Combining classifiers for credit risk prediction

Article *in* Journal of Systems Science and Systems Engineering · September 2009

DOI: 10.1007/s11518-009-5109-y

CITATIONS
12

READS
426

1 author:

Bhekisipho Twala
Tshwane University of Technology
158 PUBLICATIONS **2,181** CITATIONS

SEE PROFILE

J Syst Sci Syst Eng (Sep 2009) 18(3): 292-311 ISSN: 1004-3756 (Paper) 1861-9576 (Online)
DOI: 10.1007/s11518-009-5109-y CN11-2983/N

COMBINING CLASSIFIERS FOR CREDIT RISK PREDICTION*

Bhekisipho TWALA

CSIR, Modelling and Digital Sciences Unit, P.O. Box 395, Pretoria 0001, South Africa

btwala1@csir.co.za (✉)

Abstract

Credit risk prediction models seek to predict quality factors such as whether an individual will default (bad applicant) on a loan or not (good applicant). This can be treated as a kind of machine learning (ML) problem. Recently, the use of ML algorithms has proven to be of great practical value in solving a variety of risk problems including credit risk prediction. One of the most active areas of recent research in ML has been the use of ensemble (combining) classifiers. Research indicates that ensemble individual classifiers lead to a significant improvement in classification performance by having them vote for the most popular class. This paper explores the predicted behaviour of five classifiers for different types of noise in terms of credit risk prediction accuracy, and how could such accuracy be improved by using pairs of classifier ensembles. Benchmarking results on five credit datasets and comparison with the performance of each individual classifier on predictive accuracy at various attribute noise levels are presented. The experimental evaluation shows that the ensemble of classifiers technique has the potential to improve prediction accuracy.

Keywords: Supervised learning, statistical pattern recognition, ensemble, credit risk, prediction

1. Introduction

With the growth in financial services, there have been mounting losses from delinquent loans. For example, Manufacturer's Hanover's $3.5 Billion commercial property portfolio was burdened with $385 Million in non-performing loans (Rosenberg & Gleit 1994).

According to BIS (BIS 2004), credit risk is most simply defined as the potential that a bank borrower or counterparty will fail to meet its obligations in accordance with agreed terms. Over the last decade, a number of the world's largest banks have developed sophisticated systems in an attempt to model the credit risk arising from important aspects of their business lines. Such models are intended to aid banks in quantifying, aggregating and managing risk across geographical and product lines. The outputs of these models also play increasingly major roles in banks' risk management and performance measurement processes, including performance-based compensation, customer profitability analysis, risk-based pricing and, to a lesser (but growing) extend, active portfolio

* The work was funded by the CSIR under project MDSARR1.

management and capital structure decisions. Thus, applied finance researchers and practitioners remain concerned with prediction accuracy when building credit modelling systems.

Most techniques for predicting attributes of a credit risk system or credit data require past data from which models will be constructed and validated. One of the major problems for applying ML algorithms in credit risk prediction is the unavailability, scarcity and incompleteness of credit data, i.e., data for training the model. Most of the financial institutions do not share their data with other organizations so that a useful database with a great amount of data cannot be formed. In addition, surveys for collecting credit data are usually small but difficult and expensive to conduct.

Another important and common issue faced by researchers who use financial or credit datasets is the occurrence of noise in the data. Even if part of a well thought out measurement programme, credit datasets can be noisy for a number of reasons. These include inaccurate or non reporting of information (without a direct benefit, a project manager or developer might see data collection as an overhead they can ill afford, for example), or, where data from a number of different types of customers or from a number of banks are combined, certain fields may be blank because they are not collectable for all customers. Often data is collected either with no specific purpose in mind (i.e. it is collected because it might be useful in future) or the analysis being carried out has a different goal than that for which the data was originally collected. In research datasets, e.g. experiments on human subjects to assess the effectiveness of a new credit risk technique, say, dropout or failure to follow instructions may lead to noise in data. The relevance of this issue is strictly proportional to the dimensionality of the collected data.

Economics and finance researchers have become increasingly aware of the problems and biases which can be caused by noisy data. Moreover, many credit datasets tend to be small with many different attributes – credit risk datasets grow slowly, for example – and the numbers of available human subjects limit the size of many experimental datasets. Thus, we can ill afford to reduce our sample size further by eliminating all instances with noise. Because of the expense and difficulty of performing extensive controlled experiments on credit, case studies are often resorted to.

Various ML and statistical pattern recognition (SPR) techniques have been used in finance to predict credit risk. Reviews of the use of ML in report that ML in finance is a mature technique based on widely-available tools using well understood algorithms. A central concern of these applications is the need to increase the scoring accuracy of the credit decision. An improvement in accuracy or even a fraction of a percent translates into a significant future savings. In recent years, there has been an explosion of papers in the ML and statistics communities discussing how to combine models or model predictions. Many works in both the ML and statistical pattern recognition communities have shown that combining (ensemble) individual classifiers is an effective technique for improving classification accuracy.

An ensemble is generated by training multiple learners for the same task and then

combining their predictions. There are different ways in which ensembles can be generated, and the resulting output combined to classify new instances. The popular approaches to creating ensembles include changing the instances used for training through techniques such as bagging (Breiman 1996, Bauer & Kohavi 1999), boosting (Freund & Schapire 1996), stacking (Wolpert 1992), changing the features used in training (Ho 1995), introducing randomness in the classifier itself (Dietterich 2000).

Bagging constructs a set of classifiers by sub-sampling the training examples to generate different hypotheses. After the different hypotheses are generated, they are combined by a voting mechanism. Boosting also uses the voting system to combine the classifiers. But, instead of sub-sampling the training examples, it generates the hypotheses sequentially. In each repetition, a new classifier is generated which focus in those instances that were handled incorrectly by the previous classifier. This is achieved by giving a weight to each instance in the training examples and adjusting these weights according to its importance after every iteration. Both, bagging and boosting use classifiers generated by the same base-learning algorithm and obtained from the same data. Finally, stacking can combine classifiers obtained from different learning algorithms using a high level classifier – the met-classifier – to combine the lower level models. This is based on the fact that different classifiers are obtained from the same data and different learning algorithms use different biases to search the hypothesis space. This approach expects that the meta-classifier will be able to learn how to decide between the predictions provided by the base classifiers, in order to get accuracies better than any of them, much in the same way as a committee of experts. For purposes of this paper we follow the bagging approach.

Robustness has a twofold meaning in terms of dealing with noise using supervised classifiers. The toleration of noise in training data is one, and the toleration of noise data in test data is the other. Data presented to a given classifier, during either training or testing phase, may be noise in one or more ways. For example, attribute values and/or class labels could be noisy. For purposes of this paper we are assuming that the class labels are not noisy, i.e., only attribute values are considered as containing noise. Although the problem of noisy data has been treated adequately in various real world datasets, there are rather few published works or empirical studies concerning the task of assessing learning and classification accuracy of supervised ML algorithms given noisy data (Aha 1992). In addition, to the best of our knowledge, few researchers have carried out such an extensive study on the effect of ensemble classifiers on credit risk predictive accuracy. In this paper, we first study the robustness of five classifiers on the predictive accuracy given noisy data. Then, we propose twenty ensemble classifiers from a combination of five classifiers. Each ensemble has two classifiers as elements. The proposed method utilizes probability patterns of classification results.

There are various reasons why the five classifiers were utilized to investigate the problem considered in this paper. Despite being one of the well known algorithms from the ML and SPR communities, they are a reasonable mix of non-parametric and parametric and they work

for almost all classification problems. In addition, they can achieve good performance on many tasks.

We conduct controlled experiments capable of producing statistically significant conclusions about a relative performance of twenty ensemble classifiers given noisy data and the strategy of reliance on a single model. To further these purposes, we pose the following research questions which also represent the main contributions of this work:

1. What is the impact of poor data quality (i.e., the presence of noise in a datasets) on the analysis of the effectiveness of a five classifiers and their ensembles for credit risk prediction?
2. Which single classifier and ensemble of classifiers provide the best performance, and are the ensembles more accurate in predicting credit risk than a single classifier?
3. Is the type of attribute noise either in the training set or test set, or the proportion of noise (5%, 15%, 30%, 50%) an important factor in the credit risk prediction process?

In terms of the impact of poor data quality, our study is unique and unprecedented. From this study we conclude that the quality of the underlying data cannot be ignored when examining the efficacy of classifier procedures. Our results further show that data quality plays a major role in the effectiveness of a classifier and the ensemble of classifiers.

The following section briefly gives details of the five classifiers used in this paper. Section 3 reviews some related work to the problem of credit risk prediction in the economics and finance areas. Section 4 empirically explores the robustness and accuracy of five classifiers to five credit datasets with artificially simulated attribute noise. This section also presents empirical results from the application of the proposed ensemble procedure. We close with conclusions and directions for future research.

2. Classifiers

The most important feature of a problem domain, as far as the application of ML and SPR algorithms are concerned, is the form that the data takes and the quality of the data available. Our main focus will be on the latter. The problem of handling noise has been the focus of much attention in the ML and SPR communities. Specific ML and SPR techniques that are known to be robust enough to cope with noisy data, and to discover laws in it that may not always hold but are useful for the problem at hand, are now going to be described. These classifiers have also been used as credit scoring models (Hand & Henley 1997). Three supervised learning (artificial neural network, decision tress and naïve Bayes classifier) and two statistical (*k*-nearest neighbour and logistic discrimination) techniques are examined in the presence of increasing level of artificial noise. First, the supervised learning techniques are described and a brief description of SPR techniques is briefly introduced.

2.1 Supervised Learning Techniques

2.1.1 Artificial Neural Networks

Artificial Neural Networks (ANNs), usually nonparametric approaches, are represented by connections between a very large number of simple computing processors or elements (neurons), have been used for a variety of

classification and regression problems. These include pattern and speech recognition (Ripley 1992), credit risk prediction (Davis et al. 1992, Tam et al. 1992, Rosenberg & Gleit 1994, Lacher et al. 1995, Piramuthu 1999, West 2000, Baesens et al. 2003), software risk prediction (Neumann 2002) and so on. There are many types of ANNs, but for the purposes of this study we shall concentrate on single unit perceptrons and multi layer perceptrons also known as "backpropagation networks".

The backpropagation learning algorithm performs a hill-climbing search procedure on the weight space described above or a (noisy or stochastic) gradient descent numerical method whereby an error function is minimised. At each iteration, each weight is adjusted proportionally to its effect on the error. One cycle through the training set and on each example changes each weight proportionally to its effect on lowering the error. One may compute the error gradient using the chain rule and the information propagates backwards through the network through the interconnections, which accounts for the procedure's name.

There are two stages associated with the backpropagation method: training and classification. The ANN is trained by supplying it with a large number of numerical observations or the patterns to be learned (input data pattern) whose corresponding classifications (target values or desired output) are known. During training, the final sum-of-squares error over the validation data for the network is calculated. The selection of the optimum number of hidden nodes is made on the basis of this error value. The question of how to choose the structure of the network is beyond the scope of this paper and is a current research issue in neural networks. Once the network is trained, a new object is classified by sending its attribute values to the input nodes of the network, applying the weights to those values, and computing the values of the output units or output unit activations. The assigned class is that with the largest output unit activation.

2.1.2 Decision Trees

A decision tree (Breiman et al. 1984, Quinlan 1993) is a model of the data that encodes the distribution of the class label in terms of the predictor attributes; it is a directed, acyclic graph in a form of a tree. The root of the decision tree (DT) does not have any incoming edges. Every other node has exactly one incoming edge and zero or more outgoing edges. If a node n has no outgoing edges we call n a leaf node, otherwise we call n an internal node. Each leaf node is labelled with one class label; each internal node is labelled with one predictor attribute called the splitting attribute. Each edge e originating from an internal node n has a predicate q associated with it where q involves only the splitting attribute of n.

There are two ways to control the size of the tree. For a bottom-up pruning strategy, a very deep tree is constructed, and this tree is cut back to avoid over-fitting the training data. For top down pruning, a stopping criterion is calculated during tree growth to inhibit further construction of parts of the tree when appropriate. In this paper we follow the bottom up strategy.

A DT can be used to predict the values of the target or class attribute based on the predictor attributes. To determine the predicted value of an unknown instance, you begin at the root node of

the tree. Then decide whether to go into the left or right child node based on the value of the splitting attribute. You continue this process using the splitting attribute for successive child nodes until you reach a terminal or leaf node. The value of the target attribute shown in the leaf node is the predicted value of the target attribute.

One property that sets DTs apart from all other classifiers is their invariance to monotone transformations of the predictor variables. For example, replacing any subset of the predictor variables $\{x_i\}$ by (possible different) arbitrary strictly monotone functions of them $\{x_j \leftarrow m_j(x_j)\}$, gives rise to the same tree model. Thus, there is no issue of having to experiment with different possible transformations $m_j(x_j)$ for each individual predictor x_j to try to find the best ones. This invariance provides immunity to the presence of extreme values ("outliers") in the predictor variable space.

2.1.3 Naïve Bayes Classifier

The naïve Bayes classifier (NBC) is perhaps the simplest and most widely studied probabilistic learning method. It learns from the training data, the conditional probability of each attribute A_i, given the class label C (Ripley 1992). The strong major assumption is that all attributes A_i are independent given the value of the class C. Classification is therefore done applying Bayes rule to compute the probability of C given $A_1,\ldots,A_n$ and then predicting the class with the highest posterior probability. The probability of a class value C_i given an instance $X=\{A_1,\ldots, A_n\}$ for n observations is given by:

$$\begin{aligned} p(C_i \mid X) &= p(X \mid C_i).p(C_i) / p(X) \\ &= p(A_1,...,A_n \mid C_i).p(C_i) \\ &= \prod_{j=1}^{n} p(A_j \mid C_i).p(C_i) \end{aligned}$$

The assumption of conditional independence of a collection of random variables is very important for the above result. Otherwise, it would be impossible to estimate all the parameters without such an assumption. This is a fairly strong assumption that is often not applicable. However, bias in estimating probabilities may not make a difference in practice – it is the order of the probabilities, not the exact values that determine the probabilities. When the strong attribute independence assumption is violated, the performance of the NBC might be poor. Kononenko (1991) has developed a technique to improve the performance of the NBC. However, more work still has to be done on this independence assumption violation.

2.2 Statistical Pattern Recognition Techniques

2.2.1 *k*-Nearest Neighbour Classifier

One of the most venerable algorithms in statistical pattern recognition is the nearest neighbour. *k*-Nearest Neighbour (*k*-NN) can also be considered a supervised learning algorithm where the result of a new instance query is classified on majority of *k*-nearest neighbour category. Of late, such an algorithm has become popular in credit scoring (Henley 1995, Henley & Hand 1996, Hand & Vinciotti 2003, Islam et al. 2007). *k*-NN methods are sometimes referred to as memory-based reasoning or instance-based

learning or case-based learning techniques and have been used for classification tasks. They essentially work by assigning to an unclassified sample point the classification of the nearest of a set of previously classified points. The entire training set is stored in the memory.

To classify a new instance, the Euclidean distance (possibly weighted) is computed between the instance and each stored training instance and the new instance is assigned the class of the nearest neighbouring instance. More generally, these *k*-NNs are computed, and the new instance is assigned the class that is most frequent among the *k* neighbours. Instance-based learner's have three defining general characteristics: a similarity function (how close together the two instances are), a "typical instance" selection function (which instances to keep as examples), and a classification function (deciding how a new case relates to the learned cases) (Aha et al. 1991). The lack of a formal framework for choosing the size of neighbourhood "*k*" can be problematic. Holmes and Adams (2002) have proposed a probabilistic strategy to overcome these difficulties.

To determine the distance between a pair of instances we apply the Euclidean distance metric. Also, in our experiments, *k* is set to five.

2.2.2 Logistic Discrimination

Logistic discrimination analysis (LgD) is related to logistic regression (LR). The dependent variable can only take values of 0 and 1, say, given two classes. This technique is partially parametric, as the probability density functions for the classes are not modelled but rather the ratios between them.

Let $y \in \{0,1\}$ be the dependent or response variable and let $x = (x_{i1}, x_{i2}, \ldots, x_{ip})$ be the predictor variables vector. A linear predictor η_i is given by $\beta_0 + \beta'$ where β_0 is the constant and β' is the vector of regression coefficients $(\beta = \beta_1, \beta_2, \ldots, \beta_p)$ to be estimated from the data. They are directly interpretable as log-odds ratios or in terms of $\exp(\beta')$, as odds ratios.

The *a posteriori* class probabilities are computed by the logistic distribution:

$$P(y = 1 \mid x_{i1}, ..., x_{ip}) = \pi_i = \frac{\exp\{\eta_i\}}{1 + \exp\{\eta_i\}}$$

β' are estimated by maximising the likelihood function

$$L(\beta_0, ..., \beta_p) = \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1 - y_i}$$

The estimated predicted value $\hat{\eta}_j$ and the estimated probability $\hat{\pi}_j$ for a new observation $x_{j1}, x_{j2}, \ldots, x_{jp}$ are given by $\hat{\eta}_j = \hat{\beta}_0 + \hat{\beta}' x$ and

$$\hat{\pi}_j = \pi(x, \hat{\beta}) = \frac{\exp\{\hat{\eta}_j\}}{1 + \exp\{\hat{\eta}_j\}}$$

These terms are often referred to as "predictions" for given characteristic vector *x*. Therefore, a new element is classified as 0 if $\pi_0 \le c$ and as 1 if $\pi_0 > c$, where c is the cut-off point score. Typically, the cut off point used could be 0.5 In fact, it has been argued that the slope of the cumulative logistic probability function is steepest in the region where $\pi_i = 0.5$. For a prediction problem with more than two classes, multinomial logit models are used.

3. Related Work

Significant advances have been made in the past few decades regarding methodologies for

credit scoring. Unfortunately, these methodologies are often not available to many researchers for a variety of reasons (for example, lack of familiarity, computational challenges). Thus, researchers often resort to *ad-hoc* approaches to model credit risk, ones which may ultimately do more harm than good. Specific results are discussed below.

Altman (1968) performed a simulation study predicting corporate bankruptcy by incorporating financial ratios in a multiple discriminant model. The discriminant-ratio model was able to predict bankruptcy correctly in 94 percent of the initial sample with 95 percent of all firms in the bankrupt and non-bankruptcy groups assigned to their actual group classification. Altman further employed linear discriminant analysis (LDA) and a neural network to diagnose corporate financial distress of one thousand Italian firms and concluded that neural networks are not clearly a dominant mathematical technique compared to traditional statistical techniques such as discriminant analysis. In fact LDA compared rather well to the neural network model in terms of decision accuracy.

The pioneering work by Chatterjee & Barcun (1970) performed a simulation study to evaluate a non-parametric approach (especially nearest neighbour methods) in the context of credit scoring application. They studied personal loan applications to a New York bank and classified them on the basis of the proportion of instances with identical characteristic vectors which belonged to the same class. This work was followed up by Henley & Hand (1996) who first describe the choice of metric and number of nearest neighbour one should consider when using the nearest neighbour classifier for credit scoring applications. A mailing order company dataset was used for the task.

Wiginton (1980) gave one of the first published investigations by comparing logistic regression (LR) with discriminant analysis applied to credit scoring. His results showed LR exhibiting higher accuracy rates, however, neither method was found to be sufficiently good to be cost effective for his problem. LR was also applied by Leonard (1993) to a commercial loan evaluation process (exploring several models using random effects for bank branches).

ANNs which have proved to be a ubiquitous approach to many problems are highly suited to credit scoring applications. Neural rule extraction technique was used by Baesens et al. (2003) for credit risk evaluation in their empirical study that was conducted on three real-world credit risk datasets. Their results provided evidence that neural rule extraction and decision tables are powerful management tools that would allow one to build advanced and user-friendly decision support systems for credit evaluation. Another comparative study of several applications of neural networks to corporate credit decisions and fraud detection was described by Rosenberg & Gleit (1994). West (2000) compared such methods with alternative classifiers. Tam & Kiang (1992) studied the application of ANN to Texas bank failure prediction for the period 1985-1987. ANN was compared with LDA, LR, *k*-NN and DT model. Their results suggest that ANN is most accurate, followed by LDA, LR, DT and *k*-NN. Other researchers have used neural networks for credit scoring and reported its

accuracy as superior to that of traditional statistical methods in dealing with credit scoring problems, especially with regard to non-linear patterns (Desai et al. 1996, 1997, Mahlhotra & Mahlhotra 2003, Jensen 1991, Salchenberger et al. 1992, Piramuthu 1999).

The performances of DTs or recursive partitioning techniques when evaluating credit risk are described by Makowski (1985), Coffman (1986), Carter & Catlett (1987) and Davis et al. (1996, 1997). In fact, Makowski (1986) was one of the first to advertise the use of DTs in credit scoring. Another use of DTs in the credit scoring area was Frydman et al. (1985) who found DTs to outperform LDA. Their results show classification and regression trees (CART) outperforming LDA. Boyle et al. (1992) evaluated recursive partitioning, LDA and linear programming. Their results showed the linear discriminant classifier slightly outperforming both recursive partitioning algorithm and linear programming. A DT can also be converted into rules that could be used for prediction tasks such as credit default and bankruptcy. A very recent addition to the empirical literature dealing with the DT method is Feldman & Gross (2005). The authors use this method for mortgage default data and discuss the pros and cons of DTs in relation to traditional methods.

Genetic algorithms are one of a number of general optimization schemes based on biological analogies. In the credit scoring context one has a number of scorecards which mutate and blend together according to their fitness at classification. Fogarty & Ireson (1993) and Albright (1994) were one of the first to describe this approach. Desai et al. (1997) compared it with ANNs in the credit union environment, while Yobas et al. (1997) did a comparison of these two and classification trees using credit card data. A recent addition to the literature dealing with genetic programming (GP) is by Ong et al. (2005) whose results show GP as performing better to methods such as ANN, DTs rough sets and LR.

The NBC algorithm has been applied most recently for credit risk approval of 671 applicants by Islam (2007). Its performance was compared with *k*-nearest neighbour (*k*-NN) classifier. 5-NN was found to be the best model for the application with a misclassification rate of 9.45 per cent compared with an error rate of 12.43 per cent by NBC.

Twala (2009) evaluates the impact of seven missing data techniques on 21 datasets (including credit data) by artificially simulating three different proportions, two patterns and three mechanisms of missing data. Their results show multiple imputation (MI) achieving the highest accuracy compared with other statistical and supervised learning missing data techniques. An ensemble of missing incorporated in attributes and multiple imputation approach was shown to improve prediction accuracy when dealing with incomplete data using DTs (Twala 2008).

4. Simulation Study

4.1. Experimental Set-Up

One of the objectives of this study is to investigate the behaviour of classifiers that are well-known to be noise tolerant systems. This section describes experiments that were carried out in order to compare the noise-tolerant performance of the five different classifiers

when noise is introduced (1) to both the training and testing (unseen) sets; and (ii) to only the testing set. Then, the impact of the combination of classifiers (ensemble of classifiers) on predictive accuracy is investigated. In all experiments, the ensembles consist of five individual classifiers. For each ensemble, the corresponding training samples have been designed by using the class-dependent bagging and random selection without replacement techniques as described in Section 2. The experiments are used on five credit datasets in terms of misclassification error rate. Each dataset defines a different learning problem as described below.

LOAN PAYMENTS

The data was used to classify a set of firms into those that would default and those that wouldn't default on loan payments. Of the 32 examples for training, 16 belong to the default case and the other 16 to the non-default case. All the 16 holdout examples belong to the non-default case. For a detailed description of this data, the reader is referred to Khalik and El-Sheshai (1980).

TEXAS BANKS

Texas banks that failed during 1985–1987 were the primary source of data. Data from a year and two years prior to their failure were used. Data from 59 failed banks were matched with 59 non-failed banks, which were comparable in terms of asset size, number of branches, age and charter status. Tam and Kiang had also used holdout samples for both the 1 and 2 year prior cases. The 1 year prior case consists of 44 banks, 22 of which belong to failed and the other 22 to non-failed banks. The 2 year prior case consists of 40 banks, 20 of which belong to failed and 20 to non-failed banks. The data describes each of these banks in terms of 19 financial ratios. For a detailed overview of the data set, the reader is referred to Tam & Kiang (1992).

AUSTRALIAN CREDIT APPROVAL

This credit card applications data set has 690 observations with 15 attributes. Of the attributes, nine are discrete with two to fourteen values, and six continuous attributes. There are 307 positive instances and 383 negative instances in this data set. One or more attribute values are missing from 37 instances. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. This dataset was obtained from the UCI repository of ML (Blake et al. 1999).

GERMAN CREDIT APPROVAL

This data set was also obtained from the UCI repository of ML (Blake et al. 1999). It contains 1000 observations on 20 attributes. The class attribute describes people as either good (about 700 observations) or bad (about 300 observations) credits. Other attributes include status of existing checking account, credit history, credit purpose, credit amount, savings account/bonds, duration of present employment, instalment rate in percentage of disposable income, marital status and gender, other debtors/guarantors, duration in current residence, property, age, number of existing credits at this bank, job, telephone ownership, whether foreign worker, and number of dependents.

JAPANESE CREDIT APPROVAL

This experimental dataset is about 690 Japanese credit card application approval obtained from the UCI ML repository (Blake et al. 1999). For confidentiality all attribute names

and values have been changed to meaningless symbols. All instances with missing values were removed from the analysis, reducing the data to 653 instances (of which 357 were granted credit and remaining 296 instances were declined). Otherwise, the dataset is a good mix of 15 attributes of which six are continuous with the remaining 9 continuous.

To perform the experiment each dataset was split randomly into 10 parts (Part I, Part II, Part III, Part IV, Part V, Part VI, Part VII, Part VIII, Part IX, Part X) of equal (or approximately equal) size. 10-fold cross validation is used for the experiment. For each fold, nine of the parts of the instances in each category are placed in the training set, and the remaining one is placed in the corresponding test. The same splits of the data are used for all the ML and statistical techniques.

Since the distribution of noise among attributes and class attribute are two of the most important dimensions of this study, two suites of data are created corresponding to attribute noise and classification noise. Feature noise is introduced to both the training and testing sets and has two versions: attribute noise to both the training and testing sets ($ATTR_{TR/TS}$) and attribute noise to only the testing set ($ATTR_{TS}$).

In order to simulate noise on both attributes, a brief description is given below.

Attribute noise: To introduce $p\%$ attribute (feature) noise to a dataset of D instances, each of which has F attributes (excluding the class label), we randomly select $\frac{p*D*F}{100}$ instances with replacement and for each of them we change the value of a randomly selected feature. For nominal features, the new value is chosen randomly with equal probability from the set of all possible values. For numeric features, the new value is generated from a Normal distribution defined by the mean and the standard deviation of the given feature, which are estimated from the dataset. Feature noise was introduced to both the training and test sets.

Both of these procedures have the same percentage of noise as their parameters. These two approaches are also run to get datasets with five levels of proportion of noise p, i.e., 0%, 5%, 15%, 30% and 50%.

It is reasoned that the condition with no noise (noise-free or quiet) should be used as a baseline and what should be analysed is not the error rate itself but the increase or excess error induced by the combination of conditions under consideration. Therefore, for each combination of classifiers, the number of attributes with noise, proportion of noise, and the error rate for all data present is subtracted from each of the three five levels of noise. This will be the justification for the use of differences in error rates analysed in some of the experimental results.

In all experiments, the ensemble consists of five individual classifiers. For each ensemble, the corresponding training samples have been designed by using the class dependent bagging technique as described in Section 1. Five different configurations for the base classifiers have been tested: ANN, DT, NBC, k-NN and LgD. The results for each classifiers (i.e., with no combination) has also been included as a baseline for the twenty different combination of classifiers considered from which we shall now call ANN ensembles, DT ensembles, NBC ensembles, k-NN ensembles and LgD ensembles.

All statistical tests are conducted using the MINITAB statistical software program (MINITAB 2002). Analyses of variance, using the general linear model (GLM) procedure (Kirk 1982) are used to examine the main effects and their respective interactions. For attribute (feature) noise, this was done using a 3-way repeated measures design (where each effect was tested against its interaction with datasets). The fixed effect factors were the: ML and SPR techniques; attributes with noise in only the testing set and attributes with noise in both the training and testing sets; and proportion of noise. Attribute noise in only the training set is not considered in our experiments since one would not expect noise to be present in the training set and not in the testing set. However, it is quite common to have noise on the testing sample and not on the training sample (Hand 2008).

The five datasets used were used to estimate the smoothed error. In fact, accuracy of the tree, in the form of a smoothed error rate, was predicted using the test data. Results were averaged across ten folds of the cross-validation process before carrying out the statistical analysis. The averaging was done as a reduction in error variance benefit.

It has often been argued that selecting and evaluating a classification model based solely on its error rates is inappropriate. The argument is based on the issue of using both the false positive (rejecting a null hypothesis when it is actually true) and false negative (failing to reject a null hypothesis when it is in fact false) errors as performance measures whenever classification models are used and compared. Furthermore, in the business world, decisions (of the classification type) involve costs and expected profits. The classifier is then expected to help making the decisions that will maximise profits. For example, predicting credit risk could involve two types of errors: 1) predicting risk as high when in fact it is low, and 2) predicting risk as low when in fact it is high. Now, mere misclassification rate is simply not good enough to predict software effort. To overcome this problem and further make allowances for the inequality of mislabelled classes, variable misclassification costs are incorporated in our attribute selection criterion via prior specification for all our experiments. This also solves the imbalanced data problem. Details about how misclassification costs are used for both splitting and pruning rules are presented in Breiman et al. (1984).

4.2. Experimental Results

The results are presented in two parts. The first part of this section compares the performance of five classifiers and the effect of attribute noise on predictive accuracy. Within the first part we present results of the impact of attribute noise when it occurs on both training and testing sets, then the results of the effect of attribute noise when it is introduced on only the test set. The second part presents the overall results of the ensemble methods on predictive accuracy, with the results for each single classifier (with no combination) used as a baseline. The purpose is to verify whether the combined classifiers could achieve higher classification accuracies than their respective components. In addition, whenever the classifier is used for training purposes we shall refer to it as a training method. Similarly, whenever it is used for testing or classification purposes we

shall refer to it as a testing method.

4.2.1 Experimental Results I

From these experimental results, we have the following observations:

1. All the five classifiers significantly reduce predictive accuracy at all levels of noise from 5% to 50%. Otherwise, all the classifiers show a very good fit to the noise free problems (i.e., at the 0% level). In fact, at lower levels of noise the classifier compare favourably. Overall, DT achieves the highest accuracy rates as a classifier for handling attribute noise in the training set, followed by NBC, *k*-NN, LgD and ANN, respectively. For attribute noise in testing data, the highest accuracy rate is achieved by NBC followed by LgD, DT and ANN. The worst performance is by *k*-NN (Figure 1).
2. Attribute noise appears to have less impact when it occurs in both the training and testing sets than when it occurs in only the testing set. (Figure 2).
3. All the classifiers suffer decreases in predictive accuracy due to noise level increases. In most cases, the decrease in accuracy is linear with respect to the increase of noise level (Figure 3).

4.2.2 Experimental Results II

Figure 4-8 summarises the overall excess error rates for classifier ensembles of five ML and SPR techniques on credit risk predictive. The behaviour of these techniques is explored under varying amounts of attribute noise levels. The error rates of each technique are averaged over the 4 datasets.

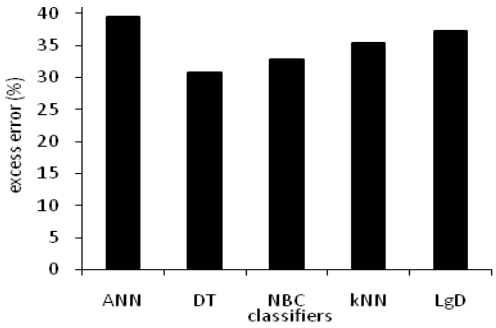

Figure 1 Overall means for classifiers for training data

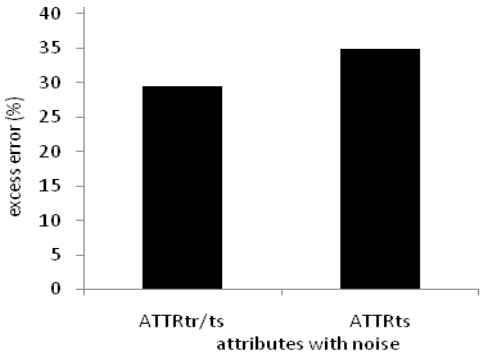

Figure 2 Overall means for attributes with noise

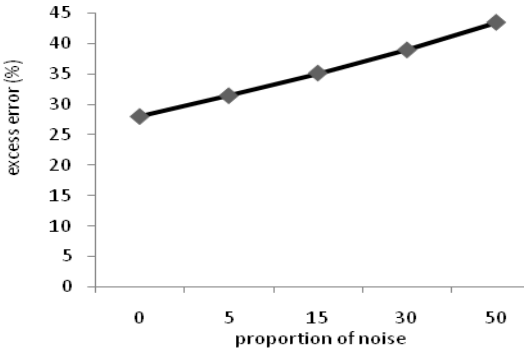

Figure 3 Overall means for noise proportions

All the main effects were found to be significant at the 5% level of significance (F = 22.78, df = 19 for ensemble classifiers; F =

36.89, df = 1 for attribute noise in training and testing sets, and on testing set only; F = 63.00, df = 3 noisy data proportions; p<0.05 for each main effect). The only significant interaction effect is the three-way between training methods, testing methods and the proportion of noise (F = 4.218; df = 48; p<0.05).

Figure 4 plots the overall excess error rates for five ANN ensembles. From the results it follows that the ensemble of ANN and NBC achieves the highest accuracy rates (at all lower levels of noise) but deteriorates with increases in noise levels, with ANN and DT exhibiting the best performance The worst performance is by the ANN and k-NN ensemble at the 5% level of significance (p<0.05).

From Figure 5 (where DT is the main component of the ensemble), it appears that the ensemble of DT and NBC has higher predictive accuracy compared with the other ensembles which involves DT as its other component. However, the DT and LgD ensemble compares favourably with DT and NBC at higher levels of noise.

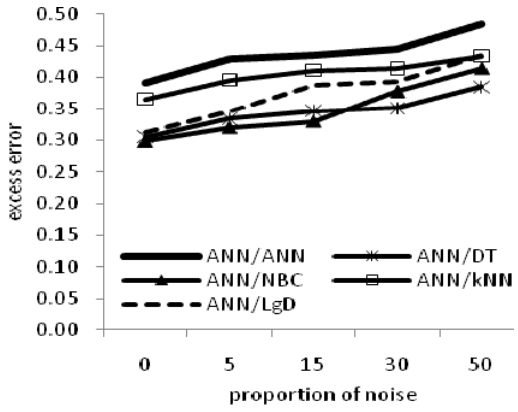

Figure 4 ANN ensembles

One again, NBC as the main component with other classifiers (especially with ANN) achieves higher predictive accuracy rates at all levels of noise. However, when the noise level is 50%, its accuracy is comparable with the NBC and DT ensemble. Poor performance is observed for the NBC and LgD ensemble (Figure 6).

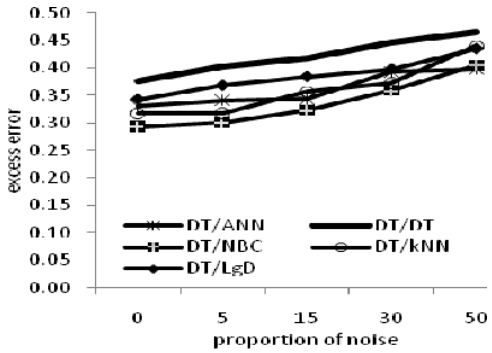

Figure 5 DT ensembles

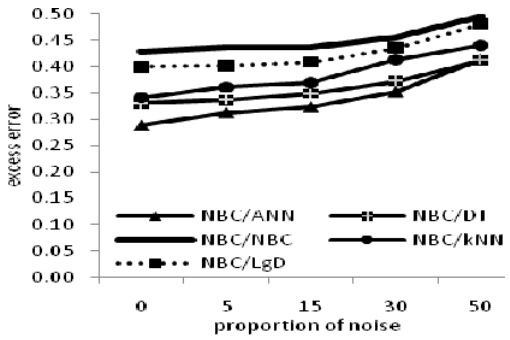

Figure 6 NBC ensembles

Figure 7 shows the effects of attribute noise on the predictive accuracies of four k-NN ensembles. A superior performance is observed for the k-NN and DT ensemble. This is the case at all noise levels. The worst performance is by k-NN and ANN, followed by k-NN and LgD.

The effects of attribute noise on predictive accuracies of ensembles with LgD as the main

component is displayed in Figure 8. The results show the LgD and NBC ensemble providing better results compared with LgD and k-NN, which clearly gives poor results.

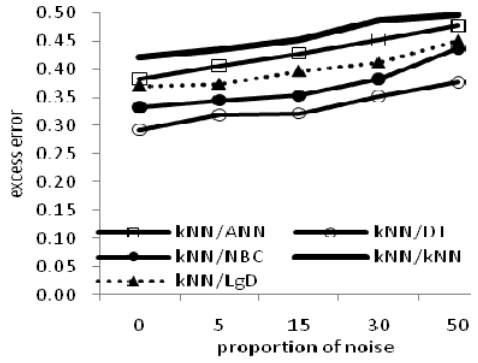

Figure 7 *k*NN ensembles

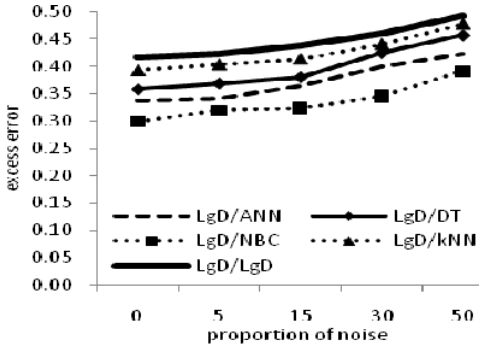

Figure 8 LgD ensembles

5. Conclusions and Discussion

In this paper, we investigated the accuracy of five classifiers applied for credit risk prediction, and tested them on five real-world datasets. Using the same datasets, we further investigated the accuracy of twenty ensembles of classifiers in terms of improving credit risk prediction.

The ensembles that achieved greater accuracy rates were NBC and DT. Based on preliminary evidence, it has been found that ensemble of classifiers improves the prediction accuracy of the baseline single classifiers (ANN, DT, NBC, *k-NN* and LgD). This is the case at all levels of attribute noise in both the training and testing sets, on the one hand, and when attribute noise is only on the testing set.

Our results also show the impact on the performance of classifiers is caused by amount of noise in either test set or in both the training and testing sets. However, as the proportion of noise increases, the major determining factor on the performance of methods is how the noise is distributed among the sets (training and testing). All classifiers yield lower accuracy rates when noise is in both the training and test sets compared with when attribute noise is only on the test set.

It was also observed that the impact of noise on predictive accuracy depends not just upon the classifier or the proportion of attribute noise but upon a combination of the two. Therefore, neither can be considered in isolation.

From our experiments, there exists threats to the validity of the results. Potential threats include the initial exclusion from the datasets of all the instances with missing values, which could have involuntarily introduced biases, especially if those missing values contained improtant information and they were not missing completely at radonm as we assumed. However, for one of the biggest datasets (German credit) the experimental results were carefully validated. For example, the experiments were conducted under the supervision of a domain expert who had a deep understanding for this underlying dataset. This was a time consumig exercise on ourselves and the expert.

The averaging of results of the five

domains/datasets (which is more like mixing datasets from different companies into one dataset) is another potential issue. Some empirical studies in finance assume that domain-and-process factors can be best accounted for within homogenous data, i.e., data produced by the same company, with an almost stable process, in the same environment. Hence, in the case of our study, it would be interesting to see if the results are consistent through the five datasets.

The issue of determining whether or not to apply the ensemble strategy to a given dataset must be considered. For the work described here, the data were artificially corrupted, i.e. noise was artificially simulated on the attributes. Unfortunately, this type of information is rarely known for most "real-world" applications. In some situations, it may be possible to use domain knowledge to determine the mechanism generating noise. For situations where this knowledge is not available, the conservative nature of the consensus ensemble dictates that noise will appear at random. In addition, fundamentally there are two different types of noise which cover whatever situation one might encounter in practice. Our approach is general since one can apply our results as appropriately. For example, a relevant part of our paper could be taken if an empirical researcher thinks he/she has only attribute noise and not class noise. In addition, it is possible to formally test for attribute noise assumption. However, as much as class noise is important, it is hard to test its assumption, which requires one to test for missingness that was deliberately created when data was collected or the data could be missing due to an unknown censoring mechanism.

Comparing all considered multiple classifiers together, we should be cautious as the number of the results on common datasets is limited in that none of the multiple classifiers is the best for all datasets. Each of the classifiers has its own area of superiority. The relative merits of the particular approaches depend on the specificity of analysed classification problems.

Several existing directions exist for future research. One area deserving of future study is to address the computational costs of constructing the particular ensemble of classifiers. In general, we would expect that they need more computations than the standard single classifier. An interesting topic would be to assess the computing time needed to construct the different ensembles.

6. Appendices

List of acronyms used in this paper.

ANN: Artificial Neural Network

$ATTR_{TR/TS}$: Attribute Noise in both Training and Test Sets

$ATTR_{TS}$: Attribute Noise in Test Set only

BIS: Bank for international Settlements

CART: Classification and Regression Trees

DT: Decision Tree

GLM: Generalised Linear Model

k-NN: *k*-Nearest Neighbour

LDA: Linear Discriminant Analysis

LgD: Logistic Discrimination

LR: Logistic Regression

MI: Multiple Imputation

ML: Machine Learning

NBC: Naïve Bayes Classifier

SPR: Statistical Pattern Recognition

UCI: University of California, Irvine

Acknowledgements

The author would also like to thank the anonymous referees for their helpful and useful comments to improve the quality of the paper.

References

[1] Aha, D.W., Kibbler, D.W. & Albert, M.K. (1991). Instance-based learning algorithms. Machine Learning, 6 (37): 37-66

[2] Aha, D.W. (1992). Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. International Journal of Man Machine Studies, 36 (2): 267-287

[3] Albright, H.T. (1994). Construction of polynomial classifier for consumer loan applications using genetic algorithms. Department of Systems and Engineering, University of Virginia, Working Paper

[4] Altman, E.I. (1968). Financial ratios, discriminant analysis and prediction of corporate bankruptcy. Journal of Finance, 23 (4): 589-609

[5] Baesens, B., Setiono, R., Mues, C.H. & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit risk evaluation. Management Science, 49: 312-329

[6] Bauer, E. & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: bagging, boosting and variants. Machine Learning, 36: 105-139

[7] BIS. (2004). International convergence of capital measurement and capital standards: a revised framework. Basel Committee of Banking Supervision, Bank for International Settlements

[8] Blake, C., Keogh, E. & Merz, C.J. (1999). UCI respiratory of machine learning databases. University of California, Irvine, Department of Information and Computer Sciences

[9] Boyle, M., Crook, J.N., Hamilton, R. & Thomas, L.C. (1992). Methods for credit scoring applied to slow payers. In: Thomas, L.C., Crook, J.N., Edelman, D.B. (eds.), Credit Scoring and Credit Control, pp. 75-90. Oxford: Clarendon

[10] Breiman, L. (1996). Bagging predictors. Machine Learning, 26 (2): 123-140

[11] Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). Classification and Regression Trees. Wadsworth

[12] Carter, C. & Catlett, J. (1987). Assessing credit card applications using machine learning. IEEE Expert, fall: 71-79

[13] Chatterjee, S. & Barcun, S. (1971). A nonparametric approach to credit screening. Journal of American Statistical Association, 65: 50-154

[14] Coffman, J.Y. (1986). The proper role of tree analysis in forecasting the risk behaviour of borrowers. MDS reports 3, 4, 7 and 9. Management Decision Systems, Atlanta

[15] Davis, R.H., Eedelman, G.B. & Gammerman, A.J. (1992). Machine learning algorithms for credit card applications. IMA Journal of Mathematics Applied Business and Industry, 4: 43-51

[16] Desai, Y.S., Crook, J.N. & Overstreet, G.A. (1996). A comparison of neural networks and linear scoring models in the credit environment. European Journal of Operations Research, 85: 24-37

[17]Desai, V.S., Convay, J.N. & Overstreet, G.A. Credit scoring models in credit-union environment using neural networks and genetic algorithms. IMA Journal of Mathematics Applied in Business and Industry, 8: 323-346

[18]Dietterich, T. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Machine Learning, 40 (2), 139-158

[19]Drucker, H., Cortes, C., Jackel, L.D., Lecun, Y. & Vapkin, V. (1994). Boosting and other ensemble methods. Neural Computation, 6: 1289-1301

[20]Feldman, D. & Gross, S. (2005). Mortgage default: classification tree analysis. Journal of Real Estate Finance and Economics, 30: 369-396

[21]Forgaty, T.C. & Ireson, N.S. (1993). Evolving Bayesian classifier for credit control – a comparison with other machine learning methods. IMA Journal of Mathematics Applied in Business and Industry, 5: 63-76

[22]Freund, Y. & Schapire, R. (1996). A decision theoretic generalization of on-line learning and an application to boosting. Journal of Computing and Systems, 55: 119-139

[23]Frydman, H.E., Altman, E.E. & Kao, D. (1985). Introducing recursive partitioning for financial classification: the case of financial distress. Journal of Finance, 40 (1): 269-291

[24]Henley, W.E. (1995). Statistical aspects of credit scoring. PhD Dissertation, The Open University, Milton Keynes, United Kingdom

[25]Hand, D.J. (2008). Private communication

[26]Hand, D.J. & Vinciotti, V. (2003). Choosing *k* for two-class nearest neighbour classifiers with unbalanced classes. Pattern Recognition Letters, 24: 1555-1562

[27]Hand, D.J. & Henley, W.E. (1997). Statistical classification methods in consumer credit scoring: a review. Journal of the Royal Statistical Society, 160: 523-541

[28]Henley, W.E. & Hand, D.J. (1996). A *k*-nearest neighbour classifier for assessing consumer risk. Statistician, 44: 77-95

[29]Ho, T. K. (1995). Random decision forests. In: Holmes, C.C., Adams, N.M. (eds.), Proceedings of the 3rd International Conference on Document Analysis and Recognition, 278-282

[30]Holmes, C.C. & Adams, N.M. (2002). A probabilistic nearest neighbour method for statistical pattern recognition. Journal of Royal Statistical Society, Series B, 64: 295-306

[31]Islam, M.J., Wu, Q.M.J., Ahmadi, M. & Sid-Ahmed, M.A. (2007). Investigating the performance of naïve Bayes classifiers and *k*-nearest neighbor classifiers. In: International Conference on Convergence Information Technology, 1541-1546, November, 21-23, 2007

[32]Jensen, H.L. (1992). Using neural networks for credit scoring. Managerial Finance, 18: 15-26

[33]Khalik, A. & El-Sheshai, K.M. (1980). Information choice and utilization in an experiment of default prediction. Journal of Accounting Research, autumn: 325-342

[34]Kirk, E.E. (1982). Experimental Design (2nd Ed.). Cole Publishing Company, Monterey, CA: Brooks

[35]Kittler, J., Hatef, M., Duin, R.P.W. & Matas, J. (1998). On combining classifiers. IEEE Transaction on Pattern Analysis and Machine Intelligence, 20 (3): 226-239

[36]Kononenko, I. (1991). Semi-naïve Bayesian classifier. In: Proceedings of European conference on Artificial Intelligence, 206-219

[37]Lacher, R.C., Oats, P.K., Sharma, S. & Fant, L.F. (1995). A neural network for classifying the financial health of a firm. European Journal of Operational Research, 85: 53-65

[38]Leonard, K.J. (1993). Empirical Bayes analysis of the commercial loans evaluation process. Statistics Probability Letters, 18: 289-296

[39]Mahlhotra, R. & Malhotra, D.L. (2003). Evaluating consumer loans using neural networks. Omega: the International Journal of Management Science, 31 (2): 83-96

[40]Makowski, P. (1985). Credit scoring branches out. Credit World, 75: 30-37

[41]MINITAB. (2002). Statistical software for windows 9.0. MINITAB, Inc., PA, USA

[42]Neumann, D.E. (2002). An enhanced neural network technique for software risk analysis. IEEE Transactions on Software Engineering, 904-912

[43]Ong, C.S., Huang, J.J. & Tzeng, G.H. (2005). Building credit scoring models using genetic programming. Expert Systems with Applications, 29: 41-47

[44]Piramuthu, S. (1999). Financial credit risk evaluation with neural and neurofuzzy systems. European Journal of Operational Research, 112: 310-321

[45]Quinlan, J.R. (1993). C.4.5: Programs for Machine Learning. Morgan Kauffman Publishers, Inc., Los Altos, California

[46]Ripley, B.D. (1992). Pattern Recognition and Neural Networks. Cambridge University Press, New York: John Wiley

[47]Rosenberg, E. & Gleit, A. (1994). Quantitative methods in credit management: a survey. Operations Research, 42 (4): 589-613

[48]Salchenberger, L.M., Cinar, E.M. & Lash, N.A. (1992). Neural networks: a new tool for predicting thrift failures. Decision Sciences, 23: 899-916

[49]Schafer, J.L. (1997). Analysis of Incomplete Multivariate Data. Chapman and Hall, London

[50]Tam, K.Y. & Kiang, M.Y. (1992). Managerial applications of neural networks: the case of bank failure predictions. Management Science, 38 (7): 926-947

[51]Twala, B. (2009). An empirical comparison of techniques for handling incomplete data using decision trees. Applied Artificial Intelligence, 23 (5): 373-405.

[52]Twala, B., Jones, M.C. & Hand, D.J. (2008). Good methods for coping with missing data in decision trees. Pattern Recognition Letters, 29: 950-956

[53]West, D. (2000). Neural network credit scoring models. Computers & Operations Research, 27: 1131-1152

[54]Wiginton, J.C. (1980). A note on the comparison of logit and discriminant models of consumer credit behavior. Journal of Financial and Quantitative

Analysis, 15: 757-770

[55]Wolpert, D. (1992). Stacked generalization. Neural Networks, 5 (2): 241-259

[56]Yobas, M.B., Crook, J.N. & Ross, P. (1997). Credit scoring using neural and evolutionary techniques. Credit Research Centre, University of Edinburgh, Working Paper 7/2

Bhekisipho Twala is a Principal Research Scientist at the CSIR in Pretoria, South Africa. His current work involves applying artificial intelligence techniques in the field of robotics. Dr Twala has a PhD in Machine Learning and Statistics from the Open University (UK), an MSc in Statistics from Southampton University (UK) and a BA degree in Economics and Statistics from the University of Swaziland. Prior to relocating to South Africa, he was a post-doctoral researcher at Bournemouth University (UK) and later at Brunel University in the UK, working on empirical software engineering research. Dr Twala is an associate editor for the Journal of Computers. His research interests are in knowledge discovery and reasoning with uncertainty, and the interface between statistics and computing. Much of his research work has been published by leading international journals and conferences. Dr Twala is also interested in applications in finance, medicine, psychology, software engineering and robotics.