Expert Systems with Applications 42 (2015) 741–750

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Credit scoring using the clustered support vector machine

Terry Harris *

Credit Research Unit, Department of Management Studies, The University of the West Indies, Cave Hill Campus, P.O. Box 64, Barbados

ARTICLE INFO

Article history:
Available online 6 September 2014

Keywords:
Credit risk
Credit scoring
Clustered support vector machine
Support vector machine

ABSTRACT

This work investigates the practice of credit scoring and introduces the use of the clustered support vector machine (CSVM) for credit scorecard development. This recently designed algorithm addresses some of the limitations noted in the literature that is associated with traditional nonlinear support vector machine (SVM) based methods for classification. Specifically, it is well known that as historical credit scoring datasets get large, these nonlinear approaches while highly accurate become computationally expensive. Accordingly, this study compares the CSVM with other nonlinear SVM based techniques and shows that the CSVM can achieve comparable levels of classification performance while remaining relatively cheap computationally.

1. Introduction

In recent years, credit risk assessment has attracted significant attention from managers at financial institutions around the world. This increased interest has been in no small part caused by the weaknesses of existing risk management techniques that have been revealed by the recent financial crisis and the growing demand for consumer credit (Wang, Yan, & Zhang, 2011). Addressing these concerns, over past decades credit scoring has become increasingly important as financial institutions move away from the traditional manual approaches to this more advanced method, which entails the building of complex statistical models (Huang, Chen, & Wang, 2007; Zhou, Lai, & Yu, 2010).

Many of the statistical methods used to build credit scorecards are based on traditional classification techniques such as logistic regression or discriminant analysis. However, in recent times non-linear approaches,[1] such as the kernel support vector machine, have been applied to credit scoring. These methods have helped to increase the accuracy and reliability of many credit scorecards (Bellotti & Crook, 2009; Yu, 2008). Nevertheless, despite these advances credit analyst at financial institutions are pressed to continually pursue improvements in classifier performance in an attempt to mitigate the credit risk faced by their institutions. However, many of the improvements in classifier performances remain unreported due to the proprietary nature of industry led credit scoring research which attempts to find more efficient and effective algorithms.

In the wider research community, the recent vintages of non-linear classifiers (e.g. the kernel support vector machine) have received a lot of attention and have been critiqued for, *inter alia*, their large time complexities. In fact the best-known time complexity for training a kernel based support vector machine is still quadratic (Bordes, Ertekin, Weston, & Bottou, 2005). As a result, when applied to credit scoring substantial computational resources are consumed when training on reasonably sized real world datasets. Accordingly, efforts to develop and apply new classifiers to credit scoring, which are capable of separating nonlinear data while remaining relatively inexpensive computationally, are well placed.

This paper investigates the suitability for credit scoring of a recently developed support vector machine based algorithm that has been proposed by Gu and Han (2013). Their clustered support vector machine has been shown to offer comparable performance to kernel based approaches while remaining cheap in terms of computational time. Furthermore, this study makes some novel adjustments to their implementation and explores the use of radius basis function (RBF) kernels in addition to the linear kernel posited by Gu and Han.

The remainder of this paper is presented as follows. Section 2 outlines a brief review of the literature concerning the field of credit scoring and sets the stage for the proposed CVSM model for credit scoring that is presented in Section 3. The details of the historic clients' loan dataset and modeling method are highlighted in Section 4. Section 5 presents the study results, and Section 6 discusses the findings, presents conclusions, and outlines possible directions for future research.

* Tel.: +1 (246) 417 4302; fax: +1 (246) 438 9167.
E-mail address: terry.harris@cavehill.uwi.edu

[1] This has been applied because credit-scoring data is often not linearly separable.

http://dx.doi.org/10.1016/j.eswa.2014.08.029
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

2. Background and related works

2.1. Overview

Credit scoring has been critical in permitting the exceptional growth in consumer credit over the last decades. Indeed without accurate, automated credit risk assessment tools, lenders could not have expanded their balance sheets effectively over this time. This section presents a brief review of the relevant literature that has emerged in this space.

2.2. What is credit scoring?

Credit scoring can be viewed as a method of measuring the risk attached to a potential customer, by analyzing their data to determine the likelihood that the prospective borrower will default on a loan (Abdou & Pointon, 2011). According to Eisenbeis (1978), Hand and Jacka (1998), and Hand, Sohn, and Kim (2005) credit scoring can also be described as the statistical technique employed to convert data into rules that can be used to guide credit granting decisions. As a result, it represents a critical process in a firm's credit management toolkit. Durand (1941) posited that the procedure includes collecting, analyzing and classifying different credit elements and variables in order to make credit granting decisions. He noted that to classify a firm's customers, the objective of the credit evaluation process, is to reduce current and expected risk of a customer being "bad" for credit. Thus credit scoring is an important technology for banks and other financial institutions as they seek to minimize risk.

2.3. Related works

Over the years, the demand for consumer credit has increased exponentially. According to Steenackers and Goovaerts (1989), this increase in the demand for credit can be attributable to the increased levels of consumption and the reliance on credit to support this activity. In the United States, this rising level of consumerism followed the introduction of the first modern credit card in 1950s, so that by the 1980s over 55% of American households owned a credit card. Crook, Edelman, and Thomas (2007) posited that by this time, in the US, the total amount of outstanding consumer credit was over $700 billion. Comparatively, at the end of June 2013 this figure had risen to a staggering $2800 billion, a 400% increase (BGFRS, 2013).

Henley (1994) noted that the increasing demand for consumer credit has led to the development of many practical the scoring models, which have adopted a wide range of statistical and non-linear methods. Similarly, Mays (2001) posited that a number of various techniques have been used to build credit scoring applications by credit analyst, researchers, and software developers. These techniques have included; discriminant analysis, linear regression, logistic regression, decision trees, neural networks, support vector machines, *k*-means, etc.

In recent times, the use of more complex non-linear techniques, such as neural networks, and support vector machines, to build credit scoring applications has seen significant increases in the reported accuracy and performance on benchmarking datasets (Baesens et al., 2003). Irwin, Warwick, and Hunt (1995) and Paliwal and Kumar (2009) both provide evidence that advanced statistical techniques yield superior performance when compared to traditional statistical techniques, such as discriminant analysis, probit analysis and logistic regression. Masters (1995) also provided evidence that the use of sophisticated techniques, such as neural networks, was essential because they had the capability to more accurately model credit scoring data that exhibits interactions and curvature. However, as pointed out by Hand (2006) the increased performance of these more advanced techniques could be illusionary and if real, diminished due to shifts in the class distribution over time. The following sub-sections present a brief discussion concerning some of the classical and advanced statistical models used for credit risk assessment.

2.4. Discriminant analysis

In his seminal paper, Fisher (1936) proposed the use of discriminant analysis to differentiate between two or more classes in a dataset. Since that time, Durand (1941) and Altman (1986) have both applied Fisher's (1936) discriminant analysis to credit scoring. Durant used discriminant analysis to assess the creditworthiness of car loan applicants, while Altman used it to explore corporate bankruptcy proposing his popular *Z*-scores (Altman, 1968). In works published separately by Desai, Crook, and Overstreet (1996), Hand and Henley (1997), Hand, Oliver, and Lunn (1998), Sarlija, Bensic, and Bohacek (2004), and Abdou and Pointon (2009), they showed that discriminant analysis is indeed a valid technique for credit scoring. Hand and Henley (1997) noted that discriminant analysis, a parametric statistical technique, was well suited to credit scoring because it was designed to classify groups and variables into two or more categories or discriminate between two groups. However, Saunders and Allen (1998) noted that with this type of method certain assumptions about the data must be met. These assumptions include, normality, linearity, homoscedasticity, non-multicollinearity, etc. Falbo (1991) and Sarlija et al. (2004) posited that despite these limitations, over the years this technique has been frequently applied to build credit scoring applications, and it remains one of the most popular approaches taken today when classifying customers as creditworthy or uncreditworthy.

Several authors have criticized the use of discriminant analysis in credit scoring. Eisenbeis (1978) point-out a number of the statistical problems in applying discriminant analysis to credit scorecard development. These problems include the following: group definition, classification error prediction, estimating population priors, and the use of linear functions instead of quadratic functions, to mention a few. Nevertheless, Greene (1998) and Abdou (2009) noted that despite these limits, discriminant analysis is one of the most commonly used techniques in credit scoring.

2.4.1. Linear regression

Another popular classical statistical technique applied to credit scoring is linear regression. This method has developed into an essential component of data analysis in general and is concerned with describing the relationship between a dependent variable and one or more independent variables. Thus, customers' historical payments, guarantees, default rates and other factors can be analyzed using linear regression to set up a score for each factor, and compare it with the bank's cut-off (threshold) score. Hence, only if a new customer's score exceeds the bank's cut-off score will credit be granted (Hand & Jacka, 1998).

In its basic form, linear regression used for credit scoring requires the establishment of a threshold score. This threshold credit score is derived from the relationships between the firm's historic clients' features and their associated weights. As can be seen in the linear equation, $Z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$, where the variable n denotes the number of features collected from past and potential clients. These features are represented by the x's, which are multi-dimensional vectors in $\mathbb{R}^m$, where m denotes the number of clients in the historical clients' database. The θ's represent the weights, and the feature variables and their weights used to calculate a credit score, $Z \in \mathbb{R}$, thus when an applicant scores below the threshold they are rejected, while an applicant scoring above may be granted credit.

Orgler (1970) was one of the first researchers to use linear regression for credit scoring. He used regression analysis to model commercial loans and to evaluate outstanding consumer loans. Conducting this work he realized that this approach was somewhat limited but could be used to review loans.

2.4.2. Logistic regression

According to Leonard (1995) and Hand and Henley (1997), logistic regression has been widely used in credit scoring applications in various disciplines worldwide and has become one of the most important statistical techniques used for this purpose. They noted that it is an important tool used in social science research for categorical data analysis. Siddiqi (2005) opined that in credit scoring, a customer's default probability is a function of certain social-economic variables such as income, marital status and others, in relation to the behavioral characteristics of the customer as indicated by their default history. Thus, this exercise can be view as a binary classification problem where logistic regression is a suitable method. Jentzsch (2006) pointed out that compared to discriminant analysis, logistic regression requires less assumptions. In fact, the only restrictions when using the logistic technique is that the output variable must be binary and that there is no multicollinearity among predictor variables.

Logistic regression can be seen as a special case of linear regression. What distinguishes logistic regression from linear regression is that with logistic regression, the outcome variable is dichotomous, while with linear regression this value can be any real number (Hosmer & Lemeshow, 1989).

2.4.3. Decision trees

One modern approach taken to develop credit scorecard is the decision tree method (Hand & Henley, 1997). Here, to achieve binary classification, a decision tree is made-up of a set of sequential binary splits on the historic clients' dataset. Hence, a dichotomous tree is built by splitting the records at each node based on a function of a single input (Coffman, 1986). There is some debate concerning the first use of a decision tree model. However, Breiman, Friedman, Olshen, and Stone (1984) wrote an early paper is widely seen as being instrumental to its widespread use and popularity; however, Rosenberg and Gleit (1994) claim that the first use of CART model was at the Harvard Business School by Raiffa and Schlaifer (1961).

A decision tree model is essentially a directed acyclic graphical model. When used for binary classification, as directed (rooted) tree, the decision tree model must satisfy the following properties, (i) there must be exactly one node that has no edges entering it, and this node is referred to as the root, (ii) every node except the root has exactly one edge entering it and (iii) there is a unique path from the root to each node.

The decision tree model is also a non-parametric method. This means that the system does not attempt to learn parameters upon which to score the applicants attributes. Rather the system memorizes certain key characteristics about the data (the binary splits and corresponding cut-off values for each feature). Arminger, Enache, and Bonne (1997) and Hand and Jacka (1998) noted that the aim of the recursive partitioning method was to minimize cost. Hence the system considers all possible splits to find the best one, and the sub-tree that is best fitted is selected based on its overall error rate or lowest cost of misclassification (Zekic-Susac, Sarlija, & Bensic, 2004). The classification of a new applicant is determined based on the classification of the resulting leaf node (a node that has no child) at after traversing the tree model using the applicant's attributes as input.

2.4.4. Neural networks

Another graphical model applied to credit scoring is the neural network. This is an advance statistical technique that calculates the weights in the form of scored points for the independent variables from past cases of creditworthy and un-creditworthy applicants (Mays, 2001). As defined by Gately (1995), neural networks are an artificial intelligence problem solving technique that learns through a trial and error training process. Smith and Gupta (2003) stated that this model can consist of three layers; input layer, hidden layer and output layer. They noted that each node in the input layer denoted one predictor variable and brings the value into the network. Nodes in the hidden layers combine and transform these values to match that of the target value(s) in the output layer. Each node in the output layer represents one dependent variable and in the case of credit scoring there is usually just one node in this layer.

Neutral networks have been applied to credit scoring due to their ability to train both linear and non-linear functions to describe data. This procedure helps analysts to achieve better decision-making outcomes when the historic clients' dataset is non-linear (Saunders & Allen, 1998). Hence, in several fields, especially banking, neutral networks have emerged as a practical technology of choice. Gately (1995) identified several areas in which neural networks can be used successfully, and these include; credit card fraud, financial accounting data fraud detection, corporate bankruptcy prediction, bank failure prediction, loan application, etc. Bishop (1995) noted that many real world problems have been addressed and essentially solved with the use of artificial neural networks. For example, digital pattern recognition successfully makes use of feed-forward network architecture.

Despite the advances derived from the use of artificial neural networks, there are some challenges. Artificial neural networks are often described as a "black box" classifier where the reason for the output is not given. This presents a particular problem in the credit scoring space where credit officers are routinely quizzed by rejected applicants as to the reason for their denial. Nevertheless, artificial neural networks remain a popular and highly accurate technique in practice.

2.4.5. Support vector machines

Another modern classification technique applied to credit scoring is the support vector machine. First developed by Cortes and Vapnik (1995) for classification problems, the support vector machine attempts to find the optimal separating hyperplane among the classes by maximizing the margin between them. Here, points lying on the boundaries are referred to as support vectors, while the middle of the margin is called the optimal separating hyperplane. It is this margin maximization property of the support vector machine that is said to lead to its state-of-the-art classification performance (Yu, 2008). However, like the artificial neutral network this technique is often criticized as being a "black box". Furthermore, it remains computationally expensive relative to traditional statistical methods.

2.5. Size and time complexity constraints

As has been noted in the preceding paragraphs a wide range of statistical and more recently non-linear methods have been used in credit scoring. Here, the use of more complex non-linear techniques, such as neural networks, and support vector machines, to build credit scoring applications has seen significant increases in the reported accuracy and performance on benchmark datasets (Baesens et al., 2003). Irwin et al. (1995) and Paliwal and Kumar (2009) agreed that such advanced statistical techniques provide a superior alternative to traditional statistical methods, such as discriminant analysis, probit analysis and logistic regression, when building practical models.

However the computational costs (time) associated with most of these nonlinear techniques can outweigh the benefits associated with increased classification performance, as the size of the historical clients dataset gets large. This is because many of these algorithms grow exponentially with increasing problem size. Furthermore, increasing computational power offers little in addressing this problem. To illustrate this, consider the fact that if the best known algorithm for solving a given credit scoring problem has a time complexity of the order of 2^n (stated mathematically $O(2^n)$), where the variable n represents the size of the training set and allowing one unit of time to equal one millisecond, then this algorithm can process in one second a maximum input size of approximately 9.96 as shown in (1);

$$\begin{aligned} &2^n = 1000, \\ &n\log(2) = \log(1000), \\ &n = \frac{\log(1000)}{\log(2)}, \\ &n = 9.96. \end{aligned} \tag{1}$$

Now, suppose that the firm wishes to increase the size of its training dataset and decides to purchase a newly designed micro-processor that is able to achieve a tenfold speedup in processing time. This new micro-processor chip would only increase the maximum solvable problem size in one second by 3.32 (as is shown in Eq. (2)).

$$\begin{aligned} &2^n = 10,000, \\ &n\log(2) = \log(10,000), \\ &n = \frac{\log(10,000)}{\log(2)}, \\ &n = 13.28. \end{aligned} \tag{2}$$

This is not very significant! Furthermore it can be contrasted with the increased performance to be derived should a better classification algorithm be applied to the problem. If a new algorithm is capable of transforming the time complexity from $O(2^n)$ to $O(n)$ then the maximum size of the problem solvable in one second would be 1000 ($n = 1000$) on the old micro-processor and 10,000 using the new micro-processor chip. Clearly, this is significantly greater than the performance possible using the older algorithm on the faster micro-processor.

As a result, the development and application of more computationally efficient algorithms in the credit scoring space is becoming increasingly more important as the sizes of historical datasets grow. The recently posited clustered support vector machine reduces the Rademacher complexity of the state-of-the-art SVM based classifier to an upper bound equivalent to the term $15k\sqrt{\log n/n}$ where k represents the number of clusters. The interested reader is invited to consult Gu and Han (2013) for further details. In the next section the author makes a contribution to literature by describing the development of the CSVM for credit risk assessment.

3. Clustered support vector machine for credit scoring

To build a CSVM classifier from a historical client dataset $S = \{(x_{(i)}, y_{(i)}); i = 1, \ldots, m\}$, where m represents the number of instances, ignoring the labels (the $y_{(i)}$'s) partition S into k clusters using K-means such that $\{c^{(j)}; j = 1, \ldots, k\}$. To do this the K-means algorithm assigns every training example $x_{(i)},$to its closest centroid.[2] Following this, the CSVM classifier for a cluster j can be represented as the linear combination of the attributes of the applicants in the cluster represented by, x's, multiplied by some cluster specific weights, w's, plus a noise term b as is shown in (3)[3].

$$z^{(j)} = w_1^{(j)}x_1^{(j)} + w_2^{(j)}x_2^{(j)} + \cdots + w_n^{(j)}x_n^{(j)} + b^{(j)}, \tag{3}$$

where the n denotes the number of client feature variables. Since the $w^{(j)}$'s and $x^{(j)}$'s can be represented as column vectors (3) can be written as;

$$z^{(j)} = w^{(j)T}x^{(j)} + b^{(j)}. \tag{4}$$

For each cluster the CSVM learns the parameters $w^{(j)}$ and $b^{(j)}$ and tries to find a hyperplane that maximizes the margin between the creditworthy and un-creditworthy individuals in the cluster. Hence, when given an individual training example $(x_{(i)}^{(j)}, y_{(i)}^{(j)})$, such that $y_{(i)}^{(j)} \in \{-1, 1\}$, the cluster specific functional margin $\hat{\gamma}^{(j)}$ can be defined for the i'th training example as follows;

$$\hat{\gamma}^{(j)} = y_{(i)}^{(j)}\left(w^{(j)T}x^{(j)} + b^{(j)}\right). \tag{5}$$

Furthermore, to confidently predict each training example in the cluster the functional margin needs to be large. And this therefore means that, $w^{(j)T}x^{(j)} + b^{(j)}$ must be a large positive number when $y_{(i)}^{(j)} = 1$, and a large negative number when $y_{(i)}^{(j)} = -1$. Thus, the functional margin with respect to the cluster $c^{(j)}$; is necessarily the smallest of the functional margins in the cluster, as in (6).

$$\hat{\gamma}^{(j)} = \min_{i=1,\ldots,m} \hat{\gamma}_{(i)}^{(j)}. \tag{6}$$

Considering a positive case, where $x_{(i)}^{(j)}$ corresponds to the label $y_{(i)}^{(j)} = 1$, the geometric distance between this point and the decision boundary, $\gamma_{(i)}^{(j)}$, is a vector orthogonal to the separating hyperplane. Thus, to find the value of $\gamma_{(i)}^{(j)}$, the corresponding point on the decision boundary is located by recognizing that $w^{(j)}/\|w^{(j)}\|$ is a unit vector pointing in the same direction as $w^{(j)}$. As a result, the relevant point on the separating hyperplane can be computed by evaluating the equation $x_{(i)}^{(j)} - \gamma_{(i)}^{(j)}w^{(j)}/\|w^{(j)}\|$. In addition, since this point is on the decision boundary, it will satisfy $w^{(j)T}x^{(j)} + b^{(j)} = 0$, as $w^{(j)T}\left(x_{(i)}^{(j)} - \gamma_{(i)}^{(j)}\frac{w^{(j)}}{\|w^{(j)}\|}\right) + b^{(j)} = 0$. And this can be reduced to, $w^{(j)T}x_{(i)}^{(j)} - \gamma_{(i)}^{(j)}\frac{w^{(j)T}w^{(j)}}{\|w^{(j)}\|} + b^{(j)} = 0$, since, $w^{(j)T}w^{(j)}/\|w^{(j)}\| = \|w^{(j)}\|^2/\|w^{(j)}\| = \|w^{(j)}\|, \gamma_{(i)}^{(j)}$ can be solved for as $\gamma_{(i)}^{(j)} = \left(\frac{w^{(j)}}{\|w^{(j)}\|}\right)^T x_{(i)}^{(j)} + \frac{b^{(j)}}{\|w^{(j)}\|}$. Hence, the general representation, taking into account cases of negative training examples, gives the equation $\gamma_{(i)}^{(j)} = y_{(i)}^{(j)}\left[\left(\frac{w^{(j)}}{\|w^{(j)}\|}\right)^T x_{(i)}^{(j)} + \frac{b^{(j)}}{\|w^{(j)}\|}\right]$. Finally, recognizing that when $\|w^{(j)}\| = 1$, the geometric margin is equal to the functional margin, the minimization problem, as in (7), can be re-expressed with respect to the geometric margin.

$$\gamma^{(j)} = \min_{i=1,\ldots,m} \gamma_{(i)}^{(j)}. \tag{7}$$

As a result, in order to find the decision boundary that maximizes the geometric margin for a cluster $c^{(j)}$ the optimization problem shown below must be solved,

$$\begin{aligned} &\max_{\gamma^{(j)}, w^{(j)}, b^{(j)}} \gamma^{(j)}, \\ &\text{s.t. } y_{(i)}^{(j)}\left(w^{(j)T}x^{(j)} + b^{(j)}\right) \geqslant \gamma^{(j)}, \quad i = 1, \ldots, m, \\ &\|w^{(j)}\| = 1. \end{aligned} \tag{8}$$

[2] The initial centroids are randomly selected.

[3] Here the subscript is used to demote the individual variables as opposed to a specific training example. In this paper the use of parenthesis in the subscript will indicate training examples while their absence will denote a specific variable. E.g. $x_{(i)}$ denotes training example i while x_i indicates independent variable i.

```
Repeat till convergence {
Select the pair $\alpha_{(i)}$ and $\alpha_{(\tau)}$ using some heuristic.
Re-optimise $W(\alpha)$ with respect to $\alpha_{(i)}$ and $\alpha_{(\tau)}$,
holding all other $\alpha$'s constant.
}
```

Fig. 1. The Sequential Minimal Optimization (SMO) algorithm.

Since the constraint $||w^{(j)}|| = 1$ is non-convex, the equation (8) is transformed thereby making it more suitable for convex-optimization. To achieve this recognize that if, $\hat{\gamma}^{(j)} = 1$, then $\hat{\gamma}^{(j)}/\|w^{(j)}\| = 1/\|w^{(j)}\|$, and maximizing this is equivalent to minimizing $\|w^{(j)}\|^2$. Furthermore, to avoid over-fitting the cluster data, a regularization term $\xi^{(j)}$, is added coupled with the constant C used to signify a turning parameter that weights the significance of misclassification. In addition, at this point the global reference vector w is added to the optimization problem to leverage information between clusters. Accordingly, the primal form of the general optimization problem is represented as follows;

$$\begin{aligned}&\min_{\gamma^{(j)},w^{(j)},w,b^{(j)}} \frac{1}{2}\|w\|^2 + \frac{1}{2}\sum_{j=1}^{k}\|w^{(j)} - w\|^2 + C\sum_{j=1}^{k}\sum_{i=1}^{m^{(j)}}\xi_{(i)}^{(j)},\\ &\text{s.t. } y_{(i)}^{(j)}\left(w^{(j)T}x_{(i)}^{(j)} + b^{(j)}\right) \geqslant 1 - \xi_{(i)}^{(j)}, \quad i = 1,\ldots,m^{(j)},\ \forall j,\\ &\xi_{(i)}^{(j)} \geqslant 0, \quad i = 1,\ldots,m^{(j)},\ \forall j.\end{aligned} \tag{9}$$

Note that this equation satisfies the Karush–Kuhn–Tucker (KKT) conditions and that $g_{(i)}^{(j)}(w) \leqslant 0$ is an active constraint. Hence, the constraint to the primal problem can be rewritten in the following form:

$$g_{(i)}^{(j)}(w) = y_{(i)}^{(j)}\left(w^{(j)T}x_{(i)}^{(j)} + b^{(j)}\right) + 1 - \xi_{(i)}^{(j)} \leqslant 0. \tag{10}$$

Accordingly, the dual form of the problem is developed. To do this a Lagrangian for the given optimization problem is constructed, as in (11) below,

$$\begin{aligned}L(v,w,b,\xi,\alpha,r) = &\frac{1}{2}\|w\|^2 + \frac{1}{2}\sum_{j=1}^{k}\|w^{(j)} - w\|^2 + C\sum_{j=1}^{k}\sum_{i=1}^{m^{(j)}}\xi_{(i)}^{(j)}\\ &- \sum_{j=1}^{k}\sum_{i=1}^{m}\alpha_{(i)}\left[y_{(i)}^{(j)}\left(w^{(j)T}x_{(i)}^{(j)} + b^{(j)}\right) - 1 + \xi_{(i)}^{(j)}\right]\\ &- \sum_{j=1}^{k}\sum_{i=1}^{m}r_{(i)}\xi_{(i)}^{(j)},\end{aligned} \tag{11}$$

where the α_i's and the r_i's are Lagrangian multipliers (constrained to be $\geqslant 0$). Equation (11) is minimized with respect to w and b by taking partial derivatives and setting them to zero. After minimizing the dual form, $W(\alpha)$, of the problem is obtained. This dual form is solved, in lieu of the primal, to derive the parameters $\alpha_{(i)}$ that maximize $W(\alpha)$ subject to the constraints. These parameters can then be used to find the optimal w's and once found w^* is found it is a trivial task to find the intercept term b using the primal problem.

In particular, Platt's (1999) Sequential Minimal Optimization (SMO) algorithm can be used to solve the dual. Here given a set of $\alpha_{(i)}$'s which satisfy the constraints, are updated, simultaneously so as to satisfying the constraints. The SMO algorithm is presented in Fig. 1 given below where $i \neq \tau$.

4. Methodology

4.1. Data

4.1.1. German and Barbados datasets

A German credit scoring dataset was taken from the UCI Machine Learning Repository. This dataset consists of 700 examples of creditworthy applicants and 300 examples of customers who should not have been granted credit. In addition, it presents twenty (20) features for each credit applicant.

Data collected from a Barbados based credit union was also used (Harris, 2013). This dataset measured 20 attributes and consisted of instances dating from 1997 to 2012. This sample data-file contained 21,117 examples of creditworthy individuals and 503 examples of clients who were un-creditworthy (please see the Appendix A).

4.2. Experimental approach

The data were pre-processed so as to transform all categorical data into numerical data for analysis. In addition, the data were normalized so as to improve the performance of the CSVM and the other seven (7) classifiers developed as comparators. All told, the classifiers developed in this paper include the following; logistic regression (LR), K means plus logistic regression (K means + LR), clustered support vector machine with a RBF kernel (CSVM-RBF), K means plus support vector machine with a RBF kernel (K means + SVM-RBF), support vector machine with a RBF kernel (SVM-RBF), linear clustered support vector machine (CSVM-linear), K means plus support vector machine with a linear kernel (K means + SVM-linear), and a linear support vector machine (SVM-linear). Fig. 2 below presents a high-level view of how these algorithms where implemented.

To begin model building, the each sample dataset was randomly split into two data-file—test (20%), and training and cross validation (80%). The withheld test dataset was exclusively used to test the performance of the classification models developed. This approach gives some intuition as to the performance of the models in real world settings. The training and cross-validation dataset was used to develop the models for each classifier type.

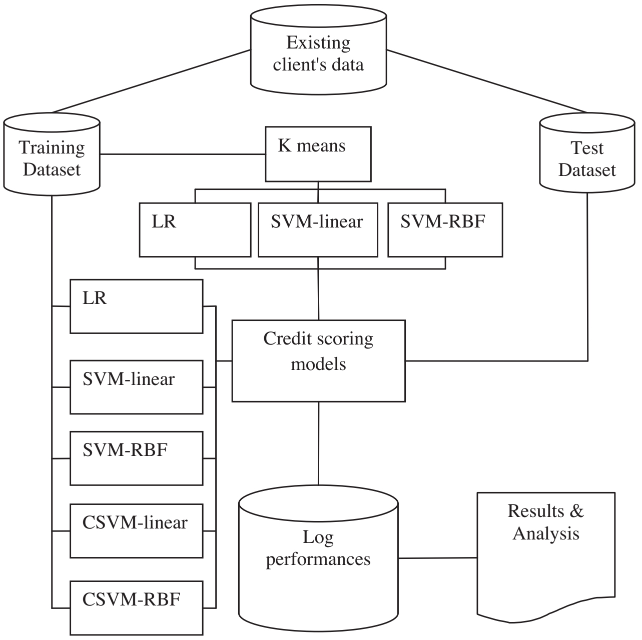

Fig. 2. High-level view of system.

Table 1
Showing comparative classifier performances on German data.

Classifier	Training accuracy	Test accuracy	BAC	AUC	Training time (s)
(1) K means + LR					
Mean	77.625	74.700	69.526	68.868	0.103
S.D	0.995	1.716	3.021	2.758	0.005
(2) LR					
Mean	70.675	68.900	71.955	70.855	0.035
S.D	0.337	1.798	2.699	2.875	0.004
(3) CSVM-RBF					
Mean	84.525	77.100	69.834	69.234	0.071
S.D	2.897	2.114	3.775	3.172	0.038
(4) K means + SVM-RBF					
Mean	83.250	76.500	69.000	68.614	0.141
S.D	2.494	1.604	3.871	3.119	0.184
(5) SVM-RBF					
Mean	83.400	78.000	70.654	69.526	0.122
S.D	2.236	1.843	3.269	2.915	0.021
(6) CSVM-linear					
Mean	79.300	76.300	71.387	70.219	0.029
S.D	0.565	2.477	2.551	2.830	0.004
(7) K means + SVM-linear					
Mean	82.233	76.381	69.238	68.752	0.107
S.D	2.514	2.105	3.790	3.089	0.046
(8) SVM-linear					
Mean	78.950	78.700	69.779	69.133	0.017
S.D	0.404	1.045	2.449	2.942	0.042

Table 2
Showing comparative classifier performances on Barbados data.

Classifier	Training accuracy	Test accuracy	BAC	AUC	Training time (s)
(1) K means + LR					
Mean	76.275	66.438	65.017	65.669	0.474
S.D	1.452	0.738	.583	.656	0.012
(2) LR					
Mean	75.903	66.122	65.055	65.896	0.131
S.D	2.054	1.683	.553	1.311	0.014
(3) CSVM-RBF					
Mean	76.452	66.442	65.022	66.053	11.328
S.D	2.086	2.009	.685	.909	0.046
(4) K means + SVM-RBF					
Mean	76.629	66.442	65.022	65.840	14.650
S.D	1.777	2.018	.673	.849	0.075
(5) SVM-RBF					
Mean	76.643	66.701	65.428	66.070	15.566
S.D	1.750	2.299	1.658	.0775	0.074
(6) CSVM-linear					
Mean	76.684	66.961	65.125	66.139	0.133
S.D	1.924	1.796	1.717	.921	0.008
(7) K means + SVM-linear					
Mean	76.737	67.004	65.068	66.054	0.492
S.D	2.026	1.775	1.737	1.034	0.037
(8) SVM-linear					
Mean	76.447	66.966	65.117	65.983	0.078
S.D	1.890	1.775	1.747	1.07	0.052

To address the imbalance nature of both datasets, the training and cross validation data-files were use to produce a new sample training and cross validation$_\alpha$ for each dataset. This sample was produced by oversampling the un-creditworthy class so as to produce an approx. 1:1 class ratio.

The training and cross validation$_\alpha$ data-file was used for parameter selection (*Gamma* and *C*). Here, five (5) fold cross-validation grid search technique was used to select the parameters *Gamma* and *C* for the support vector machine based algorithms that maximized the AUC on the training and cross validation$_\alpha$ dataset. These optimal parameters *Gamma** and C^* were used to build the credit-scoring models using the training dataset. In addition, the number of clusters, *K*, was set equals to two (2) throughout this experiment.

In total 35 credit scoring models were built for each classifier type. The performances of the models were evaluated using the withheld test dataset and these results are presented in Section 4 where each classifier's mean performance and standard deviation is reported.

Table 3
Showing summary the ANOVA computed for the eight groups of German classifiers.

	Sum of squares	df	Mean square	F	Sig.
ANOVA					
Between groups	202.732	7.000	28.962	3.284	0.002
Within groups	2398.705	272.000	8.819		
Total	2601.437	279.000			

Table 4
Showing summary the ANOVA computed for the eight groups of Barbados classifiers.

	Sum of squares	df	Mean square	F	Sig.
ANOVA					
Between groups	5.727	7.000	.818	.888	.517
Within groups	250.688	272	.922		
Total	256.414	279			

4.3. Measures

It has been previously noted that when building and reporting on credit scoring models, it is prudent to make a distinction between metrics used during (i) training phase and (ii) the reporting phase (Harris, 2013). The reason for this being that one needs to be clear as to which metric(s) was (were) used to select model parameters. Consistent with Harris (2013) the term evaluation-metric will be used when referring to the metric used during the training phase, and the term performance-metric used to refer to the measure used to report models performance at the reporting phase.

The Area under the Receiver Operating Characteristic (ROC) curve (AUC) is designated as the primary model evaluation metric and performance metric in this study. The AUC makes use of the ROC curve, which is a two dimensional measure of classification performance where the sensitivity (12) (i.e. the proportion of actual positives predicted as positive) and the specificity (13) (i.e. the proportion of actual negatives that are predicted as negative), are plotted on the Y and X axis, respectively. The AUC measure is highlighted as in (14) below where, S_1, represents the sum of the ranks of the creditworthy clients. Here, a score of 100% indicates that the classifier is able to perfectly discriminate between the classes, and a score of 50% indicates a classifier of insignificant discriminatory quality.

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False negative}}, \tag{12}$$

$$\text{Specificity} = \frac{\text{True Negative}}{\text{False Positive} + \text{True Negative}}, \tag{13}$$

$$\text{AUC} = \frac{(s_1 - \text{Sensitivity}) * [(\text{Sensitivity} + 1) * 0.5]}{\text{Sensitivity} * \text{Specificity}}. \tag{14}$$

A number of other performance metrics are also used to report the performances of the classifiers developed in this paper. For example, Test accuracy, as in (15) is also reported as it measures how accurately the credit applicants on a withheld test dataset are classified.

$$\text{Test accuracy} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} + \frac{\text{True Negative}}{\text{False Negative} + \text{True Negative}}. \tag{15}$$

An arguably more meaning full measure of classifier performance is the balanced accuracy (BAC) as in (16). This measure avoids the misleading affects on accuracy caused by imbalanced datasets by showing the arithmetic mean of sensitivity and specificity. Since skewed datasets are a common occurrence with real world credit scoring datasets this measure may be more relevant.

$$\text{BAC} = \frac{\text{Specificity} + \text{Sensitivity}}{2}. \tag{16}$$

5. Results and analysis

It has been widely noted that credit-scoring is a difficult task as credit data is very often not easily separable. The nature of the credit assessment exercise entails asynchrony of information between the applicant and the assessor. As a result, credit analysts are responsible for gathering pertinent information about the loan applicant. However, very often the best efforts of the analyst are insufficient to appraise every aspect of a client's life. Hence, credit-scoring usually results in higher misclassification rates than would normally be considered acceptable (Baesens et al., 2003). The reader is asked to bear this in mind when interpreting the results presented.

5.1. Classifier performances

Tables 1 and 2 present the performances of the CSVM classifiers in addition to seven (7) other comparator classification methods built using the German and Barbados datasets, respectively. All told, thirty-five credit-scoring models for each classifier were built for each dataset. The withheld test datasets were used to report the mean and standard deviation values for each performance metric. Here results presented in Tables 1 and 2 suggest that the models built were indeed predictive of creditworthiness as indicated by AUC on the withheld test datasets.

5.2. Significance of AUC differences

To determine the significance of the differences in performance between the models ANOVA analysis was conducted. The ANOVA analysis for the eight model types built using the German data is highlighted in Table 3; while the ANOVA for the models built using the Barbados data is shown in Table 4.

Concerning the models built using the German data, the results indicate a significant difference between one or more of the classifiers (i.e. the groups) when comparing mean AUC scores ($F = 3.284$, $p < 0.05$). However, the ANOVA for the Barbados dataset showed that there was no statistically difference in the mean AUC scores ($F = .888$, $p > 0.05$).

Next, a Bonferroni post hoc test was computed to determine which classifier(s), built on the German dataset was (were) performing significantly different from each other. Table 5 illustrates the results and shows that the only significant difference was between the mean AUC scores of the logistic regression models and the SVM models with a linear kernel function. In terms of performance the CSVM models (both linear and RBF) showed comparable AUCs to the other classifiers as there was no significant difference between them and the other classifiers in terms of AUC.

5.3. Training time

Consistent with the author's expectations, the average training times for the linear CSVM models were considerably shorter than that of the other models (Please see Tables 1 and 2), particularly the K means + SVM (linear and RBF kernels), SVM-RBF, and the K means + LR models. It is interesting that the base line SVM linear models outperform the CSVM-linear classifiers in terms of training

Table 5
Showing comparisons of the German data classifiers using Bonferroni's method.

Bonferroni	Classifier (I)	1							2						
	Classifier (J)	2	3	4	5	6	7	8	1	3	4	5	6	7	8
Mean difference (I–J)		−1.987	−0.366	0.254	−0.658	−1.351	0.254	0.735	1.987	1.621	2.241	1.329	0.636	2.241	2.722
Std. Error		0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710
Sig.		0.154	1.000	1.000	1.000	1.000	1.000	1.000	0.154	0.649	0.050	1.000	1.000	0.050	0.004
95% Confidence interval	Lower bound	−4.227	−2.606	−1.986	−2.898	−3.591	−1.986	−1.505	−0.252	−0.619	0.001	−0.911	−1.603	0.001	0.482
	Upper bound	0.252	1.874	2.493	1.582	0.889	2.493	2.974	4.227	3.861	4.480	3.569	2.876	4.480	4.961
	Classifier (I)	3							4						
	Classifier (J)	1	2	4	5	6	7	8	1	2	3	5	6	7	8
Mean difference (I–J)		0.366	−1.621	0.620	−0.292	−0.985	0.620	1.101	−0.254	−2.241	−0.620	−0.912	−1.605	0.000	0.481
Std. Error		0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710
Sig.		1.000	0.649	1.000	1.000	1.000	1.000	1.000	1.000	0.050	1.000	1.000	0.688	1.000	1.000
95% Confidence interval	Lower bound	−1.874	−3.861	−1.620	−2.532	−3.225	−1.620	−1.139	−2.493	−4.480	−2.859	−3.151	−3.844	−2.240	−1.759
	Upper bound	2.606	0.619	2.859	1.948	1.255	2.859	3.340	1.986	−0.001	1.620	1.328	0.635	2.240	2.721
	Classifier (I)	5							6						
	Classifier (J)	1	2	3	4	6	7	8	1	2	3	4	5	7	8
Mean difference (I–J)		0.658	−1.329	0.292	0.912	−0.693	0.912	1.393	1.351	−0.636	0.985	1.605	0.693	1.605	2.086
Std. Error		0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710
Sig.		1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.688	1.000	0.688	0.100
95% Confidence interval	Lower bound	−1.582	−3.569	−1.948	−1.328	−2.933	−1.328	−0.847	−0.889	−2.876	−1.255	−0.635	−1.547	−0.635	−0.154
	Upper bound	2.898	0.911	2.532	3.151	1.547	3.151	3.632	3.591	1.603	3.225	3.844	2.933	3.844	4.325
	Classifier (I)	7							8						
	Classifier (I)	1	2	3	4	5	6	8	1	2	3	4	5	6	7
Mean difference (I–J)		−0.254	−2.241	−0.620	0.000	−0.912	−1.605	0.481	−0.735	−2.722	−1.101	−0.481	−1.393	−2.086	−0.481
Std. Error		0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710
Sig.		1.000	0.050	1.000	1.000	1.000	0.688	1.000	1.000	0.004	1.000	1.000	1.000	0.100	1.000
95% Confidence interval	Lower bound	−2.493	−4.480	−2.859	−2.240	−3.151	−3.844	−1.759	−2.974	−4.961	−3.340	−2.721	−3.632	−4.325	−2.721
	Upper bound	1.986	−0.001	1.620	2.240	1.328	0.635	2.721	1.505	−0.482	1.139	1.759	0.847	0.154	1.759

time. However, the results indicate that the linear CSVM models consistently outperforms its direct comparators, which are the *K* means + SVM-linear models, SVM-RBF models, and *K* means + SVM-RBF models.

6. Conclusion

This paper introduces the use of the CSVM for credit scoring. The CSVM represents a possible solution to the limitations of the current crop of classifiers used in practice. Prior work has noted that as datasets get large nonlinear approaches become increasingly computationally expensive. As a result, the search for more computationally efficient algorithms has intensified in recent years as data analyst seek to discover patterns in datasets of increasing size and complexity without seeding classifier performance.

The results of this paper suggest that the CSVM compare well with nonlinear SVM based techniques in terms of AUC, while outperforming them in terms of training time. This is because CSVM splits the data into several clusters before training a linear support vector machine classifier to each cluster. This enables local weighting of the classifier for fast classification. In addition, the CSVM's global regularization requires that the locally weighted vectors be aligned with a global weight vector, thereby ensuring generalisability. Thus the CSVM's cutting edge performance coupled with its comparatively cheap computational cost makes it an interesting algorithm in the credit scoring space.

The future work of this author will seek to improve the classification performance of the CSVM algorithm in terms of AUC and mean model training time. In addition, other metrics will be used as the primary model evaluation metric. Furthermore, future studies will consider the impact of extending the clustered approach to other classification techniques.

Appendix A

List of client features.

German dataset
The status of the client's existing checking account
The duration of the credit period in months,
The client's credit history
The purpose for the credit
The credit amount requested
The client's savings account/bonds balance
The client's present employment status
The client's personal (marital) status and sex
Whether the client is a debtor or guarantor of credit granted by another institution
The number of years spent at present residence
The type of property possessed by client,
The client's age in years
Whether the client has other installment plans
The client's housing arrangements (i.e. own their home, rent, or live for free)
The number of existing credits the client has at the bank
The client's job
The number of people for whom the client is liable to provide maintenance for
Whether the client has a telephone
Whether the client is a foreign worker
Barbados dataset
The number of months at current address,
The applicant's marital status
The number of dependents
The age of first dependent
The age of second dependent
The age of third dependent
The age of fourth dependent
The age of sixth dependent
The age of seventh dependent
The age of eight dependent
The age of ninth dependent
The age of tenth dependent
The applicant's employment status
The number of years employed with current employer
The loan amount
The loan purpose
The loan type
The applicant's monthly income
The applicants monthly expenditure

References

Abdou, H. A. H. (2009). Credit scoring models for Egyptian banks: Neural nets and genetic programming versus conventional techniques (Ph.D. Thesis). The University of Plymouth, UK.

Abdou, H. A., & Pointon, J. (2009). Credit scoring and decision making in Egyptian public sector banks. *International Journal of Managerial Finance, 5*(4), 391–406.

Abdou, H. A., & Pointon, J. (2011). Credit scoring, statistical techniques and evaluation criteria: A review of the literature. *Intelligent Systems in Accounting, Finance and Management, 18*(2–3), 59–88.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance, 23*(4), 589–609.

Altman, E. I. (1986). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance, 23*(4), 589–609.

Arminger, G., Enache, D., & Bonne, T. (1997). Analyzing credit risk data: A comparison of logistic discrimination, classification tree analysis, and feedforward networks. *Computational Statistics, 12*(2).

Baesens, B., van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society, 54*, 1082–1088.

Bellotti, T., & Crook, J. (2009). Support vector machines for credit scoring and discovery of significant features. *Expert Systems with Applications, 36*(2), 3302–3308.

BGFRS (2013). Consumer Credit Release Board of Governors of the Federal Reserve Systemo. Document Number).

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

Bordes, A., Ertekin, S., Weston, J., & Bottou, L. O. (2005). Fast kernel classifiers with online and active learning. *The Journal of Machine Learning Research, 6*, 1579–1619.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth & Brooks.

Coffman, J. Y. (1986). The proper role of tree analysis in forecasting the risk behavior of borrowers. *Management Decision Systems, Atlanta, MDS Reports*, 3(4), 7.

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

Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research, 183*(3), 1447–1465.

Desai, V. S., Crook, J. N., & Overstreet, G. A. Jr., (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research, 95*(1), 24–37.

Durand, D. (1941). *Risk elements in consumer instalment financing*. NY: National Bureau of Economic Research.

Eisenbeis, R. A. (1978). Problems in applying discriminant analysis in credit scoring models. *Journal of Banking & Finance, 2*(3), 205–219.

Falbo, P. (1991). Credit-scoring by enlarged discriminant models. *Omega, 19*(4), 275–289.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Human Genetics, 7*(2), 179–188.

Gately, E. (1995). *Neural networks for financial forecasting: Top techniques for designing and applying the latest trading systems*. John Wiley & Sons, Inc.

Greene, W. (1998). Sample selection in credit-scoring models. *Japan and the World Economy, 10*(3), 299–316.

Gu, Q., & Han, J. (2013). Clustered support vector machines. In *Paper presented at the proceedings of the sixteenth international conference on artificial intelligence and statistics*.

Hand, D. J. (2006). Classifier technology and the illusion of progress. *Statistical science, 21*(1), 1–14.

Hand, D. J., & Jacka, S. D. (1998). Statistics in finance. Arnold London.

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society: Series A (Statistics in Society), 160*(3), 523–541.

Hand, D. J., Oliver, J. J., & Lunn, A. D. (1998). Discriminant analysis when the classes arise from a continuum. *Pattern Recognition, 31*(5), 641–650.

Hand, D. J., Sohn, S. Y., & Kim, Y. (2005). Optimal bipartite scorecards. *Expert Systems with Applications, 29*(3), 684–690.

Harris, T. (2013). Quantitative credit risk assessment using support vector machines: Broad versus Narrow default definitions. *Expert Systems with Applications, 40*(11), 4404–4413.

Henley, W. E. (1994). *Statistical aspects of credit scoring*. Open University.

Hosmer, D. W., & Lemeshow, S. (1989). The multiple logistic regression model. *Applied Logistic Regression, 1*, 25–37.

Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications, 33*(4), 847–856.

Irwin, G. W., Warwick, K., & Hunt, K. J. (1995). *Neural network applications in control* No. 53. Iet.

Jentzsch, N. (2006). *The economics and regulation of financial privacy [electronic resource]: An international comparison of credit reporting systems*. Springer.

Leonard, K. J. (1995). The development of credit scoring quality measures for consumer credit applications. *International Journal of Quality & Reliability Management, 12*(4), 79–85.

Masters, T. (1995). *Advanced algorithms for neural networks: A C++ sourcebook*. John Wiley & Sons, Inc.

Mays, E. (2001). *Handbook of credit scoring*. Chicago, IL: The Glenlake Publishing Company.

Orgler, Y. E. (1970). A credit scoring model for commercial loans. *Journal of Money, Credit and Banking, 2*(4), 435–445.

Paliwal, M., & Kumar, U. A. (2009). Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications, 36*(1), 2–17.

Platt, J. (1999). *Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods support vector learning*. Cambridge, MA: AJ, MIT Press (pp. 185–208). Cambridge, MA: AJ, MIT Press.

Raiffa, H., & Schlaifer, R. (1961). Applied statistical decision theory, Harvard Business School Publications.

Rosenberg, E., & Gleit, A. (1994). Quantitative methods in credit management: A survey. *Operations Research, 42*(4), 589–613.

Sarlija, N., Bensic, M., & Bohacek, Z. (2004). Multinomial model in consumer credit scoring. In *Paper presented at the 10th international conference on operational research KOI 2004*.

Saunders, A., & Allen, L. ((1998). *Credit risk measurement: New approaches to value at risk and other paradigms*. New York: John Wiley and Sons (March, 2002).

Siddiqi, N. (2005). *Credit risk scorecards: Developing and implementing intelligent credit scoring* (Vol. 3). Wiley.com.

Smith, K. A., & Gupta, J. N. D. (2003). Neural networks in business: Techniques and applications. *IGI Global*.

Steenackers, A., & Goovaerts, M. J. (1989). A credit scoring model for personal loans. *Insurance: Mathematics and Economics, 8*(1), 31–34.

Wang, Z., Yan, S. C., & Zhang, C. S. (2011). Active learning with adaptive regularization. *Pattern Recognition, 44*(10–11), 2375–2383.

Yu, L. (2008). *Bio-inspired credit risk analysis: Computational intelligence with support vector machines*. Springer.

Zekic-Susac, M., Sarlija, N., & Bensic, M. (2004). Small business credit scoring: A comparison of logistic regression, neural network, and decision tree models. In *Paper presented at the 26th international conference on information technology interfaces, 2004*.

Zhou, L., Lai, K. K., & Yu, L. (2010). Least squares support vector machines ensemble models for credit scoring. *Expert Systems with Applications, 37*(1), 127–133.