Applied Soft Computing 19 (2014) 31–40

Contents lists available at ScienceDirect

Applied Soft Computing

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

Churn prediction using comprehensible support vector machine: An analytical CRM application

M.A.H. Farquad [a,b,c], Vadlamani Ravi [a,*], S. Bapi Raju [b]

[a] Institute for Development and Research in Banking Technology, Castle Hills Road #1, Masab Tank, Hyderabad 500057, AP, India
[b] Department of Computer and Information Sciences, University of Hyderabad, Hyderabad 500046, AP, India
[c] School of Business, The University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 9 February 2011
Received in revised form 20 August 2013
Accepted 19 January 2014
Available online 4 February 2014

Keywords:
Churn prediction
Support vector machine
Rule extraction
Naive Bayes Tree
Machine learning and customer relationship management

ABSTRACT

Support vector machine (SVM) is currently state-of-the-art for classification tasks due to its ability to model nonlinearities. However, the main drawback of SVM is that it generates "black box" model, i.e. it does not reveal the knowledge learnt during training in human comprehensible form. The process of converting such opaque models into a transparent model is often regarded as *rule extraction*. In this paper we proposed a hybrid approach for extracting rules from SVM for customer relationship management (CRM) purposes. The proposed hybrid approach consists of three phases. (i) During first phase; SVM-RFE (SVM-recursive feature elimination) is employed to reduce the feature set. (ii) Dataset with reduced features is then used in the second phase to obtain SVM model and support vectors are extracted. (iii) Rules are then generated using Naive Bayes Tree (NBTree) in the final phase. The dataset analyzed in this research study is about Churn prediction in bank credit card customer (Business Intelligence Cup 2004) and it is highly unbalanced with 93.24% loyal and 6.76% churned customers. Further we employed various standard balancing approaches to balance the data and extracted rules. It is observed from the empirical results that the proposed hybrid outperformed all other techniques tested. As the reduced feature dataset is used, it is also observed that the proposed approach extracts smaller length rules, thereby improving the comprehensibility of the system. The generated rules act as an early warning expert system to the bank management.

1. Introduction

CRM is a process or methodology used to learn more about customers' need and behaviours in order to develop stronger relationship with them. CRM involves the continuous use of refined information about current and potential customers in order to anticipate and respond to their needs and draws on a combination of business process and Information Technology to discover the knowledge about the customers and answer questions like, "*who are the customers?*", "*what do they do?*" and "*what they like?*". Therefore the effective management of information and knowledge is central and critical to the concept of CRM for;

- Product tailoring and service innovation (web-sites tailored to customer needs, taste experience and the development of mass customisation).
- Providing a single and consolidated view of the customer.
- Calculating the lifetime value of the customer.
- Designing and developing personalized transactions.
- Multichannel based communication with the customer.
- Cross-selling/up-selling various products to customers.

Different definition of CRM put emphasis on different perspectives. CRM's technological perspective was stressed in [1], its knowledge management perspective was emphasized in [2] and its business re-engineering and continuous improvement perspective is presented in [3].

We can think about CRM at three levels, strategic, analytical and collaborative.

Strategic CRM: It is focused on development of a customer-centric business culture. Product, production and selling are the three major business orientations identified by Kotler [4].

Analytical CRM: Analytical CRM builds on the foundation of customer information. Customers' data may be found in enterprise wide repositories, sales data (purchasing history), financial data (payment history and credit score), marketing data (campaign response, loyalty scheme data) and service data. With the application of Data Mining, the industry can then interrogate this data and

* Corresponding author. Tel.: +91 40 2329 4042.
E-mail addresses: farquadonline@gmail.com (M.A.H. Farquad), rav_padma@yahoo.com, padmarav@gmail.com (V. Ravi), bapics@uohyd.ernet.in (S.B. Raju).

1568-4946/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2014.01.031

intelligent interrogation provides answers to questions, such as, "who are our most valuable customers?", "which customer have the highest propensity to switch to competitors?", "which customers would be most likely to respond to particular offer?" and so on.

Collaborative CRM: Staff members from different departments can share information collected when interacting with customers [5].

Churn prediction problem is an analytical CRM application and using the extracted rules from SVM, service providers can get transparent and efficient insight about their customers and can make better policies to retain their existing customers.

1.1. Churn prediction problem

Over the decade and half, the number of customers with banks and financial companies is increasing by the day and this has made the banks conscious of the quality of the services they offer. The phenomenon, called '*churn*' i.e. shifting loyalties from one service provider to another occurs due to reasons such as availability of latest technology, customer-friendly bank staff, low interest rates, proximity of geographical location and varied services offered. Hence, there is a pressing need to develop models that can predict which existing '*loyal*' customer is going to churn out or attrite in near future.

Service organizations need to be proactive in understanding the customers' current satisfaction levels before they attrite [6]. Research indicates that the online bank customers are less price-conscious than traditional bank customers with less probability of churning out [7]. Targeting customers on the basis of their (changing) purchase behaviour could help the organizations do better business and loyalty reward programmes helps the organizations build stronger relationships with customers [8].

In the financial services industry two "critical" churn periods are identified [9], the first period is the early years after becoming a customer and the second period is after being a customer for some 20 years. A comparative study on Logistic Regression and Neural Network for subscriber data of a major wireless carrier is carried out and it is concluded that using sophisticated neural net $93 could be saved per subscriber [10].

Machine learning techniques such as; multilayer perceptron, Hopfield neural network, self-organizing neural networks [11], decision tree [12], multivariate regression analysis [13], logistic regression and random forest [14], emergent self-organizing feature maps (ESOM) [15], neural networks [10], SVM [18], genetic algorithms and rough set theory [16], ensemble with majority voting [17] and hybrid neural networks [18] are employed to solve churn prediction problems. Glady et al., proposed a churn prediction model using customer life time value (CLV), which is defined as the discounted value of future marginal earning, based on customers' activity [19]. Hu presented a comparative study of different machine learning algorithms [20]. The trend in marketing towards building relationships with customers continues to grow and marketers have become increasingly interested in retaining customers over the long run [21]. Hyung-Su and Young-Gul suggested a performance measurement framework called CRM score card to diagnose and asses a firm's CRM practice [22].

Churn prediction problem is one of the most important applications of analytical CRM in finance. Banks would be interested to know their *about-to-churn* customers and the proposed rule extraction approach do not only provide better predictions but also comprehensibility of the system is improved. Feature selection using SVM-RFE algorithm in the first phase reduces the dimensionality of the data by yielding the key attributes in the data. Thus, less number of rules and smaller rules are extracted resulting in the improvement of the comprehensibility of the system. During the research study in this Paper, various standard balancing techniques are employed such as, random under-sampling, random over-sampling and SMOTE.

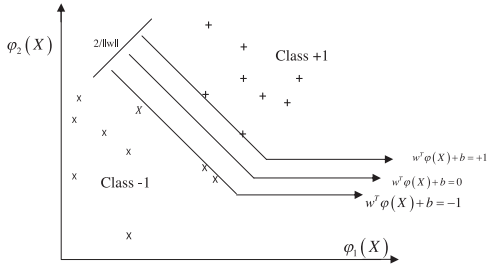

Fig. 1. Illustration of SVM optimization of the margin in the feature space.

Remaining paper is organized as follows. Section 2 provides the details about SVM and literature survey of rule extraction from SVM. Proposed rule extraction approach is then detailed in Section 3. Dataset description and experimental setup followed during this research study is presented in Section 4. Next section provides the detailed empirical analysis and observations. Section 6 concludes the paper.

2. Rule extraction from SVM

2.1. Support vector machine

The SVM is a learning procedure based on the statistical learning theory [23]. It has been used in wide variety of applications such as gene analysis [24], financial time-series forecasting [25], marketing [26], patent classification [27], face recognition [28] and predicting longitudinal dispersion coefficients in natural rivers [29].

For classification problems, the main objective of SVM is to find an optimal separating hyperplane that correctly classifies data points as much as possible and separates the points of two classes as far as possible, by minimizing the risk of misclassifying the training samples and unseen test samples [30]. The training points that are closest to the optimal separating hyperplance are called support vectors and other training examples are irrelevant for determining the binary class boundaries as shown in Fig. 1. To deal with non-linearly separable datasets problems, SVM first projects the data into a higher dimensional feature space and tries to find the linear margin in the new feature space.

Given a set of points $b \in \Re$ with $i = 1, \ldots, N$ each point x_i belongs to either of two classes with the label $y_i \in \{-1, +1\}$ [31].

The optimization problem for the SVM can be depicted as follows:

$$\min \frac{1}{2}\langle w, w\rangle \quad \text{Subject to } y_i(w \cdot x_i + b) \geq 1 \ \forall x_i \tag{1}$$

The SVM classification function for classifying linearly separable data can be written as:

$$f(x) = \langle w, x\rangle + b = \sum_{i=1}^{l} y_i \alpha_i \langle x_i, x\rangle + b; \tag{2}$$

This is also known as hard margin, where no room is given for errors. It is observed that most of the time it is linearly non-separable. Hence slack variable ξ is introduced to allow ξ error and the optimization function takes the form of (3) as shown below:

$$\min \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{l}\xi_i \quad \text{Subject to } y_i(w \cdot x_i + b) \geq 1 \ \forall x_i \tag{3}$$

To deal with the problem of non-linearly separable dataset, SVM first projects the data into a higher dimensional feature space using various kernels and tries to find the linear margin in the new feature space. The optimization function can be depicted as shown below:

$$\min \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{l}\xi_i \quad \text{Subject to } y_i(w\cdot\varphi(x)+b)\geq 1\ \forall x_i \tag{4}$$

The optimal hyperplane separating the binary decision classes is given by (5):

$$f(x) = \sum_{i=1}^{l} y_i\alpha_i K(x_i, x) + b \tag{5}$$

where $K(X_i, X) = \varphi(X_i)^T\varphi(X)$ is taken with a semi-positive definite kernel satisfying the Mercer theorem [32].

For kernel function $K(.,.)$ one typically has the following choices:

$$K(X, X_i) = (X_i^T \cdot X), \quad \text{(Linear kernel)} \tag{6}$$

$$K(X, X_i) = \left(\frac{X_i^T \cdot X + 1}{C}\right)^d, \quad \text{(Polynomial kernel of degree } d\text{)} \tag{7}$$

$$K(X, X_i) = \exp\left\{-\frac{(X - X_i)^2}{\sigma^2}\right\}, \quad \text{(RBF kernel)} \tag{8}$$

$$K(X, X_i) = \tanh(\kappa X_i^T \cdot X + \theta), \quad \text{(MLP kernel)} \tag{9}$$

where d, C, σ, κ and θ are constants.

2.2. NBTree: the hybrid algorithm

The algorithm is similar to the classical recursive partitioning schemes, except that the leaf nodes created are Naive-Bayes categorizers instead of nodes predicting a single class. A threshold for continuous attributes is chosen using the standard entropy minimization technique. The utility of a node is computed by discretizing the data and computing the 5-fold cross validation accuracy estimate of using Naive-Bayes at the node.

Intuitively, it is attempted to approximate whether the generalization accuracy for a Naive-Bayes classifier at each leaf is higher than a single Navie-Bayes classifier at the current node. To avoid splits with little value, it is defined that a split to be significant if the relative reduction in error is greater than 5% and there are at least 30 instances in the node.

2.3. The algorithm

Input: a set T of labelled instances.

Output: a decision tree with Navie-Bayes categorizers at the leaves.

1 For each attribute X_i, evaluate the utility, $u(X_i)$, of a split on attribute X_i, for continuous attributes, a threshold is also found at this stage.
2 Let $j = argmax_i(u_i)$, i.e. the attribute with the highest utility.
3 If u_j is not significantly better than the utility of the current node, create a Naive-Bayes classifier for the current node and return.
4 Partition T according to the test on X_j. If X_j is continuous, a threshold split is used; if X_j is discrete, a multi-way split is made for all possible values.
5 For each child, call the algorithm recursively on the portion of T that matches the test leading to the child.

2.4. Rule extraction from SVM

Although SVM generally predicts well, it is still a *black box* model, i.e. the knowledge learnt by SVM during training is not directly interpretable by the user. Many researchers tried to treat this *accuracy* vs. *comprehensibility* trade-off by converting the *black box*, high accurate models to transparent models via *rule extraction* in the context of neural networks [33].

Extensive work was reported in developing rule extraction techniques for neural networks [34] but less work is reported in rule extraction from SVM. In general, rule extraction techniques are divided into two major groups i.e. decompositional and pedagogical. *Decompositional* techniques view the model at its minimum (or finest) level of granularity (at the level of hidden and output units in case of ANN). Rules are first extracted at individual unit level, these subset of rules are then aggregated to form global relationship. *Pedagogical* techniques extract global relationship between the input and the output directly without analyzing the detailed characteristics of the underlying solution. The third group for rule extraction techniques is *eclectic* which combines the advantages of the decompositional and pedagogical approaches. Using rule extraction a learning system might discover salient features in the input data whose importance was not previously recognized [35].

We now briefly review the works reported in rule extraction from SVMs. SVM + prototype determine the prototype vectors (also known as cluster centres) for each input class using K-means clustering algorithm [36]. Based on the centre of the cluster, named prototype, and the farthest support vector, interval or ellipsoid, if-then rules can be created. The main drawback of this algorithm is that the extracted rules are neither exclusive nor exhaustive which results in conflicting or missing rules for classification of new data instances. Fung et al. developed a rule extraction algorithm, which extracts non-overlapping rules by constructing hyperplane with axis parallel surface [37]. They first transformed the problem to a simpler, equivalent variant and constructed the hyper-cubes by solving linear programs and each hypercube is then transformed to an if-then rule.

RulExtSVM [38] comprising three steps is used for extracting *if-then* rules using intervals defined by hyper-rectangular forms. First step is the generation of a hyper-rectangle using the intersection of the support vectors with the SVM decision boundary. During second step, the initial rule set is tuned in order to improve rule accuracy. In the final step redundant rules have been removed to obtain more concise rule set. The disadvantage of this algorithm is the construction of as many hyper-rectangles as the number of support vectors that can be computationally very expensive. Later, a hybrid rule extraction technique is proposed where after developing the SVM model using training set, they used the developed model to predict the output class labels for the training instances [39,40]. Using decision tree (C4.5) rules are generated. The quality of the extracted rules is then measured using the Area Under the Receiver Operating Characteristic Curve (AUC) [41].

Hyper-rectangle rules extraction (HRE) approach first constructs hyper rectangles according to the prototypes and the support vectors (SVs), then these hyper-rectangles are projected onto coordinate axes and *if–then* rules are generated [42]. Fuzzy rule extraction (FREx) applies triangular fuzzy membership function and determines the projection of the support vectors in the coordinate axes and each support vector is then transformed into a rule [43]. Later, a multiple kernel-support vector machine (MK-SVM) is proposed to improve the explanation capacity of SVM [44]. SQRex-SVM is used to directly extract the rules from support vectors extracted using SVM [45].

Recently, Farquad et al. proposed a hybrid rule extraction approach using SVM and the extracted rules are tested for bankruptcy prediction in banks [46,47]. They first extracted the

support vectors and then used these support vectors along with their corresponding actual target values to train fuzzy rule based system, decision tree and radial basis function network. They concluded that the hybrid SVM + FRBS (fuzzy rule based systems) outperformed the stand-alone classifiers. Then a new active learning-based approach (ALBA) to extract rules from SVM models is proposed [48]. Using the support vectors extracted they generate more number of training instances which are near to support vectors. In other words more number of instances are generated which are near the decision boundary of SVM. Late, using C4.5 and RIPPER (repeated incremental pruning to produce error reduction) rules are generated. Most Recently, Farquad et al. proposed an eclectic rule extraction approach to extract rules for solving regression problems [49]. Where, they employed CART, ANFIS and DENFIS for rule extraction purpose.

2.5. Motivation for the proposed approach

Farquad et al. proposed an eclectic procedure for extracting rules from SVM which deals with unbalanced and medium scale problems [50]. They first extracted the support vectors and their corresponding target values are then replaced by the predictions given by SVM. This modified data set is then fed to NBTree to extract rules. Their approach was evaluated using bank credit cards data for predicting churn without employing any balancing technique. They concluded that using support vectors the comprehensibility of the rules is improved. In this paper, similar framework is proposed with the addition of feature selection module and standard data balancing techniques for churn prediction problem. The study presented in this paper can be considered as an extensive analysis of rule extraction from SVM and evaluation of the efficiency of feature selection using SVM-RFE applied to churn prediction problem.

3. Proposed rule extraction approach

Churn prediction in bank credit card customers' problem is solved using the proposed approach. The churn prediction dataset is highly unbalanced with 93:7 class distributions where 93% of the samples are available for loyal customers and only 7% of the data is available to learn about churn customers. The churn prediction dataset is obtained from Chile in 2004, information about the dataset attributes is presented in Table 1. Balancing techniques such as, SMOTE, random under-sampling, random over-sampling and combined under-sampling and over-sampling are employed. The efficiency of SVM for feature selection and rule extraction from SVM using unbalanced and balanced data is analyzed. Extracting support vectors and feature selection using SVM generates vertically and horizontally reduced data. This newly generated data is then used for rule generation.

The proposed hybrid approach is composed of three phases and is depicted in Fig. 2.

1 Feature selection using SVM-RFE.
2 Support vector extraction using SVM.
3 Rule generation using NBTree.

3.1. Feature selection using SVM-RFE

SVM-RFE (recursive feature elimination) [24] algorithm is employed for feature selection purpose. Nested subsets of features are selected in a sequential backward elimination manner, which starts with all the features variables and removes one feature variable at a time. At each step, the coefficients of the weight vector w of a linear SVM are used to compute the feature ranking score. The feature say, the ith feature with the smallest ranking

Table 1
Feature description of churn prediction data set.

Feature	Description	Value
Target	Target variable	0 – Non-Churner 1 – Churner
CRED_T	Credit in month T	Positive real number
CRED_T-1	Credit in month T-1	Positive real number
CRED_T-2	Credit in month T-2	Positive real number
NCC_T	Number of credit cards in months T	Positive integer value
NCC_T-1	Number of credit cards in months T-1	Positive integer value
NCC_T-2	Number of credit cards in months T-2	Positive integer value
INCOME	Customer's income	Positive real number
N_EDUC	Customer's educational level	1 – University student 2 – Medium degree 3 – Technical degree 4 – University degree
AGE	Customer's age	Positive integer
SX	Customers sex	1 – Male 0 – Female
E_CIV	Civilian status	1 – Single 2 – Married 3 – Widow 4 – Divorced
T_WEB_T	Number of web transaction in months T	Positive integer
T_WEB_T-1	Number of web transaction in months T-1	Positive integer
T_WEB_T-2	Number of web transaction in months T-2	Positive integer
MAR_T	Customer's margin for the company in months T	Real number
MAR_T-1	Customer's margin for the company in months T-1	Real number
MAR_T-2	Customer's margin for the company in months T-2	Real number
MAR_T-3	Customer's margin for the company in months T-3	Real number
MAR_T-4	Customer's margin for the company in months T-4	Real number
MAR_T-5	Customer's margin for the company in months T-5	Real number
MAR_T-6	Customer's margin for the company in months T-6	Real number

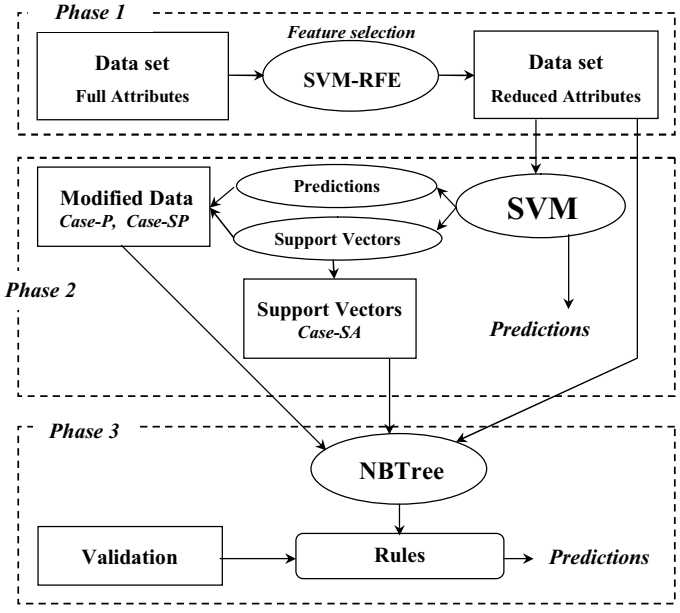

Fig. 2. (1) Rule extraction using selected attributes of data. *Note*: *Case – P*: training set with corresponding predicted target values. *Case – SA*: Support vectors with corresponding actual target values. *Case – SP*: Support vectors with corresponding predicted target values.

score $c_1 = (w)^2$ is eliminated, where w represents the corresponding component in the weight vector w. Using $c = (w)^2$ as the ranking criterion corresponds to removing the feature whose removal changes the objective function the least. This objective function is chosen to be $J = 1/2||w||^2$ in SVM-RFE.

3.2. Support vector extraction using SVM

Dealing with churn prediction data, sensitivity of the classifier is considered the most important factor for developing the SVM model and for extracting support vectors. Later, the corresponding actual target values of support vectors are replaced by the predicted target values of SVM, resulting in *Case-SP* dataset whereas support vectors with corresponding actual target values is called *Case-SA* dataset. For comparative study, the corresponding actual target values of training instances are also replaced by the predictions of SVM model, resulting in *Case-P* dataset. By using the newly generated *Case-P* and *Case-SP* we ensure that the rules extracted actually represent the knowledge learnt by the SVM.

3.3. Rule generation using NBTree

NBtree is employed for rule generation purpose [51]. It attempts to utilize the advantages of both decision trees (i.e. segmentation) and naïve bayes (evidence accumulation from multiple attributes). A decision tree is built with univariate splits at each node, but with Navie-Bayes classifiers at the leaves instead of the predictions for single class. It is concluded that NBTree's induction process is useful for larger datasets [51]. Rules are generated under 10-fold cross validation method of testing and the generated rules are later tested against the validation set.

4. Experimental setup

4.1. Dataset description

The churn prediction dataset is obtained from a Latin American Bank that suffered from an increasing number of churns with respect to their credit card customers and decided to improve its retention system. Two groups of variables are available for each customer: socio-demographic and behavioural data, which are described in Table 1. The dataset comprises of 22 variables, with 21 predictor variables and 1 class variable. It consists of 14,814 instances, of which 13,812 instances are pertaining to loyal customers and 1002 instances represent churned customers. Thus, there are 93.24% loyal customers and 6.76% churned customers. Hence, the dataset is highly unbalanced in terms of the proportion of churners vs. non-churners [52].

4.2. Data imbalance problem

In many real time problems, almost all the instances belong to one class, while far fewer instances are labelled as the other class, usually the more important class. It is obvious that traditional classifier seeking an accurate performance over a full range of instances are not suitable to deal with imbalanced learning task, since they tend to classify all the data into majority class, which is usually not the objective of the study and less important. Research studies show that many standard machine learning approaches result in poor performance, specifically dealing with large unbalanced datasets [53,54].

4.3. Random under-sampling

Under-sampling is a technique in which some of the samples belonging to the majority class are removed randomly and combined with the minority class samples. For example, 25% under-sampling means that the majority class is reduced to 25% of its original size in other words, 25% of the available majority class instances are removed randomly from data. 50% under-sampling means that the majority class is reduced to 50% of its original size.

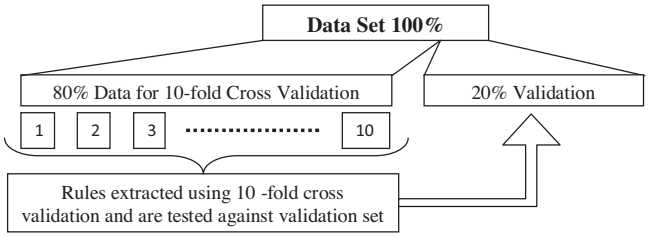

Fig. 3. Segmentation of data into validation set and 10-fold cross validation.

4.4. Random over-sampling

Oversampling is a technique in which the samples belonging to the minority class are replicated a few times and combined with the majority class samples. For example, 100% over-sampling means that the minority class instances are replicated once in other words, minority class instances are doubled, and 200% over-sampling means that the minority class is replicated twice.

4.5. Synthetic minority over-sampling technique (SMOTE)

SMOTE is an approach in which the minority class is oversampled by creating synthetic (or artificial) samples, rather than by oversampling with replacement. The minority class is oversampled by taking out each sample and introducing synthetic samples along the line segments that join any/all of the k minority class nearest neighbours. SMOTE is used to widen the data region that corresponds to minority samples. This approach effectively forces the decision region of the minority class to become more general [55].

4.6. Experimental setup

The available large scale unbalanced dataset is first divided into two parts of 80:20 ratios. 10-fold cross validation (10-FCV) is performed on the 80% of the data and 20% of the data is named as validation set and kept aside. Using validation set, efficiency of the rules generated during 10-FCV is evaluated. The class distribution proportion of the data is maintained in training data and validation data i.e. 93.24% for good customers and 6.76% for churn customers of the banks by resorting to stratified random sampling. Fig. 3 depicts the segmentation of the data and the 10-FCV structure maintained for the experiments carried out in this paper.

5. Results and discussion

Many business decision makers place high emphasis on sensitivity alone because higher sensitivity leads to greater success in correctly identifying potential churners and thereby contributing to the bottom-line of the fundamental CRM viz., retaining extant loyal customers. Consequently in this Paper, sensitivity is accorded top priority ahead of specificity and accuracy. Therefore, we evaluate and discuss the performance of our proposed hybrid approach SVM + NBTree using *Case-SP* with respect to sensitivity alone. During first phase of the proposed approach, SVM-RFE algorithm is employed for feature selection and six attributes are then selected, those are, *CRED_T* (*Credit in month T*), *CRED_T-1* (*Credit in month T-1*), *CRED_T-2* (*Credit in month T2*), *NCC_T* (*Number of Credit Cards in month T*), *NCC_T-2* (*Number of Credit Cards in month T2*) and *T_WEB_T*

Table 2
Average results obtained using original unbalanced data.

	Full features			*t*-Test	Reduced features			*t*-Test
	Sens	Spec	Acc		Sens	Spec	Acc	
SVM	64.65	80.63	79.55	3.53	82.85	72.25	74.79	1.35
NBTree	62.7	99.07	96.62	4.76	52.7	99.33	96.17	18.38
SVM + NBTree (*Case-P*)	67.65	84.9	83.74	2.39	84.7	74.59	75.25	–
SVM + NBTree (*Case-SA*)	0	100	93.25	NA	0	100	93.25	NA
SVM + NBTree (*Case-SP*)	75.45	79.03	78.79	–	82.45	75.89	76.34	1.52

Sens, sensitivity; Spec, specificity; Acc, accuracy.

Table 3
Average results obtained using 25% under-sampled data.

	Full features			*t*-Test	Reduced features			*t*-Test
	Sens	Spec	Acc		Sens	Spec	Acc	
SVM	71.75	78.42	77.92	4.38	82.5	75.58	76.06	0.26
NBTree	64.75	98.71	96.41	11.42	59.4	98.08	95.48	8.88
SVM + NBTree (*Case-P*)	78.25	78.64	78.61	0.53	82.2	76.16	76.57	–
SVM + NBTree (*Case-SA*)	0	100	93.25	NA	0	100	93.25	NA
SVM + NBTree (*Case-SP*)	79.25	76.71	76.87	–	81.2	70.98	71.67	0.41

Sens, sensitivity; Spec, specificity; Acc, accuracy.

(*Web Transaction in month T*). Balancing methods of random under-sampling, random over-sampling, combination of under-sampling and over-sampling and SMOTE are employed and an extensive study is carried out. Empirical results obtained by the proposed hybrid on validation data i.e. 20% data (refer Section 4.6) using the dataset including all features and with reduced features are presented in Tables 2–12. The classifier selected as best performer carries hyphen for *t*-test values in results tables and if the sensitivity yielded by classifier is very low or near 0% then that classifier is not considered to calculate *t*-test and it carried not-applicable (NA).

Table 2 presents the results obtained using original unbalanced data. It is observed from the empirical results that the results obtained using reduced feature data is better compared to their corresponding approaches with full feature data. It is also observed that the hybrid using *Case-P* dataset with reduced feature yielded slightly better sensitivity compared to the hybrid using *Case-SP* but *t*-statistics values reveal that their results as insignificant. As the number of rules extracted using *Case-P* data is above hundred, thus the comprehensibility of the system is very poor, hence it is not advisable. Whereas the proposed approach SVM + NBTree using *Case-SP* dataset with reduced feature yielded marginally less sensitivity compared to *Case-P* dataset but the number of rules extracted are approximately twenty, resulting in improved comprehensibility of the system. It is observed that 25% under-sampling data also resulted in similar observations such as the results of unbalanced data and are presented in Table 3. Balancing using 25% under-sample data is not facilitating any improvement towards the efficiency of the system in neither in terms of sensitivity nor the number of rules extracted. Yet again, feature selection improved the sensitivity of all the hybrid methods compared to other corresponding hybrids using full features.

Table 4 presents the results obtained using 50% under-sampling data. It is observed that considering all the features of the data, the hybrid SVM + NBTree using *Case-SP* set yielded the best sensitivity of 87.25%. Whereas the proposed hybrid SVM + NBTree using *Case-SP* set with reduced feature set obtained the sensitivity of 78.95%. It is observed that 50% under-sampling degrades the performance of the hybrids with reduced features. When under-sampling is employed it is observed that using 25% under-sampling with reduced feature data yielded better sensitivity compared to their corresponding approaches using full feature data. Whereas using 50% under-sampling the hybrid using full features data performed the best. Using 50% under-sampling the class distribution ratio become 87:14, where 87% represent loyal customers' instances and 14% of the instances represent churned customers.

Results obtained using 100%, 200% and 300% over-sampling data are presented in Tables 5–7, respectively. It is observed that using over-sampling data, the proposed approach SVM + NBTree with reduced features yielded better sensitivity compared to SVM + NBTree using full feature data. It is also observed that the hybrid SVM + NBTree using *Case-SP* data with reduced yielded statistically insignificant sensitivity and extracted less number of rules compared to SVM + NBTree using *Case-P* data with reduced features. Among various experiments conducted using different over-sampling percentages, it is observed that the proposed hybrid using 200% over-sampled data yielded the best sensitivity. Using 200% over-sampling the distribution of the classes become 82:18 where 82% of the instances are available for loyal customers and 18% instances are available for churned customers.

Table 4
Average results obtained using 50% under-sampled data.

	Full features			*t*-Test	Reduced features			*t*-Test
	Sens	Spec	Acc		Sens	Spec	Acc	
SVM	78.35	72.84	73.21	5.02	77.5	91.78	90.85	1.15
NBTree	69.4	98.21	96.26	11.9	70.5	96.33	94.59	6.29
SVM + NBTree (*Case-P*)	81	75.12	75.45	2.64	77.65	91.9	90.42	0.98
SVM + NBTree (*Case-SA*)	0.05	99.99	93.24	NA	0	100	93.25	NA
SVM + NBTree (*Case-SP*)	87.25	69.83	70.99	–	78.95	91.19	90.37	–

Sens, sensitivity; Spec, specificity; Acc, accuracy.

Table 5
Average results obtained using 100% over-sampled data.

	Full features			*t*-Test	Reduced features			*t*-Test
	Sens	Spec	Acc		Sens	Spec	Acc	
SVM	74.7	77.21	77.03	2.17	81.95	86.12	85.79	0.55
NBTree	64.9	98.11	95.87	7.89	71.45	96.51	94.82	8.09
SVM + NBTree (*Case-P*)	78.65	78.36	78.34	1.09	82.8	86.33	86.09	–
SVM + NBTree (*Case-SA*)	0.4	99.14	92.47	NA	0	100	93.25	NA
SVM + NBTree (*Case-SP*)	81.95	73.86	74.41	–	82.6	83.35	83.3	0.14

Sens, sensitivity; Spec, specificity; Acc, accuracy.

Table 6
Average results obtained using 200% over-sampled data.

	Full features			*t*-Test	Reduced features			*t*-Test
	Sens	Spec	Acc		Sens	Spec	Acc	
SVM	77.6	75.83	75.94	2.25	86.5	77.54	78.15	1.75
NBTree	66.1	97.28	95.18	10.49	72.9	96.06	94.5	16.57
SVM + NBTree (*Case-P*)	79.15	76.86	77.01	1.66	86.05	77.92	78.47	2.16
SVM + NBTree (*Case-SA*)	6.65	99.7	93.42	NA	0	100	93.25	NA
SVM + NBTree (*Case-SP*)	82.5	75.77	76.23	–	88.35	72.72	73.6	–

Sens, sensitivity; Spec, specificity; Acc, accuracy.

It is observed that over-sampling with reduced feature data improves the performance of the rules with respect to sensitivity and number of rules. But the observation using under-sampling data is of mixed nature i.e. 25% under-sampling data with reduced features yielded better results compared to full feature data, whereas 50% under-sampling data with full features yielded better results compared to reduced feature data. These observations led us to a new series of experiments where under-sampling and over-sampling is combined.

Tables 8 and 9 show the results obtained using the data set which is balanced using the combination of under-sampling and over-sampling. Table 8 presents the results obtained using the combination of 25% under-sampling and 100% over-sampling. It is observed that the hybrid SVM + NBTree using *Case-P* dataset yielded the best sensitivity of 71.07% with all the features. With reduced features, it is observed that the proposed hybrid SVM + NBTree using *Case-SP* dataset obtained an improved and best sensitivity of 86.95% and reduced rule set as well.

Results obtained using the combination of 50% under-sampling and 200% over-sampling are presented in Table 9. It is observed that the hybrid SVM + NBTree using *Case-SP* dataset yielded the best sensitivity of 78.25% using all the features. It is observed that the proposed hybrid SVM + NBTree using *Case-SP* dataset using reduced features yielded an improved sensitivity of 86%, which is almost equal to the sensitivity obtained using the hybrid SVM + NBTree using *Case-P* set i.e. 86.1%. As the number of rules extracted using *Case-SP* dataset are less than that of the rules extracted using *Case-P* dataset, it is advisable to use the hybrid SVM + NBTree using *Case-SP* dataset. Once again feature selection improves the sensitivity of the rules extracted.

As SMOTE (Chawla et al., 2002) is one of the most effective and simple approach proposed for balancing the data. We extend our research study and employed SMOTE as well to evaluate the proposed approach and to make a concrete conclusion. Table 10 presents the results yielded using SMOTE data. It is observed that using all features the hybrid SVM + NBTree using *Case-SP* dataset obtained the best sensitivity of 79.7%. The proposed hybrid SVM + NBTree using *Case-SP* dataset with reduced features yielded the best sensitivity of 91.85%. Similar to over-sampling data and combination of over-sampling and under-sampling data with SMOTE data the proposed approach SVM + NBTree with reduced feature data yielded better sensitivity compared to full feature data. It is also observed that the proposed approach yielded least number of rules when SMOTE data is used. The rule set extracted using SMOTE data is presented in Table 11.

A rule set is considered to display a high level of *fidelity* if it can *mimic* the behaviour of the machine learning technique from which it was extracted i.e. SVM in our study. Apart from accuracy, sensitivity and specificity, fidelity also is an important quantity to measure the quality of the rules. Fidelity yielded by various classifiers tested during this study are presented in Table 12. It is observed that, using SMOTE the hybrid SVM + NBTree using *Case-SP* dataset with all the features yielded that best fidelity of 93.46%, in other words, in this case the hybrid SVM + NBTree behaves 93.46% times same as the SVM. The proposed approach SVM + NBTree using *Case-SP* with reduced features using 50% under-sampling data behaves 97.63% exactly as SVM from which rules are extracted. It is observed that the rules extracted using reduced feature data behave much similar like SVM compared to the behaviour of the rules extracted using full feature data.

Table 7
Average results obtained using 300% over-sampled data.

	Full features			*t*-Test	Reduced features			*t*-Test
	Sens	Spec	Acc		Sens	Spec	Acc	
SVM	64.1	91.21	89.29	5.83	86.1	72.35	73.09	0.2
NBTree	63.58	97.29	94.98	11.35	73.4	95.41	93.92	14.5
SVM + NBTree (*Case-P*)	68.75	93.85	91.14	4.4	85.9	74.11	75.17	–
SVM + NBTree (*Case-SA*)	0	100	93.25	NA	0	100	93.25	NA
SVM + NBTree (*Case-SP*)	78.3	84.89	84.45	–	85.3	68.14	69.36	0.54

Sens, sensitivity; Spec, specificity; Acc, accuracy.

Table 8
Average results obtained using 25% under + 100% over sampled data.

	Full features			*t*-Test	Reduced features			*t*-Test
	Sens	Spec	Acc		Sens	Spec	Acc	
SVM	68.41	77.17	77.58	1.07	85.95	78.99	79.5	0.67
NBTree	68.6	97.5	95.55	1.31	71.65	96.17	94.52	11.81
SVM + NBTree (*Case-P*)	71.07	79.68	80.92	–	85.75	80.05	80.44	0.78
SVM + NBTree (*Case-SA*)	0	100	93.25	NA	0	100	93.25	NA
SVM + NBTree (*Case-SP*)	69.5	75.76	75.22	0.28	86.95	76.59	77.3	–

Sens, sensitivity; Spec, specificity; Acc, accuracy.

Table 9
Average results obtained using 50% under + 200% over sampled data.

	Full features			*t*-Test	Reduced features			*t*-Test
	Sens	Spec	Acc		Sens	Spec	Acc	
SVM	69	78.16	77.49	2.75	85.97	74.72	76.1	0.08
NBTree	72.2	94.17	92.62	2.26	76.5	93.53	92.38	6.68
SVM + NBTree (*Case-P*)	74.22	82.75	80.62	1.41	86.1	76.04	76.66	–
SVM + NBTree (*Case-SA*)	71.85	82.77	82.06	1.54	0	100	93.25	NA
SVM + NBTree (*Case-SP*)	78.25	71.53	72.04	–	86	72.61	73.43	0.05

Sens, sensitivity; Spec, specificity; Acc, accuracy.

Table 10
Average results obtained using SMOTE data.

	Full features			*t*-Test	Reduced features			*t*-Test
	Sens	Spec	Acc		Sens	Spec	Acc	
SVM	72.4	87.16	86.17	12.81	91.05	70.03	71.45	1.01
NBTree	63.95	96.55	94.35	24.07	75.35	93.92	92.67	21.56
SVM + NBTree (*Case-P*)	74.15	88.8	87.83	7.31	91.3	71.23	72.38	0.73
SVM + NBTree (*Case-SA*)	61.05	59.01	59.21	1.58	49.2	56.77	56.26	2.74
SVM + NBTree (*Case-SP*)	79.7	83.71	83.11	–	91.85	67.12	68.67	–

Sens, sensitivity; Spec, specificity; Acc, accuracy.

Table 11
Rule set extracted using SMOTE data with reduced features.

#	Antecedents	Consequent
1	If CRED_T-2 ≤ 98.379 and CRED_T-1 ≤ 99.626 and NCC_T ≤ 1.529 and CRED_T ≤ 607.095 and T_WEB_T ≤ 13.628	Churner
2	If CRED_T-2 ≤ 98.379 and CRED_T-1 ≤ 99.626 and NCC_T ≤ 1.529 and CRED_T > 607.095	Non-Churner
3	If CRED_T-2 ≤ 98.379 and CRED_T-1 ≤ 99.626 and NCC_T > 1.529	Churner
4	If CRED_T-2 ≤ 98.379 and CRED_T-1 > 95.849 and T_WEB_T ≤ 14.5	Churner
5	If CRED_T-2 ≤ 98.379 and CRED_T-1 > 95.849 and T_WEB_T > 14.5	Non-Churner
6	If CRED_T-2 ≤ 98.379 and CRED_T-1 > 99.626 and CRED_T-1 ≤ 104.8 and NCC_T-2 ≤ 1.071	Churner
7	If CRED_T-2 ≤ 98.379 and CRED_T-1 > 99.626 and CRED_T-1 ≤ 104.8 and NCC_T-2 > 1.071	Non-Churner
8	If CRED_T-2 ≤ 98.379 and CRED_T-1 > 104.8 and CRED_T-1 ≤ 161.026	Non-Churner
9	If CRED_T-2 ≤ 98.379 and CRED_T-1 > 104.8	Churner
10	If CRED_T-2 > 98.379	Non-Churner
11	If CRED_T-2 ≤ 98.379 and CRED_T-1 > 95.849 and T_WEB_T ≤ 14.5 and CRED_T ≤ 593.854	Churner
12	If CRED_T-2 ≤ 98.379 and CRED_T-1 > 95.849 and T_WEB_T ≤ 14.5 and CRED_T ≤ 593.854 and NCC_T ≤ 0.936	Churner
13	If CRED_T-2 ≤ 98.379 and CRED_T-1 > 95.849 and T_WEB_T ≤ 14.5 and CRED_T > 593.854 and NCC_T > 0.936	Churner
14	If CRED_T-2 ≤ 98.379 and CRED_T-1 > 95.849 and T_WEB_T > 14.5	Non-Churner

Table 12
Fidelity of the proposed SVM + NBTree using Case-SP.

	All features	Reduced features
Unbalanced	79.46	93.58
SMOTE	93.46	91.95
25% Under-sampling	85.04	89.82
50% Under-sampling	80.6	97.63
100% Over-sampling	83.22	94.88
200% Over-sampling	86.62	90.48
300% Over-sampling	87.66	86.91
25% Under-sampling + 100% over-sampling	71.54	92.36
50% Under-sampling + 200% over-sampling	78.34	90.52

It is observed that the hybrid SVM + NBTree using *Case-SA* yielded the worst sensitivity when compared to other classifiers. The possible reasons for such results are that the instances which stand-alone NBTree could not correctly classify are correctly classified by SVM, indeed those instances turned out to be the support vectors. Further, among these support vectors, many instances belong to churn customers. This fact is also the reason behind *Case-SP* yielding better sensitivity compared to *Case-SA* dataset.

As stated earlier the extracted rules can be used as an early warning system, it is observed from the extracted rules that the *credit value* of the customers and the number of *online transactions* a customer performs, are the main driving elements for determining a customer to be loyal or churner. Most of the rules say that the customers with less *value of the credit* in any of the month

may churn in future. Rules also indicate that the customers using *internet* (i.e. *online transactions*) less often may also churn in near future. Further, rules also imply that customers with high credit value and customers using online transactions are supposed to be loyal customers.

Overall empirical analysis induced that reduced feature data facilitates improved performance of the rules with respect to sensitivity. Where the comprehensibility of the extracted rules is concerned it is Case-SP data yielded less number of rules consistently. Based on the observation from empirical results, SMOTE data is advisable and recommended for the proposed approach SVM + NBTree using *Case-SP* data with reduced feature. It is to be noted that we employed SVM-RFE for feature selection and SVM for support vector extraction to evaluate the efficiency of SVM thoroughly and to reduce the data horizontally and vertically to simplify the rule extraction process. Other feature selection approaches also can be applied for obtaining the most important features of the data. Further, we analyzed only one dataset in this research study and results may vary with other datasets.

6. Conclusion

In this paper a hybrid rule extraction approach from SVM is presented to predict churn in bank credit card customers. Since the dataset at hand is a highly unbalanced dataset with 93.24% loyal and 6.76% churned customers; balancing techniques such as under-sampling, over-sampling, combinations of under-sampling and over-sampling and SMOTE are employed to balance the data. While solving the problems like churn prediction *sensitivity* is accorded high priority. Accordingly, by considering sensitivity alone, it is observed that the proposed hybrid SVM + NBTree using *Case-SP* with reduced features and balanced by SMOTE yielded the best sensitivity of 91.85%. The number of rules extracted using *Case-SP* data with reduced features are very less compared to the rules extracted using full feature data, resulting in improved comprehensibility of the system. Using *Case-P* and *Case-SP* data it is ensured that the extracted rules indeed represent the knowledge learnt by the SVM. The extensive study done in this paper can be considered as the study about the efficiency of SVM to deal with large scale unbalanced data with respect to rule extraction from SVM and this is an analytical CRM study applied to churn prediction problem. It is to be noted that this research study is pertaining to the dataset used in this study; results may vary with other datasets.

References

[1] D. Peppers, M. Rogers, A new marketing paradigm, Planning Review 23 (2) (1995) 14–18.
[2] A. Massey, M. Montoya-Weiss, K. Holcom, Re-engineering the customer relationship: leveraging knowledge assets at IBM, Decision Support Systems 32 (2001) 155–170.
[3] J. Anton, Customer Relationship Management: Making Hard Decisions with Soft Numbers, Prentice Hall, Englewood Cliffs, NJ, 1996.
[4] P. Kotler, Marketing Management The Millennium Edition, Prentice-Hall International, Englewood Cliffs, NJ, 2000.
[5] J. Edwards, Get It Together with Collaborative CRM, insideCRM, Tippit, 2007 http://www.insidecrm.com/features/collaborative-crm-112907/
[6] R.N. Bolton, A dynamic model of the duration of the customer's relationship with a continuous service provider: the role of satisfaction, Marketing Science 17 (1) (1998) 45–65.
[7] N.P. Mols, The behavioral consequences of PC banking, International Journal of Bank Marketing 16 (5) (1998) 195–201.
[8] R.N. Bolton, P.K. Kannan, M.D. Bramlett, Implications of loyalty program membership and service experiences for customer retention and value, Journal of the Academy of Marketing Science 28 (1) (2000) 95–108.
[9] B. Larivière, D. Van den Poel, Investigating the role of product features in preventing customer churn, by using survival analysis and choice modelling: the case of financial services, Expert Systems with Applications 27 (2) (2004) 277–285.
[10] M.C. Mozer, R. Wolniewicz, D.B. Grimes, E. Johnson, H. Kaushansky, Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry, IEEE Transactions on Neural Networks 11 (3) (2000) 690–696.
[11] K.A. Smith, J.N.D. Gupta, Neural networks in business: techniques and applications for the operations researcher, Computers and Operations Research 27 (11–12) (2000) 1023–1044.
[12] S. Garcia, A. Fernandez, F. Herrera, Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems, Applied Soft Computing 9 (4) (2009) 1304–1314.
[13] J. Bloemer, K.D. Ruyter, P. Peeters, Investigating drivers of bank loyalty: the complex relationship between image, service quality and satisfaction, International Journal of Bank Marketing 16 (7) (1998) 276–286.
[14] Y. Xie, X. Li, E.W.T. Ngai, W. Ying, Customer churn prediction using improved balanced random forests, Expert Systems with Applications 36 (2009) 5445–5449.
[15] A. Ultsch, Emergent self-organising feature maps used for prediction and prevention of churn in mobile phone markets, Journal of Targeting, Measurement and Analysis for Marketing 10 (4) (2002) 314–324.
[16] K.Y. Huang, An enhanced classification method comprising a genetic algorithm, rough set theory and a modified RBMF-index function, Applied Soft Computing 12 (1) (2012) 46–63.
[17] D.A. Kumar, V. Ravi, Predicting credit card customer churn in banks using data mining, International Journal for Data Analysis, Techniques and Strategies 1 (1) (2008) 4–28.
[18] C.-F. Tsai, Y.-H. Lu, Customer churn prediction by hybrid neural networks, Expert Systems with Applications 36 (2009) 12547–12553.
[19] N. Glady, B. Baesens, C. Croux, Interfaces with other disciplines modelling churn using customer lifetime value, European Journal of Operational Research 197 (2009) 402–411.
[20] X. Hu, A data mining approach for retailing bank customer attrition analysis, Applied Intelligence 22 (1) (2005) 47–60.
[21] K.N. Lemon, T.B. White, R. Winer, Dynamic customer relationship management: incorporating future considerations into the service retention decision, Journal of Marketing 66 (2002) 1–14.
[22] K. Huang-Su, K. Young-Gul, A CRM performance measurement framework: its development process and application, Industrial Marketing Management 38 (2009) 477–489.
[23] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York Inc., NY, USA, 1995.
[24] I. Guyon, J. Weston, S. Barnhill, V.N. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (1–3) (2002) 389–422.
[25] K.J. Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (1/2) (2003) 307–319.
[26] S. Ben-David, M. Lindenbaum, Learning distributions by their density levels: a paradigm for learning without a teacher, Journal of Computer and System Sciences 55 (1997) 171–182.
[27] C.-H. Wu, Y. Ken, T. Huang, Patent classification system using a new hybrid genetic algorithm support vector machine, Applied Soft Computing 10 (4) (2010) 1164–1177.
[28] S. Chowdhury, J.K. Sing, D.K. Basu, M. Nasipuri, Face recognition by generalized two-dimensional FLD method and multi-class support vector machines, Applied Soft Computing 11 (7) (2011) 4282–4292.
[29] H. Azamathulla, Md. Fu-Chun Wu, Support vector machine approach to for longitudinal dispersion coefficients in streams, Applied Soft Computing 11 (2) (2011) 2902–2905.
[30] V.N. Vapnik, Statistical Learning Theory, John Wiley & Sons, New York, USA, 1998.
[31] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods, Cambridge University Press, New York, NY, USA, 2000.
[32] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: D. Haussler (Ed.), Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, 1992, pp. 144–152.
[33] S. Gallant, Connectionist expert systems, Communications of the ACM 31 (2) (1988) 152–169.
[34] A.B. Tickle, R. Andrews, M. Golea, J. Diederich, The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural network, IEEE Transactions on Neural Networks 9 (6) (1998) 1057–1068.
[35] M.W. Craven, J.W. Shavlik, Using sampling and queries to extract rules from trained neural networks, in: Proceedings of the Eleventh International Conference on Machine Learning, San Francisco, CA USA, 1994.
[36] H. Núñez, C. Angulo, A. Catala, Rule-extraction from support vector machines, Proceedings of the European Symposium on Artificial Neural Networks 10 (2002) 7–112.
[37] G. Fung, S. Sandilya, R.R. Bharat, Rule extraction from linear support vector machines, in: Proceeding of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, ACM Press, New York, NY, USA, 2005, pp. 32–40.
[38] X. Fu, C.J. Ong, S. Keerthi, G.G. Hung, L. Goh, Extracting the knowledge embedded in support vector machines, in: International Joint Conference on Neural Networks (IJCNN'04), Budapest, Hungary, 2004.
[39] N.H. Barakat, J. Diederich, Learning-based rule-extraction from support vector machines, in: Proceedings of the 14th International Conference on Computer Theory and Applications ICCTA'2004, Alexandria, Egypt, 2004.

[40] N.H. Barakat, J. Diederich, Eclectic rule-extraction from support vector machines, International Journal of Computer Intelligence 2 (1) (2005) 59–62.
[41] N.H. Barakat, A.P. Bradley, Rule extraction from support vector machines: measuring the explanation capability using the area under the ROC curve, in: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, 2006.
[42] Y. Zhang, H. Su, T. Jia, J. Chu, Rule Extraction from Trained Support Vector Machines, vol. 3518, LNCS Springer, Berlin/Heidelberg, 2005, pp. 61–70.
[43] Ad.C.F. Chaves, M.M.B.R. Vellasco, R. Tanscheit, Fuzzy rule extraction from support vector machines, in: Fifth International Conference on Hybrid Intelligent Systems, Rio de Janeiro, Brazil, 6–9 November, 2005.
[44] Z. Chen, J. Li, L. Wei, A multiple kernel support vector machine scheme for feature selection and rule extraction from gene expression data of cancer tissue, Artificial Intelligence in Medicine 41 (2007) 161–175.
[45] N.H. Barakat, A.P. Bradley, Rule extraction from support vector machines: a sequential covering approach, IEEE Transactions on Knowledge and Data Engineering 19 (6) (2007) 729–741.
[46] M.A.H. Farquad, V. Ravi, R.S. Bapi, Support vector machine based hybrid classifiers and rule extraction thereof: application to bankruptcy prediction in banks, in: E. Soria, J.D. Martín, R. Magdalena, M. Martínez, A.J. Serrano (Eds.), Handbook of Research on Machine Learning Applications and Trends, Algorithms, Methods and Techniques, IGI Global, USA, 2008, pp. 404–426.
[47] M.A.H. Farquad, V. Ravi, R.S. Bapi, Rule Extraction using Support Vector Machine Based Hybrid Classifier, in: Presented in TENCON-2008, IEEE Region 10 Conference, Hyderabad, India, 19–21 November, 2008.
[48] D. Martens, B. Baesens, T.V. Gestel, Decompositional rule extraction from support vector machines by active learning, IEEE Transactions on Knowledge and Data Engineering 21 (2) (2009) 178–191.
[49] M.A.H. Farquad, V. Ravi, R.S. Bapi, Support vector regression based hybrid rule extraction methods for forecasting, Expert Systems with Applications 37 (8) (2010) 5577–5589.
[50] M.A.H. Farquad, V. Ravi, R.S. Bapi, Data mining using rules extracted from SVM: an application to churn prediction in bank credit cards, in: 12th International Conference on Rough Sets, Fuzzy Sets, Data Mining & Granular Computing (RSFDGrC'09), New Delhi, India, 16–18 December, 2009.
[51] R. Kohavi, Scaling up the accuracy of naïve-bayes classifiers: a decision-tree hybrid, in: Proceedings of Knowledge Discovery in Databases (KDD-96), Portland, USA, 1996.
[52] Business Intelligence Cup, Organized by the University of Chile, 2004, Available at: http://www.tis.cl/bicup_04/text-bicup/BICUP/202004/20public/20data.zip
[53] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intelligent Data Analysis 6 (5) (2002) 429–449.
[54] S. Visa, A. Ralescu, Issues in mining imbalanced data sets – a review paper, in: Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference (MAICS-05), 2005, pp. 67–73.
[55] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.