorithm with Different Clustering Levels

Classifier	Original Data Size	Clustering [% of Max. # of data pts.]	New Training Data Size	# SVs	Execution Time	sensitivity %
IFSVM-FCM	30992	10 %	3099	1478	4.32 (minute)	60.0 %
IFSVM-FCM	30992	20 %	6199	2654	15.02 (minute)	59.55 %

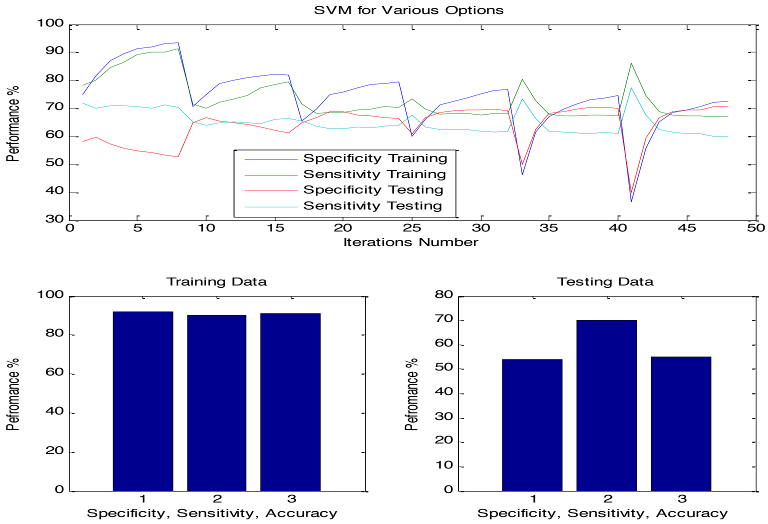
For the down-sampled dataset, the proposed IFSVM-FCM algorithm has shown a good classification performance in contrast to iterative SVM and IFSVM algorithms where 10 % to 60 % of data clustering levels are applied. Figure 6.11 shows the classification performance of the hybrid IFSVM-FCM algorithm with various data clustering ratios i.e., 10%, 20%, 30%, 40%, 50%, and 60% of the original training dataset.



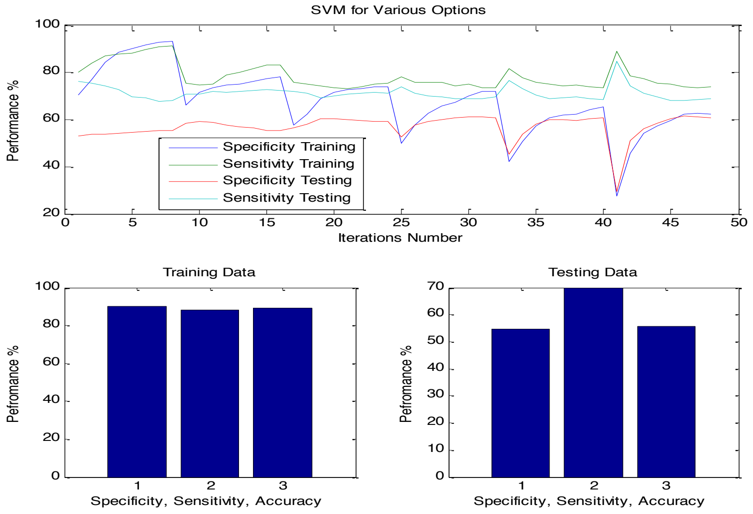
(a) Classification performance of 10% of original data



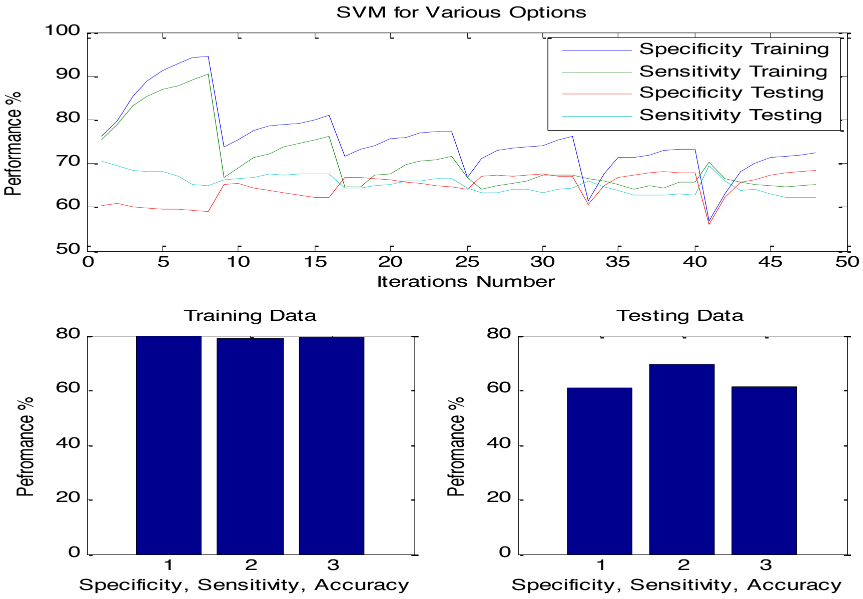
(b) Classification performance of 20% of original data



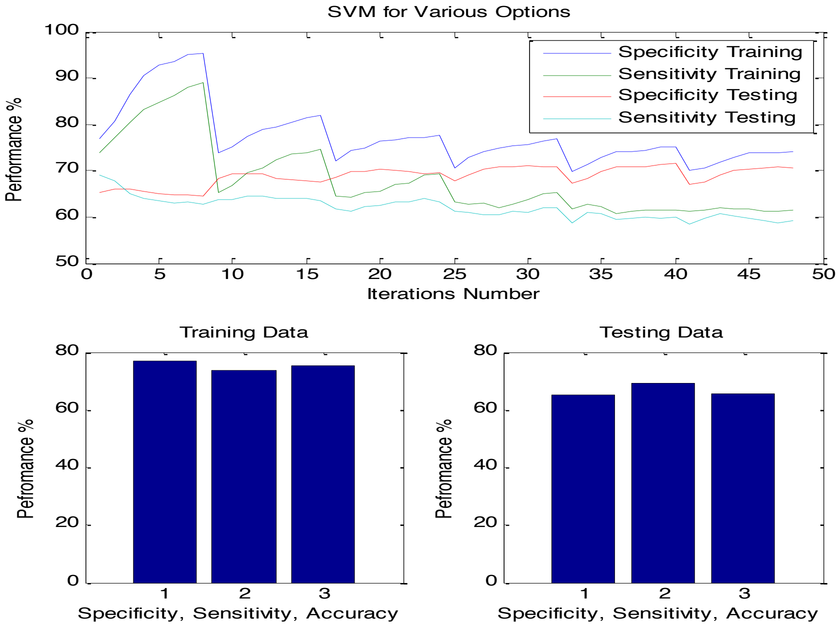
(c) Classification performance of 30% of original data



(d) Classification performance of 40% of original data



(e) Classification performance of 50% of original data



(f) Classification performance of 60% of original data

Figure 6.11: Performance of IFSVM-FCM algorithm with different clustering levels of the original training dataset, where (a) represents the highest clustering level which is 10% of the original data set and (f) represents the lowest clustering level with only 60 % of the original dataset.

The integration strategy of IFSVM and FCM clustering technique applied on the down-sampled dataset has led to promising results as shown in Table 6.4. Having the rails data clustered and fuzzified, the IFSVM is able to build the model using 10%, 20 %, …, 60% of the original training examples and therefore reducing the number of support vectors. Such reduction in the training points has also reduced the model's training time. In Table 6.4, it is clearly shown that the proposed IFSVM-FCM have a better generalization capability on down-sampled dataset with a maximum sensitivity of 71.9 % with 10 % of data points in contrast to the IFSVM which led to only 67.14 % as shown in Table 6.1. FCM clustering proves its ability to overcome the over fitting problem, reduces the number of support vectors and ensures a good classification performance.

Table 6.4: IFSVM-FCM algorithm with Different Clustering Levels

Classifier Type	Original Data Size	Clustering [% of Max. # of data pts.]	New Training Data Size	# SVs	Execution Time	sensitivity %
IFSVM-FCM	2877	10 %	288	215	2.51 (minute)	71.93 %
IFSVM-FCM	2877	20 %	575	436	2.83 (minute)	69.89 %
IFSVM-FCM	2877	30 %	864	671	2.95 (minute)	69.79 %
IFSVM-FCM	2877	40 %	1151	902	3.20 (minute)	69.48 %
IFSVM-FCM	2877	50 %	1439	1111	3.45 (minute)	69.38 %
IFSVM-FCM	2877	60 %	1726	1327	3.60 (minute)	68.77 %

6.8 Comparative Analysis and Models Evaluation

As stated earlier, the support vector machine algorithm is sensitive to the class imbalance learning (Batuwita and Palade 2010b); (Lin and Wang 2002). Class imbalance is not the only problem which tends to govern the performance of the learning algorithms, but there are other elements which potentially hinder the classification performance such as the overall size of the data set.

In this Chapter, an iterative fuzzy support vector machine IFSVM framework is proposed for rails data classification and evaluates its performance by using the confusion matrix given in Chapter 5. Two datasets, generated from bootstrapping-based over-sampling and under-sampling techniques, were utilized for data resampling. A poor generalization capability is obtained when applying the proposed IFSVM on the original dataset where the model's performance is skewed towards the majority class where the minority class is poorly classified at less than 25%.

Having the data resampled using bootstrapping-based over-sampling and under-sampling, the proposed IFSVM's classification performance is improved and a small number of support vectors were produced. Sensitivity performance improved significantly through data resampling. With under-sampling where R_{mm} is a value of 1, the IFSVM algorithm has shown a good generalization with significant classification performance increment of 67.14 %. Under-sampling technique succeeded in drastically reducing the number of support vectors of IFSVM classifier to 2206. The advantages of a fewer support vectors will mostly mean shorter computational times and smaller memory requirements (Zheng *et al.* 2013). RBF could be seen as a sensible alternative to other kernels such as polynomial function.

The employment of the Bootstrapping-based over-sampling technique causes performance degradation to 45.6 % and thus weak generalization capability, because time complexity grows dramatically as the size of the data increase. The IFSVM model built from an over-sampled data yields a large number of support vectors i.e., 27675. The experimental results show that Bootstrapping-based over-sampling increases the computational cost associated with IFSVM training algorithm.

To avoid the computational cost associated with IFSVM when classifying the over-sampled dataset, A new IFSVM-based FCM clustering has been proposed as shown in section 6.7. The superiority of the proposed IFSVM-FCM algorithm is demonstrated by two datasets. The results on the bootstrapping-based over-sampling and under-sampling datasets have remarkable increment of sensitivity rate up to 71.3 % with great reduction in training time. Figures 6.12 and 6.13 show a performance comparison of ISVM, IFSVM and IFSVM-FCM algorithms with under-sampled and oversampled data respectively.

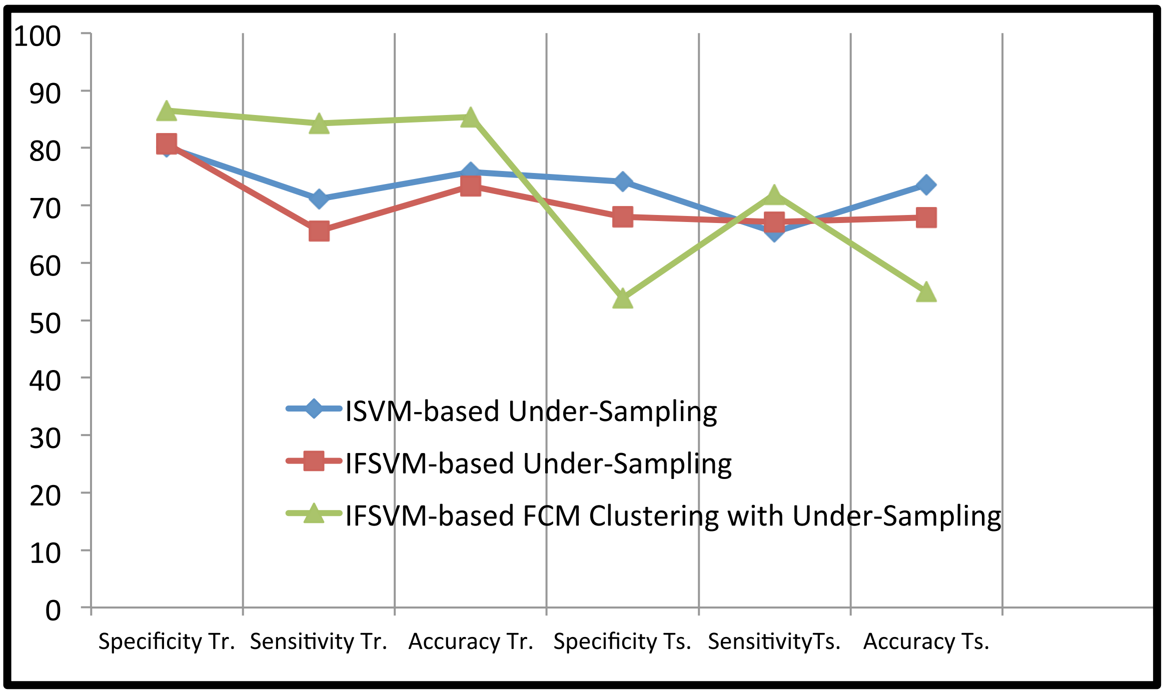


Figure 6.12: Performance comparison of ISVM, IFSVM and IFSVM-FCM algorithms with under-sampling

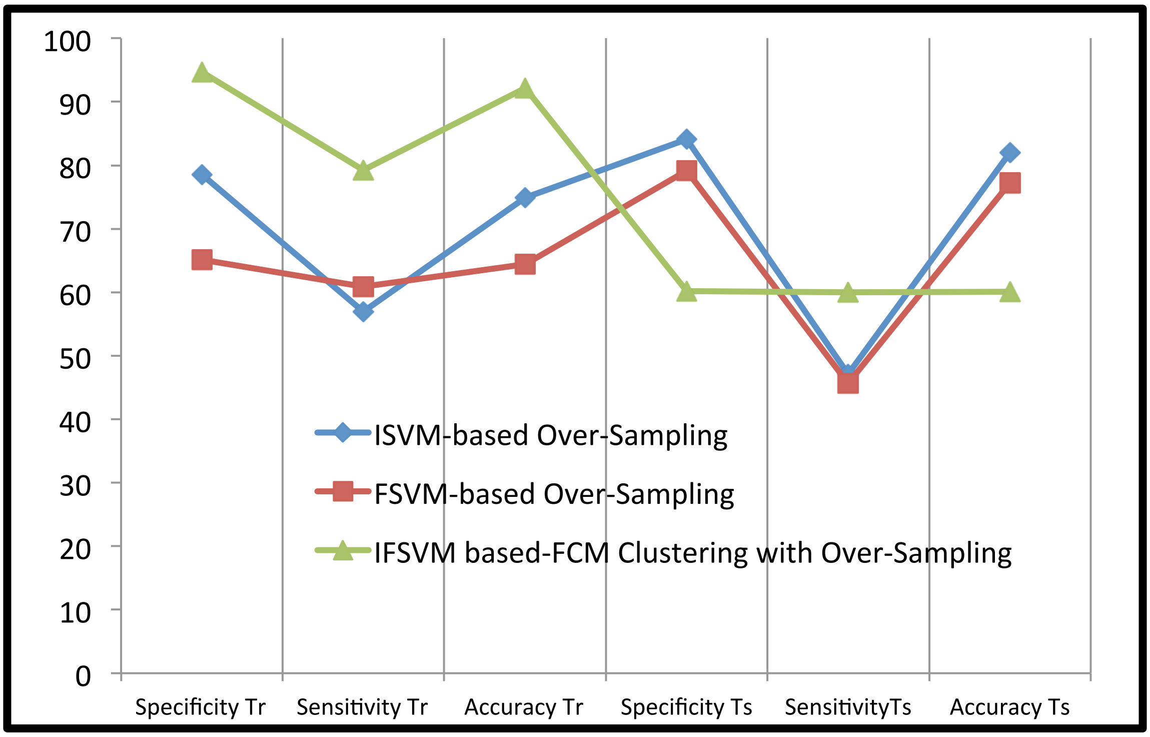


Figure 6.13: Performance comparison of ISVM, IFSVM and IFSVM-FCM algorithms with bootstrapping-based over-sampling

6.9 Summary

In this chapter, an efficient IFSVM based data resampling is proposed. Resampling strategies provide balanced rails datasets for active learning at each iterative step instead of utilizing the original imbalanced dataset. The proposed method includes the class distribution advantages of efficient data sampling and the unique learning mechanism of IFSVM. Two datasets i.e., under-sampled rails data and over-sampled rails data are employed for a successful IFSVM classification. The above-stated sensitivity analysis approves the good results achieved via under-sampling approach, finding a good trade-off between the execution time of the algorithms and the classification performance.

Another contribution presented in this chapter is defined as iterative fuzzy support vector machines-based fuzzy C-means clustering. Fuzzy C-means clustering enables IFSVM learning strategy to be applied on small scale dataset without high computational costs. Instead of using large dataset, FCMs presented in section 6.3 is used to provide the classifier with a smaller dataset.

Based on experimental results, observations have shown that not all support vectors are needed for finding an accurate separating hyper-plane. To capitalize on this fact, support Vectors Reduction method based on FCMs clustering has been developed to drop the weakest SVs. This method has significantly increased the effectiveness of the iterative FSVM and successfully yield a good generalization capability. It should also be noted that FCMs clustering inhibit the computational complexity of the proposed iterative FSVM and thus improve sensitivity analysis since the complexity of the computations is proportional to the number of support vectors SVs (Burges 1996).

All computations are carried-out on a computer with 3.10 GHz Intel core i5 processor, 64 Bits Windows 7 professional operation system and a maximum of 16.0 GB memory. All the programs are developed using Bioinformatics Toolbox in Matlab R2011a and RapidMiner 5 (free access software).

Chapter 7

Conclusions and Future Work

This chapter summarises the major contributions of this thesis and projects into the future with suggestions for further work.

7.1 Thesis Summary and Main Contributions

The overall objective of the present work is to contribute to the scientific research and technological development by investigating rails quality data modelling related problems in the engineering disciplines from control systems' perspective. This work can also be viewed as a contribution in the process of meeting the needs of real world industrial processes in terms of reducing manufacturing costs and enhancing process yields given that it highlights some of the most important aspects of complex data modelling that must be considered for the process of rails production. The achievements of this thesis can be summarized thus:

The rails manufacturing route operated by Tata Steel Europe can be considered as a complex process whose performance in terms of expected quality of delivered products and cost management are of a strategic importance. The rails manufacturing process compromise multiple interacting sub-processes with many wide-ranging characteristics. Therefore, the core challenge in the rails production line is an adversity to construct machine learning-based paradigms to predict the dynamics of the system. Knowledge-based approaches can also become challenging when one needs to specify the influence (correlation) of data variables on the final product.

Further complications can arise when the basic pre-processing tasks become computationally expensive and resource-intensive due to the volume and complexity of rails data. Consequently, modelling and optimisation procedures and algorithms need to be tailored specifically to alleviate such problem. In the light of this, the choice of designing adequate data analysis tools becomes crucial due to the fact that a single computing algorithm cannot successfully provide all requirements in the rails data modelling cycle. Therefore, an integrated software environment has been carefully designed to perform rails data analysis and modelling including:

- Clementine Data Mining (Tata Steel Corus Server);
- Microsoft Access 2007;
- Excel for data visualisation and basic pre-processing;
- RapidMiner for data mining and knowledge extraction;
- Matlab for advanced rails data modelling.

In this thesis, there are four contributions related to data mining and machine learning-based paradigms to address the aforementioned challenges in rails manufacturing process data with the goal of improving data efficiency and generalization performance of proposed learning algorithms. Figure 7.1 illustrates the main contributions of this thesis which have already been discussed explicitly within three chapters.

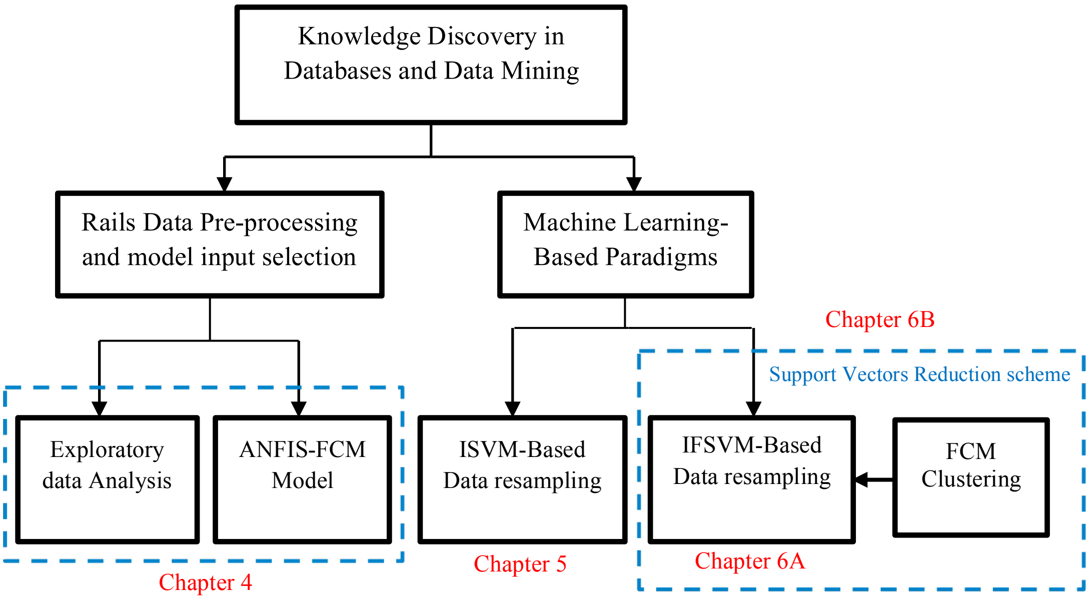

Knowledge Discovery in Databases and Data Mining
Rails Data Pre-processing and model input selection
Machine Learning-Based Paradigms
Chapter 6B
Support Vectors Reduction scheme
Exploratory data Analysis
ANFIS-FCM Model
ISVM-Based Data resampling
IFSVM-Based Data resampling
FCM Clustering
Chapter 4
Chapter 5
Chapter 6A

Figure 7.1: Illustration of the main contributions in this thesis. Chapter 4 discusses the exploratory data analysis and ANFIS-based FCM model whereas Chapters 5 and 6 describe the machine learning-based paradigms (6B illustrates the combination between machine learning and FCM clustering)

This research had been initiated by adopting the main concept of exploratory data analysis (EDA). Exploratory data analyses are capable of formation, consolidation and analysis of large amounts of data; this volume of data could not be practically visualized and analysed by hand within a reasonable timeframe. An initial inspection of rails data has revealed that simple tasks, copying, paste and plotting, cannot be executed in a straightforward manner because they often violate the specifications and norms of most commonly used software. Therefore, an adequate choice of exploratory data analysis tools i.e., RapidMiner and Matlab becomes essential since a single analysis and computing tool cannot best serve the diversified requirements needed in the data modelling cycle. A robust model input selection and rails data reduction environment based on correlation analysis and ANN has been designed to overcome the high dimensionality and complexity problem of rails manufacturing data. Data pre-processing and feature selection schemes have extracted an optimum subset of variables from the original rails data based on their influence and importance to the process.

Bootstrapping-based data over-sampling scheme is utilized to deal with the imbalanced problem in rails dataset where the distributions of class labels are not same. The imbalance problem has a serious impact on the overall performance of models and classifiers. The use of data mining and Knowledge discovery in databases i.e., RabidMiner neural networks and neuro fuzzy-based ANFIS approaches are presented as rails data modelling frameworks. The use of exploratory data analysis (data pre-processing and correlation analysis) and the feedforward neural network input selection model have proved to be very effective in selecting the most significant input variables with serious implications on the overall performance of RabidMiner Neural networks and neuro fuzzy-based ANFIS modelling approaches. The bootstrapping-ANFIS algorithm yields good prediction performance when using balanced data set with 65 %.

A new algorithm termed the iterative support vector machine (ISVM) for classification problems has been presented. The ISVM combine the good generalisation ability with a technique to address the curse of dimensionality. such combination results in a global quadratic optimisation problem where the box constraints are readily solved via sequential minimal optimization SMO method in the higher dimensional feature space. The implementation of SVM theory is not straightforward. Even though there exist specific packages tailored to solve such problem, these can effectively be applied on small-scale problems because of memory and training-time restrictions. To overcome such concerns and to make a large-scale classification possible, the sequential minimal optimization SMO technique is applied. SMO (Platt 1998) decomposes the training problem into sub-tasks of optimizing only two a_i in each optimization phase in which they are easy and fast to find where the rest of a_i s are kept fixed. SMO technique succeeded in drastically reducing the training time. However, the relation between data size and training time still remains nonlinear. Another important issue is data normalization; a successful data classification requires normalizing the training set, otherwise expensive computations will occur and in a few applications the optimization phase will fail to converge. The effectiveness of the proposed approach (ISVM) has been demonstrated on two rails datasets generated via bootstrapping-based over-sampling and under-sampling. The solution of the proposed ISVM is achieved as a sparse set of support vectors. These lie on the separating hyper-planes and as such summarise the information required to classify rails manufacturing data.

The Radial basis function (RBF) Kernel-type applied in this thesis with machine learning-based paradigms could be seen as a sensible alternative to the use of complex polynomials. Radial basis function offer superior generalization performance. Other functions such as polynomial and linear employs a feature space in a way that the number of dimensions is fixed whereas the RBF has the potential to map rails manufacturing process data into higher dimensions, which instinctively offers it better flexibility.

Inspired by the idea of margin maximization to promote the generalisation capacity of the ISVM classifier, a fuzzy margin is proposed and optimized to boost the generalisation capacity of ISVM. Following this line, a new iterative fuzzy support vector machine for rails data classification is proposed. The idea of SVMs is completely reformulated into a new fuzzy support vector machine by assigning a fuzzy membership function to each data point. This learning process is carried out immediately after the grid parameter selection of support vector machine.

In this thesis, the performance of two powerful machine learning classifiers, the proposed Iterative SVM and Iterative FSVM, is compared each with bootstrapping-based over-sampling and under-sampling and evaluate their performance by using the confusion matrix given in chapter 5. Two datasets, generated from bootstrapping-based over-sampling and under-sampling techniques, were utilized in this work.

As shown in the previous chapters, the results support the efficiency of the proposed algorithms where Sensitivity performance improved significantly through data resampling. The IFSVM algorithm gives strong generalization ability with significant classification performance of 67.14 % with the down-sampled dataset, Whereas, ISVM achieves a classification performance of 65.3%. Bootstrapping-based Over-sampling causes performance degradation for SVM and IFSVM classifiers to 47.1% and 45.6% respectively and therefore leads to a poor generalization capability.

The integration strategy with the under-sampling scheme succeeded in drastically reducing the number of support vectors of ISVM and IFSVM to 2171 and 2206 respectively. In contrast, bootstrapping-based over-sampling yields a large number of support vectors i.e. 23452 and 27675 respectively. The speed of ISVM and IFSVM classification depends on the number of support vectors (Manikandan and Venkataramani 2009), however, both the ISVM and IFSVM algorithms are sensitive to the bootstrapping-based over-sampling, which is the only shortcoming of the presented algorithm.

This work has revealed that the IFSVM algorithm with under-sampling runs quicker i.e. 3.48 minutes as compared to the bootstrapping-based over-sampling which takes 73.28 hours. The training time of ISVM on under-sampling and bootstrapping-based over-sampling is 3.67 minutes and 77.86 hours respectively. On the basis of the results drawn by these experiments, it can clearly be stated that algorithms-based Under-sampling technique are less time consuming than that with bootstrapping based over-sampling and hence are superior.

In classification problems, a distinct feature of support vector machine and fuzzy SVM is that expressing the solution by means of sparse subset of training points called support vectors that give superior computational advantages. Utilizing the epsilon intensive loss function, the existence of the global minimum solution is guaranteed with good optimization of reliable generalization bound.

Some solutions to the class imbalance problem have been presented at data level. At the data level, two re-sampling techniques are employed to balance class distribution of rails data, including bootstrapping based over-sampling minority class instances and under-sampling majority class instances. Although these external techniques have been showed to achieve success on rails data, over-sampling still confronts over-fitting problem and under-sampling has to eliminate useful information potentially and this might lead to poor generalizations when applying on other real applications.

Having the size of training data increased, the number of support vectors increases. This prolongs the time required for classification since the contribution of every support vector needs to be calculated individually. Consequently, a new iterative fuzzy support vector machines based fuzzy C-means clustering IFSVM-FCM is proposed. The problem of scaling the IFSVM to handle large-scale datasets becomes apparent when applying FCM clustering scheme. The integration strategy of IFSVM and FCM clustering led to promising results. The IFSVM is able to build the model using only 10% and 20% of the training examples and therefore decreasing the number of support vectors. Such reduction in the training points has also reduced the model's training time. The proposed IFSVM-based FCM clustering have a better generalization capability on down-sampled dataset with a maximum sensitivity of 71.9 % with 10 % of data points in contrast to the IFSVM which led to only 67.14 %.

There are several motivations that drive this research to focus on support vector machine (SVM) and fuzzy support vector machine (FSVM) approaches as supervised learning techniques. SVM is a useful tool for data analysis and classification, in the case of non-regularity in the data, for example when the data are not regularly distributed or have an indefinite distribution. It can help evaluate information hidden in the data. The SVM is derived from statistical learning theory and employs the structural risk minimization (SRM) principle which can considerably enhance SVM's generalization capability (Xia *et al.* 2005). SVM is dimensionally independent whereas other machine learning techniques such as neural networks are not. SVM modelling process is unaffected by the number of observations encountered in the training data set. Accordingly, the 'curse of dimensionality' is avoided (Trotter *et al.* 2001). Unlike previous machine learning algorithms, such as the multilayer neural network classifier which has numerous local minima, the determination of the SVM model parameters corresponds to a convex optimization problem, and so any local solution is unique global optimum

(Bishop 2006). Additionally, the SVMs have been found to be very robust to outliers. The margin parameter C controls the misclassification error and so the outliers can be supressed by choosing a proper value to C (Tang and Qu 2008). While in contrast, Multilayer neural network classifiers are vulnerable to outliers because they use the sum of square error and so the outliers need to be removed to prevent their influence before training.

7.2 Future Work

Engineering research in rails manufacturing route is still an emerging area and there are still various open problems require to be improved. Based on the finding of this research, several suggestions for future works are summarized in the following paragraphs.

Despite the fact that machine learning-based paradigms proposed in this thesis have proven to be a promising choice of rails data modelling, it is crucial to compare and evaluate the performance of these paradigms with other supervised learning techniques. Other supervised learning techniques may include artificial intelligence (logic-based techniques, perceptron-based techniques) and statistics (Bayesian networks, Instance-based techniques). This line of research may focus on decision trees, neural networks, bayes networks and other forms of SVMs such as least square support vector machines or/and lagrangian support vector machines.

Support vector machines are relatively slow when applied to large-scale problems. Training the proposed ISVMs and IFSVMs require Sequential minimal optimization SMO technique for solving the optimization problem to tackle this concern which has led to superior generalization abilities. Future research may include other solutions to solve the constrained optimization problem of SVMs such as chunking, decomposition algorithm and genetic algorithm. However, these techniques may well lead to further training complexity. A potential further study may investigate other data sampling and data compressing techniques so that the proposed algorithms can be examined extensively since the class imbalance problem and large-scale datasets could seriously detriment to the prediction performance of most classification techniques. The generalization performance, speed and complexity of machine learning-based paradigms are also governed by the size of the data. However, reducing the size of the underlying data via other data reduction techniques will drastically reduce the number of support vectors and may therefore translate into less complex models with better classification.

A further direction of research is to consider different fuzzy membership functions and kernel functions for rails data classification. The fuzzy memberships may be derived as functions of higher dimensional feature space instead of using it in the input space. However, these may require further parameter optimisation. Final important avenue of future work is to exploit the model by inverting its structure via multi-objective optimisation frameworks to create particular processing routes meant for 'right-first-time' manufacturing of rails.

Appendix A

Kernel Requirements

A.1 Symmetry Condition

Let $K(x, y)$ be a real symmetric function on a finite input space, then it is a kernel function if and only if the matrix K with components $Kx_i, y_j)$is positive semi-definite (Campbell 2000).

A.2 Mercer's Theorem

This theorem must be satisfied by a functional for a pair Φ, $\mathcal{H}$ to exist.

For a compact subset, $C \in \Re^N$, we have:

If $K(x, y)$ is a continuous is a continuous symmetric kernel of a positive integral operator T (Campbell 2000), i.e.,

$$(Tf)(y) = \int_C K(x,y)f(x)\,dx \qquad (A.1)$$

With:

$$\int_{C\times C} K(x,y)\,f(x)\,f(y)\,dx\,dy \ \geq 0 \qquad (A.2)$$

For all $f \in L_2(C)$ then it can be extended in a uniformly convergent series in the eigen functions Φ_j and positive eigenvalues λ_j of T, therefore:

$$K(x,y) = \sum_{j=1}^{N} \lambda_j\,\Phi_j(x)\Phi_j(y) \qquad (A.3)$$

Where N represents the number of positive eigenvalues.

This theorem generalizes the requirement to infinite feature space and hold for general compact spaces. The semi-positive condition for finite spaces presented in the theorem in section (A.1) is generalized via equation (A.2) whereas expression in (A.3) represents the generalization of the usual concept of an inner product in reproducing Hilbert spaces in which each dimension is scaled by $\sqrt{\lambda_j}$.

For specific practical cases, it is important to note that satisfying Mercer's condition is not straightforward. Equation (A.3) must hold for every function *f* with finite L2-norm (i.e. which satisfies (A.3)).

References

Akbani, R., Kwek, S. and Japkowicz, N., 2004. Applying Support Vector Machines to Imbalanced Datasets. *In: proceedings of European Conference on Machine Learning: ECML*, pp.39–50.

Alpaydim, E., 2010. *Introduction to Machine Learning*, The MIT Press.

Altınkaya, H., Orak, İ.M. and Esen, İ., 2014. Artificial neural network application for modeling the rail rolling process. *Expert Systems with Applications*, 41(16), pp.7135–7146.

Asiltürk, İ. and Çunkaş, M., 2011. Modeling and prediction of surface roughness in turning operations using artificial neural network and multiple regression method. *Expert Systems with Applications*, 38(5), pp.5826–5832.

Babushka, I., Chandra, J. and Flaherty, J., 1983. *Adaptive Computational Methods for Partial Differential Equations*, SIAM, Philadelphia.

Barua, S., Islam, M., Yao, Xi., Murase, K., 2014. MWMOTE - Majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering*, 26(2), pp.405–425.

Batista, G.E. a. P. a., Prati, R.C. and Monard, M.C., 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1), p.20.

Batuwita, R. and Palade, V., 2010a. Efficient resampling methods for training support vector machines with imbalanced datasets. *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*.

Batuwita, R. and Palade, V., 2010b. FSVM-CIL: Fuzzy Support Vector Machines for Class Imbalance Learning. *IEEE Transactions on Fuzzy Systems*, 18(3), pp.558–571.

Berkhin, P., 2002. *A survey of clustering and data mining techniques*, Accrue Software, San Jose, CA.

Bezdek, J.C., 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum.

Bishop, C.M., 2006. *Patteren recognision and machine learning (Vol. 4, No. 4).*, New York: Springer.

Boser, E., Vapnik, N., Guyon, I., Laboratories, T., 1992. Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th annual workshop on computational learning treory. ACM.* pp. 144–152.

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

Braha, D. and Shmilovici, A., 2002. A Data Mining for improving a cleaning process in the semiconductor Industry. *IEEE transaction on semiconductor manufacturing*, 15(1), pp.91–101.

Breiman, L., 1996. Bagging predictors. *Machine Learning*, 24, pp.123–140.

Buhmann, J.M., 1995. Data clustering and learning. *The Handbook of Brain Theory and Neural Networks*, pp.278–281.

Burges C. J. C, 1998. A Tutorial on Support Vector Machines for Pattern Recognition. , 2, pp.121–167.

Burges, C.J.C., 1996. Simplified Support Vector Decision Rules. *Machine learning-international workshop then conference. Morgan Kaufmann publisher, INC.*, pp.71–77.

Burges, C.J.C., Schoelkopf, B. and Vapnik, V., 1997. improving the accuracy and speed of support vector machines. In *Advanced in neural network processing systems 9: Proceedings of The 1996 Conference (Vol. 9)*. MIT Press, pp. 375–381.

Campbell, C., 2000. An Introduction to Kernel Methods. *in: R.J. Howlett, L.C. Jain (Eds.), Radial Basis Function Networks: Design and Applications*, pp.155–192. Available at: Springer Verlag, Berlin.

Chawla, N. V., Bowyer, K., Hall, L., Philip K., 2002. SMOTE : Synthetic Minority Over-sampling TEchnique. *Artificial Intelligence Research*, 16, pp.341–378.

Chawla, N. V, Japkowicz, N. and Kolcs, A., 2004. Editorial : Special Issue on Learning from Imbalanced Data Sets. *ACM SIGKDD Explorations*, 6(1), pp.1–6.

Chawla, N. V., Cieslak, D., Hall, L., Joshi, A., 2008. Automatically countering imbalance and its empirical relationship to cost. *Data Mining and Knowledge Discovery*, 17, pp.225–252.

Che, Z.G., Chiang, T.A. and Che, Z., 2011. Feed-forward neural networks training: A comparison between genetic algorithm and back-propagation learning algorithm. *International journal of Innovative Computing, Information and Control*, 7(10), pp.5839–5850.

Chen, D., He, Q. and Wang, X., 2010. FRSVMs: Fuzzy rough set based support vector machines. *Fuzzy Sets and Systems*, 161(4), pp.596–607.

Chen, J. and Liao, C.-M., 2002. Dynamic process fault monitoring based on neural network and PCA. *Journal of Process Control*, 12(2), pp.277–289.

Chen, M., Han, J., Yu, P., 1996. Data Mining: An Overview from a Database Perspective. , 8(6), pp.866–883.

Chen, W.J., Shao, Y.H. and Hong, N., 2014. Laplacian smooth twin support vector machine for semi-supervised classification. *International Journal of Machine Learning and Cybernetics*, 5, pp.459–468.

Choudhary, a. K., Harding, J. a. and Tiwari, M.K., 2008. Data mining in manufacturing: a review based on the kind of knowledge. *Journal of Intelligent Manufacturing*, 20(5), pp.501–521.

Cortes C. and Vapnik V., 1995. support vector networks. *Machine learning,*, 20(3), pp.273–297.

DeCoste, D. and Mazzoni, D., 2003. . Fast Query-Optimized Kernel Machine Classification Via Incremental Approximate Nearest Support Vectors. In *Machine learning-international workshop then conference. - (Vol. 20, No. 1).* pp. 115–122.

Dehariya, V.K., Shrivastava, S.K. and Jain, R.C., 2010. Clustering of Image Data Set Using K-Means and Fuzzy K-Means Algorithms. *2010 International Conference on Computational Intelligence and Communication Networks*, pp.386–391.

Dong, J., Krzyzak, A. and Suen, C.Y., 2005. Fast SVM training algorithm with decomposition on very large data sets. *IEEE transactions on pattern analysis and machine intelligence*, 27(4), pp.603–618.

Downs, T., Gates, K.E. and Masters, A., 2002. Exact Simplification of Support Vector Solutions. *The Journal of Machine Learning Research*, 2, pp.293–297.

Duan, K. and Keerthi, S.S., 2005. Which Is the Best Multiclass SVM Method ? An Empirical Study. In *Multiple Classifier Systems*. pp. 278–285.

Duda, R.O., Hart, P.E. and Stork, D.G., 2001. *Pattern Classification*, John Wiley & Sons.

Dunn, J.C., 1973. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(4), pp.32–57.

Estabrooks, A., Jo, T. and Japkowicz, N., 2004. A Multiple Resampling Method for Learning from Imbalanced Data Sets. *Computational Intelligence*, 20(1), pp.18–36.

Evgeniou, T., Pérez-Breva, L., Pontil, M., Poggio, T., 2000. Bounds on the Generalization Performance of Kernel Machine Ensembles. In *the Seventeenth International Conference on Machine Learning (ICML 2000)*. pp. 271–278.

Fayyad, Shapiro, G. and Smyth, P., 1996. Knowledge Discovery and Data Mining : Towards a Unifying Framework. In *the second International Conference on Knowledge Discovery and Data Mining (KDD)*. pp. 82–88.

Fayyad., Shapiro, G. and Smyth, P., 1996. From Data Mining to Knowledge Discovery in. *American Association for Artificial Intelligence.*, 17(3), pp.37–54.

Feng, J.C.-X. and Kusiak, A., 2006. Data mining applications in engineering design, manufacturing and logistics. *International Journal of Production Research*, 44(14), pp.2689–2694.

Fernández, A., del Jesus, M.J. and Herrera, F., 2009. Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. *International Journal of Approximate Reasoning*, 50(3), pp.561–577.

Freitas, J.A., Costa-Pereira, A. and Brazdil, P., 2007. Cost-Sensitive Decision Trees Applied to Medical Data. In *Data Warehousing and Knowledge Discovery, Proceedings of the 9th International Conference on Data Warehousing and Knowledge Discovery (DaWak 2007)*. Germany: Berlin/Heidelberg, Springer, pp. 303–312.

Friedman, J., 1996. *Another approach to polychotomous classification*, Stanford, CA.

Fung, 2001. A Comprehensive Overview of Basic Clustering Algorithms.

Fung, G. and Mangasarian, O.L., 2005. Multicategory Proximal support vector machine classifiers. *Machine learning,*, 59(1), pp.77–97.

Galar, M., Fern, Al., Barrenechea, E., Bustince, H., 2012. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *IEEE transactions on systems, man and cybernetics*, 42(4), pp.463–484.

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

Garg, A., Rachmawati, L. and Tai, K., 2013. Classification-driven model selection approach of genetic programming in modelling of turning process. *The International Journal of Advanced Manufacturing Technology*, 69(5-8), pp.1137–1151.

Garg, A., Tai, K. and Savalani, M.M., 2014. State-of-the-art in empirical modelling of rapid prototyping processes. *Rapid Prototyping Journal*, 20(2), pp.164–178.

Geebelen, D., Suykens, J. a. K. and Vandewalle, J., 2012. Reducing the Number of Support Vectors of SVM Classifiers Using the Smoothed Separable Case Approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4), pp.682–688.

Ghaedi, M., Ghaedi, a. M., Hossainpour, M., Ansari, A., Habibi, M. H., Asghari, a. R., 2014. Least square-support vector (LS-SVM) method for modeling of methylene blue dye adsorption using copper oxide loaded on activated carbon: Kinetic and isotherm study. *Journal of Industrial and Engineering Chemistry*, 20(4), pp.1641–1649.

Ghosh, S. and Dubey, S., 2013. Comparative Analysis of K-Means and Fuzzy C-Means Algorithms. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 4(4), pp.35–39.

Giles, C.L., 2007. Learning on the Border : Active Learning in Imbalanced Data Classification. In *16th ACM conf. Information and Knowledge Management*. pp. 127–136.

Giudici, P., 2003. *Applied Data Mining: Statistical Methods for Business and Industry*, New York: Wiley & Sons.

Graupe, G., 2006. Principles of Artificial Neural Networks. *world Scientific Singapore*, 6.

Guermeur, Y., 2002. Combining discriminant models with new multi-class SVMs. *Pattern Analysis and Applications*, 5, pp.168–179.

Gunn, S.R. and Kandola, J.S., 2002. Structural modelling with Sparse kernels. *Machine learning,*, 48(1), pp.137–163.

Guyon, I., Boser, B. and Vapnik, V., 1993. Automatic Capacity Tuning of Very Large VC-Dimension Classifiers. In *Advances in Neural Information Processing Systems*. San Mateo, CA: Morgan Kaufmann, pp. 147–155.

Guyon, I. and Elisseeff, A., 2003. An Introduction to Variable and Feature Selection. *Journal of machine learning research*, 3, pp.1157–1182.

Hair, J., Anderson, R. and Tatham, R., 1987. *Multivariate Data Analysis, with Readings*, New York: Macmillan Publishing Company.

Han, J., Cai, Y. and Cercone, N., 1993. Data-driven discovery of quantitative rules in relational databases - Knowledge and Data Engineering, IEEE Transactions on. *IEEE transactions on knowledge and data engineering*, 5(1).

Han, J. and Kamber, M., 2006. *data mining concepts and techniques*, Morgan Kaufmann.

Han, J. and Kamber, M., 2001. *Data Mining: Concepts and Techniques*, USA: Morgan Kaufmann Publishers.

Harding, J. A., Shahbaz, M. and Kusiak, A., 2006. Data Mining in Manufacturing: A Review. *Journal of Manufacturing Science and Engineering*, 128(4), p.969.

Haykin, S., 1999. *Neural Networks: A comprehensive foundation*, New Jersey: Prentice-Hall, Inc.

Helmuth, S., 1980. *Cluster Analysis Algorithms. For Data Reduction and Classification of Objects*,

Hornik, K., 1991. Approximation Capabilities of Muitilayer Feedforward Networks. *Neural networks*, 4(1989), pp.251–257.

Hoskins, J.C., Kaliyur, K.M. & Himmelblau, D.M., 1991. Fault diagnosis in complex chemical plants using artificial neural networks. *AIChE Journal*, 37(1), pp.137–141.

Hsu, C.W. and Lin, C.J., 2002. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2), pp.415–425.

Hu, G., Zhang, Y. and Zhu, F., 2007. Short-Term Load Forecasting Based on Fuzzy C-Mean Clustering and Weighted Support Vector Machines. In *Third International Conference on Natural Computation (ICNC 2007) IEEE*. pp. 7–12.

Hwang, J.P., Park, S. and 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(7), pp.8580–8585.

Irani, K.B., Cheng, Ji., Company, F., Fayyad, U., Services, E., 1993. Applying Machine Learning to Semiconductor Manufacturing. *IEEE Expert*, 8(1), pp.41–47.

Jain, A.K., 2010. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8), pp.651–666.

Janakiraman, V.M., Nguyen, X., Sterniak, J., Assanis, D., 2014. A System Identification Framework for Modeling Complex Combustion Dynamics Using Support Vector Machines. In *nformatics in Control, Automation and Robotics (Lecture Notes in Electrical Engineering)*. Springer, Berlin, Heidelberg, pp. 297–313.

Jang, J.R., 1993. ANFIS : Adaptive-Ne twork-Based Fuzzy Inference System. *IEEE transactions on systems, man, and cybernetics.*, 23(3), pp.665–685.

Jang, J.R., Sun, C. and Mizutani, E., 1997. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence* I. Prentice-Hall, ed., USA, NJ.

Japkowicz, N., 2000. the class imbalance problem: significance and strategies. In *the 2000 International Conference on Artificial Intelligence (IC-AI'2000): Special Track on Induction ;Learning*. Las Vigas, Nivada.

Jayadeva, J., Khemchandani, R. and Chandra, S., 2007. Twin Support Vector Machines for Pattern Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5), pp.905–910.

Jiang, X., Yi, Z. and Lv, J.C., 2006. Fuzzy SVM with a new fuzzy membership function. *Neural Computing and Applications*, 15(4), pp.268–276.

Jing, L., 2005. A Robust proximal support vector machines for classification. In *Neural Networks and Brain, ICNN&B'05. International Conference*. pp. 576–580.

Kadlec, P., Gabrys, B. and Strandt, S., 2009. Data-driven Soft Sensors in the process industry. *Computers & Chemical Engineering*, 33(4), pp.795–814.

Kang, S. and Cho, S., 2014. Approximating support vector machine with artificial neural network for fast prediction. *Expert Systems with Applications*, 41(10), pp.4989–4995.

Kaufman, L. and Rousseeuw, P.J., 1990. *Finding Groups In Data,An Introduction to Cluster Analysis*,

Khashei, M. and Bijari, M., 2010. An artificial neural network (p,d,q) model for timeseries forecasting. *Expert Systems with Applications*, 37(1), pp.479–489.

Klosgen, W. and Zytkow, J., 2002. *Handbook of data mining and knowledge discovery*, New York: Oxford University Press.

Knerr, S., Personnaz, L. and Dreyfus, G., 1990. single-layer learning revisited a stepwise procedure for building and training a neural network. In *In Neurocomputing: Algorithms, Architectures and Applications*. ED.Springer, J Fogelman.

Köksal, G., Batmaz, İ. and Testik, M.C., 2011. A review of data mining applications for quality improvement in manufacturing industry. *Expert Systems with Applications*, 38(10), pp.13448–13467.

Kuhn, H.W. and Tucker, A.W., 1951. Nonlinear Programming,. In U. of C. Press, ed. *Proceedings of Second Berkeley Symposium on Mathematical Statistics and Probability*. pp. 481–492.

Kumar, U. and Pal, P.S., 2011. Data Mining : A prediction of performer or underperformer using classification. *International Journal of Computer Science and Information Technologies, (IJCSIT)*, 2(2), pp.686–690.

Lee, D.M. and Choi, S.., 2004. Application of on-line adaptable Neural Network for the rolling force set-up of a plate mill. *Engineering Applications of Artificial Intelligence*, 17(5), pp.557–565.

Lee, L.H., Choi, S.G., 2011. An enhanced Support Vector Machine classification framework by using Euclidean distance function for text document categorization. *Applied Intelligence*, 37(1), pp.80–99.

Li, C., 2007. Classifying imbalanced data using a bagging ensemble variation (BEV). In *Proceedings of the 45th annual southeast regional conference on - ACM-SE 45*. New York, New York, USA: ACM Press, p. 203.

Li, Q. and Ren, S., 2012. A Real-Time Visual Inspection System for Discrete Surface Defects of Rail Heads. *IEEE Transactions on Instrumentation and Measurement*, 61(8), pp.2189–2199.

Liao, S.-H. and Wen, C.-H., 2007. Artificial neural networks classification and clustering of methodologies and applications – literature analysis from 1995 to 2005. *Expert Systems with Applications*, 32(1), pp.1–11.

Liao, T.W., Wang, G., Triantaphyllou, E., Rouge, B., Chang, P., 2001. A Data Mining Study of Weld Quality Models Constructed with MLP Neural Networks from Stratified Sampled Data. In *Industial Engineering Research Conference*. p. 6.

Liao, T.W., Li, D. and Li, Y., 1999. Detection of welding # aws from radiographic images with fuzzy clustering methods. *Fuzzy Sets and Systems*, 108(2), pp.145–158.

Lin, C. and Wang, S., 2004. Training algorithms for fuzzy support vector machines with noisy data. *Pattern Recognition Letters*, 25(14), pp.1647–1656.

Lin, C.-F. and Wang, S.-D., 2002. Fuzzy Support Vector Machines. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 13(2), pp.464–71.

Liu, A., Ghosh, J. and Martin, C.E., 2007. Generative Oversampling for Mining Imbalanced Datasets. In *International Conference on Data Mining*. Las Vegas, Nevada: CSREA Press, pp. 66–72.

Liu, S., Yamada, M., Collier, Ni., Sugiyama, M., 2013. Change-point detection in time-series data by relative density-ratio estimation. *Neural networks : the official journal of the International Neural Network Society*, 43, pp.72–83.

Liu, X., Wu, J. and Zhou, Z., 2009. Exploratory Undersampling for Class-Imbalance Larning. *IEEE transactions on systems, man, and cybernetics.*, 39(2), pp.539–550.

Liu, Y., An, A. and Huang, X., 2006. Boosting Prediction Accuracy on Imbalanced Datasets with SVM Ensembles. In *10th Pacific-Asia Conference on Knowledge Discovery and Data Mining*. pp. 107–118.

Luo, Y., Xiong, S. and Wang, S., 2008. A PCA Based Unsupervised Feature Selection Algorithm. In *Second International Conference on Genetic and Evolutionary Computing, WGEC'08. IEEE*. Ieee, pp. 299–302.

Maldonado, S. and López, J., 2014. Imbalanced data classification using second-order cone programming support vector machines. *Pattern Recognition*, 47(5), pp.2070–2079.

Mangasarian, O.L. and Musicant, D.R., 2001. Lagrangian Support Vector Machines. , 1(3), pp.161–177.

Manikandan, J. and Venkataramani, B., 2009. Design of a modified one-against-all SVM classifier. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, (October), pp.1869–1874.

McDonald, C.J., 1999. New tools for yield improvement in integrated circuit manufacturing : can they be applied to reliability ? *Microelectron. Reliability*, 39((6-7)), pp.731–739.

Mendel, J.M., 2003. Type-2 fuzzy sets: some questions and answers. *IEEE Connections, Newsletter of the IEEE Neural Networks Society*, 1(August), pp.10–13.

Mendel, J.M., 2001. *Uncertain Rule-based Fuzzy Logic Systems: Introduction and New Directions*, Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Micchelli, C.A. and Pontil, M., 2005. Learning the Kernel Function via Regularization. *Journal of machine learning research*, 6, pp.1099–1125.

Michell, T.M., 1997. *Machine Learning*, Boston: McGraw-Hill Series in Computer Science.

Miller, A., 2002. *Subset selection in regression*, Chapman & Hall/CRC.

Mitchell, T.M., 1999. Machine Learning and Data Mining Over the past. *Communicatio of The ACM*, 42(11), pp.30–36.

Mucherino, A., Papajorgji, P. & Pardalos, P.M., 2009. A survey of data mining techniques applied to agriculture. *Operational Research*, 9(2), pp.121–140.

Muscat, R., Mahfouf, M., Zughrat, A., Yang, Y.Y., Thornton, S., Khondabi, A.V., Sotanos, S., 2014. Hierarchical fuzzy Support Vector Machine (SVM) for Rail Data Classification. In *The 19th World Congress of the International Federation of Automatic Control, IFAC*. Cape Town, pp. 10652–10657.

Nguyen, D. and Ho, T., 2005. An efficient method for simplifying support vector machines. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*. New York, USA: ACM Press, pp. 617–624.

Oh, S., Han, J. and Cho, H., 2001. Intelligent Process Control System for Quality Improvement by Data Mining in the Process Industry. In *Data Mining for Design and Manufacturing: Methods and*. pp. 289–310.

Oliver, P., 2004. *A Frame Work of Unsupervised Learning of Dialogue Strategies*,

Ordieres Meré, J.B., González Marcos, A., González, J.a., Lobato Rubio, V., 2004. Estimation of mechanical properties of steel strip in hot dip galvanising lines. *Iron making & Steelmaking*, 31(1), pp.43–50.

Örkcü, H.H. and Bal, H., 2011. Comparing performances of backpropagation and genetic algorithms in the data classification. *Expert Systems with Applications*, 38(4), pp.3703–3709.

Öznergiz, E., Özsoy, C., Delice, I., Kural, A., 2009. Comparison of empirical and neural network hot-rolling process models. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 223(3), pp.305–312.

Perkins, S., Lacker, K. and Theiler, J., 2003. Grafting : Fast , Incremental Feature Selection by Gradient Descent in Function Space. *The Journal of machine learning research*, 3, pp.1333–1356.

Perzyk, M., Biernacki, R. and Kochański, A., 2005. Modeling of manufacturing processes by learning systems: The naïve Bayesian classifier versus artificial neural networks. *Journal of Materials Processing Technology*, 164-165, pp.1430–1435.

Ph Papaelias, M., Roberts, C. and Davis, C.L., 2008. A review on non-destructive evaluation of rails: state-of-the-art and future development. *Proceedings of the Institution of Mechanical Engineers, Journal of Rail and Rapid Transit*, 222(4), pp.367–384.

Pham, D.T. and Afify, a a, 2005. Machine-learning techniques and their applications in manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 219(5), pp.395–412.

Platt, J., Cristianini, N. and Shawe-Taylor, J., 2000. Large Margin DAGs for Multiclass Classification. *in Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press*, 12, pp.547–553.

Platt, J.C., 1998. *Sequential minimal optimization : A fast algorithm for training support vector machines ,Technical Report 98-14, Microsoft Research.*,

Prajapati, G.L. and Patle, A., 2010. On Performing Classification Using SVM with Radial Basis and Polynomial Kernel Functions. In *3rd International Conference on Emerging Trends in Engineering and Technology,ICETET*. Ieee, pp. 512–515.

Roger, S. and Girolami, M., 2011. *A first course in machine learning*, Chapman and Hall / CRC.

Rojas, A. and Nandi, A.K., 2006. Practical scheme for fast detection and classification of rolling-element bearing faults using support vector machines. *Mechanical Systems and Signal Processing*, 20(7), pp.1523–1536.

Rokach, L. and Maimon, O., 2006. Data Mining for Improving the Quality of Manufacturing: A Feature Set Decomposition Approach. *Journal of Intelligent Manufacturing*, 17(3), pp.285–299.

Ross, T., 2010. *Fuzzy Logic with Engineering Applications* Third Edit., the University of Maxico, USA: Jhon Wiley & Sons, Ltd.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986. learning representations by back-propagating errors.pdf. *nature*, 323(6088), pp.533–536.

Sadoyan, H., Zakarian, A. and Mohanty, P., 2005. Data mining algorithm for manufacturing process control. *The International Journal of Advanced Manufacturing Technology*, 28(3-4), pp.342–350.

Sahoo, P., Behera, A., Pandia, M., Dash, C., Dehuri, S., 2013. On the study of GRBF and polynomial kernel based support vector machine in web logs. *Emerging Trends and Applications in Computer Science (ICETACS),1st International Conference on.IEEE*, pp.1–5.

Schoelkopf, B., Mika, S., Burges, C., Knirsch, P., Smola, A., 1999. Input Space Versus Feature Space in Kernel-Based Methods. *IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 10, NO. 5,*, 10(5), pp.1000–1017.

Scholkopf, B. and Smols, A., 2001. *learning with kernels: upport vector machines, regularization, optimization, and beyond.* MIT Press.,

Schwenker, F., Frey, S., Glodek, M., Kachele, M., Meudt, S., Schels, M., Schmidt, M., 2014. A new multi-class fuzzy support vector machine algorithm. *in Arti cial Neural Networks in Pattern Recognition*, 8774, pp.153–164.

Sebzalli, Y.M. and Wang, X.Z., 2001. Knowledge discovery from process operational data using PCA and fuzzy clustering. *Engineering Applications of Artificial Intelligence*, 14, pp.607–616.

Shao, Y.H., Chen, W., Zhang, J., Wang, Z., Deng, N., 2014. An efficient weighted Lagrangian twin support vector machine for imbalanced data classification. *Pattern Recognition*, 47, pp.3158–3167.

Shilton, A. and Lai, D.T.H., 2007. Iterative fuzzy support vector machine classification. In *IEEE International Fuzzy Systems Conference*. Ieee, pp. 1–6.

Solomatine, D.P. and Ostfeld, A., 2008. Data-driven modelling: some past experiences and new approaches. *Journal of Hydroinformatics*, 10(1), pp.3–22.

Solomatune, D., See, L.M. and Abrahart, R.J., 2008. Data-driven modelling: concepts, approaches and experiences. *Practical Hydroinformatics: Computational Intelligence and Technological Developments in Water Application* ., (Berlin: Springer).

Suen, C.Y., 2000. Clustering combination method. *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, 2, pp.732–735.

Syed, N.A., Liu, H. and Sung, K.K., 1999. Handling concept drifts in incremental learning with support vector machines. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM*. New York, USA: ACM Press, pp. 317–321.

Tang, H.A.O. and Qu, L., 2008. Fuzzy support vector machine with a new fuzzy membership function for pattern classification. In *Machine Learning and Cybernetics, International Conference on, IEEE*. pp. 768–773.

Tang, Y., Zhang, Y., Chawla, Ni., Krasser, S., 2009. SVMs modeling for highly imbalanced classification. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 39(1), pp.281–88.

Tao, Q. and Wang, J., 2004. A New fuzzy support vector machine based on the weighted margin. *Neural Processing Letters*, 20(3), pp.139–150.

Tian, Y., Qi, Z., Ju, X., Shi, Y., Liu, X., 2013. Nonparallel Support Vector Machines for Pattern Classification. *IEEE Transactions on cybernetics*, 44(7), pp.1067–1079.

Tobon-Mejia, D.A., Medjaher, K., Zerhouni, N., Tripot, G., 2012. A Data-Driven Failure Prognostics Method Based on Mixture of Gaussians Hidden Markov Models. *IEEE Transactions on Reliability*, 61(2), pp.491–503.

Trotter, M., Buxton, B. and Holden, S., 2001. Support vector machines in combinatorial chemistry. *Measurement+ Control*, 34(8), pp.235–9.

Vapnik, V.N., 1995. *The Nature of Statistical Learning Theory*, Berlin, Germany: Springer.

Venkata Rao, 2011. *Advanced modeling and optimization of manufacturing processes: International Research and Development,* London: Springer Verlag.

Veropoulos, K., Campbell, C. & Cristianini, N., 1999. Controlling the Sensitivity of Support Vector Machines. In *in Proceedings of the International Joint Conference on n Articfial Intelligence (IJCAI)*. Stockholm, Sweden.

Wagstaff, K., Rogers, S. and Schroedl, S., 2001. Constrained K-means Clustering with Background Knowledge. *Artificial Intelligence*, pp.577–584.

Wan, S.J., Wong, S.K.M. and Prusinkiewicz, P., 1988. An algorithm for multidimensional data clustering. *ACM Transactions on Mathematical Software*, 14(2), pp.153–162.

Wang, C.-H., Kuo, W. and Bensmail, H., 2006. Detection and classification of defect patterns on semiconductor wafers. *IIE Transactions*, 38(12), pp.1059–1068.

Wang, J.Z., Chen, Y. and Wang, Z., 2003. Support vector learning for fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 11(6), pp.716–728.

Wang, S. and Yao, X., 2009. Diversity analysis on imbalanced data sets by using ensemble models. In *2009 IEEE Symposium on Computational Intelligence and Data Mining*. Ieee, pp. 324–331.

Wang, T.-Y. and Chiang, H.-M., 2007. Fuzzy support vector machine for multi-class text categorization. *Information Processing and Management*, 43, pp.914–929.

Widodo, A. and Yang, B.-S., 2007. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21, pp.2560–2574.

Wu, X., Kumar, V., Ross Q., Ghosh, J., Yang, Q., Motoda, H., 2008. Top 10 algorithms in data mining. In *Knowledge and Information Systems*. pp. 1–37.

Wu, Z. and Xie, W., 2003. Fuzzy C-means Clustering Algorithm based on Kernel Method. In *proceeding of the Fifth International Conference on Computational Intelligence and Multimedia applications (ICCIMA'03)*. pp. 1–6.

Wu, Z., Zhang, H. and Liu, J., 2014a. A fuzzy support vector machine algorithm for classification based on a novel PIM fuzzy clustering method. *Neurocomputing*, 125, pp.119–124.

Wu, Z., Zhang, H. and Liu, J., 2014b. A fuzzy support vector machine algorithm for classification based on a novel PIM fuzzy clustering method. *Neurocomputing*, 125, pp.119–124.

Xia, X., Lyu, Mi., Lok, T., Huang, G., 2005. Methods of decreasing the number of support vectors via k -Mean clustering. In *ICIC*. Springer, pp. 717–726.

Xiong, S., Liu, H. and Niu, X., 2005. Fuzzy support vector machines based on FCM Clustering. In *In Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on IEEE*. pp. 18–21.

Xu, R. and Wunsch, D., 2005. Survey of Clustering Algorithms. *IEEE transactons on neural networks*, 16(3), pp.645–678.

Yang., Zhang, G., Lu, J., Ma, J., 2011b. A Kernel Fuzzy c -Means Clustering-Based Fuzzy Support Vector Machine Algorithm for Classification Problems With Outliers or Noises. *IEEE Transactions on Fuzzy Systems*, 19(1), pp.105–115.

Yang, Y., Mahfouf, M., Panoutsos, G., Zhang, Q., Thornton, S., 2011a. Adaptive neural-fuzzy inference system for classification of rail quality data with bootstrapping-based over-sampling. *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pp.2205–2212.

Yin, S., Ding, S., Xie, X., Luo, H., 2014. A Review on Basic Data-Driven Approaches for Industrial Process Monitoring. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 61(11), pp.6418–6428.

Yoo, R., Lee, H., Chow, K., Lee, H., 2006. Constructing a Non-Linear Model with Neural Networks for Workload Characterization. *2006 IEEE International Symposium on Workload Characterization*, pp.150–159.

Yu, J., 2013. A Support Vector Clustering-Based Probabilistic Method for Unsupervised Fault Detection and Classification of Complex Chemical Processes Using Unlabeled Data. *American Institute of Chemical Engineers (AICHE)*, 59(2), pp.407–4419.

Yu, L. and Liu, H., 2003. Feature Selection for High-Dimensional Data : A Fast Correlation-Based Filter Solution. *Proceedings 20th International Conference on Machine learning. (CML-2003), Washington DC.*

Zadeh, L.A., 1965. Fuzzy Sets. , 353, pp.338–353.

Zahid, N., Abouelala, O., Limouri, M., Essaid, A., 2001. Fuzzy clustering based on - nearest-neighbours rule. *Fuzzy Sets and Systems*, 120(2), pp.239–247.

Zain, A.M., Haron, H. and Sharif, S., 2012. Applications of computational intelligence for design and operations decisions in manufacturing. *International Journal of Production Research*, 50(1), pp.191–213.

Zhang, D.-Q. and Chen, S.-C., 2003. Clustering Incomplete Data Using Kernel-Based Fuzzy C-means Algorithm. *Neural Processing Letters*, 18(3), pp.155–162.

Zhang, X., 1999. Using class-center vectors to build support vector machines. In *IEEE Proceedings of Neural the Networks and Signal Processing*. pp. 3–11.

Zheng, J., Shen, F., Fan, H., Zhao, J., 2013. An online incremental learning support vector machine for large-scale data. *Neural Computing and Applications*, 22(5), pp.1023–1035.

Zhou, Z. and Liu, X., 2006. Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. *IEEE transactions on knowledge and data ngineering*, 18(1), pp.63–77.