Contributions to Ensembles of Models for Predictive Toxicology Applications

On the Representation, Comparison and Combination of Models in Ensembles

Mokhairi MAKHTAR

Submitted for the Degree
of Doctor of Philosophy

School of Computing, Informatics and Media
University of Bradford

2012

Abstract

KEYWORDS:
Predictive Toxicology, Model Representation, Model Comparison, Ensembles of Models, Classifiers.

The increasing variety of data mining tools offers a large palette of types and representation formats for predictive models. Managing the models then becomes a big challenge, as well as reusing the models and keeping the consistency of model and data repositories. Sustainable access and quality assessment of these models become limited to researchers. The approach for the Data and Model Governance (DMG) makes easier to process and support complex solutions. In this thesis, contributions are proposed towards ensembles of models with a focus on model representation, comparison and usage.

Predictive Toxicology was chosen as an application field to demonstrate the proposed approach to represent predictive models linked to data for DMG. Further analysing methods such as predictive models comparison and predictive models combination for reusing the

models from a collection of models were studied. Thus in this thesis, an original structure of the pool of models was proposed to represent predictive toxicology models called Predictive Toxicology Markup Language (PTML). PTML offers a representation scheme for predictive toxicology data and models generated by data mining tools.

In this research, the proposed representation offers possibilities to compare models and select the relevant models based on different performance measures using proposed similarity measuring techniques. The relevant models were selected using a proposed cost function which is a composite of performance measures such as Accuracy (Acc), False Negative Rate (FNR) and False Positive Rate (FPR). The cost function will ensure that only quality models be selected as the candidate models for an ensemble.

The proposed algorithm for optimisation and combination of Acc, FNR and FPR of ensemble models using double fault measure as the diversity measure improves Acc between 0.01 to 0.30 for all toxicology data sets compared to other ensemble methods such as Bagging, Stacking, Bayes and Boosting. The highest improvements for Acc were for data sets Bee (0.30), Oral Quail (0.13) and Daphnia (0.10). A small improvement (of about 0.01) in Acc was achieved for Dietary Quail and Trout. Important results by combining all the three performance measures are also related to reducing the distance between FNR and FPR for Bee, Daphnia, Oral Quail and Trout data sets for about 0.17 to 0.28. For Dietary Quail data set the improvement was about 0.01 though, but this data set is well known as a difficult learning exercise. For five UCI data sets tested,

similar results were achieved with Acc improvement between 0.10 to 0.11, closing more the gaps between FNR and FPR.

As a conclusion, the results show that by combining performance measures (Acc, FNR and FPR), as proposed within this thesis, the Acc increased and the distance between FNR and FPR decreased.

Acknowledgements

I would like to thank Professor Daniel Neagu and Mr. Mick Ridley, my supervisors, for their great supports and guidances during this research. Their inspiring suggestions and constant encouragements had a great effect on my study. It has been a valuable opportunity to work under their dedicated supervision.

I am grateful to my parents for their encouragement and support. Most of all, I am grateful to my wife, Syadiah Nor for her patience and sacrifice. I also would like to thank my children, Syahmi, Syafiq, Syazwan and Syasya.

Finally, I had the pleasure of meeting all members of Artificial Intelligent research group in the School of Computing, Informatics and Media at University of Bradford: many thanks to Dr. Longzhi Yang and all my research colleagues for their valuable discussions.

This work is partially supported by BBSRC, TSB and Syngenta through the Knowledge Transfer Partnerships (KTP) Grant "Data and Model Governance with Applications in Predictive Toxicology". The author acknowledges the financial supports received from the the Ministry of Higher Education, Malaysia and University Sultan Zainal Abidin, Malaysia (UNisZa).

Declaration

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

Parts of the original work proposed in this thesis have appeared in the publications as listed in publications section.

The work was done under the guidance of Professor Daniel Neagu and Mr. Mick Ridley, at the School of Computing, Informatics and Media, University of Bradford, UK.

Publications

- Makhtar M, Yang L, Neagu D. and Ridley M.J. (2012): "Optimisation of Classifier Ensemble for Predictive Toxicology Application", in Proceedings of the 14th International Conference on Modelling and Simulation (UKSim2012), IEEE, pp 236-241, Cambridge, UK.

- Makhtar M., Neagu D. and Ridley M.J. (2011): "Comparing Multi Class Classifiers: On the Similarity of Confusion Matrices for Predictive Toxicology Applications", in Proceedings of the 12th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2011), 7-9 September 2011, pp. 252-261, Springer, LNCS 6936, Norwich, UK.

- Makhtar M., Neagu D. and Ridley M.J. (2011): "Binary Classification Models Comparison: on the Similarity of Datasets and Confusion Matrix for Predictive Toxicology Applications", in Proceedings of the 2nd International Conference on Information Technology in Bio and Medical Informatics (ITBAM 2011), August 29 - September 2 2011. pp. 108-122, Springer, LNCS

6865, Toulouse, France.

- Makhtar M, Neagu D. and Ridley M.J. (2010): "Predictive Model Representation and Comparison: Towards Data and Predictive Models Governance", in Proceedings of the 10th UK Workshop on Computational Intelligence (UKCI 2010), IEEE Xplore, pp 1-6, Colchester, UK.

Table of Contents

List of figures

List of tables

List of Abbreviations

Acc	Accuracy
FNR	False Negative Rate
FPR	False Positive Rate
TPR	True Positive Rate
TNR	True Negative Rate
CFS	Correlation-based Feature Selection
CRS	Classifier Ranking System
CRV	Classifier Ranking Value
DMG	Data and Model Governance
DTD	Document Type Definition
ER	Error Rate
GA	Genetic Algorithm
KDD	Knowledge Discovery in Databases
KNIME	Konstanz Information Miner
OCEM	Optimisation of Classifier Ensemble Method
PTML	Predictive Toxicology Markup Language
PMML	Predictive Model Markup Language
Sim	Similarity of Predictive Model
Weka	Waikato Environment for Knowledge Analysis

Chapter 1

Introduction

This chapter gives a brief background to the current state of data and predictive model governance. Moreover the research motivation, research aims and objectives are laid out in order to give the reader a glimpse of what inspired this research. The original contributions and the thesis structure are also covered in this chapter.

The thesis will discuss solutions for getting a better prediction in predictive toxicology problems by reusing classifiers from an existing collection. The collection of classifiers was represented using a proposed Predictive Toxicology Markup Language (PTML). The collection of classifiers will be compared using the proposed Similarity of Predictive Model (Sim) measure related to data sets, function properties and confusion matrix. Results from the comparison can be grouped together based on their similarity, for example models built using the same data set and producing the same confusion matrix although having different function properties (classifiers) are similar models. The similar models with same confusion matrix will be discarded before selecting the chosen models into the proposed en-

semble. The relevant models are the remaining models which are less similar to each other and thus introduce diversity to be used in the ensemble construction.

In this thesis, the predictive model performance measures were focused on Accuracy (Acc), False Negative Rate (FNR) and False Positive Rate (FPR). Acc is the proportion of correct predictions for all classes, $FNRate$ is the proportion of incorrect predictions for the positive class and $FPRate$ is the proportion of incorrect predictions for the negative class (e.g. No). All three performance measures will be combined as a ranking value that helps in selecting classifiers from a collection of models using a cost function (which is a composite of three performance measures: Acc, FNR and FPR) to build a high quality and robust ensemble. The Optimisation of Classifiers Ensemble Method ($OCEM$) technique which applies to ensemble selection was implemented to optimise selection of models and combination method. The method proposed was to optimise the ensemble by ranking the models using the proposed ranking system known as Classifier Ranking Value (CRV). The ensemble models consist of diverse classifiers that had been measured using diversity measures such as disagreement measure and double fault measure. Simple majority voting was applied to the combination of the models in the ensemble as a decision fusion strategy to build upon the proposed combined performance measure.

In this way the work done so far has contributed and establish new pathways in applying for the first time DMG for ensembles of classifier applied to predictive toxicology.

1.1 Background

The steps to implement Knowledge Discovery in Databases (KDD) include Selection, Preprocessing, Transformation, Data Mining, and Interpretation or Evaluation (Fayyad et al. 1996). The processes can be looped and iterated between them. Figure 1.1 is an overview of the steps that compose the KDD processes. The process starts with data cleaning (selection, pre-processing and transformation) before carrying on to data mining. The data mining process is a process where data will be analysed using a machine learning algorithm to produce knowledge.

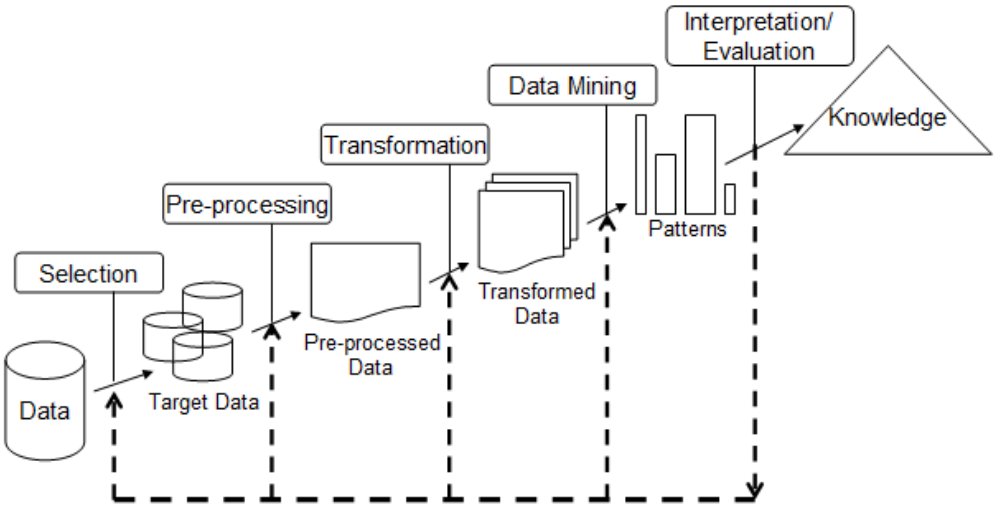

Figure 1.1: An Overview of the Steps of the KDD Process (Fayyad et al. 1996).

The tuning process in finding optimum model parameters is important. Each model from the collection of models must be trained to find the most relevant attributes and model parameters in producing a quality model. The tuning process involves selecting optimum model parameters such as number of folds for cross validation and

type of classifier. Selection of the optimum attributes from the data set is also another step of the tuning process. This will be repeated until the right combination of parameters is selected to generate the best model.

In an environment where we have data set updates, predictive models based on the older data set may become unreliable in terms of new instances added to updated versions of the data set. This is because with the new instances, a classifier may not learn accurately based on current features selected for the new data set. Feature selection process has to be applied again and retrain the model using the whole data set, thus the model will be up-to-date for future use. This evolution of training data sets always happens in application domains such as banking where transactions are updated regularly, and also in toxicology where experiment circumstances change and new compounds are added. The iteration process of tuning and finding the right combination of attributes and model parameters must take these changes into account when generating new and reliable predictive models for the updated data set. This makes it necessary to revise the predictive model generation step for an up-to-date model repository.

Models from a collection of models can be reused to speed up predictive modeling. All the models in the collection were represented using proposed representation. A method of selecting and comparing relevant models can be used to select the models from the collection. This thesis proposes the method to select and compare the models. The performance measures of the selected models

can be improved by making a combination of them. The combination of models which is known as ensemble method was considered and shown to improve the Acc as well as FNR and FPR (see the results in Chapter 6). This thesis also proposes an ensemble method by composing quality candidates in ensemble using a cost function. The cost function (CRV) is a value to rank the best classifier from a collection of models by giving a weight to each performance measures.

1.2 Motivation

The continuous process of KDD shows that there may be thousands of data mining models related to a single data set shared among data mining researchers, generating versions of predictive models on the related data sets. Thus, monitoring and maintaining changes between data and models become more challenging. The issues arisen when dealing with large collection of models are to find useful models, delete the useless models, identify the weaknesses of models, and suggest repairing actions (Liu & Tuzhilin 2008). Sometimes, the models become useless either being identical with existing ones or when FPR equals to 1.00 and FNR equals to 0.00 (or vice versa), or while Acc is poor. There is a need to define the relationship between data and models, so that the iteration process of generating new predictive models integrates consistently in the modelling framework and this evolution also needs to be recorded. The repositories of data mining models should keep information on historical developments

which are also worthy of analysis.

Another challenge here is how to share those models between researchers. Existing models are represented on various platforms in formats such as text files, relational database or different internal formats produced from data mining tools (e.g. .arff produced by Weka and .fis produced by Matlab). XML is a key to the answer where the models can be published through the web in a standard form and can be accessed easily later. For that reason Chapter 4 proposes representations in XML as a flexible bridge and a solution to deal with the current diversity of model representations.

The other challenge that arises here is whether available models can be analysed and interpreted so that information they store can be used later to generate a better performance measure of the predictive models. Since the information stored in the previous models are available in repositories, there should be a possibility of selecting the right or most suitable models from the collection of models based on individual requirements and needs. Thus Chapter 5 proposes the method for comparing the models.

This can be done in many ways such as searching the models with different criteria, comparing the performance of existing models or making a combination between models. These approaches have often been proved to achieve better predictive performance compared to producing a single predictive data mining model. Ensemble methods also offer better solutions compared to single models (Caruana et al. 2004, Kuncheva 2004, Dietterich 2000). The issues related to constructing ensembles suggested by Wang (2008) will be considered

in this thesis.

The thesis proposes a method of comparing classifiers from a collection of models. The collection of models were represented using the proposed representation $PTML$). The method proposed is to optimise selection of models to be included in the ensemble method by reusing and selecting diverse classifiers to make a combination between them. The selection of candidate models in the ensemble is done by using a cost function proposed (CRV).

1.3 Problem Statement

The growing diversity of data mining tools offers a large palette of types and representation formats for predictive models. The growing diversity of data mining tools offers a large palette of types and representation formats for predictive models. Various predictive models that have been generated on the same or similar data sets are valuable assets that should be managed properly, to allow reusing these models for further work. Such models could be recorded and retrieved for future classification tasks on the same domains. A numbers of processes can be done to the collection of models such as searching of models, comparison between models and finding the most suitable models in model repositories have become big challenges. Prediction of toxicology data is a critical issue where the toxic instances should be precisely classified.

Furthermore, an ensemble method that focuses on a single performance measure such as Acc may return biased classifiers on cer-

tain classes. There is a need to produce a combination of other performance measures such as FNR and FPR in order to have a more generalised and better performance measure. Thus new methods for selecting the most relevant predictive models and making combination between them to increase the prediction of toxic classes will be considered in this research.

1.4 Research Framework and Scope

The research focuses on selecting the relevant classifier from a collection of models to predict new chemical substance in classifying the toxic or non toxic class of a chemical compound. The comparison of models were done by calculating the similarity of predictive model using the proposed technique. The relevant model obtained from the comparison stage can be used as a single model or in combination with other models in an ensemble for prediction of new chemical substance.

By reusing the models from the collection, an ensemble method can be applied in order to get better performance measures compared to single models. Experiments were conducted using binary classification models on predictive toxicology data sets. In addition there are also experiments done on the data sets from UCI (UC Irvine Machine Learning Repository, 2012).

1.5 Research Aim and Objectives

The aim of this research is to come up with a new method for comparing and searching relevant classifiers from a collection of models to be used as a model for predicting toxic classes of new chemical substance. The relevant models will be combined together in order to get the highest Accuracy (Acc) and lowest False Negative Rate (FNR) and False Positive Rate (FPR) by giving a weight to each of the performance measures. This is a specific problem in Predictive Toxicology, where predicting with good overall accuracy is not enough, especially for a high FNR, e.g. a chemical is classified wrongly to be not toxic, when actually it is toxic. The aim can be achieved by following these objectives.

The objectives of this research are:

1. To construct a framework for data and model governance in predictive toxicology.
2. To develop a knowledge representation for data and predictive toxicology models.
3. To construct a new technique for comparing the similarity of models from a collection of models based on Input (Training Set), Function (Classifier Properties) and Output (Confusion Matrix).
4. To construct a new technique for comparing the elements of a predictive model which are similarity of Input (Training Set), Function (Classifier Properties) and Output (Confusion Matrix).

5. To construct a new technique for ranking the classifiers with a composite of performance measures such as Acc, FNR and FPR.

6. To develop a new algorithm for optimising the selection and combination of classifiers.

From the objectives, a structured research methodology was designed and will be discussed in the next section.

1.6 Research Methodology

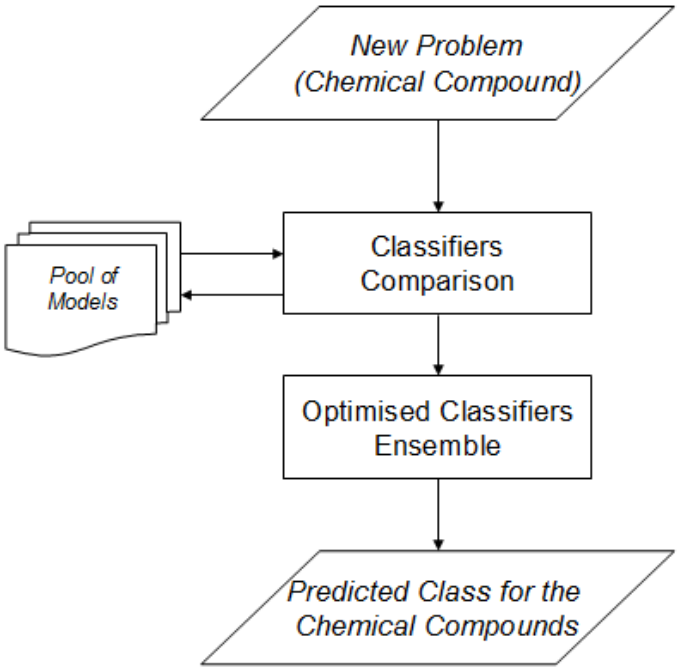

Figure 1.2: The General Method for the Research Study

The research focuses on toxicology data sets where the problem is to predict whether a chemical compound (test set) is toxic or non

toxic to animals. The prediction can be done by testing the test set against the training set. In this thesis, the model for prediction will be selected from a collection of models.

Figure 1.2 depicts the process of predicting a new problem using a pool of models. The pool of models is represented using a standard format as proposed in Chapter 4. To access those models, a method for comparing the relevant classifiers will be applied as proposed in Chapter 5. The results from the comparison are the relevant models that can be used to predict the toxicity.

To improve the prediction performance, the selected relevant models can be optimised by making a combination between them known as ensemble method. The ensemble method proposes that the candidate models be selected using a cost function which is a composite of three performance measures such as Acc, FNR and FPR. The optimisation technique used was Genetic Algorithm (GA). Diversity measures and a simple majority voting technique were applied in the ensemble.

1.7 Research Contributions

The contributions of this research are:

- A new framework for data and model governance (Chapter 3).
- A new knowledge representation for predictive toxicology data and models (Predictive Toxicology Markup Language - PTML) (Chapter 4).

- A novel technique to compare the similarity of models (Chapter 5).
  - A technique to compare data sets (training set) (Data set Similarity Coefficient - DSC)
  - A technique to compare the similarity of functions' property used to generate the predictive models.
  - A technique to compare the similarity of confusion matrices.
  - A technique to compare the similarity of multi class confusion matrices.
- A technique using a cost function (composite of Acc, FNR and FPR) to rank classifiers from a collection of models (Chapter 6).
- A new algorithm to optimise the selection and combination of classifiers (Chapter 6).
- An Improved results of Accuracy, with minimise False Negative Rate and False Positive Rate for all data sets compared to other ensemble method such as Bagging, Boosting and Stacking (Chapter 6).

All contributions have been published in peer-reviewed conference papers as detailed above.

1.8 Outline of the Thesis

This thesis is divided into seven chapters. The first chapter gives an overall picture of the thesis, aim and objectives of the research.

Chapter 2 is about related work and literature review in the area of study. Definition and some examples of the techniques used in the study are also discussed in this chapter.

Chapter 3 discusses the proposed methodology of the research and the concept of data and model governance. The chapter discusses briefly the process of model management.

Chapter 4 discusses the proposed standard representation for model management called Predictive Toxicology Markup Language (PTML). The PTML was used in the following chapter for model comparison and models combination.

In Chapter 5, a technique to compare predictive models is proposed. The chapter starts with comparison of three elements of a predictive model (Input, Function and Output) by calculating their similarity. The similarity of models can be calculated by combining all the three similarity measures. The results from the comparison which is the relevant model, can be ranked using a proposed cost function (CRV) to find the best model. The model can be used as a single model or combining them in the proposed ensemble.

Chapter 6 proposes an ensemble method by selecting models with a composite of three performance measures (Acc, FNR and FPR). The experiment shows that the prediction results from the ensemble method was better compared to a single model. The optimisation of the combination method is also proposed in this chapter.

Chapter 7 discusses and evaluates all the findings and outcomes achieved in this research.

Lastly, in Chapter 8, the conclusions of the research will be dis-

cussed. In order to improve the research in the future, some ideas for further work are stated.

1.9 Summary

The awareness of the safety and health of products make modelling toxicology model an important domain. All chemicals should be tested to minimise their affect on living things such as humans and animals, as well as the environment which must be kept safe. There is a need to properly model the toxicology data carefully by applying data mining processes.

Therefore in this thesis, a data and flexible model representation was proposed in a more general framework towards data and model governance. Within this thesis, the aim is to develop an ensemble method that makes an improved prediction by reusing the quality models from a collection of models.

The proposed method for comparing models from a collection of models will help in optimising the ensemble process where only relevant models to be included in an ensemble. For the ensemble process, the ranking system will select the model using a cost function. The cost function which is defined as Classifiers Ranking Value (CRV) is a composite of three performance measures (Acc, FNR and FPR). This will ensure that only best models will be included in an ensemble. Based on the thesis objectives and the contributions made, this thesis can be explored by others to enhance the knowledge in the future.

Chapter 2

State of the Art

2.1 Introduction

This chapter aims to discuss the subjects, techniques and algorithms covered within this thesis to provide an overview of computational toxicology approaches. Existing work related to the thesis subject will be referred to and discussed. An overview of the machine learning algorithms used in the experimental work to support this thesis will also be introduced and briefly explained. The shortcomings of existing approaches will be examined as a basis for justifying the original merit of the work proposed in this thesis.

A classification model is a model that holds the information of a function (classifier) that classifies the instances to targeted classes (Tan et al. 2005). The results of the classified instances are stored as a confusion matrix. The number of classes are differentiated between binary classification model and multi class classification model. The binary classification models have only two classes, normally represented as true and false class. For multi class classifica-

tion models, the number of classes will be more than two.

The most useful performance measure for classification models is the accuracy (Acc). There are other performance measures that can be calculated using a confusion matrix for the binary classification model such as True Negative Rate (TNR), True Positive Rate (TPR), False Negative Rate (FNR) and False Positive Rate (FPR).

This chapter is structured as follows: Section 2.2 presents the overview of Knowledge Discovery Process and model management. Section 2.3 describes definition of predictive toxicology model, the representation, the performance measures and data mining functions applied. Related work on model comparison will be discussed in Section 2.7. The taxonomy for model combination and ensemble methods is discussed in Section 2.8. This chapter ends with genetic algorithms and summary.

2.2 Knowledge Discovery Process

The processes of generating predictive models involve data preparation, checking of data quality, feature selection process, modelling, prediction, and analysis of results. The whole process of data mining is known as knowledge discovery. The steps of knowledge discovery (shown as Figure 1.1) are described in Section 1.1.

There are many freeware data mining tools such as Weka (Waikato Environment for Knowledge Analysis) (Witten et al. 1999, Bouckaert et al. 2010) and KNIME (Konstanz Information Miner) (Berthold et al. 2009). Commercial tools are also available such as SPSS Modeler

provided by IBM SPSS and Oracle Data Mining by Oracle allow similar functionality in generating predictive models. For this thesis, Weka was used as a data mining tool to generate the pool of models.

2.2.1 Model Management

Any collection of models generated using data mining tools needs proper management. Liu & Tuzhilin (2008) studied the problem of how to develop automated model base analysis tools. There is a problem because of the amount of data that has been collected and the real world problems studied have become more complex. Before this, a data mining application may have only required a few models built to solve a problem. They raised the issues in model management as follows:

1. Models building and storing

   For example, how to automate the models generation and the storage of the models.

2. Models reusing

   The models stored in the repositories can be retrieved and further analysed.

This research has moved toward the objective of generating collection of models. The models can be selected by analysing how to compare the models in the context of data and model governance. The relevant models selected can be improved by making an ensemble from them. The performance may be improved and an end user may get benefits from the model management processes.

2.3 Predictive Toxicology

A predictive model is a model that can be used to predict or estimate the target values of future cases (Fayyad et al. 1996). Predictive toxicology is the discipline of predicting the toxic effects of chemical compounds against human, animal and environmental health (Trundle 2008). The predictive model may predict whether a new chemical compound is toxic or non toxic to living organisms.

In Predictive Toxicology, the goal is to describe the relations between chemical structure of a molecule and its biological and toxicological processes (Neagu et al. 2005). The relation is used to predict the behaviour of a new unknown chemical compound.

The toxicity level may vary from one organism to another. For example a chemical compound may have greater toxic effect on some animal species than others. In the production of products, for example, the level of pesticides in a chemical compound is very important because it can be harmful not only to living things but also to the wider ecosystem.

2.3.1 Predictive Model Representations

Data mining tools have been developed to produce one model for a single data set and the model produced is based on a single technique. In a real situation, there are many data and models available in different sources. The main issue is how to manage the data and the models. Two possible approaches are via the use of Object Oriented Database (OODB) which can represent predictive models

when it comes to huge amount of data because it can normalize and represent the record by objects or classes. Another alternative is the use of XML to map the data and the models.

The results from data mining processes can have different types (such as class type or pattern) and models can be represented in many ways. A standard representation of a predictive model is needed to access these predictive models developed with different resources. XML can be used as the basic format of representation and provides a method to represent and describe the information. The purpose of an integrative approach for data and model representation is to visualize the model, extract the parameters of the models, process and manage the models in relation to the available data. More significant processing can be done further to the models, such as comparison, selection and combination between them to respond to subsequent tasks. Languages that represent predictive models based on XML are listed below:

- PMML (Predictive Model Markup Language) is a standard XML-based language used to represent predictive models and allow sharing of models to compliance applications. It was established by the Data Mining Group (DMG) and has 4 components: Data Dictionary, Mining Model, Transformation Dictionary and Model Statistics (DMG 2012). PMML is still in the development process. There is a workshop on PMML modeling held in year 2011 to discuss the issues and to enhance the representation.

- Chaves et al. (2006) developed a PMML compliant scoring engine called Augustus. It is an open source PMML-compliant

scoring engine designed and developed using Python. The standard components used by Augustus are from PMML and added other new components such as data management component, utilities for processing PMML files and run time support.

- PMQL (Predictive Modelling Query Language) is a specialized query language for interacting with PMML documents. It is embedded within DeVisa framework developed by Gorea (2008), which provides functions such as scoring, model comparison, model composition, model searching, statistics and administration through a web service interface.

- The Hybrid Intelligent Systems Markup Language (HISML) is a XML proposal for knowledge representation, data exchange and analysis of experimental data, based on a modular implicit and explicit knowledge-based intelligent system It was proposed by Neagu, Craciun, Chaudhry & Price (2007).

- ToxML is an XML database standard based on toxicity controlled vocabulary for use in database standardization. It was developed by scientists at Leadscope Inc. for application in areas such as genetic toxicity, carcinogenicity and chronic toxicity (Leadscope 2012).

From all the representations, XML is used as a basis to represent information in the standard format. The flexibility in defining tags make it easy to construct. Furthermore, XML files can be published through the Internet and become accessible. The proposed representation to represent predictive toxicology models can be found in

Chapter 4.

2.3.2 Retrieving XML Documents

XML file is a document containing information that is represented in a standard format. Thus, it can be processed and manipulated similar to a database. The following are technologies that can be used to retrieve the information from an XML file.

2.3.2.1 XQuery : An XML Query Language

XML documents can be queried using an XML query language called XQuery. XQuery is similar to Structure Query Language (SQL). The flexibility of XML allows it to represent diverse sources of information. XQuery also offers flexibility in retrieving and interpreting represented tags information.

The main features of the XQuery are:

- To extract and manipulate data from XML documents.
- To use SQL-like "FLWOR expression" which are FOR, LET, WHERE, ORDER BY, RETURN.
- To provide syntax to construct new XML documents.

XQuery is still in the development process. It does not yet allow the update of XML documents or databases and lacks full text search capability. Figure 2.1 is an example of the XQuery language used to retrieve a $nameofemployee$ with salary more than £30 from an xml file named $employees.xml$

```
for $i in doc("employees.xml")/company/data
    where $i/salary > 30
        order by $i/name
return $i/name
```

Figure 2.1: An Example of XQuery Statement

In this thesis, all PTML models from the collection of models were retrieved using XML parser. The flexibility of XML parser functions such as allows to build XML documents, navigate XML structure, and add, modify, or delete elements and content of an XML documents make it suitable for a large number of PTML models and its structure.

2.3.2.2 XML Parser

An XML parser is a software that reads XML files and is able to parse all the data from the files using tags defined. It is a language that provides classes to process XML files. It is suitable for huge documents and able to parse complex XML structures. It is under the package of $javax.xml.parsers$. Figure 2.2 is an example of a function inherited from $javax.xml.parsers$ to parse all information from an XML file and store in variable $tempEmp$. The value in the $tempEmp$ can be saved into database or can be printed to the screen.

```
public void endElement {
    if(qName.equalsIgnoreCase("header")) {
        myEmpls.add(tempEmp);
    }else if (qName.equalsIgnoreCase("version")) {
        tempEmp.setversion(tempVal.trim());
    }else if (qName.equalsIgnoreCase("date")) {
        tempEmp.setdate(tempVal.trim());
    }else if (qName.equalsIgnoreCase("author")) {
        tempEmp.setauthor(tempVal.trim());
    }else if (qName.equalsIgnoreCase("source")) {
        tempEmp.setsource(tempVal.trim());
    }else if (qName.equalsIgnoreCase("comments")) {
        tempEmp.setcomments(tempVal.trim());
    }
}
```

Figure 2.2: An Example of XML Parser Statement

2.4 Confusion Matrices

Confusion matrix is the raw output generated from a classification model. The output shows the correctly and incorrectly classified of instances. Table 2.1 is a representation of confusion matrix for binary class classifiers and Table 2.2 is a confusion matrix for multi class classifiers. From the confusion matrix, various performance measures can be calculated as discussed in the next section.

2.4.1 Binary Confusion Matrices

Kohavi & Provost (1998) defined a confusion matrix that contains information about actual and predicted classifications done by a classification model. Performance of such models is commonly evalu-

ated using the data in the matrix (see Table 2.1). Table 2.1 shows the confusion matrix for a binary class classifier.

Table 2.1: Confusion Matrix of Binary Classification: True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP).

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

TP is the number of correct predictions for positive output (e.g. Yes),

FP is the number of incorrect predictions for the negative output (e.g. No),

FN is the number of incorrect prediction for the positive output, and

TN is the number of correct predictions for the negative output.

2.4.2 Multi Class Confusion Matrices

The confusion matrix for multi class classifiers is shown in Table 2.2. The intersection of the first column (Class A) with the first row is the True Positive (TP) value for Class A. True positives for second, third and forth columns are the diagonal values of the confusion matrix.

Table 2.2: Confusion Matrix for a Multi Class Classifier.

	Class A	Class B	Class C	Class D
Class A	$TP_{AA(1,1)}$	$e_{AB(1,2)}$	$e_{AC(1,3)}$	$e_{AD(1,4)}$
Class B	$e_{BA(2,1)}$	$TP_{BB(2,2)}$	$e_{BC(2,3)}$	$e_{BD(2,4)}$
Class C	$e_{CA(3,1)}$	$e_{CB(3,2)}$	$TP_{CC(3,3)}$	$e_{CD(3,4)}$
Class D	$e_{DA(4,1)}$	$e_{DB(4,2)}$	$e_{DC(4,3)}$	$TP_{DD(4,4)}$

2.5 Classifier Performance Measure

Performance measures for binary class classifiers can be calculated using Accuracy (Acc), False Negative Rate (FNR), False Positive Rate (FPR), True Positive Rate (TPR) and True Negative Rate (TNR). The performance measures applied within this thesis will be discussed in the following section.

2.5.1 Binary Class Performance Measures

The performance measures can be calculated as follows (Kohavi & Provost 1998, Fawcett 2004):

$$TPRate = \frac{TP}{(TP + FN)} \tag{2.1}$$

$$FPRate = \frac{FP}{(FP + TN)} \tag{2.2}$$

$$FNRate = \frac{FN}{(FN + TP)} \tag{2.3}$$

$$TNRate = \frac{TN}{(TN + FP)} \tag{2.4}$$

$$Acc = \frac{TP + TN}{(TP + FP + FN + TN)} \tag{2.5}$$

$TPRate$ is the proportion of correct predictions for positive class (e.g. Yes),

$FPRate$ is the proportion of incorrect predictions for the negative class (e.g. No),

$FNRate$ is the proportion of incorrect prediction for the positive class,

$TNRate$ is the proportion of correct predictions for the negative class, and

Acc is the proportion of correct predictions for all classes.

2.5.2 Multi Class Performance Measures

The classification accuracy of a multi class classifier is the ratio of the sum of the principal diagonal values to the total of values in the confusion matrix. If C indicates the confusion matrix, Prasanna et al. (2007) defined the classification accuracy Acc as follows:

$$Acc_C = \left(\frac{\sum_{i=1}^{N} C_{ii}}{\sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij}} \right) \tag{2.6}$$

where:

N is the number of classes,

i refers to the row index, and

j refers to the column index for the confusion matrix C.

The Error Rate (ER) for the classifier is the complement of the

Acc. Error rate can be calculated as follow:

$$ErrorRate = 1 - Acc \tag{2.7}$$

2.6 Generating Predictive Models

2.6.1 Weka

Weka (Waikato Environment for Knowledge Analysis), is a Java based tool that incorporates many well known machine learning algorithms for data mining (Witten et al. 1999, Bouckaert et al. 2010). The tasks offered in this tool are data pre-processing, classification, regression, clustering, association rules, and visualization (Witten. et al. 2011). The tools are able to make predictions using the interface provided or as a package in a Java development environment. Detail of functions used within this research will be explained in Section 3.5.

2.6.2 Feature Selection Algorithms

Feature selection is a technique to identify the most relevant features or attributes which are used to generate predictive models on a training data set. By using a raw data set (with no feature selection), the model will have to learn from all the features available. For data sets that have hundreds of features such as toxicology data sets, the Acc of the models may be lower because most of the features have no relationship to target classes (see Table 5.22) and the Acc is improved when the feature selection algorithms are applied (Neagu, Guo, Trundle & Cronin 2007). It is because the model learns better

about the data using the relevant attributes selected using a feature selection algorithm, while irrelevant features do not enter noise anymore during the learning stage. Trundle (2008) has studied the importance of feature selection in toxicology data sets. The feature selection process can reduce noise and insignificant attributes in the training data set and this will be improved the classification accuracy (Luukka 2011). In this thesis, the feature selection algorithms chosen are briefly explained in Section 3.5.

2.6.3 Classification Algorithms

Weka offers collection of machine learning algorithms and functionality of classification algorithms. The classification algorithms were used within this thesis are listed below.

- K-Nearest neighbors classifier (weka.classifiers.lazy.IBk)
- Decision trees (weka.classifiers.trees.J48)
- Numerical prediction (weka.classifiers.rules.JRip)
- Naive Bayes (weka.classifiers.bayes.NaiveBayesUpdateable)
- Multilayer Perceptron (weka.classifiers.functions.MultilayerPerceptron)
- Bagging (weka.classifiers.meta.Bagging)
- Boosting (weka.classifiers.meta.AdaBoostM1)
- Stacking (weka.classifiers.meta.StackingC)
- Ensemble Selection (weka.classifiers.meta.EnsembleSelection)

- Random Forest (weka.classifiers.trees.RandomForest)

The description of each classification algorithm will be discussed in Section 3.

2.7 Predictive Model Comparison

Comparison of predictive models can be accomplished by measuring the similarity between them. Similarity and distance metrics are complementary to each other. For example the Hamming distance is one of the distances used to calculate the dissimilarity between two strings (Hamming 1950). Todeschini et al. (2004) proposed a new measure to calculate a distance between two models using hamming distance.

Choi et al. (2010) surveyed similarity of 76 binary similarity and distance measures. They had grouped the similarity and distance techniques using hierarchical clustering to estimate the similarity among the measures. Researchers can refer to a group for selecting the appropriate similarity measure to be applied, depending on the data.

Lesot et al. (2009) explored the similarity measures of different data types. They found that the nature of data is the main factor when deciding which similarity measure is to be applied. In that paper, they studied similarity measures for binary and numerical data.

Sequeira & Zaki (2007) explored similarities across data sets using a two step solution: constructing a condensed model of the data

set and identifying similarities between the condensed models. Their technique is limited to finding similar subspaces based on the structure of the data set alone, without sharing the data sets.

In this thesis the similarity of predictive models was proposed by comparing the element of predictive models that will be discussed in Chapter 5.

2.8 Predictive Model Combination

The technique of model combination has appeared under various names such as hybrid method, decision combination, multiple experts, mixture of experts, classifier ensembles, cooperative agents, opinion pool, decision forest, classifier fusion, and combinational systems (Parvin et al. 2009).

2.8.1 Model Ensemble

The idea of a model ensemble is to have more expertise (predictive models) involved in decision making rather than a single model used in predicting the output (Rokach 2009). It is more effective to use a collection of predictive models for large data sets, or for data sets which are diverse to select the relevant predictive model in the collection of models. Diverse data sets can be produced by applying different feature selection algorithms as explored in Chapter 5.

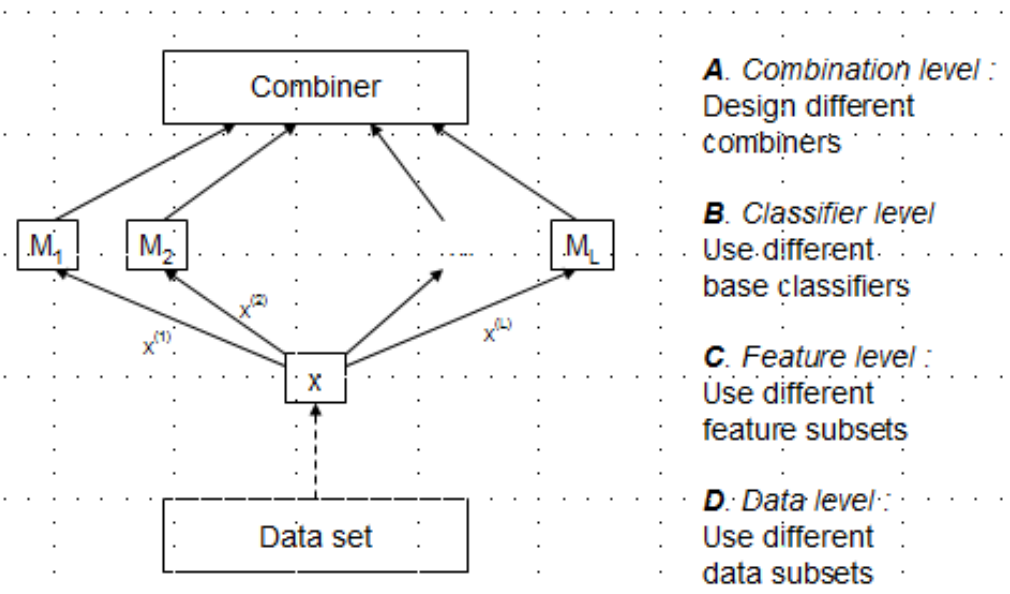

Figure 2.3: Approaches to Building Classifier Ensembles (Kuncheva 2004)

2.8.2 Ensemble Methods

Figure 2.3 shows that an ensemble classifier has to go through a number of the processes (Kuncheva 2004). At the data level (Level D), different subsets of data set are created in order to make independent classifiers. Each classifier will be used for the next step in Level B. Diversity of an ensemble model can be obtained by using different subsets of feature selection (Level C) and different base classifiers (Level B). Finally, Level A represents the different ways of combining classifier decisions. The final predictive model from the ensemble learning has proved to be a better performance compared to single predictive model (Woloszynski & Kurzynski 2011). Thus, this technique will often increase the performance of a predictive model (Dietterich 2000).

Wang (2010) implemented strategies for selecting the models to be included in an ensemble based on performance measures such as Acc, $Sensitivity$ and $Specificity$, and/or diversity. $Sensitivity$ is similar to FNR and $specificity$ is similar to FPR. This thesis applied model selection strategies using a cost function that is a composite of Acc, FNR and FPR. With the composite of Acc, FNR and FPR, the ensemble constructed will be improved in a specific class. Thus, the strategies may be useful for unbalanced data or the number of misclassifying the samples of assigned class is higher than the other (Wang 2010).

There are many studies which have implemented various techniques to construct an ensemble (Sirlantzis et al. 2008). The diversity measure and decision strategy are most important factors that effect the accuracy of an ensemble. Table 2.3 summarises the studies that have been done for constructing an ensemble.

Table 2.3: Techniques Use for Constructing an Ensemble.

Author	Diversity Measures	Decision Fusion Strategy	Performance measures
Masisi et al. (2008)	Kohavi-wolpert Variance	Voting	Acc
Mehmood et al. (2010)	Error Rate	Weight Majority Voting	Acc
Wang (2010)	Coincident Failure Diversity (CFD)	Voting	Acc, $Sensitivity$, $Specificity$
Khakabimamaghani et al. (2010)	Disagreement Measure	Thresholded Voting	Acc
Nabiha et al. (2011)	Correlation Coefficient, Q Statistics, Disagreement Measure	Voting, Weighted Voting	Acc

2.8.3 Ensemble Learning Algorithms

There are many ensemble learning techniques discovered and implemented by researchers. Every technique has its own pros and

cons but the performance of ensemble technique always give a high impact to the performance of predictive model generated.

Ensemble methods have been applied in many applications such as Arabic handwritten recognition (Nabiha et al. 2011), classifying spam email (Wang 2010), dynamic signature authentication (Al-Muhanna & Meshoul 2011), human face and voice recognition (Xiaoyan et al. 2009). All the studies agreed that the accuracy improved when applying an ensemble method compared to a single classifier (Chitra & Uma 2010, Bakar et al. 2011).

Polikar (2006) suggested that all ensemble systems consist of two main processes. The first process is related to diversity of ensemble and the second process is related to combining the outputs of individual classifiers in an ensemble. For the first process, the strategy to generate the most diverse classifiers is important. There are different parameters that can be used to generate diverse classifiers such as different feature selection and machine learning algorithms. The second process is related to the decision fusion strategy such as majority voting and weighted majority voting. The established ensemble methods will be discussed in the following section.

2.8.3.1 Bagging

Bagging takes each model in the ensemble and gives it an equal weight. It trains each model in the ensemble using a randomly-drawn subset of the training set in order to promote model variance and diversity. As an example, to achieve very high classification accuracy, the random forest algorithm combines random decision

trees with this algorithm (Breiman 1996). Majority voting is used as the decision strategy. Normally, bootstrap aggregating is often abbreviated as bagging.

2.8.3.2 Boosting

Boosting is different from bagging. It focuses on the instances for data set that are used to generate predictive models. Boosting is a general method for improving the performance of any learning algorithm (Freund & Schapire 1996). In theory, boosting can be used to significantly reduce the error of any "weak" learning algorithm that consistently generates classifiers which need only be a little bit better than random guessing. Additionally, this algorithm involves incrementally building an ensemble by training each new model instance to emphasize on the training instances that previous models mis-classified. Sometimes, this technique will be more likely to over fit the training data but it has proved to get better accuracy than bagging. Boosting shares similar decision fusion strategy to bagging which is majority voting.

2.8.3.3 Stacking

Stacking assumes that the model generated is adequately flexible to represent any of the ensemble algorithms. The flexible model generated from stacking starts by giving training to a master model to make a final decision based on the decisions of another collection of models. The idea is to combine multiple models in a different way by introducing the concept of a meta learner. The meta learner uses the

output from a base classifier as an input to make the final decision. The base classifiers are trained with a different training set Polikar (2006).

2.8.3.4 Bayes Optimal Classifier

The Bayes Optimal Classifier is generally considered to be the principle ensemble among ensemble learning algorithms. It is because this classifier is an ensemble that takes all hypotheses in the hypothesis space. A vote proportional is given for each hypothesis to the possibility that the training data set would be sampled from a system if that hypothesis was true. The vote of each hypothesis is also multiplied by the prior probability of that hypothesis (Parvin et al. 2009).

2.8.3.5 Hybrid Intelligent System

Hybrid intelligent systems involve a combination of local and global models as ensemble experts by mixing technologies in hybrid systems. The objective of this approach is to improve the prediction accuracy, and also to provide reasonable training response time by using parallel processing (Neagu, Craciun, Chaudhry & Price 2007). Santos & Sabourin (2011) proposed a hybrid search algorithm to select a population of classifier ensemble.

2.8.3.6 Ensemble Selection from Library of Models

The method of ensemble selection from a library of models was proposed by Caruana et al. (2004). The library is a collection of mod-

els generated using different learning algorithms and parameter settings. They used forward stepwise selection for adding the models into the ensemble to maximise the performance. The performance measures were focussing on accuracy, cross entropy, mean precision, and receiver operating characteristic (ROC).

The ensemble selection procedure proposed by (Caruana et al. 2004) is as follow:

1. Start with the empty ensemble and a library of models.
2. Add to the ensemble the model in the library that maximises the ensemble's performance to the error metric on a hillclimb (validation) set.
3. Repeat Step 2 for a fixed number of iterations or until all the models in the library have been used.
4. Return the ensemble from the nested set of ensembles that has maximum performance on the hillclimb (validation) set.

The method combines all possibilities of models in collection and does not consider a diversity measure. In this thesis, the ensembles were optimised by selecting relevant models using a cost function (composite of Acc, FNR and FPR) and combining the classifiers using a diversity measure such as disagreement measure and double fault measure.

2.9 Diversity Measures

Diversity measure is important in an ensemble. One of the issues in building an ensemble is to have diverse models in an ensemble. There are no rule to indicate which diversity is suitable for certain problems or data sets (Polikar 2006).

There are many definitions about the diversity of models, but they are all grouped into two categories which are pair wise and non pair wise (Kuncheva 2005). The agreement for a relationship between two binary classifiers i and k is presented in table 2.4.

Table 2.4: Relationship Between a Pair of Classifiers

	$D_k correct(1)$	$D_k wrong(0)$
$D_i correct(1)$	N^{11}	N^{10}
$D_i wrong(0)$	N^{01}	N^{00}

where:

N^{11} is the number of correct predictions made both classifiers i and k,

N^{10} is the number of correct predictions made by classifier i and incorrect predictions made by classifier k,

N^{01} is the number of correct predictions made by classifier k and incorrect predictions made by classifier i,

N^{00} is the number of incorrect predictions for both classifiers i and k,

The diversity measures considered in this proposed ensemble was summarised from Kuncheva & Whitaker (2003) as well as Bian & Wang (2007). The diversity can be grouped into two groups: pair wise and non pair wise measures. The diversity of pair wise measure

can be calculated between two base leaners in an ensemble. For an ensemble, the diversity can be calculated by averaging all the values from each pair of classifiers. Non pair wise diversity measures the diversity by averaging all the performance measures such as Acc of all base learners in the ensemble.

Bian & Wang (2007) had studied and investigated the diversity measures. They grouped the similar diversity measures into three groups as follow:

- Group 1 : consists of Disagreement Measure, Kohavi-Wolpert Variance and Entropy Measure.
- Group 2 : consists of General Diversity and Coincident Failure Diversity.
- Group 3 : consists of Double-fault Measure, Q Statistic, Correlation Coefficient, Measure of Difficulty and Interrater Agreement Measure.

The diversity of classifiers in the ensemble may produce better result in prediction giving higher accuracy compared to a single classifier (Kuncheva & Whitaker 2003).

2.9.1 Disagreement Measure

The Disagreement measure is to calculate the diversity between two classifiers which are a base classifier and a complimentary classifier. It is the ratio of correctly classified samples between two classifiers for both classes. The Disagreement Measure between two classifiers is as follows:

$$Dis_{i,k} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{2.8}$$

2.9.2 Double-fault Measure

The Double-fault measure calculates the diversity between classifiers to find which classifiers are least related to a base classifier. It is the ratio of incorrect predictions by both classifiers. The Double-fault measure between two classifiers is as follows:

$$DF_{i,k} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{2.9}$$

In this thesis, the diversity measures applied in ensemble are from two different groups as suggested by Bian & Wang (2007). The diversity measures are disagreement measure from Group 1 and double-fault measure from Group 3. Chapter 6.2.1 will discuss the implementation of diversity measures that were applied in the ensemble proposed.

2.10 Decision Fusion Strategies

Decision fusion strategies give an important impact to the final prediction results of an ensemble of classifiers. Ghosh et al. (2011) discussed the decision fusion strategies available and classified them into two methods. The methods are:

- Utility-based

  Utility-based methods provide the function to combine the deci-

sion based on the output generated from each classifier. It does not consider any knowledge or evidence from previous predictions. Some of the methods included are simple average and voting techniques.

- Evidence-based

  Evidence-based is, in contrast to utility-based, that decision needs knowledge or evidence from previous prediction of output generated from each classifier. Some of the methods included are Bayesian and the Dempster-Shafer methods.

This thesis applies simple majority voting because of the criticality in predicting a toxic class. Use of predictive toxicology models with high confidence rely on low FNR. Thus the decision fusion strategy must carefully predict the toxic class. The decision of the voting technique to predict a chemical's toxicity has to be fifty percent or more to vote the chemical compound as toxic while less than fifty percent will be non toxic. The results of the prediction may give a high confidence in predictive toxicology. The methods for simple majority voting applied in this thesis will be discussed in Section 6.2.3.

2.11 Optimisation Technique

The process of constructing an ensemble of models requires computational time because of the complex processes such as generation of models, selection of models, combination of models and performance evaluation (see Figure 2.3). The optimisation technique plays

an important role in optimising the selection of relevant models from a collection of models to be included in constructing an ensemble. At the same time, optimisation technique will maintain the objective of ensemble in getting higher Acc compared to a single classifier.

The main objective of optimisation technique is to choose the most relevant parameters in order to get the best results. In this case, the relevant parameters are:

- the most relevant models,
- the most diverse classifier,
- the highest Acc, and
- the optimum number of candidates in ensemble.

The value of parameter depends on the situation and the problem. Thus, the value of the objective function in this case can be to maximise or minimise the value of the objective function. For example in this thesis, the proposed optimisation technique will maximise the performance measures (Acc, FNR and FPR) and minimise the number of candidates in an ensemble to speed up the ensemble process as discuss in Chapter 6.

2.11.1 Genetic Algorithms

The Genetic Algorithm (GA) is a technique to find the optimum solution by applying the principle of evolutionary biology. The technique tries to mimic the same biology processes of generating human genetic. The method of selection, recombination and mutation, and reproduction will be repeated to find a solution to a problem.

There are many studies that apply GA in their ensemble construction such as Khakabimamaghani et al. (2010), Mehmood et al. (2010), Musehane et al. (2008) and Masisi et al. (2008). In this thesis, GA was proposed as the optimisation technique applied in the ensemble . Section 6.4 will discuss further on the proposed optimisation that has been implemented in this research.

2.12 Summary

In this chapter the methods and techniques that are relevant to improving the computation of predictive toxicology models were discussed. The literature review starts with a predictive toxicology definition, the performance measures and techniques to calculate the similarity of models. Methods to solve the problem in selecting the most relevant models in a collection of models and make them into an ensemble were briefly stated and reviewed. The detailed process and issues in ensemble building were reviewed carefully.

The methods used within this research were justified in this chapter. In the following chapters, all the proposed methods implemented will be introduced and discussed.

Chapter 3

Methodology and Proposed Framework for Data and Model Governance

3.1 Introduction

The increasing variety of data mining tools offers a large palette of types and representation formats for predictive models. Managing the models then becomes a big challenge, as well as reusing the models and keeping the consistency of model and data repositories because of the lack of an agreed representation across the models. The flexibility of XML representation makes it easier to provide solutions for Data and Model Governance (DMG) and support data and model exchange. Predictive Toxicology was chosen as an application field to demonstrate the proposed approach to represent predictive models linked to data for DMG.

In this chapter, the framework of data and model governance will be discussed. Furthermore the detailed methods of the research design are briefly explained. All the data sets and data mining tools used in the research will also be explained. The contribution for this chapter is a new framework for data and model governance.

3.2 Methodology and Research Design

The objectives of this research can be accomplished by implementing a structured research design. This research follows the methodology proposed as shown in Figure 3.1. The main research problem is to find the relevant predictive model to be used as an expertise to predict new chemical compound.

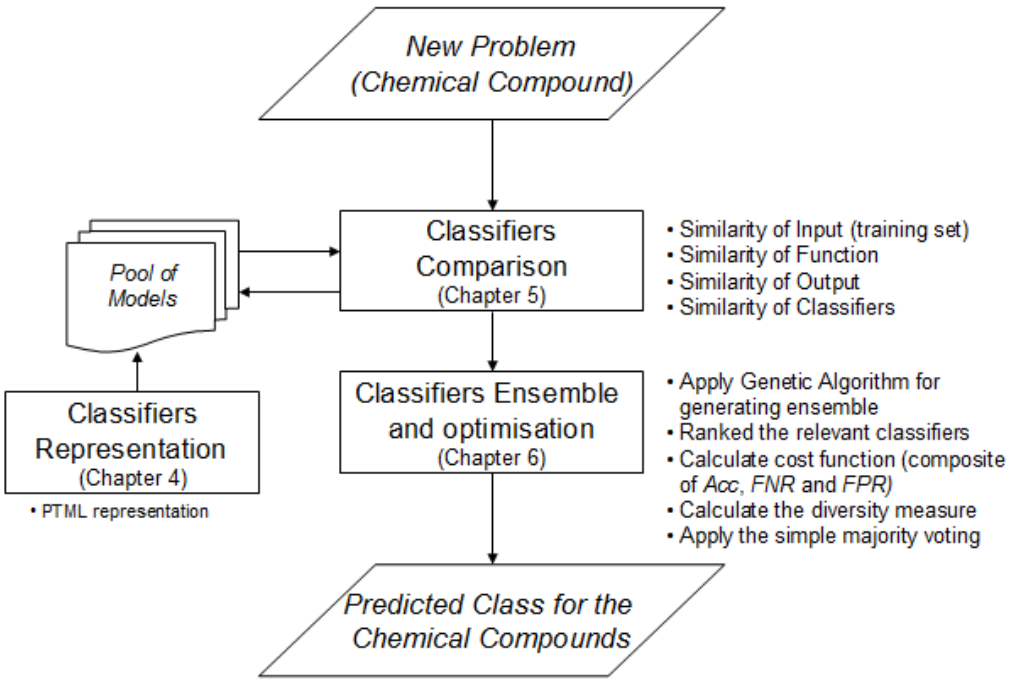

Figure 3.1: The Method Followed by the Research Study

Collections of classifiers are available from different sources and in many formats. Some of them are in standard format such as PMML, PToxML and PMQL. Others were generated using lots of available data mining tools such as Weka or data mining functions from Matlab. Usually, the models generated will have their own format and not be in a standard format.

A standard representation is needed in order to access those valuable models. In this research, a standard representation in the from of Predictive Toxicology Markup Language (PTML) was proposed. The representation will be explained in Chapter 4. It uses Extensible Markup Language (XML) as a representation and can be used to represent any predictive models. The models will be represented with minimal tags that are necessary to describe the models and for further analysis. Using the PTML, a collection of models can be accessed and manipulated.

Before the models can be selected, comparison of the classifiers is important as only the relevant models should be selected for a certain problem. In Chapter 5, the method of classifiers comparison is proposed to compare the similarity of predictive models from the collection of models. The similarity method proposed considers three elements of a predictive model such as Input (training set), Function (classifier properties) and Output (confusion matrix). Chapter 5 will demonstrate with experiments that the method is able to find the relevant models for any problem. The similarity method is also able to compare the binary and multi classes models.

For the last part of the research, the selected relevant models can

be optimised for better performance in prediction. In predictive toxicology, the important performance measure is False Negative Rate (FNR). A lower FNR tells us that the model is able to correctly predicted toxic class. That is very critical performance measure to be aware of compared to False Positive Rate (FPR) where FPR is a performance measure related to prediction of non-toxic class. The core performance measure of a predictive model is Acc.

Ensemble methods have been shown to achieve better accuracy compared to a single classifier (Bakar et al. 2011). Most ensembles focus on accuracy as their main performance measure. For this research, the domain of predictive toxicology requires that the toxicity of a chemical compound be predicted correctly. Thus FNR is important to be considered and the models should be able to predict with lower FNR.

With this objective in mind, the proposed ensemble was developed and will be discussed in Chapter 6. The proposed ensemble was optimised and able to predict the new chemical compounds with lower FNR and lower FPR thus increasing the accuracy of the models. The optimisation technique used was a Genetic Algorithm (GA) and a simple majority voting technique was used as a decision making strategy.

In order to find the ensemble candidates, the models were ranked by using a cost function that combines Acc, FNR and FPR as a composite performance measure. Diversity measures such as disagreement measure and double fault measure were applied in constructing the ensemble method. By combining all the methods pro-

posed, the results of the prediction for all experiments conducted outperform other ensemble techniques such as Bagging, Boosting, Stacking and Bayes.

3.3 Data and Model Governance

The processes of generating predictive models involve data preparation, checking of data quality, reduction, modelling, prediction, and analysis of results. Each benchmark model is trained to find which attributes and model parameters are most important to producing a quality model. The tuning process involves selecting optimum model parameters such as the number of fold-cross validation and classifier type. Selection of the optimum attributes from the data set is another step of the tuning process. This will be iterated until the optimum combination of parameters is found to generate a better quality model.

In the case of data set updates, predictive models related to the older data set become unreliable. To generate new and reliable predictive models for the updated data set, the iteration process of tuning and finding the most relevant combination of attributes and model parameters must take these changes into account. This makes it necessary to recall the predictive model generation step for an up-to-date model repository.

This continuous process shows that there may be thousands of data mining models related to a single data set shared among data mining researchers, generating versions of predictive models and re-

lated data sets. Thus, to monitor and maintain changes between data and models becomes even more challenging. There is a need to define the relationship between data and models, so that the iteration process of generating new predictive models integrates consistently in the modelling framework and this evolution also needs to be recorded. These repositories of data mining models should keep information on historical developments which are also valuable for analysis.

The models were reused from a repository of existing models as a more efficient way of choosing the relevant models and reusing existing knowledge in the field of predictive toxicology. In predictive toxicology, there is a great emphasis paid to the development of QSARs (Quantitative Structure Activity Relationship) validated by some experts and used to specific tasks e.g. recommended by regulatory bodies for testing chemical from a particular class of chemical compounds. Such models are used at different times but currently, there is not a consolidated approach on maintaining them for future use, and this gap motivates some of our research. Development of such models requires expertise and time consuming procedures for validation and are later reported for use by industry and regulatory bodies. In this work we presume such models are part of the collection of models we make use of.

Existing models are represented in various platforms, for example text files, relational database or different internal format produced from data mining tools (e.g. .arff produced by Weka and .fis produced by Matlab). The challenge is how to share those models

between researchers. One of the solutions is to represent the model in XML format where the models can be published through the web in standard form and can be accessed easily later. For that reason the representation in XML was proposed to be a bridge and a flexible solution to deal with the current diversity of model representations.

Searching the best model from a collection of models can be done with a selection of criteria. Further analysis of the model is focussed on comparing the performance of existing models or creating a combination between models. Caruana et al. (2004) found that these approaches have often been proved to achieve better predictive performance compared to producing a single predictive data mining model.

3.3.1 Data and Model Governance Framework

Data quality management (DQM) focuses on collecting, organising, storing, processing and presenting high quality data to the stakeholders for organization. For data governance, it is part of DQM which specifies the framework for decision rights and accountabilities (Wende 2007).

A global view of predictive modelling must involve data and models. Thus this valuable combination of data and models needs proper management. Data Governance is defined by IBM as the quality control discipline for assessing, managing, using, improving, monitoring, maintaining, and protecting organizational information (IBM 2012).

The process of generating predictive models involves Data Preparation, Feature Selection, Data Modelling and Prediction, Evaluating

and Validating the Model, and Implementing and Maintaining the Model. The process of predictive modelling needs to be properly managed and controlled because of the consequences in the decision making. This research is moving towards Data and (predictive) Model Governance. DMG is defined as the set of quality control processes for assessing, managing, using, improving, monitoring, maintaining, and protecting data and (predictive) model information.

(Fu et al. 2011) studied data governance issues and proposed a framework for data governance related to data storage management for example accuracy, completeness and integrity. Besides data governance, models should also be the main assets that needs to be managed properly. The governance process complements the management process. The management process focuses on the decision and implementation to be made within the organisation, but in the governance process, the most important is the accountability of the decision made by the management process (Khatri & Brown 2010, DGI 2010).

From the view of predictive modelling governance, the data and model have to be properly managed to achieve quality prediction. In this research, the accountability of the model selection and combination is refered to the user's requirement. For example a user may want a model with high Acc and low FNR, so the selection of models and ensemble proposed in this thesis will follow the user's request. The proposed framework for data and model governance can be depicted as Figure 3.2.

Figure 3.2 represents data and model governance framework for

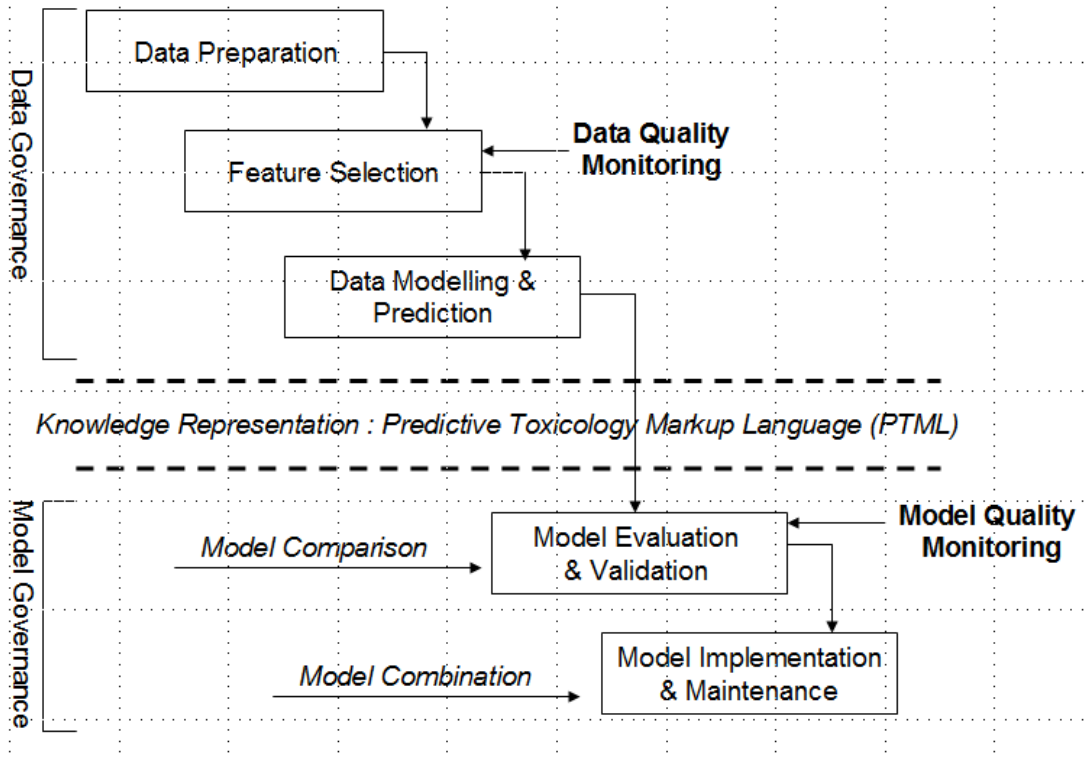

Figure 3.2: Data and Model Governance Framework

predictive modelling. From the framework, it shows that the whole process of predictive modelling involves governance tasks at every single stage for predictive modelling. Furthermore the framework emphasizes that the process of quality checking must be engaged at every task involved in predictive modelling.

The reuse of models from a collection of models will be categorised as model governance. The relevant classifiers from a collection can be chosen by making comparison between them. The methods was proposed to select and compare the classifiers from the collection. The detailed technique for the proposed classifiers comparison can be found in Chapter 5. Later the process will make a combination between them. Chapter 6 will discusses the ensemble method pro-

posed.

3.4 Data Sets

This research is focus on predictive toxicology models. The data sets used were five real data sets freely available from Demetra Project (Demetra). Data sets from UCI repository were also used in the experiments as a benchmark before the methods proposed can be applied to the real toxicology application.

3.4.1 Demetra Data Sets

The five data sets were used repeatedly throughout the experiments and the results are reported within this thesis. Trundle (2008) had used the same data sets and they are formally defined as:

1. Trout is defined as the acute toxicity for Rainbow Trout (Oncorhynchus mykiss) measured as a LC50 over 96-hours of exposure.

2. Daphnia is defined as the acute toxicity for Water Flea (Daphnia Magna) measured as a LC50 over 48-hours of exposure.

3. Oral Quail is defined as the acute oral toxicity for Bobwhite Quail (Colinus virginianus) measured as a LD50 over 14-days of exposure.

4. Dietary Quail is defined as the dietary toxicity for Bobwhite Quail (Colinus virginianus) measured as a LD50 over 8-days of exposure.

5. Bee is defined as the acute contact toxicity for Honey Bee (Apis melifera) measured as a LD50 over 48-hours of exposure.

Table 3.1 shows basic information on each of the endpoints: the number of instances indicating how many chemicals are included in each data set; the number of classes indicating how many different target classes exist for each endpoint (with numerical target values being discretised according to the Global Harmonisation System); and the class distributions indicating the number of instances belonging to each class overall, and in the training and testing sets (GHS 2012). Note that for the Demetra source, there is an overlap in the chemical instances in each of the endpoints i.e. a chemical may have a recorded toxicity value for more than one of the endpoints.

Table 3.1: Summary of the Five Demetra Data Sets Used in the Experimental Work Presented within this Thesis.

Data set	No. of Instances	No. of Features	No. of Classes	Class Distribution
Trout	282	250	3	129:89:64
Oral Quail	116	255	4	4:28:24:60
Daphnia	264	184	4	4:28:24:60
Dietary Quail	123	256	5	8:37:34:34:10
Bee	105	254	5	13:23:13:42:14

3.4.2 UCI Data Sets

There is a repository of data sets maintained by University of California Irvine (UCI) to facilitate research in data mining and knowledge discovery (Bay et al. 2000). This open archive consists of a wide variety of data types and application areas. For this research, the UCI

data sets were used to verify and validate the methods proposed within this thesis. We have selected five data sets also reported by other researchers and similar (to Demetra data sets) in terms of number of classes and coming from a medical domain.

Table 3.1 is the distribution of the UCI data sets (UCI 2012).

Table 3.2: Summary of the Five UCI Data Sets Used in the Experimental Work Presented within this Thesis.

Data set	No. of Instances	No. of Features	No. of Classes	Class Distribution
Blood Transfusion	748	5	2	178:570
Breast Cancer	699	11	2	458:241
Hepatitis	155	20	2	32:123
Liver Disorder	345	7	2	145:200
Pima Indian Diabetes	768	9	2	500:268

3.4.3 Collection of Predictive Models

The research aim was to find relevant models from a collection of models and to use them alone or as part of ensemble for predictions on new problems. To presume the collection of models, this section will describe the methodology used to generate the collection of models. All the predictive models were generated automatically based on PTML representation.

The collection of models was generated using numbers of classifiers to make the model diverse. Thus, the construction of an ensemble using those models will ensure that the ensemble is heterogeneous. Wang (2010) found that results and reliable classifications improved the Acc significantly when using a heterogeneous ensemble compared to other single model and ensemble filters.

The steps taken to automate the generation of PTML based on representation for predictive models are as follows:

1. Data Preparation

   The important step in the data mining process to generate a predictive model is data preparation. The huge amount of data is normally checked for mistakes, out of range values or impossible data combinations. Results can be misleading if the data is not properly prepared.

2. Model and Parameter Settings Selection

   Different combinations of input settings will be used to generate predictive models. The settings such as type of classifier and number of folds may affect the performance of the generated models.

3. Model Generation and Performance Test

   In this case, Weka, a Java based data mining tool, has been used to generate the data mining models but many other model generation tools may be also used (e.g. Oracle Data Mining and SAS Analytics). From the input given (data and model parameters), models are generated automatically and tested against test sets. These models are stored in text files of internal format (e.g. Weka Generated Model with file extension .model).

4. XML Model Generation

   The internal storage representation has to be converted into XML format for later processing and analysis. Predictive Toxicology Markup Language will be a bridge between various model

formats.

5. Model Representation Testing

   Models in XML form will be tested by retrieving the model using an XML Parser to check there are no syntactical errors in their representation.

6. Models Publishing

   The XML model can be published and stored in the repository for further processing tasks. In this case, the PTML model is used for data and model repositories.

3.5 Weka's Functions

Weka is a data mining tool that offer lots of functions related to knowledge discovery and data mining processes. The tool can be integrated in a Java based environment to make it flexible to developers. The Weka package can be included in Java sources and runs on various platforms. For the end users, Weka has a Graphical User Interface (GUI) with the user manual that makes it easy to use.

In this research, all the models generated were using Weka package being called into a Java program. Functions that were used in this research were select attribute (feature selection) and classify. Feature selection functions are used for finding the most significant attributes to be used for prediction. Classify functions are machine learning functions that will use to classify classes of the new problem. The output of the prediction are performance measures (Acc, FNR, FPR) and confusion matrix.

Following are the functions used for attribute evaluation (feature selection) within this research:

- CfsSubsetEval

  Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.

- Classifier subset evaluator

  Evaluates attribute subsets on training data or a separate hold out testing set.

- ConsistencySubsetEval

  Evaluates the worth of a subset of attributes by the level of consistency in the class values when the training instances are projected onto the subset of attributes.

All the attributes were searched using these algorithms:

- BestFirst

  Searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility.

- Genetic Search

  Performs a search using the simple genetic algorithm.

- Greedy Step Wise

  Performs a greedy forward or backward search through the space of attribute subsets.

For the machine learning algorithms, the classifier functions applied within this thesis were as follows:

- K-Nearest neighbors classifier (weka.classifiers.lazy.IBk)
  K-nearest neighbours classifier. Can select appropriate value of K based on cross-validation. Can also do distance weighting.

- Decision trees (weka.classifiers.trees.J48)
  Class for generating a pruned or unpruned C4.5 decision tree.

- Numerical prediction (weka.classifiers.rules.JRip)
  This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction.

- Naive Bayes (weka.classifiers.bayes.NaiveBayesUpdateable)
  Class for a Naive Bayes classifier using estimator classes. This is the updateable version of NaiveBayes.

- Multilayer Perceptron
  (weka.classifiers.functions.MultilayerPerceptron)
  A Classifier that uses backpropagation to classify instances. This network can be built by hand, created by an algorithm or both. The network can also be monitored and modified during training time.

- Bagging (weka.classifiers.meta.Bagging)
  Class for bagging a classifier to reduce variance. Can do classification and regression depending on the base learner.

- Boosting (weka.classifiers.meta.AdaBoostM1)
  Class for boosting a nominal class classifier using the Adaboost M1 method. Only nominal class problems can be tackled. Often dramatically improves performance, but sometimes overfits.

- Stacking (weka.classifiers.meta.StackingC)

  Implements StackingC (more efficient version of stacking).

- Ensemble Selection (weka.classifiers.meta.EnsembleSelection)

  Combines several classifiers using the ensemble selection method (see Section 2.8.3.6).

- Random Forest (weka.classifiers.trees.RandomForest)

  Class for constructing a forest of random trees.

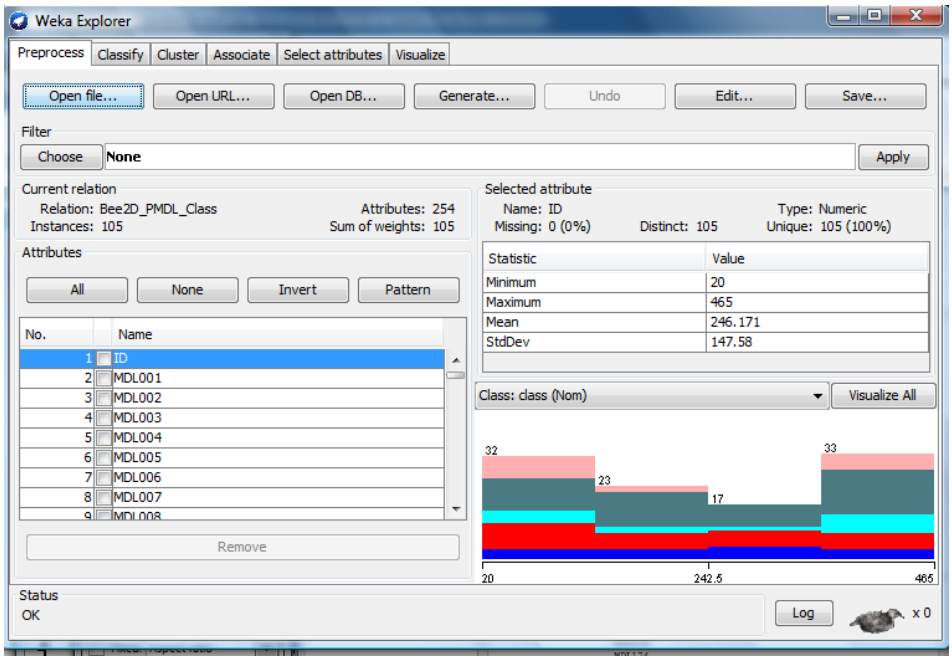

Figure 3.3: Weka Data Set Preparation Screen

Figure 3.3 is a screen shot from Weka. The screen is the first screen used to select a data set and display the information of a data set such as number of attributes, number of instances and all the attributes with the values.

After loading the data set as in Figure 3.3, the feature selection algorithms can be applied to find the most significant attributes of

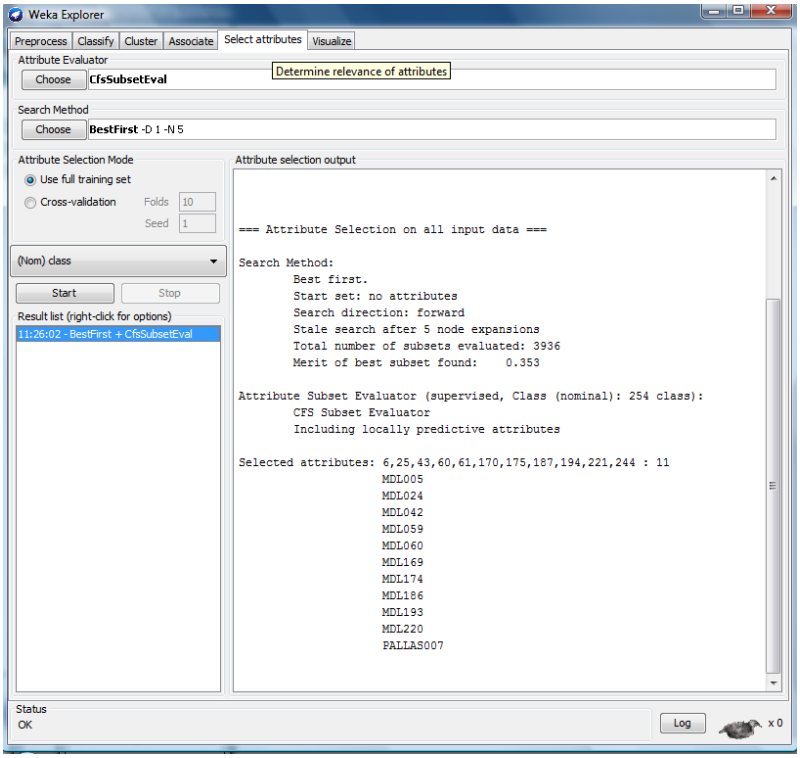

Figure 3.4: Weka Attribute Evaluator Screen

the data set. There are numbers of algorithms can be selected. The screen is depicted in Figure 3.4.

The screen (Figure 3.5) is the modelling of the data that had been selected. Here a large number of updated machine learning algorithms can be selected. The attribute selection mode such as 10-fold cross validation and classifiers can be selected. The results of the generated models will appear in the classifier output box. The main results are performance measures and confusion matrix.

3.6 Summary

Generated predictive models are valuable assets to the user because they can be used to predict new problems based on current training

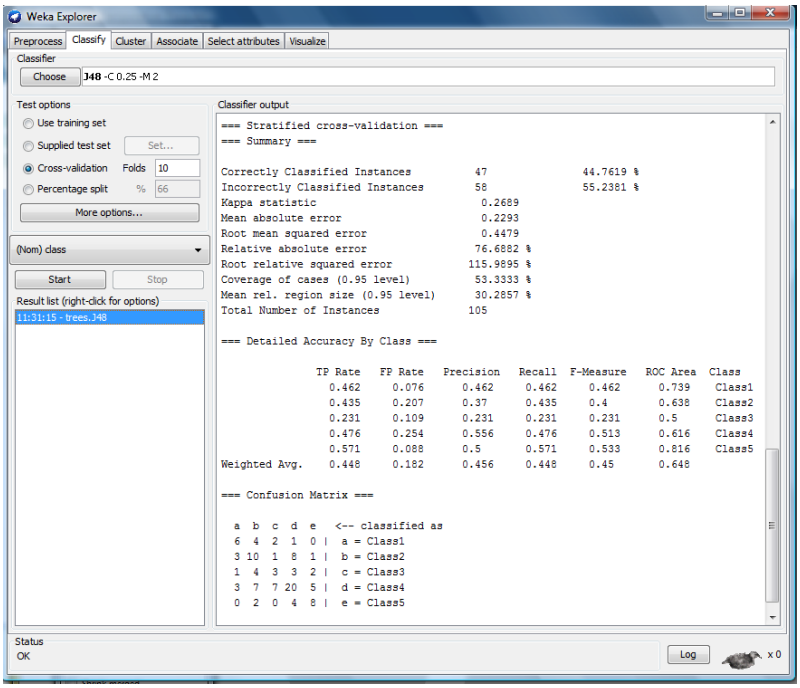

Figure 3.5: Weka Classifier and Prediction Results Screen

data set. They are most valuable if the models offer the possibility to be analysed for knowledge and are manageable. With proper representation using a PTML representation, these models can be processed further by selecting, comparing and combination of the classifiers which will be discussed in the following chapters. This thesis is moving toward data and model governance where there are huge amounts of data and large numbers of models to be stored and maintained. The proposed representation, selection, comparison and combination of models will be discussed in the following chapters.

Chapter 4

Classifiers Representation

4.1 Introduction

Currently there are emerging solutions for data governance, but there is no consistent approach to a sustainable data and related model governance framework. The lack of an agreed representation across data mining tools for models makes difficult to analyse and interpret them. Extensible Markup Language (XML) structure has the potential to describe this information about data and associated models.

In this chapter, Predictive Toxicology Markup Language (PTML), an original structure for representing predictive toxicology model is proposed. It offers a representation scheme for predictive toxicology data and models generated by data mining tools. The representation also offers possibilities to compare the models by calculating their similarity using the proposed models comparison technique that will be introduced and discussed in Chapter 5. The objective of PTML is to store the main information that needs to be captured so that

further processes of comparison and combination of classifiers can be implemented. The contribution of this chapter is a new knowledge representation for predictive toxicology data and models (Predictive Toxicology Markup Language).

The lack of a standard representation means that attributes are not consistent. The success of the model comparison depends on the standard naming of attributes. A pool of classifiers was generated from a data set produced using software called Dragon. Other software may produce different attributes for the same data set. Different names of attributes with the same meaning can be mapped using ontology. The ontology process is not considered in this research.

4.2 Model Structures for PTML

The main objective of standard representation for predictive models is to make it easier to process and understand. In this thesis, the representation is based on XML. All the predictive models (PTML) can be stored in a database. Thus, the basic process of Database Management System (DBMS) can be manipulated against the models. The DBMS processes are query, searching, add, delete and update.

This section will briefly explain the proposed model representation for Data and Model Governance in Predictive Toxicology. The model representation proposed is called Predictive Toxicology Markup Language (PTML) it represents predictive models generated from different data mining tools. The representation is part of the whole

research for Data and Model Governance. From the representation, further processing can be done to the predictive models in order to find the most relevant models from a collection of models.

The proposed Predictive Toxicology Markup Language is an extension to the model proposed by Neagu, Craciun, Chaudhry & Price (2007) to provide solutions in data and model representation for toxicology data. PTML represents data mining models in a standard format and can be simply manipulated for searching and comparing. It also describes predictive toxicology data and the associated model generated by data mining processes.

PMML (Predictive Model Markup Language) is a standard XML-based language used to represent predictive models and allow sharing of models to compliant applications. PMML is still still under development because it is attempting to represent the complete information of data mining processes. That is why there are other parties building on PMML models such as representations proposed by Chaves et al. (2006) and Gorea (2008).

Chaves et al. (2006) developed a PMML compliant scoring engine called Augustus. Augustus used components from PMML and added other new components such as a data management component, utilities for processing PMML files and run time support. Gorea (2008) proposed PMQL (Predictive Modelling Query Language), a specialized query language for interacting with PMML documents. It is embedded within DeVisa framework which provides functions such as scoring, model comparison, model composition, model searching, statistics and administration through a web service interface for the

PMML. Both agents rely on the PMML to have a pool of models and cannot be used with other models.

The difference with the proposed PTML is that it can be a bridge to models that are represented differently. The difference with other representations are as follows:

- Simpler representation but yet able to hold predictive models information,
- Integrative approach for data and model representation, and
- Process and manage the models in relation to the available data.

The next section will discuss further the functions and elements of $PTML$.

Weka and Java are the two main tools used in this thesis in generating predictive models and converting them to PTML representation. Java functions that was developed within this research are called $generateWekaModel$ and used to retrieve output from Weka while $PTMLTranslation$ is used to translate the output from Weka to PTML structure.

The $generateWekaModel$ function will invoke Weka and generate predictive models with diverse feature selection algorithms and classifiers as discussed in Section 3.5. The hundreds of models generated for all Demetra data sets were stored in Weka's standard representation (*.model*).

The Weka's models generated (*.model*) then will be translated and represented in PTML format using $PTMLTranslation$ which was de-

veloped as part of this thesis research. The conversion to PTML is based on the proposed representation as discussed in this section. The PTML models that have XML tagging can be used later for model comparison and combination.

PTML was proposed to make predictive models easier to analyse and interpret. This section gives an overview and explanation of the components of the PTML (Predictive Toxicology Markup Language) model. The PTML structure currently consists of 6 elements: Model Description, Model Parameter, Model Attributes, Model Performance, Class Attribute and Confusion Matrix (See Figure 4.1). Document Type Definition (DTD) for PTML can be found in Appendix A.1. The DTD is an XML schema that allow different format of predictive models to be imported using PTML standard.

1. Model Description

   This section describes the general information of the predictive model. Attributes such as the date when the model was generated, descriptions of model and file name for Weka model type can be found in this section. (see Figure 4.2).

2. Model Parameter

   Another important part regarding the generation of a predictive model is the parameter settings. Information such as type of classifier, number of folds and seed are used to distinguish between models. This information is useful for describing or regenerating predictive models. (see Figure 4.3).

```
<dataMiningModel>
    <modelDescription>
    :
    </modelDescription>
    <modelAttributes>
    :
    </modelAttributes>
    <modelParameter>
    :
    </modelParameter>
    <modelPerformance>
    :
    </modelPerformance>
    <classAttribute>
    :
    </classAttribute>
    <ConfusionMatrix>
    :
    </ConfusionMatrix>
</dataMiningModel>
```

Figure 4.1: The PTML Document Structure

```
<modelDescription>
    <Name>DM</Name>
    <Date>25-12-2008</Date>
    <Version>Ver1.1</Version>
    <Author>Mokhairi</Author>
    <Description>Testing Autogenerated
                 Model From Weka
    </Description>
    <wekaModel>wekaModel20.model</wekaModel>
</modelDescription>
```

Figure 4.2: Model Description

```
<modelParameter>
    <option Classifier=
    "weka.classifiers.trees.J48">
    </option>
    <option TrainingType>
    10fold-cross-validation</option>
    <option Fold="10"></option>
    <option Seed="1"></option>
</modelParameter>
```

Figure 4.3: The PTML Model Parameter

3. Model Attributes

   This section describes the data set and attributes used for the generation of the predictive model. The information includes file name of data set the model is generated from, number of instances and number of attributes. Without these, the predictive model cannot be generated or used to make predictions (see Figure 4.4). The representation emphasizes the inclusion of the data source into the model representation, thus further operations to compare models by the source data can be defined for model comparison.

4. Model Performance

   The element of model performance is a wrapper around various elements that can illustrate the overall quality of the model. This element holds related results generated from a specific

```
<modelAttributes>
    <DataSet>APC_Recon-(C)Mallard_Duck-
             Raw_Data-Training.arff
    </DataSet>
    <FeatureSelectionAlgorithm>
        CfsSubsetEval
    </FeatureSelectionAlgorithm>
    <FeatureSearchMethod>
        BestFirst
    <FeatureSearchMethod>
    <TotalNumberInstances>
        24.0
    </TotalNumberInstances>
    <NumberOfAttributes>
        184
    </NumberOfAttributes>
    <NumberOfAttributesSelected>
        7
    </NumberOfAttributesSelected>
    <Features>
        <Name>Del(Rho)NA5</Name>
        <Type>Numeric</Type>
    </Features>
    :
</modelAttributes>
```

Figure 4.4: Model Attributes

data set although it is possible to regenerate and recalculate the model. Statistical performances for the model generated are shown in this section such as correctly classified instances, mean absolute error and root mean squared error. From the performance, conclusions can be made about the model's quality. (see Figure 4.5).

```
<modelPerformance>
    <modelType>
        Classification
    <modelType>
    <CorrectlyClassifiedInstances>20.0
    </CorrectlyClassifiedInstances>
    <IncorrectlyClassifiedInstances>4.0
    </IncorrectlyClassifiedInstances>
    <Accuracy>83.33</Accuracy>
    <Kappa>0.71</Kappa>
    <MeanAbsoluteError>0.15
    </MeanAbsoluteError>
    <RootMeanSquaredError>0.33
    </RootMeanSquaredError>
    <RelativeAbsoluteError>40.77
    </RelativeAbsoluteError>
    <RootRelativeSquaredError>76.59
    </RootRelativeSquaredError>
</modelPerformance>
```

Figure 4.5: Model Performance

5. Class Attribute

   Further performance for each class attribute is stated in this section. The information included for each class are true pos-

itive rate, false positive rate, precision, recall and receiver operating characteristic (ROC) area (see Figure 4.6). This performance is based on newest test set.

```
<classAttribute>
    <Name>contact-lenses</Name>
    <Class>soft</Class>
        <Details>
            <TPRate>1.0</TPRate>
            <FPRate>0.053</FPRate>
            <Precision>0.833</Precision>
            <Recall>1.0</Recall>
            <FMeasure>0.909</FMeasure>
            <ROCArea>0.947</ROCArea>
        </Details>
    <Class>hard</Class>
        :
        :
</classAttribute>
```

Figure 4.6: Class Attribute

6. Confusion Matrix

   The confusion matrix is the most important element in generating classification models. It can give an overview of correct and incorrect classifications to the class attribute. (see Figure 4.7).

```
<ConfusionMatrix>
    <Array> Class1  Class2          </Array>
    <Array> 5       0       Class1 </Array>
    <Array> 0       3       Class2 </Array>
</ConfusionMatrix>
```

Figure 4.7: Confusion Matrix

4.3 Generated Predictive Models

The method of generating a collection of predictive models was described in Chapter 3. The automatic generation of predictive models is well addressed in the literature mainly in the work related to Hybrid Intelligent Systems (Neagu et al. 2005). One of the main motivations is the tuning of generated models and adapting them to further problems is not an easy task.

Caruana et al. (2004) addressed model generation for use in ensembles of models. They generated diverse sets of models by using seven different algorithms. About 2000 models were trained using these algorithms and applied to test sets. All algorithms used had different parameter settings. The algorithms they used were:

- Support Vector Machines (SVMs)
- Artificial Neural Nets (ANNs)
- Memory based Learning algorithms: k-NN
- Decision Trees (DT)
- Bagged Decision Trees (BAG-DT)

- Boosted Decision Trees (BST-DT)
- Boosted Stumps (BST-STMP).

For this research, collections of models were generated by using different algorithms implemented in Weka such as:

- K-Nearest neighbors classifier (weka.classifiers.lazy.IBk)
- Decision trees (weka.classifiers.trees.J48)
- Numerical prediction (weka.classifiers.rules.JRip)
- Naive Bayes (weka.classifiers.bayes.NaiveBayesUpdateable)
- Multilayer Perceptron (weka.classifiers.functions.MultilayerPerceptron)
- Bagging (weka.classifiers.meta.Bagging)
- Boosting (weka.classifiers.meta.AdaBoostM1)
- Stacking (weka.classifiers.meta.StackingC)
- Ensemble Selection (weka.classifiers.meta.EnsembleSelection)
- Random Forest (weka.classifiers.trees.RandomForest)

The description of each classification algorithm was discussed in Chapter 3.

Figure 4.8 shows the generated PTML models in XML format. The models were a collection of models that can be accessed and manipulated for prediction. The models were linked to their training data sets as shown in Figure 4.9. The comparison methods will access

and retrieve those PTML files and select the relevant models to be used for prediction.

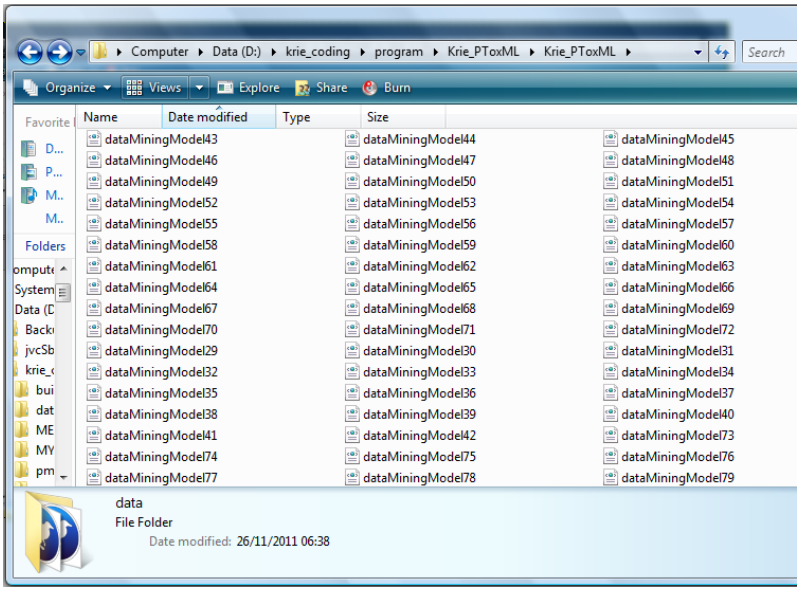

Figure 4.8: Example of the PTML Models Collection.

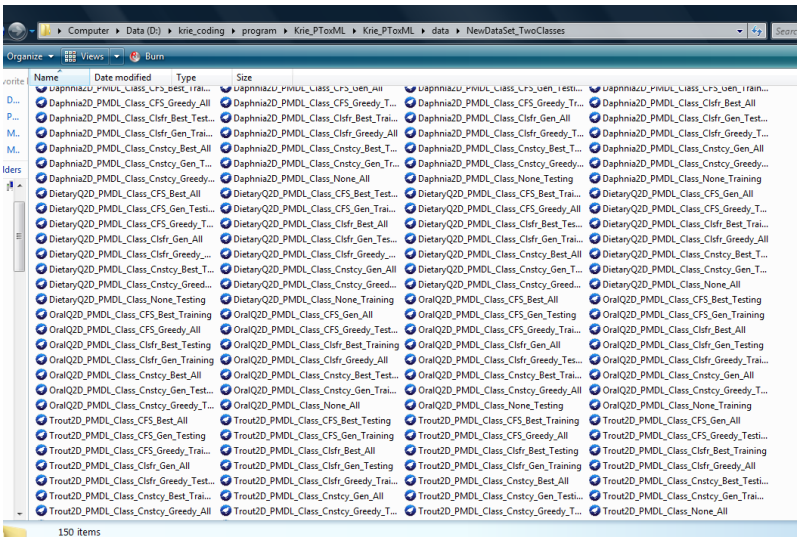

Figure 4.9: Example of the Collection of Training Data Sets Linked to PTML Models

4.4 Retrieving PTML Models

The generated models as shown in Figure 4.8 can be retrieved using a suitable technique. As discussed earlier in Section 2.3.2, there are a few technologies that can be used to retrieve the XML documents. In this research, all generated PTML models were retrieved using XML parser. The XML parser is a class built for Java and is suitable for a large number of PTML models and their structure. To further speed up the processing of the models, all the retrieved PTML documents were stored in a structured database (MS Access).

The relevant PTML models from the collection are retrieved using the proposed techniques that will be discussed in the next chapter. The model will be compared based on the element of predictive model (input, property and output). The steps of retrieval process are as follows:

1. Read new data set (attributes, instances).
2. Compare the attributes and instances of new data set with the models from the pool using methods proposed in Chapter 5.
3. Sort the models with the highest similarity.
4. Return the relevant models.

The method proposed for the comparison of the new data set uses Data set Similarity Coefficient (DSC). It uses a measure of the similarity of two-dimensional data sets to generate predictive models. The method based on DSC would measure the overall similarity of data sets between predictive models. Hence, the relevant models

related to the new problem will be retrieved from the collection of models.

4.5 Limitations

In this chapter, the PTML representation proposed focused on classification models with three element of Input (data set properties), Function (classifier properties) and Output (Confusion Matrix). The thesis may be developed by extending the representation to apply regression model in the future. In addition, the representation may be enhanced by including other elements and properties of predictive model such as quality factors.

4.6 Summary

In this chapter, an original representation of predictive toxicology models structure is presented. An implementation of the structure using real toxicity data has been generated and represented in the PTML format. The models were stored in the repository for easier sharing that allows faster access and simpler formalised structured format.

The representation of models also offers the possibility of automated searching and retrieval of classifiers based on some specific criteria. The criteria will be based on the comparison between the collection and the problem. The comparison technique will be proposed in the next chapter.

In the next chapter, the collection of models in the repository

is used throughout the thesis, with the aim of comparing relevant classifiers and use in an ensemble to make predictions.

Chapter 5

Proposed Method for Classifiers Comparison

5.1 Introduction

Generating predictive models by applying machine learning and model ensembles techniques has become an easy task when facilitated by development of more user-friendly data mining tools. However, such progress raises issues related to model management: once developed many classifiers for example become accessible in collections of models. Choosing the relevant model from the collection may be a faster task: calculating the similarity of predictive models is the key to rank them, which may improve model selection or combination.

Furthermore, calculating the similarity of predictive models helps to characterize the model diversity and to identify relevant models from a collection of models. The relevant models are considered based on their performance which is calculated using their confu-

sion matrix.

This chapter will introduce a methodology to measure the similarity of classifiers by comparing their data sets, functions and confusion matrices. The results show that the methodology proposed performs well in measuring model similarity from a collection of classifiers.

The contributions in this chapter are:

- A technique to compare the similarity of classifiers.
  - A technique to compare data sets (training set) (Data set Similarity Coefficient - DSC)
  - A technique to compare the similarity of functions used to generate the predictive models.
  - A technique to compare the similarity of confusion matrices.
  - A technique to compare the similarity of multi class confusion matrices to solve binary class problem.

The rest of the chapter is structured as follows: Section 5.3 presents the concept of predictive toxicology models comparison and motivation to compare classifiers with a composite similarity metric. Section 5.4.1 defines the technique of comparison of (toxicology) input data sets. The similarity measure of Predictive Model Functions is proposed in Section 5.4.2. Section 5.4.3 introduces and exemplifies the technique to compare the output of predictive models represented by their confusion matrices. A composite measure of the

similarity of predictive models is proposed in Section 5.5. Experiments and results are discussed in Section 5.6.

5.2 Classification Models

For each classification model, the number of classes will differentiate between binary and multi class models. Classification models can be grouped into two:

1. Binary Classification Models

   Binary classification models have only two classes which are first class and second class which normally represent true and false classes. The important performance measure is Acc. Other critical performance measures related to binary models are TNR, TPR, FNR and FPR. As mentioned earlier, in the toxicology domain the most critical issue in prediction is to find whether the chemical compound is toxic or non toxic. Thus, the prediction should have high confidence in FNR.

2. Multi class Classification Models

   In multi class classification models, the number of classes will be more than two. The important performance measure will be Acc as described in Section 2.5.2. For the same target, which is to have models with low FNR, the thesis proposed that the confusion matrix of multi class classification models be re-grouped into binary class in order to solve the binary class problem. This is because, in predictive toxicology, the chemical compound predicted using multi class classifiers can be categorised

by toxic level such as class1, class2, class3 and class$n$. All the classes can be grouped into toxic or non toxic classes.

Through this chapter, experiments were conducted on both binary classification and multi class classification models to solve binary problem. The objective is to find the relevant model from the collection of models that will be chosen to make a new prediction.

5.2.1 Classifier Elements

The proposed definition of predictive model structure that consists of Input, Function and Output can be found in Chapter 4. The Input consists of data collections used by machine learning algorithms to get the prediction output (see Figure 5.1). The Output is obtained by using the model confusion matrix, in the case of classification models. Function represents the machine learning algorithm properties used to generate predictive model. Information such as classifiers, feature selection algorithms, number of folds and seed are used to distinguish between models. This information is useful for describing or regenerating predictive models. The performance of the classification models is related to correctly classified instances. Such information can be found from the model confusion matrix which is useful for classifier performance evaluation.

In this thesis, a methodology was proposed to measure the similarity of classifiers as predictive models by comparing their input data sets, functions and confusion matrices. In this work, to assure the compatibility of a model selection and combination, the models were compared with the models built on similar input data sets.

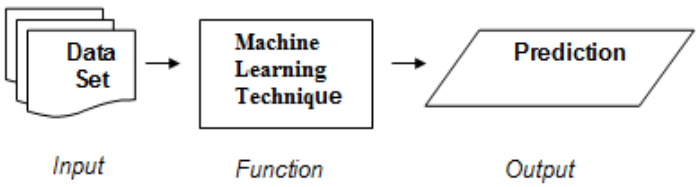

Figure 5.1: Predictive Modelling Framework

The confusion matrix provides information on the performance of each class for a trained classifier. However, in order to measure the similarity of predictive models as a whole, there is a need to measure the similarity of predictive model elements (Input, Function and Output) independently.

For the first structure element (Input) of the predictive model, the Data set Similarity Coefficient (DSC), a measure of the similarity of two-dimensional data sets used to generate predictive models, was proposed. The method based on DSC would measure the overall similarity of data sets between predictive models.

Similarity of Predictive Model Function ($SimF$) method was also proposed to measure similarity of algorithms applied in generating the predictive models.

For the last structure element (Output), the Similarity of Output ($SimO$) method to measure the similarity of confusion matrices between compared predictive models was also proposed. Performance measures such as Accuracy or False Negative Rate can be derived from the confusion matrix, which can also be used to derive further metrics.

Thus, to calculate the Similarity of Model (Sim), similarities of

each predictive structure (DSC, $SimF$ and $SimO$) were combined together. To give more flexibility to the users to calculate the similarity of predictive models, the Sim method allows the user to select which structure is important in this composite metric. For the performance measures, the study was focused on the importance of False Negative Rate (FNR) for predictive toxicology models, which is an important metric for the application domain: low value of the FNR means the predictive model is able to predict the toxicity of chemical compounds in a safer way.

5.3 Classifiers Comparison

Predictive models comparison helps in finding how similar models are. However, relying on only standard performance indicators such as accuracy may not give much clue on the overall or specific quality of a predictive model. Sometimes, the accuracy might be biased for a certain class and this may not provide a good indication of the overall performance of the predictive model (Khoussainov et al. 2005). In this case, the accuracy is not necessarily the only measurement for predictive models, whereas the confusion matrix is still the most valuable source of performance indicators from classifiers to be analyzed.

The motivation of this thesis is given by the need to analyse the multi class classifier models for selected classes. In toxicology, users are mostly interested in the toxic class being predicted correctly. Using the confusion matrix as the information source of classifiers

performance, users can derive more useful measurements related to their objective. The classifiers can either be binary class or multi class models.

The technique proposed for comparing classifiers can be used for both binary class and multi class classifiers. But the solution requires much effort in converting data sets to new binary class sets and retraining the models with the new data sets. Since there maybe thousands of models in a collection of models, to be practical, the proposed technique to transform the multi class confusion matrices into binary confusion matrices is done without updating the data sets and re-generating the models. This means that the Acc, FNR and FPR can be calculated using current confusion matrices for multi class predictive models. This will confirm that the original structures and information the predictive models learned remain unchanged. It is done by combining the multi class data set into a new data set with only binary classes of toxic and non-toxic output and re-generating new predictive models related to the new data sets.

Comparison of predictive models is different to other similarity domains such as sequence similarity in bio-informatics or information retrieval. Todeschini et al. (2004) used variable cross-correlation matrix to find the relationship of features and reduce the similar features in order to find simpler models. They modified the Hamming distance technique to calculate the distances between predictive toxicology models.

In real cases, data sets are constantly updated. The changes in

data sets will make previously generated predictive models obsolete if not updated to the current content. This situation will have to be considered when comparing predictive models to calculate their similarity. The changes of instances could play a crucial role when finding similar models because it would affect the performances of learning models. Consequently, the big challenge is to measure model similarity in a collection of models. There are four cases of data set update to be considered:

1. Different sets of records (instances) and similar variables (descriptors).
2. Different sets of records (instances) and different variables (descriptors).
3. Similar sets of records (instances) and similar variables (descriptors).
4. Similar sets of records (instances) and different variables (descriptors).

From the current literature, there are no comparisons of classifiers that calculate their similarity by incorporating the Input, Function and Output values in order to rank the classifiers from a collection of models. Many studies have been done by comparing the confusion matrices properties in ensembles of models such as those by Prasanna et al. (2007) and Freitas et al. (2007). However, a more integrated approach to consider model development (training data and function) is still necessary to improve model management and reuse for related tasks.

5.4 Similarity of Predictive Models' Element

This section will describe the method for calculating the similarity for each predictive model element (Input, Function and Output).

5.4.1 Similarity of Toxicology Data Sets

This section introduces the technique to compare similarity of data sets from a collection of models. The models are generated using Weka based on the four cases introduced above. The aim is to analyse if similar models would predict similar results.

In this research, the data sets are composed of rows (chemical compounds) and columns (descriptors): the descriptors are calculated values to describe the chemical compound properties, whereas the outputs are toxicity values obtained from testing chemical compounds against in vivo or in vitro end points.

Simpler predictive models can be generated by following a feature selection process, which is applied to find the most relevant descriptors of the data set. For the evaluation of the similarity between the two sets, the Jaccard Similarity Coefficient (JSC) can be used. It is defined as the size of the intersection divided by the size of the union of the sets A and B:

$$JSC\left(A,B\right)=\frac{|A\bigcap B|}{|A\bigcup B|} \tag{5.1}$$

where, $A=\{a_i\}$ and $B=\{b_i\}$ containing $i=1..n$ tuples. In the data sets, there were two-dimensional sets (see Table 5.1 to Table 5.4 for examples), which cannot be measured using JSC. This study

addresses the cases where the values for the same descriptors and chemical compounds are the same (data quality check having been previously done) for all the data sets experimented on.

The new method proposed was Data set Similarity Coefficient (DSC) to measure the similarity of two-dimensional data sets used to generate predictive models. The DSC between Model a (Ma) and Model b (Mb) data sets is:

$$DSC_{(M_a,M_b)} = \frac{|C_{Ma} \bigcap C_{Mb} \,||\, R_{Ma} \bigcap R_{Mb}|}{|C_{Ma} \bigcup C_{Mb} \,||\, R_{Ma} \bigcup R_{Mb}|} \tag{5.2}$$

where:

C_{Ma} is the set of all descriptor names (columns) for the data set used in Model a,

C_{Mb} is the set of all descriptor names (columns) for the data set used in Model b,

R_{Ma} is the set of all chemical compound attributes (rows) for the data set used in M_a,

R_{Mb} is the set of all chemical compound attributes (rows) for the data set used in M_b.

Equation 5.2, DSC($_{Ma}$, $_{Mb}$) is a Data set Similarity Coefficient used to measure similarity of Input data set for Model a ($_{Ma}$) and Model b ($_{Mb}$).

To exemplify the use of the proposed DSC: consider four models with their particular data sets which are $DS1$, $DS2$, $DS3$ and $DS4$ (see Table 5.1 to Table 5.4). Data set $DS1$ is the main data set whereas the other data sets are subsets of $DS1$. There are seven descriptors

(ID, $D1$, $D2$, $D3$, $D4$, $D5$ and $Class$) in the data set.

Table 5.1: Example of Data Set DS1

ID	D1	D2	D3	D4	D5	Class
1	9.5	7.5	10	7.5	21.5	Yes
2	1.1	2	4.1	10	20	No
3	7	10	11	10.6	20.5	Yes
4	10	15	20	15	20	Yes
5	9	14	19	14	10	N0

Table 5.2: Example of Data Set DS2

ID	D1	D2	Class
1	9.5	7.5	Yes
2	1.1	2	No
3	7	10	Yes

Table 5.3: Example of Data Set DS3

ID	D2	D3	Class
3	10	11	Yes
4	15	20	Yes
5	14	19	No

Table 5.5 shows the similarity coefficient matrix of data sets $DS1$, $DS2$, $DS3$ and $DS4$ which were calculated by using DSC: $DS2$ seems to be most similar to $DS1$, followed by $DS3$ and $DS4$. For example, the similarity between $Modela$ (Ma) and $Modelb$ (Mb) that used data sets $DS1$ and $DS2$ is:

$$DSC_{(M_a,M_b)} = \frac{|2\,||\,3|}{|5\,||\,5|} = \frac{6}{25} = 0.24 \tag{5.3}$$

DSC may provide an effective measure in calculating the similarity of data sets used in predictive models. The data set similarity for

Table 5.4: Example of Data Set DS4

ID	D4	D5	Class
2	10	20	No
3	10.6	20.5	Yes
4	15	20	Yes

Table 5.5: Data Set Similarity Coefficient Matrix of Data Set DS1, DS2, DS3 and DS4

	DS1	DS2	DS3	DS4
DS1	1.00	0.24	0.24	0.24
DS2	0.24	1.00	0.06	0.00
DS3	0.24	0.06	1.00	0.00
DS4	0.24	0.00	0.00	1.00

two predictive models will give us an indication of what the predictive model may derive from similar data sets.

5.4.2 Similarity of Predictive Model Functions

This section introduces the similarity measure for the second element of the predictive model, the Function F. Using this method, the models generated with similar functions can be searched. The method proposed was to apply the Jaccard Similarity Coefficient (JSC) to calculate the similarity for F; it is defined as the size of the intersection divided by the size of the union of the set F_{Ma} and set F_{Mb}:

$$SimF_{(M_a,M_b)} = \frac{|F_{Ma} \bigcap F_{Mb}|}{|F_{Ma} \bigcup F_{Mb}|} \tag{5.4}$$

For consistency the method assumes all parameter names of predictive models come from the same representation such as Predic-

tive Toxicology Markup Language (PTML) or Predictive Model Markup Language (PMML). For example, given two models Model a (Ma) and Model b (Mb):

F_{Ma} = { "Decison Tree", "10-Folds, "classification" } and

F_{Mb} = { "NeuralNet", "10-Folds","classification" }.

The similarity of the two properties of learning functions is,

$$SimF_{(M_a,M_b)} = \frac{|2|}{|3|} = 0.67 \tag{5.5}$$

The result for $SimF$ (M_a, M_b) shows 67% intersection between the function sets of the two models.

5.4.3 Similarity of Confusion Matrices

In this section, a novel technique was proposed to compare predictive models performance based on their confusion matrices. The confusion matrix stresses the raw results of the classification generated by the classification algorithm. The result contains information on correct and incorrect classification determined by the machine learning algorithm to predict the output.

5.4.4 Similarity of Confusion Matrices for Binary Classifiers

5.4.4.1 Binary Class Confusion Matrices

Kohavi & Provost (1998) defined a confusion matrix that contains information about actual and predicted classifications by a classification model. Table 2.1 shows the confusion matrix for a two class

classifier. The performance measures for two class classifiers can be calculated from the confusion matrix as follows: sensitivity or TPR = TP/(TP+FN) is the rate of correct predictions for the positive output (e.g. Yes or True), FPR = FP/(FP+TN) is the rate of incorrect predictions for the positive output (e.g. No or False), specificity or TNR =TN/(TN+FP) is the rate of correct predictions for the negative output, and the rate of incorrect predictions for the negative output is FNR = FN / (TP+FN). Acc = (TP+TN) / (TP+FP+FN+TN) measures the correct predictions for all classes.

Table 5.6: Example of Confusion Matrix for Model M1.

		Actual	Actual
		Yes	No
Predicted	Yes	1	2
Predicted	No	3	4

Table 5.7: Example of Confusion Matrix for Model M2.

		Actual	Actual
		Yes	No
Predicted	Yes	3	4
Predicted	No	1	2

Table 5.8: Example of Confusion Matrix for Model M3.

		Actual	Actual
		Yes	No
Predicted	Yes	2	3
Predicted	No	2	3

For example, consider the confusion matrices for three models $M1$, $M2$ and $M3$ (shown in Table 5.6 to Table 5.8) with the same binary output classes from the same input data set. The following

confusion matrices resulted from the classifiers learning whether a chemical compound is toxic (class "Yes") or non-toxic (class "No"). All models show the same accuracy value (see Table 5.9) although the confusion matrices are different. The first classifier ($ModelM1$ in Table 5.6) successfully classifies 5 out of 10 cases. However, an alarming 3 chemical compounds will be given the all clear when they are actually toxic. Also, the 2 chemical compounds said to be toxic despite being non-toxic will be rejected although it is incorrect.

Table 5.7 shows the confusion matrix for the second classifier (Model M2). This time the model classifies well the class "Yes" but worse the class "No". Overall, it correctly classifies 50% of all cases and shows a very different confusion matrix compared to Model $M1$.

The confusion matrix of the model $M3$ shows a more balanced behavior than the first two classifiers, according to the TP, FP, FN and TN values. However its accuracy is the same as $M1$ and $M2$. This shows that comparing models may require a more detailed and composite performance measure, since accuracy alone does not define fully the predictive models performance.

5.4.4.2 Similarity of Confusion Matrix for Binary Classifiers

The confusion matrices help to evaluate classifiers in a more detailed way than just using the accuracy score and also can provide a tool to compare models' performance. Following is the method proposed to compare confusion matrices. Table 5.9 contains Acc, TPR, TNR, FNR and FPR values for models $M1$, $M2$ and $M3$, calculated from their confusion matrices: although accuracy is the same for

all models, it fails to describe differences in their performance. The other four performance indicators (TPR, TNR, FNR and FPR) help providing detailed performance for each class and are more realistic tools for comparing the performance of the predictive models.

Table 5.9: Performance Measures (Acc, TPR, TNR, FNR and FPR) for Models $M1$, $M2$ and $M3$.

Models	Acc	TPR	TNR	FNR	FPR
M1	0.50	0.25	0.67	0.75	0.33
M2	0.50	0.75	0.33	0.25	0.67
M3	0.50	0.50	0.50	0.50	0.50

The Euclidean Distance was used to calculate the difference between performances of the two models. In this example TPR and TNR were chosen to measure the distance between the models performances. For the performance measures in Table 5.9, the notations $k1...kn$ were used. In this case $k1$ is TPR, $k2$ is TNR, where n equals 2. The following steps illustrate the calculation of the distance between the confusion matrices of two predictive models.

Step 1: Save the selected performance measure/s in a 1-dimension (vector) format.

The selected model's performance measures were saved into two rows vectors. The vectors of performance measures for $M1$ and $M2$ where $k1$ is TPR and $k2$ is TNR are: V_{Ma} = (0.25, 0.67) and V_{Mb} = (0.75, 0.33). From the vectors V_{Ma} and V_{Mb}, the distance between them can be calculated by using the distance technique.

Step 2: Calculate the distance between the vectors.

The distance is calculated using the Euclidean Distance:

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \tag{5.6}$$

The distance O (Output) between Model A (Ma) and Model B (Mb) is the average of distances between the confusion matrix elements:

$$DisO_{(Ma,Mb)} = \left(\frac{\sqrt{\sum_{k=1}^{n} (V_{Mak} - V_{Mbk})^2}}{n} \right) \tag{5.7}$$

Similarity and distance measures complement each other. In this case, the similarity of output O ($SimO$) between two models will be:

$$SimO_{(Ma,Mb)} = 1 - \left(\frac{\sqrt{\sum_{k=1}^{n} (V_{Mak} - V_{Mbk})^2}}{n} \right) \tag{5.8}$$

where:

k is the index of performance measures selected, n equals to number of k, $V(Ma)$ is the vector for Model A (Ma), and $V(Mb)$ is the vector for Model B (Ma).

The value for $SimO(Ma, Mb)$ in the example above is 0.70. Table 5.10 contains the values for $SimO(Ma, Mb)$ related to the similarity of the three classifiers using TPR and TNR. The models might have learned from identical data sets but were generated using different classification algorithms.

Table 5.10: Similarity Matrix for Model $M1$, $M2$ and $M3$

	M1	M2	M3
M1	1.00	0.70	0.85
M2	0.70	1.00	0.85
M3	0.85	0.85	1.00

5.4.5 Similarity of Confusion Matrices for Multi Class Classifiers

5.4.5.1 Multi Class Confusion Matrices

Sometimes, multi class classification problems can still be solved with binary classifiers. Such a solution may divide the original multi class data set into two class subsets, learning a different binary model for each subset. These techniques are known as binarisation strategies (Hashemi et al. 2009, Liu & Zheng 2005). Galar et al. (2011) reported that there are three main approaches: One-vs-All (OVA), One vs-One (OVO), and Error Correcting Output Codes (ECOC).

All of these techniques decompose a complex multi class to a simpler binary class problem. Hence this strategy may improve the performance because the classifiers have an easier task to distinguish between only two classes rather than many classes.

The experiment in this chapter focused on multi class classifiers for toxicology applications. The performance measures of confusion matrices of multi class classifiers are regrouped into a binary classification problem. Such an approach may result in selecting multi class classifiers with lower False Negative Rate (FNR) for example. Consequently, the methodology for model comparison based on the

similarity of confusion matrices provides a working method to select models from a collection of classifiers.

5.4.5.2 Reducing Multi Class to Binary Classification Problems

In this section, the aim was to investigate whether there are any differences in performance between binarisation strategies by regenerating new binary classifiers from multi class classifiers. It was done by calculating the performance measures using multi class classifiers confusion matrices without retraining new binary classifiers.

In the next section, the discussions will be on the performance measures related to binary classification classifiers and proposal of a methodology to reduce multi class problems to a binary version while calculating the performance measures of the multi class classifiers with a focus on lower False Negative Rate (FNR) for example, as required in toxicity prediction problems.

5.4.5.3 Performance Measures and Confusion Matrix for Multi Class Classifiers

The confusion matrix for a multi class classification problem is a generalization of the binary case. The properties and the performance measures derived from a multi class confusion matrix will be discussed below. Table 5.11 is an example of a multi class confusion matrix. For the first column (Class A) the intersection with the first row is the True Positive (TP) value for Class A. The sum of values from remaining cells of the column is the False Negative (FN) value for Class A. True positives for second and third columns are

the diagonal values of the confusion matrix.

Table 5.11: Confusion Matrix for a 3-Class Classifier.

		Actual		
		Class A	Class B	Class C
Predicted	Class A	$TP_{AA(1,1)}$	$e_{AB(1,2)}$	$e_{AC(1,3)}$
	Class B	$e_{BA(2,1)}$	$TP_{BB(2,2)}$	$e_{BC(2,3)}$
	Class C	$e_{CA(3,1)}$	$e_{CB(3,2)}$	$TP_{CC(3,3)}$

The classification accuracy of a multi class classifier is the ratio of the sum of the principal diagonal values to the total of values in the confusion matrix. If C indicates the confusion matrix, Prasanna et al. (2007) defined that the classification accuracy Acc is as follow:

$$Acc_C = \left(\frac{\sum_{i=1}^{N} C_{ii}}{\sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij}} \right) \tag{5.9}$$

where:

N is the number of classes,

i refers to the rows index and,

j refers to the columns index for the confusion matrix C.

The Error Rate (ER) for the classifiers is the complement of the accuracy: $ER = (1 - Acc)$.

Beside the Acc and the (ER), other performance measures can be derived and used to measure the performance of multi class classifiers. Moreover the performance measures of the two-class classification problem can be applied by regrouping the multi class confusion matrix into two-class classification measures.

In predictive toxicology applications, the interest is more on the false negative rate (FNR) measurement in the case where the model fails to correctly classify the instances to the appropriate classes. To

give more flexibility for such applications for multi class classifiers comparison, the thesis proposed that the positive (toxic) class and negative (non-toxic) class to be selected by regrouping them into a two class problem. Freitas et al. (2007) and Prasanna et al. (2007) found that this technique is also highly recommended in classifier ensembles where good combination of classes and models will make the binary prediction more accurate.

The performances measures for the positive (toxic) class in predictive multi class classifiers are described below. Consider the selected toxic classes are Class A (e.g. Very Toxic, column 1) and Class B (e.g. Toxic, column 2) in Table 5.11. The selected class indexes are stored into the one-row vector V. Thus V= (1, 2). The proposed TPR, FNR, FPR and TNR measures for the selected classes are as follow:

$$TPR_{SelectedMa} = \left(\frac{\sum_{x=1,j=V_x}^{C} \sum_{y=1,i=V_y}^{C} R_{ij}}{\sum_{x=1,j=V_x}^{C} \sum_{i=1}^{N} R_{ij}} \right) \tag{5.10}$$

$$FNR_{SelectedMa} = \left(\frac{\sum_{y=1,j=V_x}^{C} \sum_{y=1,i\neq V_y}^{N} R_{ij}}{\sum_{x=1,j=V_x}^{C} \sum_{i=1}^{N} R_{ij}} \right) \tag{5.11}$$

where:

N is the number of samples of all classes in the confusion matrix R,

C is the number of selected class samples for the confusion matrix R,

i is the row index in the confusion matrix R,

j is the column index in the confusion matrix R,

x and y are counters for columns and rows, and,

V is a vector of selected class indexes.

$$TNR_{SelectedMa} = \left(\frac{\sum_{y=1,j=V_x}^{C} \sum_{y=1,i\neq V_y}^{N} R_{ij}}{\sum_{x=1,j=V_x}^{C} \sum_{i=1}^{N} R_{ij}} \right) \tag{5.12}$$

$$FPR_{SelectedMa} = \left(\frac{\sum_{x=1,j=V_x}^{C} \sum_{y=1,i=V_y}^{C} R_{ij}}{\sum_{x=1,j=V_x}^{C} \sum_{i=1}^{N} R_{ij}} \right) \tag{5.13}$$

The performance measures for the non-toxic class, False Positive Rate (FPR) and True Negative Rate (TNR), can be derived by adapting Equation 5.12 and Equation 5.13. The vector V for non-toxic class is 3 because there is only one non-toxic class in column 3, thus, $V = (3)$ and $C = 1$.

Table 5.12: Confusion Matrix (MA) for Model A.

		Actual		
		Class A	Class B	Class C
Predicted	Class A	$10_{AA(1,1)}$	$21_{AB(1,2)}$	$33_{AC(1,3)}$
	Class B	$24_{BA(2,1)}$	$53_{BB(2,2)}$	$26_{BC(2,3)}$
	Class C	$17_{CA(3,1)}$	$18_{CB(3,2)}$	$19_{CC(3,3)}$

Consider that Model A produced a confusion matrix MA (see Table 5.12). Referring to the Equation 5.10 to 5.13, the next equation demonstrates how to calculate the TPR, FNR, FPR and TNR of toxic classes. For these examples, two classes were selected as toxic classes (Class A and Class B). The index for Class A is 1 and the index for Class B is 2. Thus the vector for $V = (1, 2)$. For example by using the values from Table 5.12, the performance measures TPR, FNR FPR and TNR are as follows:

$$TPR_{MA} = \frac{(10 + 24) + (21 + 53)}{(10 + 24 + 17) + (21 + 53 + 18)} = 0.76 \tag{5.14}$$

$$FNR_{MA} = \frac{(17) + (18)}{(10 + 24 + 17) + (21 + 53 + 18)} = 0.24 \tag{5.15}$$

$$TNR_{MA} = \frac{(19)}{(33 + 26 + 19)} = 0.13 \tag{5.16}$$

$$FPR_{MA} = \frac{(33) + (26)}{(33 + 26 + 19)} = 0.76 \tag{5.17}$$

From the results above, TPR and FNR complement each other in the confusion matrix. In the next section the methodology to measure the similarity between confusion matrices for multi class classifiers will be demonstrated.

5.4.5.4 Similarity of Confusion Matrices for Multi Class Classifiers

In this section, the same technique as Section 5.4.4.2 is proposed and applied to compare multi class classifiers' confusion matrices. For example, three predictive models generated by different classifiers using the same data set. The model M1 generates the confusion matrix MA (see Table 5.12), the model M2 generates the confusion matrix MB (see Table 5.13), and the model M3 generates confusion matrix MC (see Table 5.14).

Table 5.13: Confusion Matrix (MB) for Model B.

		Actual		
		Class A	Class B	Class C
Predicted	Class A	$24_{AA(1,1)}$	$18_{AB(1,2)}$	$33_{AC(1,3)}$
	Class B	$10_{BA(2,1)}$	$53_{BB(2,2)}$	$19_{BC(2,3)}$
	Class C	$17_{CA(3,1)}$	$21_{CB(3,2)}$	$26_{CC(3,3)}$

Table 5.14: Confusion Matrix (MC) for Model C.

		Actual		
		Class A	Class B	Class C
Predicted	Class A	$34_{AA(1,1)}$	$4_{AB(1,2)}$	$9_{AC(1,3)}$
	Class B	$10_{BA(2,1)}$	$80_{BB(2,2)}$	$10_{BC(2,3)}$
	Class C	$7_{CA(3,1)}$	$8_{CB(3,2)}$	$59_{CC(3,3)}$

Table 5.15 shows the performance measures TPR, FNR and Acc calculated using Equation 5.9, 5.11 and 5.10. The values of performance measures were calculated by grouping the selected toxic classes A and B. Thus, $V = (1, 2)$. From the results depicted in Table 5, model M3 is the best model compared to M1 and M2: TPR is the highest value and FNR is the lowest value for model M3.

Table 5.15: Performance Measures (TPR and FNR) for Models M1, M2 and M3).

Models	TPR	FNR	Acc
M1	0.76	0.25	0.37
M2	0.73	0.27	0.47
M3	0.90	0.10	0.78

For the similarity measurement, in this example FNR was chose to measure the distance between the models' performances. For the performance measures in Table 5.15, the notations $k1...kn$ were used. In this case $k1$ is FNR. The following steps illustrate the calculation of the distance between the confusion matrices of two predictive models:

Step 1: Save the selected performance measure/s in a 1-dimension (vector) format.

Save the selected performance measures into two rows vectors;

in this case the vectors for M1 (V_{MA}) and M2 (V_{MB}) have just 1 element: V_{MA} = (0.25) and V_{MB} = (0.27).

Step 2: Calculate the distance between the vectors.

The distance between the vectors V_{MA} and V_{MB} is calculated using the Euclidean Distance. The distance O (Output) between model A (MA) and model B (MB) is the average of distances between the confusion matrix elements. Similarity and distance measures are complementary. In this case, the similarity of output O ($SimO$) between two models will be:

$$SimO_{SelectedClass(Ma,Mb)} = 1 - \left(\frac{\sqrt{\sum_{k=1}^{n} \left(V_{Mak} - V_{Mbk}\right)^2}}{n} \right) \quad (5.18)$$

where: k is the order of performance measures selected, n equals to number of k, VMA is the index vector for model A (MA), and VMB is the index vector for model B (MB). The value for $SimO(MA, MB)$ in the example above is 0.98. Table 5.16 contains the values for $SimO(MA, MB)$ related to the similarity of the three classifiers using FNR. The result shows that the similarity of confusion matrices between models M1 and M2 is 0.98.

Table 5.16: Similarity Matrix for Models M1, M2 and M3.

Models	M1	M2	M3
M1	1.00	0.98	0.85
M2	0.98	1.00	0.83
M3	0.85	0.83	1.00

5.5 Similarity of Predictive Models

In the previous sections, the proposal was to evaluate the similarity of a predictive model based on the similarity of input (I) data sets, the performance measured by the Confusion Matrix as the output (O), and the similarity of the functions (F). To calculate the similarity between classifiers, Table 5.17 is the example with the similarity value for I (DSC), F ($SimF$) and O ($SimO$) for models M1, M2 and M3.

Table 5.17: Value of I, F and O for Model M1, M2 and M3.

Models	M1	M2	M3
M1	-	I=0.0,F=1.0,0=0.3	I=0.9,F=0.2,0=0.2
M2	I=0.0,F=1.0,0=0.3	-	I=0.0,F=0.9,0=0.7
M3	I=0.9,F=0.2,0=0.2	I=0.0,F=0.9,0=0.7	-

To find the similarity between models, the idea is to combine all similarity values for Input (I), Function (F) and Output (O) according to the definition of the predictive models' structure. To provide more flexibility in calculating the similarity of predictive models, each structure of a predictive model has its own weight α, β, γ. The proposed Similarity of Models:

$$Sim_{(Ma,Mb)} = \frac{\alpha \times I_{(Ma,Mb)} + \beta \times F_{(Ma,Mb)} + \gamma \times O_{(Ma,Mb)}}{\alpha + \beta + \gamma} \tag{5.19}$$

where:

$I(Ma, Mb)$ is the Data set Similarity Coefficient ($DSC(Ma, Mb)$) between Model A (Ma) and Model B (Mb),

$F(Ma, Mb)$ is the Similarity of Function ($SimF(Ma, Mb)$) between Model A (Ma) and Model B (Mb),

$O(Ma, Mb)$ is the Similarity of Confusion Matrix ($SimO(Ma, Mb)$) between Model A (Ma) and Model B (Mb), and

α, β, $\gamma \in [0, 1]$ are real numbers and their sum is between 0 to 3.

The values of α, β or γ can be handled depending on the priority given to the predictive model's elements. Consider the weight value for I (α=1), F (γ =0) and O (β=1). The similarity ($Sim(Ma, Mb)$) between models is shown in Table 5.18 where models M1 and M2 are less similar than models M2 and M3.

Table 5.18: Similarity Values of Models M1, M2, M3 Given I (α =1), F (γ =0) and O (β =1)

Models	M1	M2	M3
M1	1	0.15	0.55
M2	0.15	1	0.85
M3	0.55	0.85	1

5.6 The Implementation of Proposed Classifiers Comparison Method

This section will show the experiments and results for conducting the proposed classifier comparison method. The experiments were done on both binary class and multi class models. The predictive models were applied to various toxicology data sets as described in Chapter 3.

Table 5.19: The Mapping of the Old Classes to the New Binary Classes in Each Data Sets.

Data sets	Old Classes (Multi classes)	New Classes (Binary Classes)	Instances
Trout	Class1, Class2 Class3	Yes (Toxic) No (Non-toxic)	218 64
Oral Quail	Class1, Class2,Class3 Class4	Yes (Toxic) No (Non-toxic)	56 60
Daphnia	Class1, Class2 Class3,Class4	Yes (Toxic) No (Non-toxic)	187 77
Dietary Quail	Class1, Class2,Class3 Class4,Class5	Yes (Toxic) No (Non-toxic)	101 22
Bee	Class1, Class2,Class3,Class4 Class5	Yes (Toxic) No (Non-toxic)	76 29

5.6.1 The Study on the Binary Class Data Set

Every data set had originally more than two classes to predict the toxicology levels for every compound. The mapping multi class to binary class is a technique to solve general problem (see Table 5.19). The models were generated from a group of predictive toxicology data sets whereby each group of data set was run through data preparation and reductions processes. The data sets had been grouped into 3 segments which were raw data set (100%), training data set (70%) and testing data set (30%). The group of data sets were divided in the first 70% as training set and remaining 30% as test set order by chemical ID. The feature selection algorithms were applied to the data sets to find sets of attributes that are highly correlated with the target classes. Each data set was run using Weka with 10-fold cross validation and various classifiers as discussed in Chapter 3).

5.6.1.1 The Implementation of Similarity of Predictive Model (Sim) to All Demetra Data Sets.

The objective of this experiment is to calculate the proposed similarity of predictive models using ($Sim_{(Ma,Mb)}$). with the values of I (α =1), F (β =0) and O (γ =1); this means that the similarity was focused on the data sets and confusion matrices. False Negative Rate (FNR) was set in the ($Sim_{(Ma,Mb)}$) to justify the importance of it from the viewpoint of toxicology data sets, where the aim was to have a model with low FNR. This means that the models were chosen on the basis of minimum FNR. For example in Table 5.20, similar data sets are likely to predict similar FNR although using different machine learning algorithms, such as $Model1$ and $Model151$, and $Model4$ and $Model154$. In Table B.1 to Table B.4 (see Appendix B) similar results were obtained for other toxicity data sets (Daphnia, Dietary Quail, Oral Quail and Trout).

Table 5.20: Results of Model Similarity from Bee Data Set

Machine Learning	IBK						J48					
Feature Selection	None			CFS			None			CFS		
Split(%)	100	70	30	100	70	30	100	70	30	100	70	30
ModelID	1	2	3	4	5	6	151	152	153	154	155	156
1	1	0.84	0.62	0.46	0.48	0.44	0.97	0.84	0.64	0.45	0.46	0.49
2	0.84	1	0.48	0.43	0.47	0.43	0.80	0.99	0.50	0.44	0.44	0.48
3	0.62	0.48	1	0.45	0.45	0.41	0.59	0.47	0.98	0.42	0.43	0.46
4	0.46	0.45	0.43	1	0.83	0.63	0.49	0.46	0.45	0.99	0.84	0.62
5	0.48	0.47	0.45	0.83	1	0.46	0.49	0.48	0.47	0.82	0.98	0.49
6	0.44	0.43	0.41	0.63	0.46	1	0.47	0.44	0.43	0.64	0.48	0.95
151	0.97	0.80	0.59	0.49	0.49	0.47	1	0.81	0.61	0.48	0.49	0.47
152	0.84	0.99	0.47	0.46	0.48	0.44	0.81	1	0.49	0.45	0.45	0.49
153	0.64	0.50	0.98	0.45	0.47	0.43	0.61	0.49	1	0.44	0.44	0.48
154	0.45	0.44	0.42	0.99	0.82	0.64	0.48	0.45	0.44	1	0.84	0.61
155	0.46	0.44	0.43	0.84	0.98	0.48	0.49	0.45	0.44	0.84	1	0.46
156	0.49	0.48	0.46	0.62	0.49	0.95	0.47	0.49	0.48	0.61	0.46	1

5.6.1.2 The Implementation of Data Set Similarity Coefficient (DSC) to All Demetra Data Sets.

The objective of the this study is to find the similarity of data sets between five end points. The five Demetra data sets are Bee, Daphnia, Dietary Quail, Oral Quail and Trout. For this experiment, I (α =1), F (γ =0) and O (β =0). From the result (see Table 5.21), all data sets share over 50% similar descriptors and chemical compounds: the highest data set similarity is 63% between Daphnia and Trout, while Bee and Oral Quail have about 48% chemical compounds in common.

Table 5.21: Results of Similarity for All Data Sets

Data sets	Bee	Daphnia	Dietary Quail	Oral Quail	Trout
Bee	1.00	0.54	0.59	0.48	0.58
Daphnia	0.54	1.00	0.59	0.53	0.63
Dietary Quail	0.59	0.59	1.00	0.56	0.59
Oral Quail	0.48	0.53	0.56	1.00	0.50
Trout	0.58	0.63	0.59	0.50	1.00

5.6.1.3 The Comparative Study of Feature Selection Algorithms Applied to Demetra Data Sets.

This experiment was designed to show that the performance of models rely on the functions used to generate the predictive models. This experiment compares the results if the models using feature selection algorithms with different classifiers (see Table 5.22).

From Table 5.22, generally the accuracy of the models increased when a feature selection algorithm was used. This experiment used the Correlation-based Feature Selection (CFS) as the feature selec-

Table 5.22: Results of the Accuracy (Acc) and the False Negative Rate (FNR) for All Data Sets.

Data set	Bee		Daphina		Dietary Quail		Oral Quail		Trout	
Feature	None	CFS	None	CFS	None	CFS	None	CFS	None	CFS
Selection										
IBK										
Model	M1	M4	M31	M34	M61	M64	M91	M94	M121	M124
Acc	82.85	88.57	73.11	74.62	76.42	81.3	67.24	63.79	78.72	80.49
FNR	0.12	0.04	0.19	0.2	0.19	0.15	0.32	0.38	0.14	0.12
J48										
Model	M151	M154	M181	M184	M211	M214	M241	M244	M271	M274
Acc	86.67	89.52	74.62	78.41	74.8	82.11	67.24	73.28	74.11	81.92
FNR	0.06	0.02	0.19	0.12	0.2	0.13	0.36	0.3	0.17	0.07

tion algorithm. From the toxicology point of view, the interest is on the FNR, whether the models are able to minimise the error in predicting the toxic class. From the results, models with feature selection and using J48 classifier seem have the right combination in correctly predicting the toxicity class.

5.6.2 The Study on the Multi Class Data Sets

In this study the models were generated using multi class Demetra data sets. The predictive models were generated using different combinations of data sets, algorithms, and model parameters. The feature selection algorithms were applied to the original full data sets as discussed in Chapter 4. Each data set was generated using Weka with 10-fold cross validation and the same classifiers.

5.6.2.1 The Similarity of Confusion Matrices for Multi Class Classifiers.

This experiment was conducted to make sure the proposed method for binarisation will solve the binary problem as discussed in sec-

tion 5.4.5. The multi class confusion matrices were compared using Similarity of Confusion Matrices.

In Table 5.23, the confusion matrix for a decision tree was applied to the Bee data set with the 5 classes provided. Considering the fusion of Class1, Class2, Class3 as toxic classes and Class4, Class5 as non toxic classes. By applying the method proposed to calculate TPR and FRN in Section 5.4.5.3, the performance for a randomly chosen model M154c are shown in Table 5.24. From the results it shows that, Equation 5.9, 5.12 and 5.11 are correct.

Table 5.23: A Confusion Matrix Generated Using Multi Class Data Set With Feature Selection (CFS), 10-fold Cross Validation and Using Classifiers (weka.classifiers.trees.J48).

	Class1	Class2	Class3	Class4	Class5
Class1	7	4	2	3	0
Class2	4	7	4	8	2
Class3	0	2	1	4	0
Class4	2	10	4	23	4
Class5	0	0	2	4	8
Total Instances	13	23	13	42	14

Table 5.24: Performance Measures Calculated Based on the Confusion Matrix Using Table 5.23.

Performance Measures	Results
TPRate (All Classes) and Accuracy (See Eq. 5.9 and Eq. 5.12)	0.44
Error Rate (All Classes)	0.56
FNRate (selected toxic class; Class1,Class2,Class3,Class4) (See Eq. 5.11)	0.07

5.6.2.2 The Comparative Study on Error Rate and FNR for Demetra Data Sets.

This experiment was designed to compare the use of error rate for all classes vs. false negative rate for selected toxic classes in multi class classifiers. The data sets had been grouped into 3 segments which were raw data set (100%), training data set (70%) and testing data set (30%) order by chemical ID. For Table 5.25 (ER vs. FNR results were measured using the selected classes) it shows that models with similar ER Rate can exhibit a range of FNR values:

Table 5.25: Error Rate (ER) and FNR of Multi Class Classifiers Applied to the Demetra Data Sets.

Data sets	Toxic Classes (Lowest FNR)	All Classes (ER)	Toxic Classes (Highest FNR)	All Classes (ER)
Bee	0.04 - M304c	0.60 - M304c	0.12 - M1c	0.61 - M1c
Daphnia	0.07 - M334c	0.56 - M334c	0.20 - M31c	0.56 - M31c
Dietary Quail	0.19 - M364c	0.59 - M364c	0.25 - M211c	0.61 - M211c
Oral Quail	0.30 - M91c	0.60 - M91c	0.52 - M244c	0.61 - M244c
Trout	0.12 - M271c	0.51 - M271c	0.17 - M274c	0.52 - M274c

5.6.2.3 The Comparative Study of FNR for Multi Class Demetra Data Sets.

This study will investigate how the relationship of the numbers of toxic classes will affect the performance of the classifier. In this experiment, toxic classes were mapped into two categories: binary class (Toxic and Non-toxic) and multi class (class A, class B .. class N). More than 500 models were selected from the collection based on their lowest FNR for each data sets and classifiers. The models shown in Table 5.26 were IBK, J48 and JRip classifiers that applied feature selection algorithms (CFS) or without feature selection algo-

rithm to compare the FNR between them. They were chosen having lowest FNR and the missing models in the collection had higher FNR. From the results shown in Table 5.26 it can be concluded that:

- Data sets with feature selection algorithms (such as CFS) applied are better in FNR performance measurement compared to data sets with no feature selection. Example of such models are M4a and M1a.
- The classifiers performance are highest in Bee data set and lowest in Oral Quail data set.
- Some performance (FNR) of models with selected class for more than 1 toxic class (e.g. M4c) is poor compared to binary model with only 1 toxic class (e.g. M4a), but in contrast some of the multi class classifiers are better than binary classifiers (e.g. M34c vs. M34a and M271c vs. M271a).
- On average, models that applied binarisation strategies (models named ended with 'a') are better than multi class classifiers that apply calculation of FNR to their confusion matrices (models named ending in 'c'). This proved that multi class classifiers for Daphnia data sets such as M334c are better than binary classifiers (e.g. M331a). For Oral Quail data set, both binary and multi class had the same performance (0.30) of FNR (e.g. M91c vs. M244a).

From the results shown in Table 5.26, if the objective is to discriminate between two binary classes, in this case Toxic and Non-

toxic, then the classifiers with binary class format have better performance compared to multi class classifiers. For some models, re-grouping classes in a single toxic class may increase the accuracy as compared to re-generating binary class classifiers.

Table 5.26: Results of FNR for All Data Sets with Feature Selection Algorithms (CFS) and Without CFS Generated (None) Using Classifiers (IBK, J48 and JRip).

Algorithms	IBK	J48	JRip
Data sets	FNR - ModelID	FNR - ModelID	FNR - ModelID
Bee	0.12 - M1a	0.06 - M151a	0.06 - M301a
(None)	0.12 - M1c	0.09 - M151c	0.04 - M301c
Bee	0.04 - M4a	0.02 - M154a	0.04 - M304a
(CFS)	0.11 - M4c	0.07 - M154c	0.04 - M304c
Daphnia	0.19 - M31a	0.19 - M181a	0.10 - M331a
(None)	0.20 - M31c	0.20 - M181c	0.11 - M331c
Daphnia	0.20 - M34a	0.12 - M184a	0.12 - M334a
(CFS)	0.16 - M34c	0.14 - M184c	0.07 - M334c
Dietary Quail	0.19 - M61a	0.20 - M211a	0.23 - M361a
(None)	0.19 - M61c	0.25 - M211c	0.24 - M361c
Dietary Quail	0.15 - M64a	0.13 - M214a	0.20 - M364a
(CFS)	0.19 - M64c	0.15 - M214c	0.19 - M364c
Oral Quail	0.32 - M91a	0.36 - M241a	0.54 - M391a
(None)	0.30 - M91c	0.34 - M241c	0.62 - M391c
Oral Quail	0.37 - M94a	0.30 - M244a	0.47 - M394a
(CFS)	0.36 - M94c	0.52 - M244c	0.61 - M394c
Trout	0.14 - M121a	0.17 - M271a	0.10 - M421a
(None)	0.16 - M121c	0.12 - M271c	0.09 - M421c
Trout	0.12 - M124a	0.07 - M274a	0.05 - M424a
(CFS)	0.14 - M124c	0.17 - M274c	0.12 - M424c

5.6.2.4 The Implementation of Similarity of Predictive Model (Sim) to Multi Class Demetra Data Sets

In this experiment, models from Table 5.26 were selected to calculate their similarity. From the results in Table 5.27, it shows that the models have a large spread of performance value of FNR.

The similarity values between confusion matrices shows that similar FNR values between models indicate similar performance among them although using different classifier algorithms. Example of such models are models M4a, M304a, M31c and M181c.

However, the results only show a single element of the similarity evaluation for predictive models' performance. In order to have more accurate results of the similarity of predictive models, the comparison of multi class confusion matrices can be applied using the proposed methodology for calculating the similarity of binary predictive models.

Table 5.27: Similarity Matrix for Models (M4a, M304a, M151c and M154c).

Models	M4a	M304a	M151c	M154c
M4a	1.00	1.00	0.95	0.97
M304a	1.00	1.00	0.97	0.97
M151c	0.95	0.97	1.00	0.98
M154c	0.97	0.97	0.98	1.00

5.7 Limitations

In this thesis, the classifier comparison was proposed by comparing their similarity. The comparison consists of three elements which are Input (data set properties), Function (classifier properties) and Output (confusion matrix). The comparison of input was based on one to one matching assuming that the descriptor names and chemical compounds had already gone through a quality check. The thesis can be improved by considering predictive models from different sources by integrating an ontology in matching criteria so that more

models from different sources can be included in the pool of models. In addition, the element of functions also can be enhanced by further analysing their properties rather than by making a simple comparison.

5.8 Summary

This study shows that comparing predictive models is an important issue since it can help users to minimise the cost of generating new predictive models by reusing an existing ones. The comparison of models from huge repositories of models would help to find relevant and good quality models based on comparison algorithms. The confusion matrix provides a more useful quality indicator for the performances of predictive classifier models. The analysis and understanding of their relationships will make the classifier selection more reliable.

The comparison of models from large repositories of models would help to find the relevant model based on optimisation of comparison functions. The experiments show that the similarity of models will help in classifying models for further analysis and customised selection and combination of the relevant model according to the user's needs.

This study also shows that comparing predictive models' confusion matrices will help users to choose similar models based on FNR as a performance measure. From the experiments presented, regrouping multi class classifiers' confusion matrices to binary is

another solution to analyse and categorise the performance of multi class classifiers from a collection of models. This methodology can be integrated in ensembles of classifiers by further analysing the diversity of classes of selected models which will be discussed in the next chapter.

Chapter 6

Proposed Method for Optimisation of Classifier Ensemble

6.1 Introduction

Ensembles of classifiers have proved their potential in getting higher accuracy compared to a single classifier (Santos & Sabourin 2011, Al-Muhanna & Meshoul 2011, Bian & Wang 2007). High diversity in an ensemble may improve the performance results significantly. This chapter proposes an ensemble approach which has diversity calculated using measures of classification output such as disagreement measure and double fault measure. A Classifier Ranking System (CRS) is introduced for the selection of relevant classifiers. The Optimisation of Classifiers Ensemble Method ($OCEM$) technique which applies to the ensemble selection was implemented

to optimise selection of models and combination method. The results show that the proposed method performs well in selecting the relevant ensemble model to improve the prediction from a collection of classifiers.

Wenjia (2010) and optimised in selecting the member of candidates in ensemble by balancing the FNR and FPR to minimise the error.

This thesis addresses two possible scenarios during the stage of generating predictive toxicology models to be stored in a collection of models as follows:

- Domain experts developed as good as possible models based on the data sets (offline approach)
- Models are developed during data study (online approach)

The ensemble method proposed was not focused on generating models during training because the input is complex and the added value of domain experts may be lost. Thus, the thesis proposed a way to construct an ensemble by reusing an existing collection of models. The collection of models is presumed that all models come from the same scenarios. Some such models are unbalanced in prediction performance results, e.g. FNR=0.00 and FPR=1.00 or vice versa although with high Acc. Thus there is a need to combine and balance the performance measures FNR and FPR while maintaining highest Acc. In this chapter the ensemble methods were developed by considering ensemble diversity issues suggested by Wang (2010) and optimised in selecting candidates by balancing the FNR and FPR to minimise the ensemble error.

The contributions in this chapter are:

- A technique using a cost function (combination of Acc, FNR and FPR) to rank classifiers from the pool of classifiers.
- A new algorithm to optimise the selection and combination of classifiers.
- Improved results of overall Accuracy, False Negative Rate and False Positive Rate for all data sets.

The process of generating quality models can use many data mining tools, but the management and selection of relevant models from a pool of classifiers is still an open issue. Some models are useful with a test set but might be worse in another domain or with other test sets. To speed up the process of generating predictive models, the models will be selected from a collection of models. The relevant models have to be chosen from a previously built collection of models to work on new problems.

The application used to demonstrate the advantages of the method proposed is to find a better solution for predicting whether a chemical compound is toxic or non-toxic. The prediction is made by selecting relevant classifiers from a collection of existing classifiers. The issue arising from this is how to find the relevant classifiers, to rank them and then use them individually or to combine the models in an ensemble for better prediction results. In order to have better results in performance measures, the selections of classifiers from the collection are based on three performance measures (Accuracy, False Negative Rate and False Positive Rate). The classifiers then

will be combined as an ensemble to meet the performance measures criteria.

The rest of the chapter is structured as follows : Section 6.2 presents the concept of model diversity, model selection, model ranking and decision fusion strategy. Section 6.3 will describe the proposed method to rank the classifiers from a pool for model selection. Section 6.4, introduces and exemplifies the technique to optimise the ensemble method of classifiers. The implementation of the proposed method for optimisation of the classifier ensemble method will be discussed in Section 6.5.

6.2 Classifier Ensemble Method

Selection of classifiers can be done in many different ways. The main objective of classifier selection is to get the most accurate model for a given new data set (Sewell 2011). A model ensemble is a well known technique to improve accuracy. Many ensemble methods have been developed for different applications: such as bagging (Breiman 1996) and boosting (Schapire & Freund 1998). The aim of those techniques is to improve only the accuracy. Better performances have been reported by researchers combining the classifiers in hybrid ensembles (Neagu et al. 2005) and (Wang et al. 2001).

Also, systems for predictive models management that offer online services such as scoring, model composition, model comparison, search and statistics exist (Gorea 2008). The high-quality of predictive model management systems that allow users to get differ-

ent performance measures (such as FNR and FPR) can be developed by considering several issues of model representation, model comparison and model ensemble.

This thesis proposed an ensemble technique which intends to improve model performance in three aspects: $ER = (1 - Acc)$, false negative rate (FNR) and false positive rate (FPR). The similarity techniques proposed in Chapter 5 are applied to classify the similarity models by their Input (data set), Function (classifiers properties) and Output (confusion matrix). From the similarity of models, we can get diversity of models, where similar models were grouped together. To rank the classifiers, the Classifier Ranking System (CRS) proposed in Section 6.3.1 was used. The flexibility of the selection models in an ensemble using a composite of three performance measures by applying a weight to Acc, FNR and FPR were produced better quality prediction models. The following issues should be considered when implementing ensemble learning.

6.2.1 Classifiers Diversity

The main objective of having diverse classifiers in an ensemble is to ensure that the classifiers selected would not make the same (common) mistake. Wang (2008) studied the problem and listed the factors that affect the accuracy of an ensemble. The factors studied include the accuracy of individual models, the diversity among the individual models in an ensemble, the decision making strategy, and the number of members used for constructing an ensemble.

The diversity measures considered in this proposed ensemble

were the disagreement and double fault measure. Following are the equations to calculate both diversity measure. The notation of the equations are refer to Table 2.4. The Disagreement measure is as follows:

$$Dis_{i,k} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{6.1}$$

and the Double Fault measures is as follows:

$$DF_{i,k} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{6.2}$$

6.2.2 Classifiers Selection

Currently few studies on choosing the relevant models from a pool of models exist. This research considers that classifiers were trained and stored in a collection of models using standard representation such as PTML. From the collection, the most relevant models based on user's requirements can be selected. This is generally an optimization problem aiming to find the model with higher Acc, lower FNR, and lower FPR for a given new problem.

The models were reused from a pool of existing models as a more efficient way of choosing the relevant models and reuse existing knowledge. To rank the classifiers, the performance measures used were Acc, FNR and FPR. However, this solution is open to criticism when it comes to the decision on which classifier to be applied and chosen from a collection of classifiers. In prediction problems, classifiers built upon a good ensemble combination performed better compared to a single classifier. One of the methods to rank the clas-

sifiers is to calculate the similarity of models as proposed in Chapter 5.

Aho et al. (2008) proposed a ranking method for models selection from a pool of models. They proposed ranking selection by comparing the training distributions of classifiers with the input distribution. Instead, this research proposes CRV to rank the relevant classifiers to be selected in the final ensemble because the models selected were based on a combination of three performance measures of Acc, FNR and FPR.

6.2.3 Decision Fusion Strategy

The final output of an ensemble method of classifiers will depend on the decision fusion strategies. Ghosh et al. (2011) summarised in their paper that approaches for decision fusion can be categorised into two classes, which are utility-based and evidence-based. The utility-based techniques include simple average, voting techniques, and their variants. These techniques are the simplest way to fuse decisions and do not utilize any prior knowledge or evidence from previous prediction. For the evidence-based approach, the decision to be made will incorporate a priori information from the past predictions.

In this study, simple majority voting was used in predicting the chemical compound toxicity for ensemble classifiers. In simple majority voting the chemical compound will be predicted as toxic if the vote is 50% or more, to give a high confidence in predicting toxic class. Following is the simple majority voting technique applied in

this study.

$$SMVoting_i = \frac{C_i \sum_1^n}{n} \tag{6.3}$$

where:

i is the index of Instances in a classifier,

n is number or classifiers in an ensemble,

C is classifier.

If the value of $SMVoting_i \geq 0.5$ then the predicted class will be toxic and if $SMVoting_i < 0.5$ the predicted class will be non toxic.

Consider Table 6.1 where simple majority voting for two classifiers is demonstrated. To simplify the calculation of simple majority voting as Equation 6.3, the toxic class which is "Yes" was mapped to a value of 1 and non toxic class which is "No" was mapped to a value of 0.

The last column (New Predictive Classes) of the table show the new prediction of the classes obtained. The new toxic class was predicted whenever only one output from a classifier predicted toxic. This means, in a pair of classifiers, to predict non toxic class, both classifiers have to predict "No" to get a non toxic class. If one of the classifier predicts "Yes", then the final result will be "Yes" or toxic class.

6.3 Proposed Classifiers Ranking System

The method proposes that the classifiers can be selected based on their performance measures such as Acc, FNR, FPR, True Negative

Table 6.1: Simple Majority Voting for Two Classifiers

InstancesID	$C1$ $Output$	$C2$ $output$	$SMVoting$ $Values$	New Predictive Classes
1	1	1	2/2 = 1.0	1 ("Yes")
2	0	0	0/2 = 0.0	0 ("No")
3	1	0	1/2 = 0.5	1 ("Yes")
4	0	1	1/2 = 0.5	1 ("Yes")

Rate (TNR) and True Positive Rate (TPR). The most critical performance in predictive toxicology is classifier with low FNR. The selected classifiers were resulted from the composite performance measures. Selecting relevant classifiers from the pool of models can be done by comparing the classifiers and selecting the right performance measures.

6.3.1 Classifiers Ranking Value

A well known method to predict future outcome with higher accuracy for classification problems is an ensemble method. The system will allow the models to be chosen among the diversity of models and make a combination of techniques to make it hybrid. The literature proved that this technique is able to successfully predict the truth and achieve very high accuracy compared to single classifiers.

Diversity of classifiers in an ensemble may produce better prediction models. A process of constructing ensembles starts with data manipulation, model generation, selection of models and selection of decision fusion strategy (Wang 2008, Bian & Wang 2007).

This study proposed that to select the most relevant classifier to be a member of an ensemble, a classifier rating system should be

used. The performance measures of Acc, FNR and FPR may be included in the classifiers selection in the ensemble. The weight can be assigned to all of the performance measures. In Equation 6.4, the user can use the weights of performance measures by embedding them in the Classifier Ranking Value (CRV):

$$CRV = (w_1 * (1 - Acc)) + (w_2 * FNR) + (w_3 * FPR) \tag{6.4}$$

where:

CRV is a ranking value for a classifier,

$w1$, $w2$ and $w3$ are the weights for Acc, FNR and FPR, respectively.

The values of $w1$, $w2$ and $w3$ are between 0 and 1.

The sum of ($w1$ + $w2$ + $w3$) equals to 1.

The method proposes the classifier rating system by giving a classifier rating value to each classifier in the pool of classifiers. Using the CRV, the best classifiers can be selected and consequently have diversity of classifiers in the ensemble for their combination. The combination technique will be introduced in the next section by optimising the diversity of classifiers. This approach can also be applied to ensemble classifiers. Of course, before the CRV for an ensemble is calculated, the classification output must be combined by certain aggregator, such as simple majority voting.

6.4 Proposed Ensemble Method

In this section, the optimisation of the hybridisation of the classifiers in an ensemble will be discussed. The aim is to combine the best classifiers against the most relevant and diverse classifier in the pool. The optimisation technique of ensemble process applied was similar to GA in three phase:

- Initialisation

  In the first phase, the classifiers from a collection were compared using the proposed Similarity Measure (Sim). The objective of this is to eliminate any similar models, thus it can speed up the ensemble process. The CRV will apply to all the divergent models related to problems to be predicted. The lowest CRV will be initialised as a base classifier.

- Selection

  In the second phase, selection will be applied to find the most diverse classifier from the base classifier. The most divergent will be paired with the base classifier. The pair will be fused together to find the new combination output. The results of the combination will be compared with the base CRV. If the results improve the previous CRV, the pair will be selected and if it does not improve the CRV, the process will start with mutation. In this case, another divergent classifier will be a paired with the base classifier.

- Reproduction

  The last step is the repetition process. The process of finding

a suitable pair will be repeated until it improve the previous CRV. The first and second steps will be repeated until the new ensemble attained the minimum CRV.

This study presumes that a pool of classifiers with different parameter settings has been generated previously. The parameters of classifiers were generated using feature selection algorithms and machine learning algorithms. The pool of classifiers were diverse where there are no similar models in the pool calculated using Sim as proposed in Chapter 5. The pool had gone through clean up process using the proposed Sim in previous chapter. This will ensure that the selected classifiers are diverse and there are no similar models in the pool.

The selection of the classifiers that are to be included in an ensemble from a large set of classifiers requires high computational complexity. For example more models to be added in the ensemble will lead to more processing time in making combinations of them. To optimise the selection, the method presumed that the best classifier (lowest CRV) is an initial classifier (C_1). By measuring the diverse of C_1 from other relevant classifiers (C_i where i = 1, 2, ... , n), we can find the best complement of C_1 that can be combined in the ensemble in order to achieve optimal performance measures (Acc, FNR and FPR). This study developed an ensemble of complementary classifiers initially in order to optimise their diversity.

Input: A set of classifiers M with elements C_i ($i \in \{1, 2, \ldots, n\}$, $n \geq 2$), where each C_i is a classifier. A dataset D for classification.

Output: A set of classifiers (in ensemble) E, $E \subseteq M$.

1. E = $\{C_k\}$, where C_k is the classifier with the smallest CRV, denoted as CRV_E
2. M = M - C_k
3. **DO** {
4. **Sort (M)** in descending order according to Distance Measures and **PUSH** them into a stack S
5. **DO** {
6. Cl = **POP(S)**
7. E' = E $\bigcup$ C_l
8. Calculate the classification result of D using E'
9. Calculate CRV for E'
10. } **WHILE** ($CRV_E < CRV_{E'}$)
11. **IF** ($CRV_{E'} > CRV_E$) {
12. E = E $\bigcup$ C_l
13. M = M – C_l
14. }
15. } **WHILE** ($|M| > 0$ && $CRV_{E'} > CRV_E$)

Figure 6.1: The *OCEM* Algorithm

6.4.1 The $OCEM$ Algorithm

The algorithm developed is as depicted in Figure 6.1. Given a set of classifiers, a pool of diverse classifiers is selected by eliminating the similar ones. This is achieved by the Sim technique proposed in the Chapter 5.

The algorithm starts by calculating CRV for all the selected classifiers in the pool which correspond to a set M. The model with the lowest CRV will be assigned as a base model C_k (line 3). To improve the performance measure of the base classifier, it will be combined with the most diverse of relevant classifiers. Thus in line 6, to find a divergent classifier against the base classifier, the diversity between C_k and M using classifiers diversity $Dis_{i,k}$ will be calculated.

There will be a repetition to find good ensemble classifiers by combining the base classifier with the diverse classifier and calculating the performance of the combined classifiers. If the performance measures of the ensemble improve the CRV of the ensemble, then the classifiers will be retained in the ensemble. If the CRV is not improved, the next diverse classifier will be added into the ensemble.

In particular, a temporary stack (S) is utilised to host the remaining classifiers in the pool in descending order for convenience. The classifier with the highest diversity value will always pop out first to tentatively combine with the existing ensemble to examine if the combination produces higher CRV values. The last step is selecting the ensemble that obtains the highest performance measures.

For this study, collections of models were generated using a series of algorithms implemented in Weka, such as:

- k-nearest neighbours classifier (weka.classifiers.lazy.IBk),
- decision trees (weka.classifiers.trees.J48),
- numerical prediction (weka.classifiers.rules.JRip),
- random forest (weka.classifiers.randomforest) and
- neural network (MultilayerPerceptron).

For the ensemble classifiers, bagging, boosting (adaboostM1), stacking (StackingC), and Bayes were used.

The predictive models were applied to various toxicology data from the Demetra project and also to the UCI data sets. More than 1000 predictive models were generated with different combinations of data sets, algorithms, and model parameters. The models were generated from a group of predictive toxicology data sets whereby each group of data sets was run through data preparation and reductions processes. All the models were validated using 10-Folds Cross Validation.

Feature selection was used to find sets of attributes that are highly correlated with the target classes. The feature selection algorithms applied to the data sets were Correlation-based Feature Selection (CFS), CfsSubsetEval, ClassifierSubsetEval and ConsistencySubsetEval with the BestFirst, GeneticSearch and GreedyStepwise as searching methods using Weka. The following studies were conducted to implement the proposed $OCEM$ algorithm.

6.5 The $OCEM$ Applied to Demetra Data Sets

This section studies the performance of $OCEM$ applied to Demetra data sets with the aim to achieve the highest Acc and at the same time minimise the FNR and FPR to give the ensemble produced more balance in predicting both classes. The experiment of Demetra data sets with Bagging, AdaBoost, Stacking and Bayes will be a benchmark to evaluate the performance of $OCEM$. The main objective of the $OCEM$ is to optimise the combination of three performance measures (Acc, FNR and FPR) so that the ensemble constructed will have minimal but balance between FNR and FPR and maintain the highest Acc. The CRVs were assigned individual weights such as $w1$ for Acc, $w2$ for FNR and $w3$ for FPR. For the diversity measures, the disagreement measure were denote as $OCEM_D$ and double fault measure as $OCEM_{DF}$.

The investigations were divided into 3 subsections as follows:

1. Applied $OCEM$ with a single performance measure where:
    - $w1$=0.0, $w2$=1.0 and $w3$=0.0 (focused on FNR) denote as $CRV1$.
    - $w1$=0.0, $w2$=0.0 and $w3$=1.0 (focused on FPR) denote as $CRV2$, and
    - $w1$=1.0, $w2$=0.0 and $w3$=0.0 (focused on Acc) denote as $CRV3$,
2. Applied $OCEM$ by combining two performance measures where:
    - $w1$=0.5, $w2$=0.5 and $w3$=0.0 (focused on Acc and FNR) denote as $CRV4$, and

- $w1$=0.0, $w2$=0.5 and $w3$=0.5 (focused on FNR and FPR) denote as $CRV5$.

3. Applied $OCEM$ by combining three performance measures where:

   - $w1$=0.3, $w2$=0.5 and $w3$=0.2 (focused on more weight to FNR followed by Acc and FPR) denote as $CRV6$, and
   - $w1$=0.6, $w2$=0.2 and $w3$=0.2 (focused on more weight to Acc but considered to minimised the FNR and FPR) denote as $CRV7$.

6.5.1 The Comparative Study on Ensembles Methods (Bagging, Boosting, Stacking and Bayes)

Table 6.2: Acc, FNR and FPR for Bagging, AdaBoost, Stacking and Bayes.

Data Set	Bagging	AdaBoost	Stacking	Bayes
Bee	Acc = 0.90 FNR = 0.01 FPR = 0.70	Acc = 0.90 FNR = 0.00 FPR = 0.75	Acc = 0.86 FNR = 0.00 FPR = 1.00	Acc = 0.90 FNR = 0.02 FPR = 0.65
Daphnia	Acc = 0.81 FNR = 0.07 FPR = 0.45	Acc = 0.81 FNR = 0.09 FPR = 0.45	Acc = 0.71 FNR = 0.00 FPR = 1.00	Acc = 0.77 FNR = 0.20 FPR = 0.29
Dietary Quail	Acc = 0.87 FNR = 0.10 FPR = 0.20	Acc = 0.84 FNR = 0.11 FPR = 0.28	Acc = 0.63 FNR = 0.00 FPR = 1.00	Acc = 0.85 FNR = 0.07 FPR = 0.30
Oral Quail	Acc = 0.71 FNR = 0.42 FPR = 0.13	Acc = 0.68 FNR = 0.51 FPR = 0.13	Acc = 0.42 FNR = 0.80 FPR = 0.20	Acc = 0.69 FNR = 0.48 FPR = 0.14
Trout	Acc = 0.82 FNR = 0.06 FPR = 0.49	Acc = 0.80 FNR = 0.08 FPR = 0.55	Acc = 0.77 FNR = 0.00 FPR = 1.00	Acc = 0.77 FNR = 0.00 FPR = 1.00

The well known ensemble methods (Bagging, AdaBoost, Stacking

and Bayes) focused on getting highest Acc. From the results (see Table 6.2), it really shows that Bagging is the best method compared to AdaBoost, Stacking and Bayes. Adaboost and Bayes are average methods to classify Dametra data sets. The worst method was Stacking where it can not classify most of the non toxic class for 4 data sets (Bee, Daphnia, Dietary Quail and Trout) e.g. FNR=0.00, FPR=1.00.

Although Bagging performs best on the Acc for all data sets, the huge distances between its FNR and FPR are still an important issue. For example, Bagging has difference between FNR and FPR for data set Bee 0.69, Daphnia 0.38 and Trout 0.37. This shows that the classifier was biased in certain classes.

To overcome the big distances between FNR and FPR, an ensemble proposed should be able to close the gap. Thus, the following sections will study different combinations of CRV weights to give different results to Acc, FNR and FPR.

6.5.2 The Implementation of $OCEM$ on a Single Performance Measure)

Table 6.3 show results for 3 different parameters given to $OCEM$. $OCEM_D$ is using disagreement and $OCEM_{DF}$ is using double fault as diversity measures.

From the results, $CRV1$ given $w1$=0.0, $w2$=1.0 and $w3$=0.0, the $OCEM_{DF}$ using double fault measure is able to achieve the lowest FNR compared to $OCEM_D$ for data sets Bee (see columns $CRV1$). The same results were achieved for Daphnia, Dietary Quail and

Table 6.3: $OCEM$ that Focused on Single Performance Measures.

Diversity CRVs	$OCEM_D$ $CRV1$	$OCEM_{DF}$ $CRV1$	$OCEM_D$ $CRV2$	$OCEM_{DF}$ $CRV2$	$OCEM_D$ $CRV3$	$OCEM_{DF}$ $CRV3$
Bee	Acc = 0.86 FNR = 0.03 FPR = 0.79	Acc = 0.86 FNR = 0.00 FPR = 1.00	Acc = 0.88 FNR = 0.06 FPR = 0.50	Acc = 0.87 FNR = 0.08 FPR = 0.43	Acc = 0.92 FNR = 0.01 FPR = 0.50	Acc = 0.92 FNR = 0.01 FPR = 0.62
Daphnia	Acc = 0.71 FNR = 0.00 FPR = 1.00	Acc = 0.71 FNR = 0.00 FPR = 1.00	Acc = 0.55 FNR = 0.61 FPR = 0.07	Acc = 0.57 FNR = 0.59 FPR = 0.05	Acc = 0.88 FNR = 0.07 FPR = 0.36	Acc = 0.84 FNR = 0.09 FPR = 0.35
Dietary Quail	Acc = 0.63 FNR = 0.00 FPR = 1.00	Acc = 0.63 FNR = 0.00 FPR = 1.00	Acc = 0.85 FNR = 0.12 FPR = 0.20	Acc = 0.82 FNR = 0.13 FPR = 0.31	Acc = 0.85 FNR = 0.11 FPR = 0.23	Acc = 0.85 FNR = 0.09 FPR = 0.29
Oral Quail	Acc = 0.78 FNR = 0.09 FPR = 0.47	Acc = 0.78 FNR = 0.09 FPR = 0.47	Acc = 0.67 FNR = 0.60 FPR = 0.07	Acc = 0.70 FNR = 0.20 FPR = 0.50	Acc = 0.84 FNR = 0.16 FPR = 0.17	Acc = 0.84 FNR = 0.16 FPR = 0.17
Trout	Acc = 0.78 FNR = 0.00 FPR = 1.00	Acc = 0.78 FNR = 0.00 FPR = 1.00	Acc = 0.65 FNR = 0.42 FPR = 0.10	Acc = 0.66 FNR = 0.41 FPR = 0.10	Acc = 0.84 FNR = 0.08 FPR = 0.50	Acc = 0.84 FNR = 0.08 FPR = 0.46

Trout using both diversity measures, disagreement and double fault measure. The drawback here is when focused on certain performance, in this case FNR ($CRV1$), the error can be minimised but the error of other performance measures will be increased. For example the Acc is not optimum and FPR up to 1.0 for most of the data sets.

When we focused on FPR given $w1$=0.0, $w2$=0.0 and $w3$=1.0, the lowest FPR was achieved, but the Acc and FNR were not optimum for example data set Daphnia (see columns $CRV2$).

The same problem was found when we focused on Acc given $w1$=1.0, $w2$=0.0 and $w3$=0.0, FNR and FPR are not minimised and the different between them is noticeable (see columns $CRV3$). All the Acc for five data sets were maximised but there is a big gap between FNR and FPR. Thus, there is motivation to combine performance measures to close the gap between FNR and FPR but maintain the highest Acc. The optimised results by combining performance measures will be discussed in the following sections.

6.5.3 The Implementation of $OCEM$ on Two Performance Measures)

Table 6.4: Results by Combining Two Performance Measures

Diversity CRVs	$OCEM_D$ $CRV4$	$OCEM_{DF}$ $CRV4$	$OCEM_D$ $CRV5$	$OCEM_{DF}$ $CRV5$
Bee	Acc = 0.91 FNR = 0.00 FPR = 0.64	Acc = 0.91 FNR = 0.00 FPR = 0.64	Acc = 0.92 FNR = 0.01 FPR = 0.62	Acc = 0.93 FNR = 0.01 FPR = 0.43
Daphnia	Acc = 0.81 FNR = 0.06 FPR = 0.51	Acc = 0.81 FNR = 0.06 FPR = 0.51	Acc = 0.82 FNR = 0.12 FPR = 0.32	Acc = 0.75 FNR = 0.32 FPR = 0.08
Dietary Quail	Acc = 0.82 FNR = 0.09 FPR = 0.32	Acc = 0.85 FNR = 0.09 FPR = 0.29	Acc = 0.86 FNR = 0.09 FPR = 0.23	Acc = 0.80 FNR = 0.13 FPR = 0.37
Oral Quail	Acc = 0.78 FNR = 0.09 FPR = 0.47	Acc = 0.84 FNR = 0.16 FPR = 0.17	Acc = 0.79 FNR = 0.12 FPR = 0.38	Acc = 0.73 FNR = 0.34 FPR = 0.21
Trout	Acc = 0.83 FNR = 0.03 FPR = 0.63	Acc = 0.83 FNR = 0.05 FPR = 0.63	Acc = 0.74 FNR = 0.30 FPR = 0.13	Acc = 0.75 FNR = 0.29 FPR = 0.13

Table 6.4 shows results with different weights of w given to $OCEM$. $OCEM_D$ is using disagreement and $OCEM_{DF}$ is using double fault as diversity measures.

From the results, $CRV4$ given $w1$=0.5, $w2$=0.5 and $w3$=0.0, the $OCEM_D$ using disagreement measure is able to achieve the lowest FNR and high Acc compared to $OCEM_{DF}$ for most of the data sets. The problem here is when focused on two performance measures in this case Acc and FNR, the other performance measure will be higher. For example the FPR for data sets Bee, Daphnia and Trout were over 0.5 (see first column of $CRV4$).

The same problem was found when we focused on FNR and FPR

(see columns $CRV5$ given $w1$=0.0, $w2$=0.5 and $w3$=0.5), FNR and FPR were minimised and the difference between them is lower and improved but the Acc is not optimal for all data sets accept Bee (see last column). Thus, to give more balance and robust ensemble performance, the combination of all performance measures to close the gap between FNR and FPR but maintain the highest Acc were to be considered. The next section will demonstrate the results for the combining Acc, FNR and FPR into CRV.

6.5.4 The Implementation of $OCEM$ to Combine the Three Performance Measures)

Table 6.5: Results by Combining Three Performance Measures

Diversity $CRVs$	$OCEM_D$ $CRV6$	$OCEM_{DF}$ $CRV6$	$OCEM_D$ $CRV7$	$OCEM_{DF}$ $CRV7$
Bee	Acc = 0.92 FNR = 0.00 FPR = 0.51	Acc = 0.92 FNR = 0.01 FPR = 0.50	Acc = 0.92 FNR = 0.01 FPR = 0.50	Acc = 0.93 FNR = 0.01 FPR = 0.43
Daphnia	Acc = 0.85 FNR = 0.07 FPR = 0.36	Acc = 0.84 FNR = 0.09 FPR = 0.35	Acc = 0.82 FNR = 0.12 FPR = 0.32	Acc = 0.82 FNR = 0.12 FPR = 0.32
Dietary Quail	Acc = 0.85 FNR = 0.11 FPR = 0.23	Acc = 0.85 FNR = 0.09 FPR = 0.29	Acc = 0.86 FNR = 0.09 FPR = 0.23	Acc = 0.88 FNR = 0.11 FPR = 0.20
Oral Quail	Acc = 0.78 FNR = 0.09 FPR = 0.47	Acc = 0.84 FNR = 0.16 FPR = 0.17	Acc = 0.84 FNR = 0.16 FPR = 0.17	Acc = 0.84 FNR = 0.16 FPR = 0.17
Trout	Acc = 0.82 FNR = 0.12 FPR = 0.38	Acc = 0.82 FNR = 0.12 FPR = 0.38	Acc = 0.82 FNR = 0.12 FPR = 0.38	Acc = 0.83 FNR = 0.12 FPR = 0.37

Table 6.5 shows the results for combining all the performance measure. From the results, $CRV6$ was given $w1$=0.3, $w2$=0.5 and $w3$=0.2 to improved on FNR followed by Acc and FRP (see columns $CRV6$). Although it achieved the lowest FNR and high Acc using disagreement measure ($OCEM_D$) for data sets Bee, Daphnia and Oral Quail, the gap compared to the FPR is still high.

The gaps were improved where FPR were lowest for those data sets by adjusted the weights given $w1$=0.6, $w2$=0.2 and $w3$=0.2 (see $CRV7$). This shows that the performance measures were optimise for all data sets using the weights given and double fault measure as diversity measure.

The proposed algorithm for optimisation and combination of Acc, FNR and FPR of ensemble models using double fault measure as the diversity measure improves the Acc between 0.01 to 0.30 for all toxicology data sets compared to other ensemble methods such as Bagging, Stacking, Bayes and Boosting. The highest improvements for Acc were for data sets Bee (0.30), Oral Quail (0.13) and Daphnia (0.10). A small improvement in Acc was achieved for Dietary Quail and Trout of about 0.01. The most important results in this finding by combining all the three performance measure were able to reduce the distance between FNR and FPR for Bee, Daphnia, Oral Quail and Trout data sets between 0.17 to 0.28. The Dietary Quail improved for about 0.01 though, but this data set is well known as a difficult learning exercise (Neagu, Guo, Trundle & Cronin 2007). For five UCI data sets tested, similar results achieved with Acc improvement between 0.10 to 0.11 and were closing more gaps between FNR and FPR.

Figure 6.2 is a chart that used data from Table 6.5. The chart combined the error rate of each performance measures. It can be seen that the most stable and balance performances of ensemble constructed for all Demetra data sets is $OCEM_{DF}$ and $CRV7$. It was given weight of $w1$=0.6, $w2$=0.2 and $w3$=0.2 to get highest Acc and lowest FNR and FPR using double fault measure as diversity measures. The $OCEM_{DF}$ and $CRV7$ ($OCEMDFCRV7$) will be used as the most optimise ensemble for Demetra data sets to be compared with other ensemble methods. The next section will compare ($OCEMDFCRV7$) with other well know ensemble methods such as

Bagging, Boosting, Stacking and Bayes.

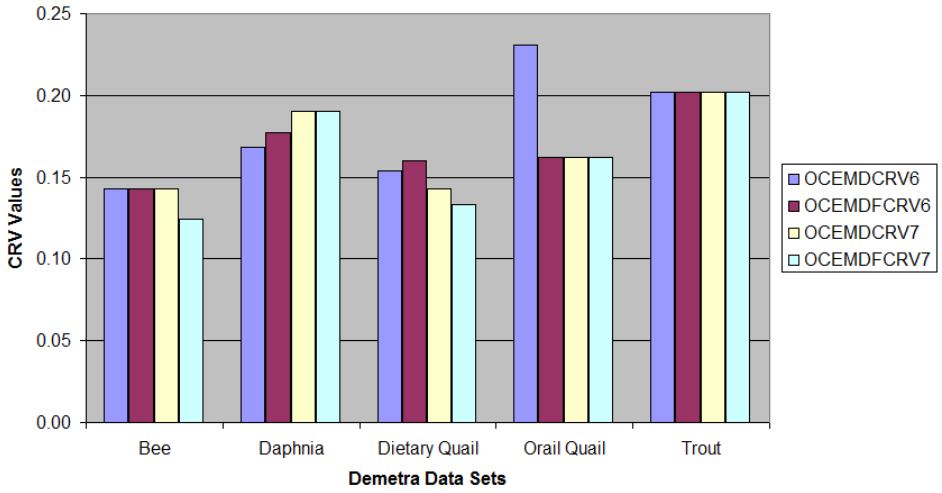

Figure 6.2: Optimised CRV Values

6.5.4.1 The Study on Number of Members in the Ensemble for Demetra Data sets

The objective of this study is to find the optimised number of members in an ensemble. Obviously fewer members in an ensemble will lead to faster processing to find the optimised performance measures. In this study, the voting strategy used is simple majority voting.

Figure 6.3 is graph of CRV values calculated from Table 6.5 (see column $OCEM_{DF}$ and $CRV7$). The results show that the optimum number of members in an ensemble with highest performance measures of Acc, and lowest FNR and FPR is between 2 to 6 members. The Acc dropped, while FNR and FPR increased when the number of members in an ensemble is more than 6 for Demetra data set using the double fault measure. As a conclusion 2 or more member

improves other performance measures.

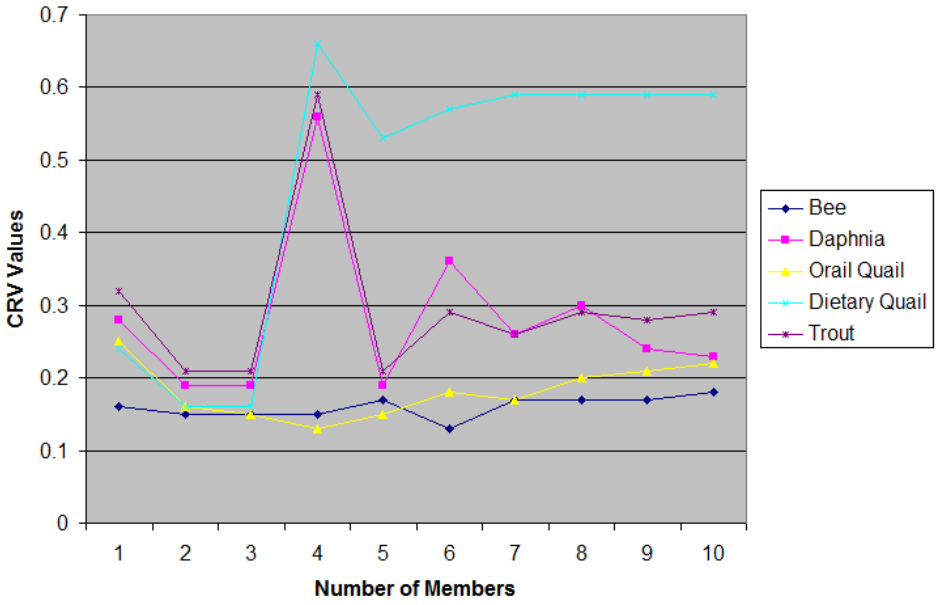

Figure 6.3: The CRV values given $w1$=0.6, $w2$=0.2 and $w3$=0.2 for All Data Set for $OCEM_{DF}$ up to 10 Member in an Ensemble

6.5.5 Comparative Study between $OCEM$ and other Ensemble Methods

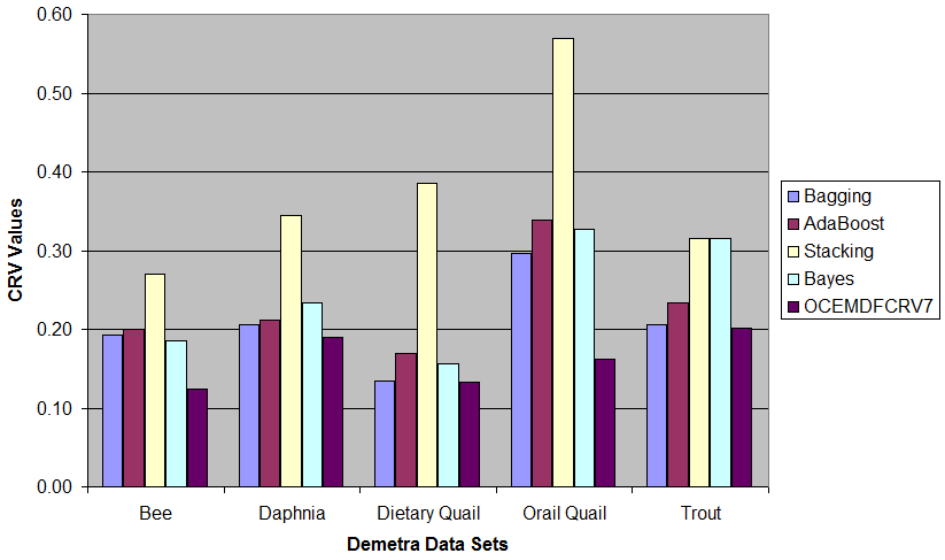

Figure 6.4: $OCEM$ Performance Compared to other Ensembles Methods

This section will study the methods to optimise the three performance measures using proposed $OCEMDFCRV7$ compared to other ensemble methods such as Bagging, Boosting, Stacking and Bayes. The bar graph shows CRV values for $OCEMDFCRV7$ and data from Table 6.2 calculated given $w1$=0.6, $w2$=0.2 and $w3$=0.2. Figure 6.4 shows $OCEMDFCRV7$ consistently getting lowest CRV that optimise the Acc, FNR and FPR for all Demetra data sets. The diversity measure used was double fault measure.

6.5.6 The Implementation of $OCEM$ to Different Group of Demetra Data Sets

This study was conducted on the Demetra data sets that had been divided into different groups. The objective is to investigate whether $OCEM$ algorithm performed well on the different groups of data sets. Original data sets were divided into four groups (Set A, Set B, Set C and Set D). Sun (2005) also divided data sets into groups during training. The portion of every group of data sets was as depicted in Figure 6.5). The $OCEM$ applied Disagreement Measure as a diversity measure and Set B, Set C and Set D were used during training.

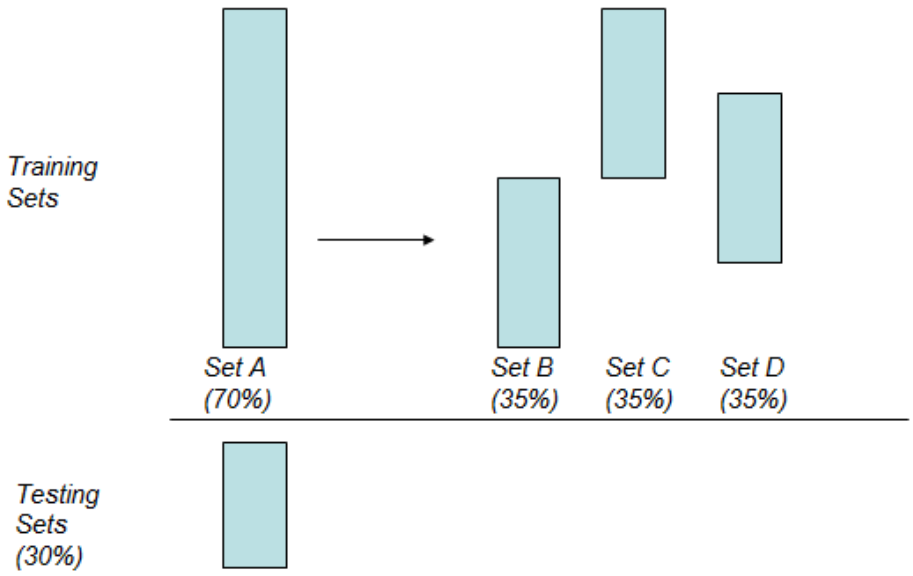

Figure 6.5: Groups of Demetra Data Set

6.5.7 Performance of $OCEM$ to Data Sets Split into Training and Testing Sets

There were studies conducted on data sets that had been divided into a training set (70%) and a testing set (30%). The results show similar conclusions as obtained in previous experiments. The detail results can be found in Table B.5 to Table B.8.

Table 6.6: Acc, FNR and FPR for Different Ensemble Methods for Training and Testing Sets.

Data Set	Bagging	AdaBoost	Stacking	Bayes	$OCEM_DF$	$OCEM_D$
Bee	Acc = 0.90 FNR = 0.07 FPR = 0.25	Acc = 0.90 FNR = 0.03 FPR = 0.50	Acc = 0.87 FNR = 0.00 FPR = 1.00	Acc = 0.78 FNR = 0.17 FPR = 0.50	Acc = 0.73 FNR = 0.69 FPR = 0.13	Acc = 0.96 FNR = 0.00 FPR = 0.25
Daphnia	Acc = 0.62 FNR = 0.10 FPR = 0.30	Acc = 0.83 FNR = 0.06 FPR = 0.33	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.83 FNR = 0.26 FPR = 0.00	Acc = 0.97 FNR = 0.04 FPR = 0.01	Acc = 0.88 FNR = 0.14 FPR = 0.06
Dietary Quail	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.88 FNR = 0.14 FPR = 0.11	Acc = 0.97 FNR = 0.00 FPR = 0.07
Oral Quail	Acc = 0.60 FNR = 0.38 FPR = 0.41	Acc = 0.57 FNR = 0.55 FPR = 0.29	Acc = 0.48 FNR = 1.00 FPR = 0.00	Acc = 0.65 FNR = 0.55 FPR = 0.11	Acc = 0.54 FNR = 0.81 FPR = 0.22	Acc = 0.82 FNR = 0.16 FPR = 0.17
Trout	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.76 FNR = 0.17 FPR = 0.37	Acc = 0.91 FNR = 0.04 FPR = 0.26

This experiment was conducted to study the performance of $OCEM$ compared to other ensemble methods by generating the collection of models based on training set and testing set. The training sets, 70% sequential splits of each Demetra data set. The remaining instances (30%) were the test sets. The process of generating the models to be saved in the collection follows previous methods. From the results (see Table 6.6), the accuracy of OCEM using disagreement measure ($OCEM_D$) outperform other ensemble methods. The OCEM using double fault ($OCEM_D$) did not perform well in training and test sets. This is because the number of instances in the Demetra data sets

were small and splitting them into training sets makes the data sets much smaller for classifier to learn from.

For Oral Quail's data set, $OCEM$ outperforms other ensemble methods. The $OCEM$ result for disagreement measure can be found in column $OCEM_D$ and column $OCEM_{DF}$ is for double fault measure. The Acc is improved using disagreement measure compared to double fault measure. The most significant results were obtained for Oral Quail: highest Acc and lowest FNR. For data sets Bee, Daphnia and Dietary Quail, Acc is highest while FNR lowest compared to other ensemble methods by using $OCEM_{DF}$. The improvement of the FNR is the main objective which means that this method is able to minimise the ER of predicting the toxic class. So as a conclusions, for Demetra data sets that split into training and test set should applied disagreement measure as a diversity measure and given weight of $w1$=0.6, $w2$=0.2 and $w3$=0.2 to the CRV.

6.5.8 Performance of $OCEM$ to by Partitioning Training Sets

Table 6.5 shows the results that different performance measures obtained by giving different weights to each performance measure. By giving more weight to certain measures, that performance could be increased. In this study, the most optimised were the combination of $w1$=0.6, $w2$=0.2 and $w3$=0.2, the ensemble has reached a highest Acc and lowest FNR and FPR using disagreement measures. The Acc for all data sets improved and outperforms all other ensemble methods. This scenario shows that the $OCEM$ method is able to gain higher accuracy for the data sets (see Table 6.7).

Table 6.7: Acc, FNR and FPR for Different Ensemble Methods for Different Partition Data Sets.

Data Set	Bagging	AdaBoost	Stacking	Bayes	$OCEM_D$
Bee	Acc = 0.87 FNR = 0.03 FPR = 0.75	Acc = 0.78 FNR = 0.17 FPR = 0.50	Acc = 0.87 FNR = 0.00 FPR = 1.00	Acc = 0.81 FNR = 0.14 FPR = 0.50	Acc = 0.96 FNR = 0.00 FPR = 0.25
Daphnia	Acc = 0.85 FNR = 0.10 FPR = 0.23	Acc = 0.87 FNR = 0.10 FPR = 0.16	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.85 FNR = 0.22 FPR = 0.03	Acc = 0.90 FNR = 0.08 FPR = 0.13
Dietary Quail	Acc = 0.89 FNR = 0.08 FPR = 0.14	Acc = 0.89 FNR = 0.17 FPR = 0.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.91 FNR = 0.13 FPR = 0.00	Acc = 1.00 FNR = 0.00 FPR = 0.00
Oral Quail	Acc = 0.74 FNR = 0.44 FPR = 0.05	Acc = 0.71 FNR = 0.50 FPR = 0.05	Acc = 0.48 FNR = 1.00 FPR = 0.00	Acc = 0.74 FNR = 0.44 FPR = 0.05	Acc = 0.82 FNR = 0.16 FPR = 0.17
Trout	Acc = 0.88 FNR = 0.04 FPR = 0.46	Acc = 0.87 FNR = 0.02 FPR = 0.06	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.91 FNR = 0.01 FPR = 0.40

6.6 The Implementation of $OCEM$ to UCI Data Sets

For this study, collections of models were generated using the same technique and algorithms as applied to Demetra data sets. All the models were validated using 10-Folds Cross Validation. The objective of this study is to see if the proposed ensemble method (OCEM) perform well to the benchmark data sets from UCI repositories. The results validate that the propose ensemble slightly improve the Acc but lowest FNR and FPR compared to other ensemble method (Bagging, Boosting, Stacking and Bayes).

6.6.1 The Study to Improve Acc and Minimise FNR and FPR

From this experiment, it shows that the Acc for Breast Cancer, Hepatitis, Liver Disorder and Pima Indian Diabetes improved and outperforms all other ensemble methods given $w1$=0.6, $w2$=0.2 and $w3$=0.2 using double fault as diversity measure. The Acc for Blood Transfusion slightly improved by 0.01 compared to other ensembles (see Table 6.8). For the UCI data sets, to have an improvement in a certain performance, more weight has to be given to that performance measure such as results obtained in Table 6.8 to improve Acc.

As a conclusion, the similar result with Demetra data sets also obtained for UCI data sets where the ensemble constructed were able to get highest Acc and minimise FNR and FPR compared to Bagging, Boosting, Stacking and Bayes. The results also show that

the $OCEM$ proposed by given $w1$=0.6, $w2$=0.2 and $w3$=0.2 and double fault as diversity measure can also be applied to other domain and data sets.

Table 6.8: Acc, FNR and FPR for Different Ensemble.

Data Set	Bagging	AdaBoost	Stacking	Bayes	$OCEM_D$	$OCEM_{DF}$
Blood Transfusion	Acc = 0.75 FNR = 1.00 FPR = 0.00	Acc = 0.75 FNR = 1.00 FPR = 0.00	Acc = 0.75 FNR = 0.90 FPR = 0.00	Acc = 0.71 FNR = 1.00 FPR = 0.09	Acc = 0.76 FNR = 0.69 FPR = 0.13	Acc = 0.76 FNR = 0.13 FPR = 0.69
Breast Cancer	Acc = 0.95 FNR = 0.03 FPR = 0.05	Acc = 0.94 FNR = 0.03 FPR = 0.09	Acc = 0.65 FNR = 0.00 FPR = 1.00	Acc = 0.96 FNR = 0.03 FPR = 0.08	Acc = 0.97 FNR = 0.01 FPR = 0.03	Acc = 0.97 FNR = 0.01 FPR = 0.03
Hepatitis	Acc = 0.80 FNR = 1.00 FPR = 0.00	Acc = 0.83 FNR = 0.57 FPR = 0.08	Acc = 0.80 FNR = 1.00 FPR = 0.00	Acc = 0.87 FNR = 0.48 FPR = 0.11	Acc = 0.94 FNR = 0.03 FPR = 0.07	Acc = 0.94 FNR = 0.07 FPR = 0.03
Liver Disorder	Acc = 0.53 FNR = 0.73 FPR = 0.17	Acc = 0.53 FNR = 0.73 FPR = 0.17	Acc = 0.57 FNR = 1.00 FPR = 0.00	Acc = 0.56 FNR = 0.80 FPR = 0.22	Acc = 0.59 FNR = 0.90 FPR = 0.10	Acc = 0.59 FNR = 0.90 FPR = 0.10
Pima Indian Diabetes	Acc = 0.74 FNR = 0.17 FPR = 0.39	Acc = 0.75 FNR = 0.16 FPR = 0.40	Acc = 0.65 FNR = 0.00 FPR = 1.00	Acc = 0.74 FNR = 0.17 FPR = 0.29	Acc = 0.76 FNR = 0.17 FPR = 0.37	Acc = 0.76 FNR = 0.17 FPR = 0.37

6.7 Limitations

In this chapter, the ensemble method proposed was optimised on their performance measures using CRV. The diversity measures studied were disagreement measure and double fault measure. The method can be improved by considering other diversity measures. In addition the decision fusion strategy applied was simple majority voting. It can be broadened to other decision fusion strategy such as weight voting technique.

6.8 Summary

In this chapter, a method to optimise the selection of classifiers from a pool of models and make an ensemble between the classifiers to

obtain higher performance in Acc, FNR and FPR was proposed. The selection process involves selecting the relevant and diverse classifiers by ranking them using a proposed Classifier Rating System (CRS) and calculates their diversity. The experiments show that an ensemble approach is better than a single classifier for predicting the toxic class of chemical compounds. It proved that the method proposed for Optimising Classifier Ensemble Method ($OCEM$) outperforms other four ensemble methods such as bagging, stacking, bayes and boosting.

The results show different performance measures obtained by giving different weights to each performance measures. By giving more weight to certain measures, their performance could be increased. With the combination of $w1$=0.6, $w2$=0.2, $w3$=0.2, the ensemble has reached a optimal Acc, FNR and FPR.

Different results obtained using double fault measure (see Table 6.5). Refer to the table, the results performed well for three data sets such as Bee, Daphnia and Oral Quail. As conclusions, the highest Acc obtained using the double fault measure as a diversity measure and with the combination of $w1$=0.6, $w2$=0.2, $w3$=0.2 applied in $OCEM$.

Chapter 7

Evaluation and Discussion

7.1 Introduction

This chapter will evaluate and discuss the research outcomes within this thesis. The problems of reusing models from collections of models to predict the toxicity of new classes of chemical compounds were the main aims of this research. The work conducted was divided into four main chapters.

Chapter 3 discussed the proposed methodology of the research and the concept of data and model governance. Chapter 4 discussed the proposed standard representation for model management called Predictive Toxicology Markup Language (PTML) that were used in Chapter 5 and Chapter 6 for model comparison and models combination. In Chapter 5, a technique to compare predictive models was proposed by calculating their similarity. The model can be used as a single model or by combining them in the proposed ensemble. Chapter 6 proposed an ensemble method by selecting models with a combination of three performance measures (Acc, FNR and FPR) to

improved the performance.

The outcomes of the chapters will be evaluated and discussed in the following sections below within the context of a framework for model and data governance.

7.2 Methodology and Proposed Framework for Data and Model Governance

The research starts from the availability of models that have been trained by the domain experts. The models can be selected to be used as a single model or in combination to predict new toxicology problems. The research process followed the structured methods as discussed in Chapter 3 and shown in Figure 3.1. There are 3 main processes that integrate together toward getting a quality prediction by reusing a collection of predictive toxicology models. The processes are model representation, model comparison and ensemble construction of models.

The models were assumed to be generated by domain experts and stored in a collection of models. The model comes from different ways of representation developed using various data mining tools. The collection of models needs a proper management system to keep the models updated and corrected so that it can be accessed when needed to be used for new predictive toxicology problems.

As discussed in Chapter 3, from the view of predictive modelling governance, the data and models have to be properly managed to achieve higher prediction. Liu & Tuzhilin (2008) raised the issues in

model management of how to automate the models generation, and the storage of the models. Another issue raised is how the repositories can be retrieved and further analysed. (Fu et al. 2011) also studied data governance issues and proposed a framework for data governance related to data storage management, for example accuracy, completeness and integrity. Besides data governance, models should also be the main asset that needs to be managed properly. Thus, there is a need to define the data and model governance framework.

From that, a new framework for data and model governance was proposed and defined in section 3.3.1. The framework was defined as Data and Model Governance (DMG). It is a set of quality control processes for assessing, managing, using, improving, monitoring, maintaining, and protecting data and (predictive) model information (see Figure 3.2). By defining DMG, the models have to be represented in a standard format so that the quality control process for DMG is possible and will be discussed in the next section.

7.3 Proposed Classifiers Representation

As discussed earlier in Chapter 4, the proposed representation of predictive toxicology models was called Predictive Toxicology Markup Language (PTML). PTML represents data mining models in a standard format using XML and can be simply manipulated for searching and comparing. It also describes predictive toxicology data and the associated model generated by data mining processes. There were

model representations proposed but there are limitations with them as follows:

- PMML (Predictive Model Markup Language) is a standard XML-based language used to represent predictive models and allow sharing of models to compliant applications. PMML is still under development because it is attempting to represent the complete information of data mining processes.

- Chaves et al. (2006) developed a PMML compliant scoring engine called Augustus. Augustus used components from PMML and added other a new components such as data management component, utilities for processing PMML files and run time support.

- Gorea (2008) proposed PMQL (Predictive Modelling Query Language) is a specialized query language for interacting with PMML documents. It is embedded within DeVisa framework which provides functions such as scoring, model comparison, model composition, model searching, statistics and administration through a web service interface for the PMML.

That is why there are other parties building on PMML models such as representations proposed by Chaves et al. (2006) and Gorea (2008). Both rely on the PMML models to have a collection of models and cannot be used with other models.

The difference with proposed $PTML$ is that it can be a bridge to different models that represented in various ways depends on the data mining tools. The difference of PTML with other representations

are as follows:

- Simpler representation but yet able to hold predictive models information,
- Integrative approach for data and model representation, and
- Process and manage the models in relation to the available data.

The PTML structure currently consists of 6 elements: Model Description, Model Parameter, Model Attributes, Model Performance, Class Attribute and Confusion Matrix (See Figure 4.1). Document Type Definition (DTD) for PTML can be found in Appendix A.1. The DTD is an XML schema that allows different formats of predictive models to be imported using PTML standard. All the models generated were stored in the collection using proposed PTML. All the studies and experiments from Chapter 5 and Chapter 6 used collections of models stored based on PTML structures.

The PTML representation proposed focused on classification models with three element of Input (data set), Function (classifier properties) and Output (Confusion Matrix). The representation can be extended to apply regression model in the future. In addition, the representation may be enhanced by including other elements and properties of predictive model such as quality factors.

The next process of data and model governance is models comparison and models combination discussed in the next sections.

7.4 Proposed Method for Classifiers Comparison

Chapter 5 discussed the method proposed for models comparison from a collection of models. By comparing the models, the models can be selected and reused for prediction. This is a model governance process. Choosing the relevant model from the collection may be a easier task: calculating the similarity of predictive models is the key to rank them, which may improve model selection or combination. Furthermore, calculating the similarity of predictive models helps to characterize the model diversity and to identify relevant models from a collection of models.

Comparison of predictive models can be accomplished by measuring the similarity between them. Similarity and distance metrics are complementary to each other. Todeschini et al. (2004) proposed a new measure to calculate a distance between two models based in training sets. The proposed representation of predictive model (PTML) consist of Input (data set), Function (classifier properties) and Output (confusion matrix). From the definition, that is why the proposed models comparison were integrated to compare output and models properties as well.

The proposed models comparison consists of three elements as follows:

- For the first element of PTML which is input, a novel technique to compare data sets (Data set Similarity Coefficient - DSC) was proposed in Section 5.4.1. Using this technique, the models

based on similar data sets can be found or relevant models to the test set (problem) can be searched.

- The second element of PTML is function. The comparison technique for this element was proposed in Section 5.4.2 to find the models that used similar functions.

- The last part of PTML structure is the confusion matrix. To compare the confusion matrices, performance measures such as Acc, FNR and FPR were used. The method was proposed in section 5.4.3. From this method, performance of similar models can be grouped together.

To compare predictive models as a whole, the similarity of each PTML element will be combined as proposed in Section 5.5. The method can be used to compare the similarity of models or to find the relevant models related to new problems.

The outcomes from the studies and experiments conducted were as follows:

- The flexibility of using weight of α, γ and β. To calculate the proposed similarity of predictive models using ($Sim_{(Ma,Mb)}$) with the values of I (α =1), F (β =0) and O (γ =1). False Negative Rate (FNR) was set in the ($Sim_{(Ma,Mb)}$) to justify the importance of it from the viewpoint of toxicology data sets, where the aim was to have a model with low FNR. This means that the models were chosen on the basis of minimum FNR. The detailed results were shown in Section 5.6.1.1.

- The weight can be modified to calculate the similarity of data

sets used between two models. The experiment in Section 5.6.1.2 was to find the similarity of data sets between five end points. The five Demetra data sets are Bee, Daphnia, Dietary Quail, Oral Quail and Trout. For this experiment, I (α =1), F (γ =0) and O (β =0). From the result (see Table 5.21), all data sets share over 50% similar descriptors and chemical compounds: the highest data set similarity is 63% between Daphnia and Trout, while Bee and Oral Quail have about 48% chemical compounds in common.

- From Table 5.22, generally the accuracy of the models increased when a feature selection algorithm was used. The use of the Correlation-based Feature Selection (CFS) as the feature selection algorithm and using J48 classifier seem to have the right combination in correctly predicting the toxicity class with low FNR.

- The binarisation strategies were discussed in Sections and the outcomes were as follows based on results on Table 5.26 and it can be concluded that:

  - Data sets with feature selection algorithms (such as CFS) applied are better in FNR performance measurement compared to data sets with no feature selection.

  - The classifiers performance are highest in Bee data set and lowest in Oral Quail data set.

  - Some performance (FNR) of models with selected class for more than 1 toxic class (e.g. M4c) is poor compared to

binary model with only 1 toxic class (e.g. M4a), but in contrast some of the multi class classifiers are better than binary classifiers (e.g. M34c vs. M34a and M271c vs. M271a).

- On average, models that applied binarisation strategies (model names ended with 'a') are better than multi class classifiers that apply calculation of FNR to their confusion matrices (models names ending in 'c'). This proved that multi class classifiers for Daphnia data sets such as M334c are better than binary classifiers (e.g. M331a). For Oral Quail data set, both binary and multi class had the same performance (0.30) for FNR (e.g. M91c vs. M244a).

- From the results shown in Table 5.26, if the objective is to discriminate between two binary classes, in this case Toxic and Non-toxic, then the classifiers with binary class format have better performance compared to multi class classifiers. For some models, regrouping classes in a single toxic class may increase the accuracy as compared to re-generating binary class classifiers.

The comparison proposed consists of three elements which are Input (data set), Function (classifier properties) and Output (confusion matrix). The problem for this method is the comparison of input was based on one to one matching assuming that the descriptor names and chemical compounds had already gone through a quality check. Improvements can be done by considering predictive models from different sources by integrating an ontology in matching crite-

ria so that more models from different sources can be included in the pool of models. In addition, the element of functions also can be enhanced by further analysing their properties rather than by making a simple comparison.

7.5 Proposed Method for Optimisation of Classifier Ensemble

The last process in model governance is to combine the relevant models to improve three performance measures (Acc, FNR and FPR). The proposed ensemble method was discussed in Chapter 6. Most ensemble methods proposed such as by Masisi et al. (2008), Mehmood et al. (2010), Khakabimamaghani et al. (2010) and Nabiha et al. (2011) were focused only on the Acc. In this thesis, there are experiments that demonstrated to focused on single performance measures (see 6.5.2). The drawback here is when focused on certain performance in this case FNR ($CRV1$), the error of the single performance measure can be minimised but the error of other performance measures will be increased. For example the Acc is not optimum and the worst FPR values are up to 1.0 for most of the Demetra data sets when focusing on minimising FNR . To overcome the big distances between FNR and FPR, an ensemble proposed by combining all the performance measures was able to close the gap.

Table 6.5 shows the results for combining all the performance measures. From the results, $CRV6$ was given $w1$=0.3, $w2$=0.5 and $w3$=0.2 to improved on FNR followed by Acc and FRP (see columns

$CRV6$). Although it achieved the lowest FNR and high Acc using disagreement measure ($OCEM_D$) for data sets Bee, Daphnia and Oral Quail, the gap compared to the FPR is still high.

The gaps were improved where FPR were lowest for those data sets by adjusted the weights given $w1$=0.6, $w2$=0.2 and $w3$=0.2 (see $CRV7$). This shows that the performance measures were optimised for all data sets using the weights given and double fault measure as diversity measure.

Figure 6.2 is based on data from Table 6.5 and shows the error rate for each performance measures. It can be seen that the most stable and balance performances of ensemble constructed for all Demetra data sets is $OCEM_{DF}$ and $CRV7$. It was given weight of $w1$=0.6, $w2$=0.2 and $w3$=0.2 to get highest Acc and lowest FNR and FPR using double fault measure as diversity measures. As a conclusion, the most optimised parameter for $OCEM$ is $OCEM_{DF}$ and $CRV7$ ($OCEMDFCRV7$) for Demetra data sets to be compared with other ensemble methods. A study using those parameters was done and compared with other ensemble methods such as Bagging, Boosting, Stacking and Bayes. The results from Figure 6.4 shows that $OCEMDFCRV7$ consistently getting lowest CRV that optimise the Acc, FNR and FPR for all Demetra data sets. The diversity measure used was double fault measure.

A study was conducted on the Demetra data sets that had been divided into different groups. The objective is to investigate whether $OCEM$ algorithm performed well on the different groups of data sets. Original data sets were divided into four groups (see Section 6.5.6).

The portion of every group of data sets was as depicted in Figure 6.5). From the results the most optimised performance measures were the combination of $w1$=0.6, $w2$=0.2 and $w3$=0.2. The ensemble constructed has reached a highest Acc and lowest FNR and FPR using disagreement measures. The Acc for all data sets improved all other ensemble methods. This scenario shows that the $OCEM$ method is able to gain higher Acc for the data sets (see Table 6.7).

Lastly the ensemble method proposed ($OCEM$) tested to the benchmark data sets from UCI repositories. The results validate that the proposed ensemble slightly improve the Acc but lowest FNR and FPR compared to other ensemble method (Bagging, Boosting, Stacking and Bayes). From the experiment in Section 6.6, it shows that the Acc for Breast Cancer, Hepatitis, Blood Transfusion and Pima Indian Diabetes improved all other ensemble methods given $w1$=0.6, $w2$=0.2 and $w3$=0.2 using double fault as diversity measure (see Table 6.8).

As a conclusion, the similar result with Demetra data sets also obtained for UCI data sets where the ensemble constructed were able to get highest Acc and minimise FNR and FPR compared to Bagging, Boosting, Stacking and Bayes. The results also show that the $OCEM$ proposed by given $w1$=0.6, $w2$=0.2 and $w3$=0.2 and double fault as diversity measure can also be applied to other domain and data sets.

7.6 Summary

This chapter concludes the process of models ensemble building toward data and model governance as discussed in Chapter 3. The relevant models to be included in the ensemble were selected from a collection of models. The models from the collection were represented using proposed PTML representation as discussed in Chapter 4. The PTML models were compared to find the similarity with test sets using proposed similarity measuring techniques as demonstrated in Chapter 5. The ensemble processes combining Acc, FNR and FPR were demonstrated in Chapter 6. As a conclusion the integration of all methods show that the Acc improved and the FNR and FPR were minimised compared to Bagging, Boosting, Stacking and Bayes.

The ensemble method improves the Acc between 0.01 to 0.30 for all toxicology data sets compared to other ensemble methods. The highest improvements for Acc were for data sets Bee (0.30), Oral Quail (0.13) and Daphnia (0.10). A small improvement in Acc was achieved for Dietary Quail and Trout of about 0.01. The most important results in this finding by combining all the three performance measure were able to reduce the distance between FNR and FPR for Bee, Daphnia, Oral Quail and Trout data sets between 0.17 to 0.28. The Dietary Quail improved for about 0.01 though, but this data set is well known as a difficult learning exercise (Neagu et. al. 2007). For five UCI data sets tested, similar results achieved with Acc improvement between 0.10 to 0.11 and were closing more gaps between FNR and FPR. As a conclusion, the results show that by

combining performance measures (Acc, FNR and FPR), as proposed hereby the Acc increased and the distance between FNR and FPR decreased.

Chapter 8

Conclusions and Future Work

This chapter will conclude the research activities within this thesis. The first section will summarise the research method while the second section will discuss the original contribution of the thesis followed by some limitations of the methods proposed. The last section will suggest future work that can be considered to extend the research.

8.1 Introduction

There are lots of available models generated in different formats by a number of data mining tools. All of these models can be used for prediction of new unknown situations. From the scenario, the research starts with the problem of how to represent those models in a structured format. Later, the models were represented using the proposed XML standard format and were able to be analysed for further processing such as comparison and combination between models. General aims and objectives of this research were implemented

in the domain of predictive toxicology.

At the beginning of the thesis, the aim of this research was to establish a new method for searching relevant classifiers from a collection of models and make an ensemble out of them. The aim was achieved by meeting the objectives as stated in section 1.5.

The objectives of this research were:

1. To construct a framework for data and model governance.
2. To develop a knowledge representation for data and predictive toxicology models.
3. To construct a new technique for comparing the similarity of models from a collection of models.
4. To construct new techniques for comparing the elements of a predictive model which are similarity of Input (Training Set), Function (Classifier Properties) and Output (Confusion Matrix).
5. To construct a new technique for ranking the classifiers with a composite of performance measures such as Acc, FNR and FPR.
6. To develop a new algorithm for optimising the selection and combination of classifiers.

From the aim and objectives outlined, the research was designed to follow a structured methodology of research as discussed in section 3.1. The methods were carefully designed and split into chapters and contributions.

The research moved toward the management of the predictive models and how to make a comparison between them. When there is a new problem to be predicted, the relevant classifiers from a collection of models will be selected by comparing all the models. In this stage, the research developed a proposed similarity measure to compare predictive models from a collection of models. The results show that the technique was able find the most relevant model for prediction. The prediction measures were focused on the Acc, FNR and FPR.

As discussed earlier, FNR plays an important indicator in predictive toxicology performance where low FNR means that the model is able to predict toxic class in a safer way. This was the motivation that made the research move forward on how to improve the prediction with the combination of three performance measures of Acc, FNR and FPR. A novel ensemble method was proposed in this stage which applies a composite of the performance measures in order to get the highest quality ensemble models. The other issues related to ensemble construction such as diversity measure and classifier ranking were included as well as optimising the ensemble process.

The whole research process followed the structured methods as shown in Figure 8.1. It shows that there are 3 main processes that contribute toward getting a quality prediction by reusing a collection of predictive toxicology models. The processes are model representation, model comparison and ensemble construction of models. The original contributions to the fields of predictive toxicology and machine learning made by the author within the thesis will be dis-

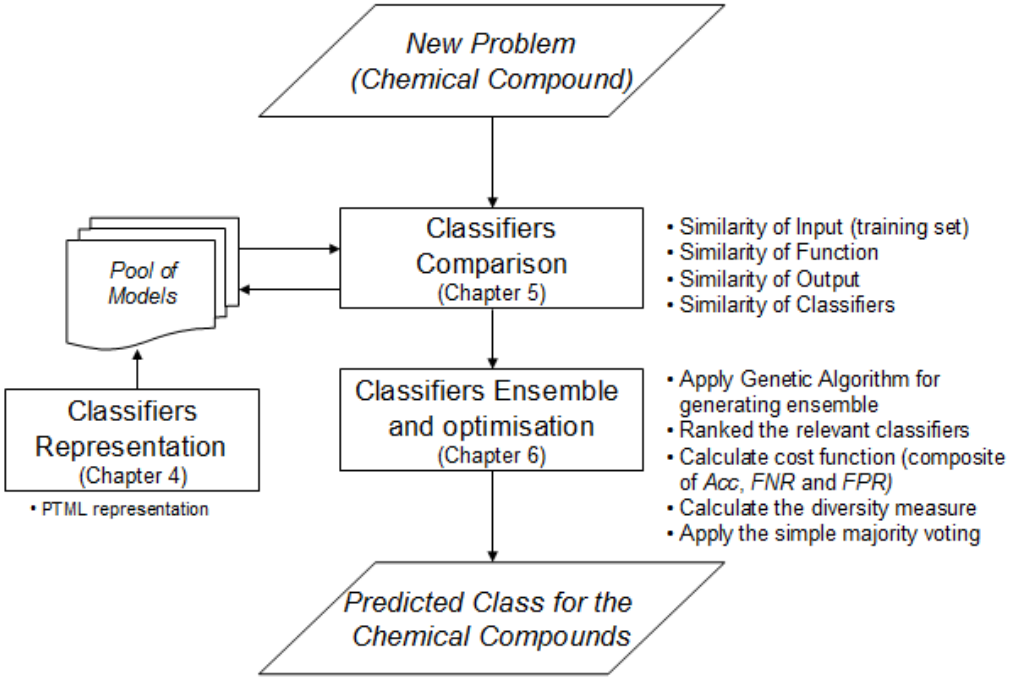

Figure 8.1: The Method Followed for the Research Study

cussed in the next section. The last section will discuss the future directions of how this research can be expanded.

8.2 Original Contributions of the Thesis

This section discusses the contributions in detail and how the aims and objectives were fulfilled.

- A new framework for data and model governance was proposed and defined in section 3.3.1. The framework was defined as a Data and Model Governance (DMG). DMG is defined as a set of quality control processes for assessing, managing, using, improving, monitoring, maintaining, and protecting data and (predictive) model information. The collection of models

need a proper management system to keep the models updated and corrected so that it can be accessed when needed to be used for predicting new problem. The proposed framework was published by the author in (Makhtar et al. 2010): Makhtar, M., Neagu, D. C. and Ridley, M. J. (2010), Predictive Model Representation and Comparison: Towards Data and Predictive Models Governance, in Proceedings of the 10th Annual Workshop on Computational Intelligence (UKCI2010), IEEE Xplore, pp. 1-6.

- By defining the DMG, the models have to be represented in a standard format so that the quality control process for DMG is possible. A new knowledge representation for predictive toxicology data and models called Predictive Toxicology Markup Language (PTML) was proposed in Chapter 4. The PTML was constructed based on the elements of predictive models (input, function and output). The representation was published by the author in (Makhtar et al. 2010): Makhtar, M., Neagu, D. and Ridley, M. J. (2010), Predictive Model Representation and Comparison: Towards Data and Predictive Models Governance, in Proceedings of the 10th Annual Workshop on Computational Intelligence (UKCI2010), IEEE Xplore, pp. 1-6.

- Relevant models can be searched from the collection of models. The searching methods were proposed in Chapter 5 by comparing the predictive models. The flexibility of the comparison is that the similarity measure is grouped into three elements of the PTML (input, function and output). Thus, the comparison

was proposed for each element of PTML as discussed in Chapter 5. The comparison techniques was published by the author in (Makhtar et al. 2011*a*): Makhtar M., Neagu D. and Ridley M.J. (2011): "Binary Classification Models Comparison: on the Similarity of Datasets and Confusion Matrix for Predictive Toxicology Applications", in Proceedings of the 2nd International Conference on Information Technology in Bio and Medical Informatics (ITBAM 2011), Springer LNCS 6865, pp. 108-122.

- For the first element of PTML which is input, a novel technique to compare data sets (Data set Similarity Coefficient - DSC) was proposed in Section 5.4.1. Using this technique, the models with similar data set can be calculated or relevant models to the test set (problem) can be searched.

- The second element of PTML is function. The comparison technique for this element was proposed in Section 5.4.2 to find the models that used similar functions.

- The last part of PTML structure is the confusion matrix. To compare the confusion matrices, performance measures such as Acc, FNR and FPR were used. The method was proposed in section 5.4.3. From this method, performance of similar models can be grouped together.

- In this research, the study was conducted to solve the binary problem with the multi class models. The comparison was proposed in Section 5.4.5 by regrouping the multi class to binary class. The comparison technique of multi class classifiers was

published in (Makhtar et al. 2011*b*) : Makhtar M., Neagu D. and Ridley M.J. (2011): "Comparing Multi Class Classifiers: On the Similarity of Confusion Matrices for Predictive Toxicology Applications", in Proceedings of the 12th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2011), Springer LNCS 6936, pp. 252-261.

- To compare predictive models as a whole, the similarity of each PTML element will be combined as proposed in Section 5.5. The method can be used to compare the similarity of models or to find the relevant models for new problems.

- The last part of the research is the method to improve the performance measures by making a combination of models as proposed in Chapter 6. In order to get a quality model, the ranking technique was proposed in Section 6.3 by using a composite of three performance measures (Acc, FNR and FPR). This will ensure that the models in the ensemble were highest Acc and, minimise FNR and FPR.

- An algorithm was developed to optimise the ensemble methods by optimising the number of candidates in the ensemble and selecting the ensemble using a cost function. The algorithm was discussed in section 6.4. The algorithm was published in (Makhtar et al. 2012): Makhtar M, Yang L, Neagu D. and Ridley M. (2012): "Optimisation of Classifier Ensemble for Predictive Toxicology Application", in Proceedings of the 14th International Conference on Modelling and Simulation (UKSim2012),

IEEE, pp 236-241.

- The ensemble methods proposed lead to the achievement of improved results of Accuracy and minimise False Negative Rate and False Positive Rate for all data sets compared to other ensemble methods such as Bagging, Boosting and Stacking. The results were briefly discussed in Chapter 6.

As a summary, the contributions of this research were:

- A new framework for data and model governance (Chapter 3).
- A new knowledge representation for predictive toxicology data and models (Predictive Toxicology Markup Language - PTML) (Chapter 4).
- A novel technique to compare the similarity of models (Chapter 5) which includes:
  - A technique to compare data sets (training set) (Data set Similarity Coefficient - DSC)
  - A technique to compare the similarity of functions' property used to generate the predictive models.
  - A technique to compare the similarity of confusion matrices.
  - A technique to compare the similarity of multi class confusion matrices.
- A technique using a cost function (composite of Acc, FNR and FPR) to rank classifiers from a collection of models (Chapter 6).

- A new algorithm to optimise the selection and combination of classifiers (Chapter 6).

- An improved results of Accuracy, with minimise False Negative Rate and False Positive Rate for all data sets compared to other ensemble method such as Bagging, Boosting and Stacking (Chapter 6).

Although this research contributes to the domain of knowledge, it can still be improved in the future. The next section will give an outline of some of the limitations of the proposed methods which can be enhanced.

8.3 Research Limitations

The contributions listed and the results presented previously show that the thesis contributes to the domain of the knowledge. However, the research has some limitations which are highlighted as follow:

- The PTML representation proposed was focussing on classification models with three element of Input (data set properties),function (classifier properties) and Output (Confusion Matrix). Other elements and properties of predictive model such as quality factors may be added to the representation. The representation should consider other types of data mining model such as regression model.

- The classifier comparison was proposed by comparing the similarity of them. The comparison consists of three elements

which are Input (data set properties),function (classifier properties) and Output (confusion matrix). The comparison of input was based on one to one matching assuming that the descriptor names and chemical compounds had already gone through quality checks. Ontology can be added to give more flexibility in the comparison method.

- Only two diversity measures were studied which are disagreement measure and double fault measure. The decision fusion strategy applied was simple majority voting. This can be broadened to other methods in the future.

8.4 Recommendations for Further Research

- In Chapter 4 the PTML representation proposed focused on classification models. The thesis may be improved by considering applying a regression model to the representation in the future. In addition, the representation may be enhanced by including other elements and properties of predictive model such as quality factors. The ontology may be considered when comparing the predictive models.

- The one to one matching in comparison method assumes that the descriptor names and chemical compounds had already gone through quality checks. The work can be improved by considering predictive models from different sources by integrating an ontology in matching criteria so that more models from different sources can be included in the pool of models.

In addition, the element of functions can also be enhanced by further analysing the properties of models rather than making a simple comparison.

- The diversity measure techniques is one of the issues that should be considered in ensemble methods. The method can be improved by considering other diversity measure. In addition, the decision fusion strategy applied was simple majority voting. It can be broadened to other decision fusion strategies such as majority voting and weight voting technique.

- The research was focussing on binary class classifiers. In the future the methods such as diversity measure, decision fusion strategy and comparison of classifiers can be applied to enhance multi class classifiers.

- Although the work is promising, the approach can be improved in several directions. The weight ($w1$, $w2$ and $w3$) allocated to each performance measure (Acc, FNR and FPR) is done manually. It is interesting to investigate how to automate this process.

Lastly, the aims and objectives outlined were carried out by following the structured research design proposed. From that, methods were proposed for each objective and related studies and experiments were conducted. The listed contributions show that the aim and objectives were fulfilled. Apart from that, there are some limitations that may be improved in the future with regards to continuing this research domain.

Bibliography

Aho, T., Elomaa, T. & Kujala, J. (2008), Unsupervised classifier selection based on two-sample test, *in* 'Proceedings of the 11th International Conference on Discovery Science', Springer-Verlag, Berlin, Heidelberg, pp. 28–39.

Al-Muhanna, N. & Meshoul, S. (2011), 'Ensemble classifiers for dynamic signature authentication', *IEEE International Conference on Computer Science and Automation Engineering (CSAE)* pp. 700–704.

Bakar, A. A., Kefli, Z., Abdullah, S. & Sahani, M. (2011), Predictive models for dengue outbreak using multiple rulebase classifiers, *in* 'International Conference on Electrical Engineering and Informatics (ICEEI)', IEEE, pp. 1–6.

Bay, S. D., Kibler, D., Pazzani, M. J. & Smyth, P. (2000), The uci kdd archive of large data sets for data mining research and experimentation, *in* 'SIGKDD Explorations', pp. 14–18.

Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Thiel, K. & Wiswedel, B. (2009), 'Knime - the kon-

stanz information miner: version 2.0 and beyond', *SIGKDD Explor. Newsl., 11* (1), 26–31.

Bian, S. & Wang, W. (2007), 'On diversity and accuracy of homogeneous and heterogeneous ensembles', *Int. J. Hybrid Intell. Syst., 4* pp. 103–128.

Bouckaert, R. R., Frank, E., Hall, M. A., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. (2010), 'WEKA–experiences with a java open-source project', *Journal of Machine Learning Research, 11* pp. 2533–2541.

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

Caruana, R., Niculescu-Mizil, A., Crew, G. & Ksikes, A. (2004), Ensemble selection from libraries of models, *in* 'In Proceedings of the 21st International Conference on Machine Learning', ACM Press, pp. 137–144.

Chaves, J., Curry, C., Grossman, R. L., Locke, D. & Vejcik, S. (2006), 'Augustus: The design and architecture of a pmml-based scoring engine 1'.

Chitra, A. & Uma, S. (2010), 'An ensemble model of multiple classifiers for time series prediction', *International Journal of Computer Theory and Engineering, 2* (3), 454–458.

Choi, S.-S., Cha, S.-H. & Tappert, C. C. (2010), 'A survey of binary similarity and distance measures', *Journal of Systemics, Cybernetics and Informatics, 8* pp. 43–48.

Demetra (2012), 'Demetra project official web site', http://www.demetra-tox.net/.

DGI (2010), 'Data governance institute', http://www. datagovernance. com/ adg_data_governance_definition.html.

Dietterich, T. G. (2000), Ensemble methods in machine learning, *in* J. Kittler & F. Roli, eds, 'First International Workshop on Multiple Classifier Systems', Lecture Notes in Computer Science, Springer-Verlag, New York, NY, USA, pp. 1–15.

DMG (2012), 'Data mining group rfc documents', http://www.dmg.org/rfc/.

Fawcett, T. (2004), Roc graphs: Notes and practical considerations for researchers, hp laboratories, http://http://binf.gmu.edu/mmasso/roc101.pdf, Technical report.

Fayyad, U., Piatetsky-shapiro, G. & Smyth, P. (1996), 'From data mining to knowledge discovery in databases', *AI Magazine, 17* pp. 37–54.

Freitas, C. O. A., De Carvalho, J. a. M., Oliveira, Jr., J. J., Aires, S. B. K. & Sabourin, R. (2007), Confusion matrix disagreement for multiple classifiers, *in* 'Proceedings of the Congress on pattern recognition 12th Iberoamerican conference on Progress in pattern recognition, image analysis and applications', CIARP'07, Springer-Verlag, Berlin, Heidelberg, pp. 387–396.

Freund, Y. & Schapire, R. E. (1996), 'Experiments with a new boosting algorithm', pp. 148–156.

Fu, X., Wojak, A., Neagu, D., Ridley, M. & Travis, K. (2011), 'Data governance in predictive toxicology: A review.', *J Cheminform, 3* (1), 24.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H. & Herrera, F. (2011), 'An overview of ensemble methods for binary classifiers in multi class problems: Experimental study on one-vs-one and one-vs-all schemes', *Pattern Recogn., 44* pp. 1761–1776.

Ghosh, K., Ng, Y. S. & Srinivasan, R. (2011), 'Evaluation of decision fusion strategies for effective collaboration among heterogeneous fault diagnostic methods', *Computers and Chemical Engineering, 35* (2), 342–355.

GHS (2012), 'Global harmonisation system', http://www.hse.gov.uk/ghs/index.htm.

Gorea, D. (2008), 'Dynamically integrating knowledge in applications. an online scoring engine architecture', *Advances in Electrical and Computer Engineering, 8* pp. 44–49.

Hamming, R. W. (1950), 'Error detecting and error correcting codes', *Bell System Technical Journal, 29* (2), 147–160.

Hashemi, S., Yang, Y., Mirzamomen, Z. & Kangavari, M. (2009), 'Adapted one-versus-all decision trees for data stream classification', *IEEE Transactions on Knowledge and Data Engineering, 21* pp. 624 – 637.

IBM (2012), 'Ibm- information governance solutions', http://www-01.ibm.com/software/tivoli/governance/servicemanagement/data-governance.html.

Khakabimamaghani, S., Barzinpour, F. & Gholamian, M. R. (2010), 'A high diversity hybrid ensemble of classifiers', *2nd International Conference on Software Engineering and Data Mining (SEDM)* pp. 461 – 466.

Khatri, V. & Brown, C. V. (2010), 'Designing data governance', *Communication of the ACM, 53* pp. 148–152.

Khoussainov, R., Heß, A. & Kushmerick, N. (2005), Ensembles of biased classifiers, *in* 'Proceedings of the 22nd international conference on Machine learning', ICML '05, pp. 425–432.

Kohavi, R. & Provost, F. (1998), 'Glossary of terms. editorial for the special issue on applications of machine learning and the knowledge discovery process', *Machine Learning, 30* pp. 271–274.

Kuncheva, L. I. (2004), *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience.

Kuncheva, L. I. (2005), *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience.

Kuncheva, L. I. & Whitaker, C. J. (2003), 'Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy', *Machine Learning, 51* pp. 181–207.

Leadscope (2012), 'Leadscope - chemoinformatics platform for drug discovery', http://www.leadscope.com/toxml/.

Lesot, M.-J., Rifqi, M. & Benhadda, H. (2009), 'Similarity measures for binary and numerical data: a survey', *International Journal of Knowledge Engineering and Soft Data Paradigms, 1* pp. 63–84.

Liu, B. & Tuzhilin, A. (2008), 'Managing large collections of data mining models', *Communication of the ACM, 51* pp. 85–89.

Liu, Y. & Zheng, Y. F. (2005), One-against-all multi-class svm classification using reliability measures, *in* 'International Joint Conference on Neural Networks', IEEE Xplore, pp. 1–4.

Luukka, P. (2011), 'Feature selection using fuzzy entropy measures with similarity classifier', *Expert Syst. Appl., 38* pp. 4600–4607.

Makhtar, M., Neagu, D. & Ridley, M. (2011*a*), Binary classification models comparison : on the similarity of datasets and confusion matrix for predictive toxicology applications, *in* 'Proceedings of the Second international conference on Information technology in bio- and medical informatics', ITBAM'11, Springer-Verlag, Berlin, Heidelberg, pp. 108–122.

Makhtar, M., Neagu, D. & Ridley, M. J. (2010), Predictive model representation and comparison: Towards data and predictive models governance, *in* 'In Proceedings of the 10th Annual Workshop on Computational Intelligence (UKCI2010)', IEEE Xplore, pp. 1–6.

Makhtar, M., Neagu, D. & Ridley, M. J. (2011*b*), Comparing multi-class classifiers: On the similarity of confusion matrices for predictive toxicology applications., *in* 'IDEAL'11', pp. 252–261.

Makhtar, M., Yang, L., Neagu, D. & Ridley, M. J. (2012), Optimisation of classifier ensemble for predictive toxicology application, *in* 'In Proceedings of the 14th International Conference on Modelling and Simulation (UKSim2012)', IEEE, pp. 1–6.

Masisi, L. M., Nelwamondo, F. V. & Marwala, T. (2008), 'The effect of structural diversity of an ensemble of classifiers on classification accuracy', *Computing Research Repository, abs/0804.4741* .

Mehmood, Y., Ishtiaq, M., Tariq, M. & Arfan Jaffar, M. (2010), Classifier ensemble optimization for gender classification using genetic algorithm, *in* 'International Conference on Information and Emerging Technologies (ICIET)', IEEE Xplore, pp. 1–5.

Musehane, R., Netshiongolwe, F., Nelwamondo, F. V., Masisi, L. & Marwala, T. (2008), 'Relationship between structural diversity and performance of multiple classifiers for decision', *Computing Research Repository, abs/0810.3* pp. 109–114.

Nabiha, A., Nadir, F. & Mokhtar, S. (2011), 'Progressive algorithm for classifier ensemble construction based on diversity: Application to the arabic handwritten recognition', *The 2nd International Conference on Information and Communication Systems* pp. 1–6.

Neagu, D., Craciun, M., Chaudhry, Q. & Price, N. (2007), 'Knowledge representation for versatile hybrid intelligent processing applied in predictive toxicology', *Life Science Data Mining* pp. 213–238.

Neagu, D., Craciun, M. V., Stroia, S. A. & Bumbaru, S. (2005), 'Hybrid intelligent systems for predictive toxicology - a distributed ap-

proach', *5th International Conference on Intelligent Systems Design and Applications* pp. 26–31.

Neagu, D., Guo, G., Trundle, P. & Cronin, M. (2007), 'A comparative study of machine learning algorithms applied to predictive toxicology data mining', *Alternatives to Laboratory Animals ATLA, 35* pp. 25–32.

Parvin, H., Alizadeh, H. & Minaei-Bidgoli, B. (2009), 'A new method for constructing classifier ensembles.', *International Journal of Digital Content Technology and its Applications* pp. 62–66.

Polikar, R. (2006), 'Ensemble based systems in decision making', *Ieee Circuits And Systems Magazine, 6* (3), 21–45.

Prasanna, S. R. M., Yegnanarayana, B., Pinto, J. P. & Hermansky, H. (2007), Analysis of confusion matrix to combine evidence for phoneme recognition, Idiap-RR Idiap-RR-27-2007, IDIAP. Submitted for publication.

Rokach, L. (2009), 'Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography', *Comput. Stat. Data Anal., 53* (12), 4046–4072.

Santos, E. M. D. & Sabourin, R. (2011), Classifier ensembles optimization guided by population oracle, *in* 'IEEE Congress on Evolutionary Computation (CEC)', IEEE, pp. 693–698.

Schapire, R. E. & Freund, Y. (1998), 'Boosting the margin: a new explanation for the effectiveness of voting methods', *The Annals of Statistics, 26* pp. 322–330.

Sequeira, K. & Zaki, M. J. (2007), Exploring similarities across high-dimensional datasets, *in* D. Taniar, ed., 'Research and Trends in Data Mining Technologies and Applications', Idea Group, Inc., chapter 3, pp. 53–85.

Sewell, M. (2011), 'Ensemble learning', http://machine-learning.martinsewell.com/ ensembles/ ensemble-learning.pdf.

Sirlantzis, K., Hoque, S. & Fairhurst, M. C. (2008), 'Diversity in multiple classifier ensembles based on binary feature quantisation with application to face recognition', *Appl. Soft Comput., 8* (1), 437–445.

Sun, F.-S. (2005), Error prediction for multi-classification, *in* 'Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel Distributed Computing', IEEE Xplore, pp. 140 – 143.

Tan, P.-N., Steinbach, M. & Kumar, V. (2005), *Introduction to Data Mining, (First Edition)*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Todeschini, R., Consonni, V. & Pavan, M. (2004), 'A distance measure between models: a tool for similarity/diversity analysis of model populations', *Chemometrics and Intelligent Laboratory Systems, 70* pp. 55–61.

Trundle, P. (2008), Hybrid Intelligent Systems Applied to Predict Pesticides Toxicity - a Data Integration Approach, PhD thesis in Computer Science, School of Computing Informatics and Media,

University of Bradford, Richmond Road, Bradford, West Yorkshire, BD7 1DP, UK.

UCI (2012), 'University of california, irvine, machine learning repository', http://archive.ics.uci.edu/ml/.

Wang, W. (2008), Some fundamental issues in ensemble methods, *in* 'Proceedings of the International Joint Conference on Neural Networks (IJCNN 2008)', IEEE, pp. 2243–2250.

Wang, W. (2010), Heterogeneous bayesian ensembles for classifying spam emails, *in* 'Proceedings of Neural Networks (IJCNN), The 2010 International Joint Conference on', IEEE, pp. 1–8.

Wang, W., Partridge, D. & Etherington, J. (2001), Hybrid ensembles and coincident-failure diversity., *in* 'International Joint Conference on Neural Networks', IEEE, pp. 2376–2381.

Wende, K. (2007), 'A model for data governance - organising accountabilities for data quality management'.

Witten., I., Frank, E. & Hall, M. (2011), *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann series in data management systems, Elsevier Science & Technology.

Witten, I. H., Frank, E., Trigg, L., Hall, M., Holmes, G. & Cunningham, S. J. (1999), Weka: Practical machine learning tools and techniques with java implementations, *in* 'In Proceedings of ICONIP/ANZIIS/ANNES'99 Workshop on Emerging Knowledge Engineering and Connectionist-Based Information Systems', pp. 192–196.

Woloszynski, T. & Kurzynski, M. (2011), 'A probabilistic model of classifier competence for dynamic ensemble selection', *Pattern Recogn., 44* (10-11), 2656–2668.

Xiaoyan, M., Jiangfeng, L. & Watta, P. Hassoun, M. (2009), 'Weighted voting-based ensemble classifiers with application to human face recognition and voice recognition', *International Joint Conference on Neural Networks (IJCNN)* pp. 2168–2171.

XQuery (2012), 'Xquery 1.0: An xml query language (second edition)', http://www.w3.org/TR/xquery/.

Appendix A

DTD and PTML Model

A.1 DTD for PTML Models

Following is an example of PTML model

A.2 An Example of PTML Model

Following is an example of PTML model.

Table A.1: The DTD for PTML Document Structure

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!-- definition of modelDescription -->
<xs:element name="Classifier">
    < xs:element  ref = "Name" type="xs:string"/>
    < xs:element  ref = "Date" type="xs:Date"/>
    < xs:element  ref = "Author" type="xs:string"/>
    < xs:element  ref = "Description" type="xs:string"/>
    < xs:element  ref = "WekaModel" type="xs:string"/>
</xs:element>

<!-- definition of modelParameter -->
<xs:element name="Classifier">
    < xs:element  ref = "ClassifierName" type="xs:string"/>
    < xs:element  ref = "Fold" type="xs:Integer"/>
    < xs:element  ref = "Seed" type="xs:Integer"/>
</xs:element>

<!-- definition of modelAttributes -->
<xs:element name="DataSet">
    < xs:element  ref = "DataSet␣Name" type="xs:string"/>
    < xs:element  ref = "TotalNumberInstances"  type="xs:Integer"/>
    < xs:element  ref = "NumberOfAttributes" type="xs:Integer"/>
    < xs:element  ref = "FeatureSelectionAlgorithm" type="xs:string"/>
    < xs:element  ref = "FeatureSearchMethod" type="xs:string"/>
    < xs:element  ref = "NumberOfAttributesSelected" type="xs:Integer"/>
    < xs: Features >
        <xs:element ref="FeatureName" type="xs:string"/>
        <xs:element ref="Type" type="xs:string"/>
    </xs:sequence>
</xs:element>

<!-- definition of modelPerformance -->
<xs:element name="ClassificationModelPerformance">
    < xs:element  ref = "CorrectlyClassifiedInstances" type="xs:Integer"/>
    < xs:element  ref = "Accuracy" type="xs:Decimal"/>
    < xs:element  ref = "MeanAbsoluteError" type="xs:Decimal"/>
    < xs:element  ref = "RootMeanSquaredError" type="xs:Decimal"/>
    < xs:element  ref = "RelativeAbsoluteError" type="xs:Decimal"/>
</xs:element>

<!-- definition of classAttribute -->
<xs:element name="ClassName">
    < xs:element  ref = "TPRate" type="xs:Decimal"/>
    < xs:element  ref = "FPRate" type="xs:Decimal"/>
    < xs:element  ref = "FNRate" type="xs:Decimal"/>
    < xs:element  ref = "TNRate" type="xs:Decimal"/>
</xs:element>

<!-- definition of ConfusionMatrix -->
<xs:element name="Class">
    < xs:element  ref = "ClassName" type="xs:String"/>
    < xs:element  ref = "Value" type="xs:Integer" />
</xs:element>
```

Table A.2: The PTML Document Structure

```
<?xml version="1.0" encoding="UTF-8"?>
<PTML>

<modelDescription><Name>DM</Name>
<Date>25-12-2008</Date>
<Version>Ver1.1</Version>
<Author>Mokhairi</Author>
<Description>Testing Autogenerated Model From Weka</Description>
<wekaModel>wekaModel13.model</wekaModel>
</modelDescription>


<modelData>
<DataSetName>CFS_APC_Recon-(C)Mallard_Duck-Raw_Data.arff</DataSetName>
<LastUpdatedDate>10/01/2010</LastUpdatedDate>
<AttributeEvaluator>CfsSubsetEval</AttributeEvaluator>
<SearchingMethod>BestFirst</SearchingMethod>
<SplitType>100%</SplitType>
<TotalNumberInstances>60.0</TotalNumberInstances>
<NumberOfAttributes>6</NumberOfAttributes>

<DataSetAttributes>
<Attributes><Name>Del(Rho)NA4</Name><Type>Numeric</Type></Attributes>
<Attributes><Name>PIP6</Name><Type>Numeric</Type></Attributes>
<Attributes><Name>FPIP12</Name><Type>Numeric</Type></Attributes>
<Attributes><Name>Class</Name><Type>Nominal</Type></Attributes>
</DataSetAttributes>
</modelData>


<modelParameter>
<Classifier>weka.classifiers.lazy.IBk</Classifier>
<Fold>10</Fold>
<Seed>1</Seed>
</modelParameter>


<modelPerformance>
<CorrectlyClassifiedInstances>18.0</CorrectlyClassifiedInstances>
<PctCorrectlyClassifiedInstances>30.0</PctCorrectlyClassifiedInstances>
<IncorrectlyClassifiedInstances>42.0</IncorrectlyClassifiedInstances>
<PctIncorrectlyClassifiedInstances>70.0</PctIncorrectlyClassifiedInstances>
<Kappa>0.057</Kappa>
<MeanAbsoluteError>0.352</MeanAbsoluteError>
<RootMeanSquaredError>0.572</RootMeanSquaredError>
<RelativeAbsoluteError>94.515</RelativeAbsoluteError>
<RootRelativeSquaredError>132.55</RootRelativeSquaredError>
</modelPerformance>
```

```
<classAttribute><Name>Class</Name>
<Class>I</Class>
<Details><TPRate>0.824</TPRate>
<FPRate>0.116</FPRate>
<TNRate>0.884</TNRate>
<FNRate>0.176</FNRate>
<Precision>0.737</Precision>
<Recall>0.824</Recall>
<FMeasure>0.778</FMeasure>
<ROCArea>0.854</ROCArea>
</Details>
<Class>II</Class>
<Details><TPRate>0.0</TPRate>
<FPRate>0.341</FPRate>
<TNRate>0.659</TNRate>
<FNRate>1.0</FNRate>
<Precision>0.0</Precision>
<Recall>0.0</Recall>
<FMeasure>0.0</FMeasure>
<ROCArea>0.33</ROCArea>
</Details>
<Class>III</Class>
<Details><TPRate>0.188</TPRate>
<FPRate>0.295</FPRate>
<TNRate>0.705</TNRate>
<FNRate>0.812</FNRate>
<Precision>0.188</Precision>
<Recall>0.188</Recall>
<FMeasure>0.188</FMeasure>
<ROCArea>0.446</ROCArea>
</Details>
<Class>IV</Class>
<Details><TPRate>0.091</TPRate>
<FPRate>0.184</FPRate>
<TNRate>0.816</TNRate>
<FNRate>0.909</FNRate>
<Precision>0.1</Precision>
<Recall>0.091</Recall>
<FMeasure>0.095</FMeasure>
<ROCArea>0.454</ROCArea>
</Details>
</classAttribute>
<ConfusionMatrix>
<Array>I        II      III     IV      </Array>
<Array>14       1       0       2       I       </Array>
<Array>3        0       9       4       II      </Array>
<Array>1        9       3       3       III     </Array>
<Array>1        5       4       1       IV      </Array>
</ConfusionMatrix>
</PTML>
```

Appendix B

Results

B.1 Results of PTML Model Similarity

Table B.1: Results of Model Similarity from Daphnia Data Set

Machine Learning	IBK						J48					
Feature Selection	None			CFS			None			CFS		
Split(%)	100	70	30	100	70	30	100	70	30	100	70	30
ModelID	31	32	33	34	35	36	181	182	183	184	185	186
31	1	0.84	0.61	0.5	0.5	0.49	1	0.83	0.62	0.46	0.45	0.5
32	0.84	1	0.44	0.48	0.49	0.49	0.84	1	0.45	0.48	0.47	0.48
33	0.61	0.44	1	0.46	0.45	0.45	0.61	0.44	0.99	0.42	0.41	0.46
34	0.5	0.48	0.46	1	0.84	0.64	0.49	0.48	0.47	0.96	0.8	0.65
35	0.5	0.49	0.45	0.84	1	0.5	0.5	0.49	0.46	0.81	0.96	0.5
36	0.49	0.49	0.45	0.64	0.5	1	0.5	0.49	0.46	0.62	0.46	0.99
181	1	0.84	0.61	0.49	0.5	0.5	1	0.84	0.62	0.47	0.46	0.49
182	0.83	1	0.44	0.48	0.49	0.49	0.84	1	0.45	0.48	0.47	0.48
183	0.62	0.45	0.99	0.47	0.46	0.46	0.62	0.45	1	0.43	0.42	0.47
184	0.46	0.48	0.42	0.96	0.81	0.62	0.47	0.48	0.43	1	0.84	0.61
185	0.45	0.47	0.41	0.8	0.96	0.46	0.46	0.47	0.42	0.84	1	0.45
186	0.5	0.48	0.46	0.65	0.5	0.99	0.49	0.48	0.47	0.61	0.45	1

Table B.2: Results of Model Similarity from Dietary Quail Data Set

Machine Learning	IBK						J48					
Feature Selection	None			CFS			None			CFS		
Split (%)	100	70	30	100	70	30	100	70	30	100	70	30
Model	61	62	63	64	65	66	211	212	213	214	215	216
61	1	0.82	0.62	0.48	0.49	0.47	0.99	0.85	0.64	0.47	0.5	0.47
62	0.82	1	0.44	0.45	0.46	0.44	0.83	0.97	0.46	0.44	0.47	0.44
63	0.62	0.44	1	0.49	0.48	0.5	0.61	0.47	0.98	0.5	0.47	0.5
64	0.48	0.45	0.49	1	0.85	0.64	0.47	0.48	0.49	0.99	0.83	0.64
65	0.49	0.46	0.48	0.85	1	0.48	0.48	0.48	0.49	0.83	0.98	0.48
66	0.47	0.44	0.5	0.64	0.48	1	0.46	0.47	0.48	0.65	0.47	1
211	0.99	0.83	0.61	0.47	0.48	0.46	1	0.85	0.64	0.46	0.5	0.46
212	0.85	0.97	0.47	0.48	0.48	0.47	0.85	1	0.49	0.47	0.5	0.47
213	0.64	0.46	0.98	0.49	0.49	0.48	0.64	0.49	1	0.48	0.49	0.48
214	0.47	0.44	0.5	0.99	0.83	0.65	0.46	0.47	0.48	1	0.82	0.65
215	0.5	0.47	0.47	0.83	0.98	0.47	0.5	0.5	0.49	0.82	1	0.47
216	0.47	0.44	0.5	0.64	0.48	1	0.46	0.47	0.48	0.65	0.47	1

Table B.3: Results of Model Similarity from Oral Quail Data Set

Machine Learning	IBK						J48					
Feature Selection	None			CFS			None			CFS		
Split (%)	100	70	30	100	70	30	100	70	30	100	70	30
Model	91	92	93	94	95	96	241	242	243	244	245	246
91	1	0.83	0.62	0.47	0.47	0.36	0.98	0.84	0.64	0.49	0.44	0.44
92	0.83	1	0.45	0.46	0.49	0.34	0.82	0.97	0.48	0.49	0.46	0.42
93	0.62	0.45	1	0.49	0.44	0.39	0.63	0.48	0.97	0.46	0.41	0.47
94	0.47	0.46	0.49	1	0.79	0.53	0.49	0.48	0.48	0.96	0.77	0.62
95	0.47	0.49	0.44	0.79	1	0.33	0.45	0.46	0.46	0.83	0.97	0.41
96	0.36	0.34	0.39	0.53	0.33	1	0.37	0.37	0.36	0.5	0.3	0.92
241	0.98	0.82	0.63	0.49	0.45	0.37	1	0.84	0.64	0.47	0.43	0.46
242	0.84	0.97	0.48	0.48	0.46	0.37	0.84	1	0.5	0.48	0.43	0.45
243	0.64	0.48	0.97	0.48	0.46	0.36	0.64	0.5	1	0.49	0.44	0.44
244	0.49	0.49	0.46	0.96	0.83	0.5	0.47	0.48	0.49	1	0.8	0.58
245	0.44	0.46	0.41	0.77	0.97	0.3	0.43	0.43	0.44	0.8	1	0.38
246	0.44	0.42	0.47	0.62	0.41	0.92	0.46	0.45	0.44	0.58	0.38	1

Table B.4: Results of Model Similarity from Trout Data Set

Machine Learning	IBK						J48					
Feature Selection	None			CFS			None			CFS		
Split (%)	100	70	30	100	70	30	100	70	30	100	70	30
Model	121	122	123	124	125	126	271	272	273	274	275	276
121	1	0.84	0.62	0.49	0.49	0.47	0.98	0.84	0.63	0.47	0.48	0.49
122	0.84	1	0.46	0.48	0.5	0.47	0.84	1	0.47	0.46	0.47	0.48
123	0.62	0.46	1	0.47	0.46	0.49	0.6	0.46	0.99	0.5	0.48	0.48
124	0.49	0.48	0.47	1	0.84	0.63	0.48	0.49	0.49	0.97	0.84	0.65
125	0.49	0.5	0.46	0.84	1	0.47	0.49	0.5	0.48	0.81	0.98	0.48
126	0.47	0.47	0.49	0.63	0.47	1	0.46	0.47	0.49	0.64	0.49	0.99
271	0.98	0.84	0.6	0.48	0.49	0.46	1	0.84	0.61	0.45	0.46	0.47
272	0.84	1	0.46	0.49	0.5	0.47	0.84	1	0.48	0.46	0.48	0.48
273	0.63	0.47	0.99	0.49	0.48	0.49	0.61	0.48	1	0.49	0.5	0.49
274	0.47	0.46	0.5	0.97	0.81	0.64	0.45	0.46	0.49	1	0.84	0.63
275	0.48	0.47	0.48	0.84	0.98	0.49	0.46	0.48	0.5	0.84	1	0.49
276	0.49	0.48	0.48	0.65	0.48	0.99	0.47	0.48	0.49	0.63	0.49	1

B.2 The Study of $OCEM$ to Demetra Data Sets Using Training Set (70%) and Testing Set (30%)

The study were conducted for training set and testing set for all Demetra data set using disagreement measure as a diversity measure. The results were as follows:

Table B.5: FNR for Different Ensemble.

Data Set	Bagging	AdaBoost	Stacking	Bayes	OCEM
Bee	Acc = 0.90 FNR = 0.07 FPR = 0.25	Acc = 0.90 FNR = 0.03 FPR = 0.50	Acc = 0.87 FNR = 0.00 FPR = 1.00	Acc = 0.78 FNR = 0.17 FPR = 0.50	Acc = 0.93 FNR = 0.00 FPR = 0.50
Daphnia	Acc = 0.62 FNR = 0.10 FPR = 0.30	Acc = 0.83 FNR = 0.06 FPR = 0.33	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.83 FNR = 0.26 FPR = 0.00	Acc = 0.62 FNR = 0.00 FPR = 1.00
Dietary Quail	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00
Oral Quail	Acc = 0.60 FNR = 0.38 FPR = 0.41	Acc = 0.57 FNR = 0.55 FPR = 0.29	Acc = 0.48 FNR = 1.00 FPR = 0.00	Acc = 0.65 FNR = 0.55 FPR = 0.11	Acc = 0.51 FNR = 0.05 FPR = 0.94
Trout	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00

Table B.6: FPR for Different Ensemble.

Data Set	Bagging	AdaBoost	Stacking	Bayes	OCEM
Bee	Acc = 0.90 FNR = 0.07 FPR = 0.25	Acc = 0.90 FNR = 0.03 FPR = 0.50	Acc = 0.87 FNR = 0.00 FPR = 1.00	Acc = 0.78 FNR = 0.17 FPR = 0.50	Acc = 0.90 FNR = 0.07 FPR = 0.25
Daphnia	Acc = 0.62 FNR = 0.10 FPR = 0.30	Acc = 0.83 FNR = 0.06 FPR = 0.33	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.83 FNR = 0.26 FPR = 0.00	Acc = 0.63 FNR = 0.58 FPR = 0.00
Dietary Quail	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.97 FNR = 0.00 FPR = 0.07
Oral Quail	Acc = 0.60 FNR = 0.38 FPR = 0.41	Acc = 0.57 FNR = 0.55 FPR = 0.29	Acc = 0.48 FNR = 1.00 FPR = 0.00	Acc = 0.65 FNR = 0.55 FPR = 0.11	Acc = 0.48 FNR = 1.00 FPR = 0.00
Trout	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.45 FPR = 0.00

Table B.7: Acc for Different Ensemble.

Data Set	Bagging	AdaBoost	Stacking	Bayes	OCEM
Bee	Acc = 0.90 FNR = 0.07 FPR = 0.25	Acc = 0.90 FNR = 0.03 FPR = 0.50	Acc = 0.87 FNR = 0.00 FPR = 1.00	Acc = 0.78 FNR = 0.17 FPR = 0.50	Acc = 0.96 FNR = 0.00 FPR = 0.25
Daphnia	Acc = 0.62 FNR = 0.10 FPR = 0.30	Acc = 0.83 FNR = 0.06 FPR = 0.33	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.83 FNR = 0.26 FPR = 0.00	Acc = 0.88 FNR = 0.14 FPR = 0.06
Dietary Quail	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.62 FNR = 0.00 FPR = 1.00	Acc = 0.97 FNR = 0.00 FPR = 0.07
Oral Quail	Acc = 0.60 FNR = 0.38 FPR = 0.41	Acc = 0.57 FNR = 0.55 FPR = 0.29	Acc = 0.48 FNR = 1.00 FPR = 0.00	Acc = 0.65 FNR = 0.55 FPR = 0.11	Acc = 0.82 FNR = 0.16 FPR = 0.17
Trout	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.82 FNR = 0.00 FPR = 1.00	Acc = 0.91 FNR = 0.04 FPR = 0.26

Table B.8: Acc, FNR, FPR of $OCEM$ Given Different Weight of w

Data Set	$w1$=0.5, $w2$=0.5 $w3$=0.0	$w1$=0.0, $w2$=0.5 $w3$=0.5	$w1$=0.3, $w2$=0.5 $w3$=0.2	$w1$=0.6, $w2$=0.2 $w3$=0.2
Bee	Acc = 0.96 FNR = 0.00 FPR = 0.25	Acc = 0.90 FNR = 0.07 FPR = 0.25	Acc = 0.90 FNR = 0.07 FPR = 0.25	Acc = 0.90 FNR = 0.07 FPR = 0.25
Daphnia	Acc = 0.87 FNR = 0.04 FPR = 0.26	Acc = 0.88 FNR = 0.14 FPR = 0.06	Acc = 0.88 FNR = 0.10 FPR = 0.13	Acc = 0.88 FNR = 0.14 FPR = 0.06
Dietary Quail	Acc = 0.97 FNR = 0.00 FPR = 0.07	Acc = 0.97 FNR = 0.00 FPR = 0.07	Acc = 0.97 FNR = 0.00 FPR = 0.07	Acc = 0.97 FNR = 0.00 FPR = 0.07
Oral Quail	Acc = 0.80 FNR = 0.11 FPR = 0.29	Acc = 0.82 FNR = 0.16 FPR = 0.17	Acc = 0.82 FNR = 0.16 FPR = 0.17	Acc = 0.82 FNR = 0.16 FPR = 0.17
Trout	Acc = 0.91 FNR = 0.00 FPR = 0.46	Acc = 0.91 FNR = 0.04 FPR = 0.26	Acc = 0.91 FNR = 0.04 FPR = 0.26	Acc = 0.91 FNR = 0.04 FPR = 0.26