An Enhanced Binary Classifier Incorporating Weighted Scores

Deepali Virmani
Computer Science Engineering Department
Bhagwan Parshuram Institute of Technology
Delhi, India
deepalivirmani@gmail.com

Nikita Jain
Computer Science Engineering Department
Bhagwan Parshuram Institute of Technology
Delhi, India
nikitajain1210@gmail.com

Abhishek Srivastav
Computer Science Engineering Department
Bhagwan Parshuram Institute of Technology
Delhi, India
srisaiabhi2196@gmail.com

Mahima Mittal
Computer Science Engineering Department
Bhagwan Parshuram Institute of Technology
Delhi, India
mahimamittal01@gmail.com

Surbhi Mittal
Computer Science Engineering Department
Bhagwan Parshuram Institute of Technology
Delhi, India
surbhimittal19970@gmail.com

Abstract—**In this study, an approach is being proposed which will predict the output of an observation based on several parameters which employ the weighted score classification method. We will use the weighted scores concept for classification by representing data points on graph with respect to a threshold value found through the proposed algorithm. Secondly, cluster analysis method is employed to group the observational parameters to verify our approach. The algorithm is simple in terms of calculations required to arrive at a conclusion and provides greater accuracy for large datasets. The use of the weighted score method along with the curve fitting and cluster analysis will improve its performance. The algorithm is made in such a way that the intermediate values can be processed for clustering at the same time. The proposed algorithm excels due to its simplistic approach and provides an accuracy of 97.72%.**

Keywords-weighted score; classification; clustering; deviation; threshold; SVM; decision tree

I. INTRODUCTION

Classification is a data mining technique in which a collection of data is categorized into classes in which the training dataset may or may not have class labels. A dataset may have two or more class labels. In this work we are focusing on binary classification using clustering technique based on curve analysis and weighted score method followed by verification. To illustrate with an example, let us suppose that we have a dataset containing data about spam from a repository. We want to identify the data points above and below the threshold level which are classified as spam and not spam respectively. The threshold level can be obtained through sorting and processing of the dataset. The proposed algorithm preprocesses the dataset to find the weighted score as well as to predict the threshold value, which is then represented graphically. After this step, verification is done using cluster analysis. Observations on various datasets were found to be accurate to a high degree. Deviations of various data points from the threshold values were obtained and various inferences were found. We have also calculated the individual effect of an attribute with respect to its effect on classification. Data provided by datasets contain hidden information that might not be known to the user. An effort has been made to develop a new algorithm to facilitate mining techniques. The simplistic approach of the algorithm is easy to understand and implement.

The proposed algorithm is based on clustering which acts as a stable preprocessing method for binary classification. Weighted score method assigns different importance degrees to the instances of a dataset. The proposed classifier calculates the mean of each sample which is multiplied with each attribute's value summed up to assign a weight to that sample. A threshold value is taken and plotted data points fall below or above it. The minimum and maximum values among the weighted sample sums are subtracted from the threshold value which is halved to obtain the centers of two clusters. Clustering is performed using these centers and by taking maximum distance into consideration which will be the same with the distance between a center and the threshold value. The clusters obtained correspond to the binary class labels which classify the dataset. Observations are cross verified using the clustering method. Weighted score is a simple technique and also incorporates the individual contribution of an attribute consisting of its weighted score in its contribution to the deviation of the data point from the threshold.

II. LITERATURE REVIEW

Various studies have been proposed for classifying datasets into two categories. Previous researchers utilized different classification approaches. SVM (support vector machines), is basically used for classification and regression analysis and employs supervised learning techniques. In SVM algorithm, new examples are assigned or classified into categories and therefore it is regarded as a non-probabilistic classifier. SVM can be thought of as a clustering algorithm in space in which points belonging to a cluster are distant from points of other clusters. In that space a hyper-plane divides the points in groups. A particular hyper-plane with the characterization of minimizing the total distance of the data points on either of its sides is selected. This is also called a linear classifier. There are various variations to the basic approach of the SVM namely linear kernel SVM, polynomial kernel SVM and radial kernel SVM. The most efficient method for fitting SVM is the sequential minimal optimization (SMO) method. It breaks the problem down into sub-problems that can be solved analytically rather than numerically. There are various SVM applications like the recognition of standing people in a picture. Authors in [1] used SVM along with k nearest neighbor (KNN) for visual category recognition. Authors in [2] used variations of SVM to predict the future popularity of social media messages. The disadvantages of SVM are that the theory only covers the determination of the parameters for a given value of the regularization and kernel parameters and is dependent on the kernel choice. As a result, SVM comes up with the problem of overfitting from optimizing the parameters to model selection. Kernel models can be quite sensitive to overfitting the model selection criterion [3]. In [4], local space time features were used for recognizing complex motion patterns using SVM.

A decision tree is a predictive model to go from observations and related choices about an item to possible outcomes about the item's target value. It has various applications in statistics, data mining and machine learning. In this structure, each node denotes a test on an attribute, leaves represent class labels and branches represent conjunctions of features that denote the test outcome. Besides being simple to interpret and understand, decision trees are able to handle both categorical and numerical data [5]. To solve the problem of fragmentation and replication, a notion of decision graphs has been introduced which allows disjunctions or joins. There are assumptions taken into consideration regarding decision tree algorithm. At the beginning, the whole training set is considered as the root. Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model. Records are distributed recursively on the basis of attribute values. The decision tree algorithm is sensitive to root selection. If the dataset consists of n attributes then the decision of which attribute to place at the root or at different tree levels as internal nodes is a complicated step. Any random node cannot be placed at the root. If the random approach is followed, it may give bad results with low accuracy. Placing attributes is done by statistical approach. A variation of this weighted class based decision tree [6] has been proposed in which weights are easy assigned according to the importance of class labels which are further classified using a decision tree. The dataset is split in this approach which might potentially introduce bias where small changes in the dataset can introduce big impact. Decision-tree can lead to over-complex trees that do not generalize well the training data.

Studies have shown that classification issues are often more precise when using a combination of classifiers which outperform a highly specific classifier [7]. Using a combination of classifiers noisy data can be handled in a better way with augmented accuracy and speed even though complexity issues may emerge [8]. Weighted score method assigns different importance degrees to instances of a dataset and is often used as a pre-processing method. Automated weighted sum (AWSum) uses a weighted sum approach where feature values are assigned weights that are summed and compared to a threshold in order to classify an example. It provides insight into the data [9]. Authors in [10] dealt with the weighted score fusion method which involves the classification of a fruit based on the diverse and complementary features that can be used to describe it. The algorithm has various steps which involve preprocessing, multiple feature selection, optimal feature selection and SVM. However, the approach requires improvements in the real world environment. A quadratic classifier is used in statistical classification to separate measurements of two or more classes of objects or events using a quadric surface. It is a more general version of the linear classifier. Statistical classification considers a set of vectors of observations x of an object or event, each of which has a known type y referred to as the training set. The problem is then to determine the class of a new observation vector. The correct solution is quadratic in nature. In the special case where each observation consists of two measurements, this means that the surfaces separating the classes will be conic sections, thus the quadratic model is the generalization of the linear approach developed to incorporate the conic separating surfaces for classification. Quadratic discriminant analysis (QDA) is closely related to linear discriminant analysis (LDA), where it is assumed that the measurements from each class are normally distributed. Unlike LDA however, in QDA there is no assumption that the covariance of each of the classes is identical. Classification error rate ranges around 20%-30%.

An artificial neural network consists of units (neurons), arranged in layers, which convert an input vector into some output. Each unit takes an input, applies a (most probably a nonlinear) function to it and then passes the output on to the next layer. The networks are defined to be feed-forward, which means that a unit feeds its output to all the units on the next layer, but there is no feedback to the previous layer. Weightings are applied to the signals passing from one unit to another. These weightings are tuned in the training phase (learning phase) to adopt the neural network to the particular problem. The network processes records one at a time, and learns by comparing their classification with the known actual classification. The errors from the initial classification are fed back into the network and used to modify the network's algorithm for further iterations. Neurons are organized into layers: input, hidden and output. The input layer is composed not of full neurons but rather consists simply of the record's values that are inputted to the next layer of neurons. Next there are one or more hidden layers. The final layer is the output

layer, where there is one node for each class. A single sweep forward through the network results in the assignment of a value to each output node, and the record is assigned to the class node with the highest value. A key feature of neural networks is the iterative learning process in which records (rows) are presented to the network one at a time, and the weights associated with the input values are adjusted each time. Authors in [12] used the neural network classifier for diagnosis and classification of medical problems. Authors in [13] modified the output of the neural network classifier in the form of Bayes classifier values.

The search for a confirmed improvement of classification techniques has been a continuing topic in data mining field. The proposed algorithm is advantageous in the terms of required calculations to arrive to a conclusion and accuracy on large datasets. Secondly, the use of weighted scores for data pre-processing improves the clustering results [11]. The algorithm is made in such a way that the intermediate values can be processed for clustering at the same time. The pre-processing of the data and its representation allowed the clustering and cluster production at the same time.

TABLE I. COMPARISON OF BINARY CLASSIFICATION TECHNIQUES

Name	Complexity
SVM-KNN	Highest
Weighted Classification / Decision tree	Medium
Binary Classification / Quantum Adiabatic Algorithm	Medium
AWSum	Lowest
Quadratic Classifier	Low
Neural Network Classifier	Moderate

III. METHODOLOGY

1. WEIGHTED SCORE(x,n) //function to find row average of dataset, x is the column values, n is the number of columns
   1.1. LOOP1: for i in 1 to n
      1.1.1. $wi = \sum_{i=1}^{n} x_i / n$
   1.2. LOOP2: for i in 1 to n
      1.2.1. $r_i = \sum_{i=1}^{n} w_i * x_i$ // r_i is the weighted score.
   1.3. return r_i.
2. FIND THRESHOLD(r_i,n,k) //function that finds the threshold value by clustering (k means)
   2.1. LOOP3: for i in 1 to k //where k is the number of clusters
      2.1.1. LOOP4: for j in 1 to n // n is the number of data points
   2.2. $min \sum_{i=0}^{k} \sum_{x \varepsilon ci}^{n} \left| x - \frac{1}{|c_i|} \sum_{rj \varepsilon ci} rj \right|$ //assign each $c_1 \cup c_2 \cup c_3 \ldots \cup c_n = \rho$ datapoint to nearest center
   2.3. END LOOP3,LOOP4
3. THRESHOLD_VERIFY(r_i,n,K,K_i) //verifying the threshold value

//T(−X)=max(t such that #{s∈T | s≥t}=X) then
//T(X)={t∈T | t≥T(−X)} where X=n(no of dataset entries).

   3.1. LOOP5: for i in 1 to n
      3.1.1. if $T_{MAX} < r_i$
      3.1.2. set $T_{MAX} = r_i$
   3.2. LOOP6:for i in 1 to n
      3.2.1. if $T_{MIN} > r_i$
      3.2.2. set $T_{MIN} = r_i$
   3.3. threshold _value $T_H = (T_{MAX} - T_{MIN})/2$.

//this T_H coincides with k means center provided appropriate scaling is there.

   3.4. $O_1 = (T_{MAX} - T_H)/2$ //initial centroids
   3.5. $O_2 = (T_H - T_{MIN})/2$
   3.6. LOOP7: for i in 1 to K //where k is the number of clusters
      3.6.1. LOOP8: for j in 1 to Ki // K_i the number of objects of the cluster i
         3.6.1.1. $F\{C_1, C_2, \ldots C_k\} = i=1K\ j=1K i T_{ii\text{-}Oi}$

//Tij is the j -th object of the i -th cluster and Oi is the centroid of the i-th cluster which is defined.

   3.7. The dataset is divided into two parts one part above the threshold other below it.
4. MEASURE DEVIATION(x_i, T_H,r_i,n) //measures deviation of weighted score from threshold value
   4.1. dev=$T_H - r_i$
   4.2. LOOP9: for i in 1 to n
      4.2.1. dom factor[i]=x_i/r_i //influence of each attribute value on weighted score
   4.3. End LOOP9

A flow chart of the algorithm is shown in Figure 1.

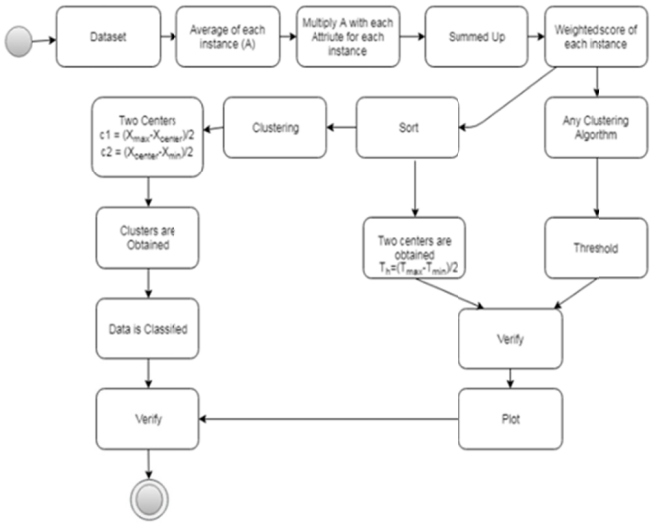

Fig. 1. Flow chart of the algorithm used

IV. RESULTS

In this paper we focused on the modification of weighted score and processing of weighted scores has been done to perform clustering. The proposed algorithm was applied to various datasets and the observations were recorded. The data sets were of different types and varied in complexity. The execution time of the algorithm on the datasets is provided in the results (Table II). There were 7 datasets selected and each row entry or data point of the dataset was classified to one of

the two levels either below or above the threshold level. The algorithm can classify the data points of the dataset in two parts as soon as it is provided with an input. Figures 2-8 show the algorithm application on the dataset (each upper image) and the application of the clustering algorithm on weighted scores (each lower image). Results of both algorithms are verified and it was found that the threshold value or the center coincided in both cases provided appropriate scaling is present. The threshold value is represented in the Figures with an orange dot which divides the dataset into two parts: the data points are depicted in red and black dots. The points above the threshold are black and the points below are red. Chosen datasets show a wide range of variations. Every dataset is unique in itself because of its nature and entries. Some of the datasets, like [12] and [17] contain a smaller number of data points and the algorithm was highly accurate in that case. In the case of a slightly larger number of data points the accuracy was a little less. When the data points were scattered along the graph, the accuracy was slightly less due to variations in the dataset and deviation from an ideal behavior.

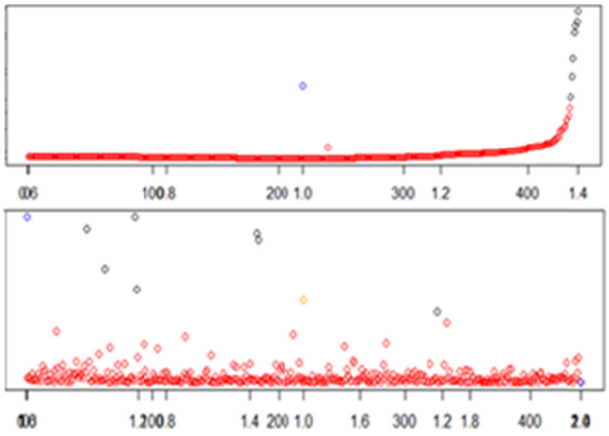

Fig. 2. Up: Algorithm application on sorted dataset_1 [14]. Down: Clustering on unsorted dataset_1 [14]

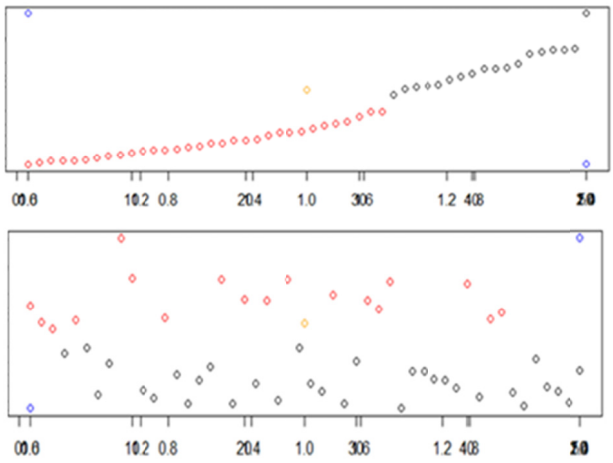

Fig. 3. Up: Algorithm application on sorted dataset_2 [15] Down: Clustering on unsorted dataset_2 [15].

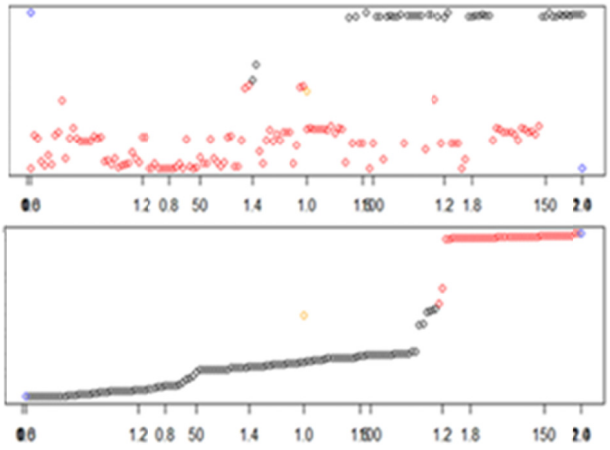

Fig. 4. Up: Algorithm application on sorted dataset_3 [16] Down: Clustering on unsorted dataset_3 [16]

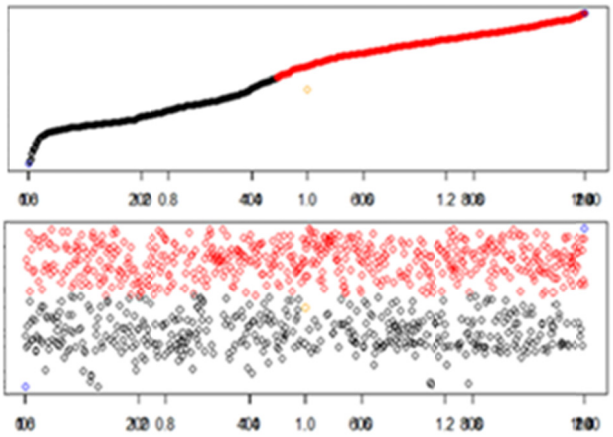

Fig. 5. Up: Algorithm application on sorted dataset_4 [17] Down: Clustering on unsorted dataset_4 [17]

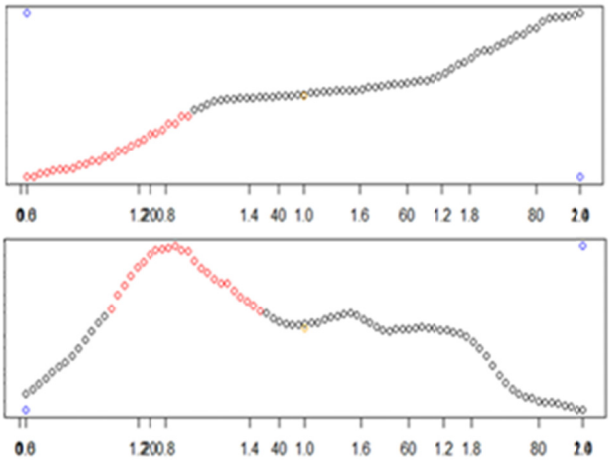

Fig. 6. Up: Algorithm application on sorted dataset_5 [18] Down: Clustering on unsorted dataset_5 [18]

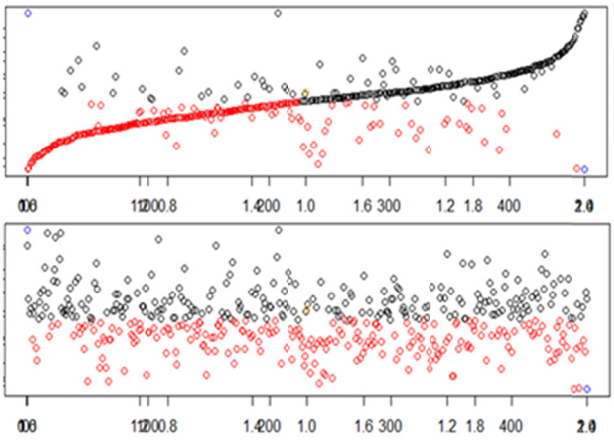

Fig. 7. Up: Algorithm application on sorted dataset_6 [19] Down: Clustering on unsorted dataset_6 [19]

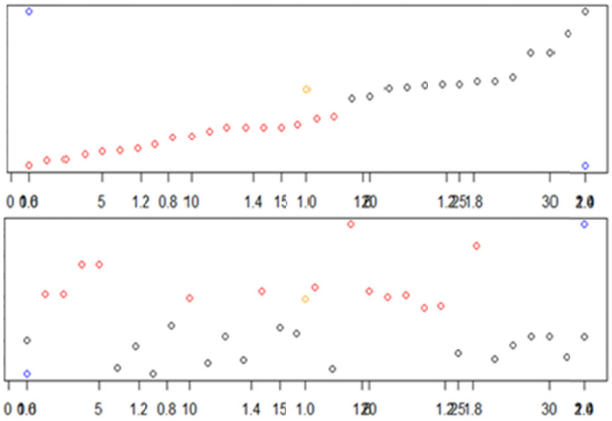

Fig. 8. Up: Algorithm application on sorted dataset_7 [20] Down: Clustering on unsorted dataset_7 [20]

TABLE II. RESULTS

Dataset	Error (%)	Accuracy	Execution Time
Dataset_1 [14]	0.227%	99.72%	2.143s
Dataset_2 [15]	2%	98%	1.022s
Dataset_3 [16]	2.197%	97.802%	1.561s
Dataset_4 [17]	2.3%	97.7%	4.820s
Dataset_5 [18]	5.2%	94.8%	2.69s
Dataset_6 [19]	4.329%	95.67%	3.01s
Dataset_7 [20]	3.125%	96.875%	0.992s

V. Error Analysis and Dominating Factor

The error percentage was calculated on the basis of the data points that were incorrectly clustered as a deviation from the expected behavior. The number of data points that were incorrectly clustered was divided with the total number of data points to obtain the error percentage. The execution time gives the processing of the dataset by the algorithm. It was observed that the datasets in which there were there smaller distances between individual data points the error percentage was relatively larger than the datasets in which data points had larger distances between them. No data point was found to be lying on the threshold value and therefore each data point belongs to either below or above the threshold level.

VI. Conclusion

On the results obtained by the implementation of the algorithm, it was found that the algorithm is having 97.72% average accuracy. The algorithm allowed pre-processing of the data and its weighted representation allowed the simultaneous clustering and cluster production. The precision rates for a dataset can be simply calculated using the number of points that are above or below the threshold value and belong to the other cluster divided by the total number of the clusters. It was found that fixing of initial centroids and maximum values can lead to effective clustering. Apart from using k means clustering we can also involve other clustering algorithms for verification to achieve higher degree of precision such as fuzzy k means. It is important that the plotting of the data points and cluster centers should be done on the same graph with proper scale to obtain useful results. The threshold point is of great significance, it gives a center point for classifying the data points of the dataset. There were two data point representations: one was with respect to the proposed algorithm and the other pertained to the clustering algorithm employed the accuracy of the classification depending on the selection of the threshold value. It was observed that the center coincided in both cases therefore the obtained threshold value was accurate. The algorithm is different from traditional weighted score as it utilizes the weighted score of data point for the clustering and classification at the same time.

VII. Future Work

The proposed classifier can be enhanced in various ways. Some other efficient clustering algorithm can be applied to the weighted score such as fuzzy k means, probabilistic k means to get a higher level of accuracy in clustering as they are more efficient in handling initial centroids. The algorithm can also be used to function as a neural classifier to automatically identify the threshold level and perform the classification.

Acknowledgment

The authors would like to thank Dr. Payal Pahwa, principal of the Bhagwan Parshuram Institute of Technology for her support and encouragement during the research and providing necessary facilities at institute.

References

[1] H. Zhang, A. C. Berg, M. Maire, J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition", 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, USA, pp. 2126-2136, June 17-22, 2006

[2] B. Yu, C. Miao, L. Kwok, "Toward predicting popularity of social marketing messages", in: Social Computing, Behavioral-Cultural Modeling and Prediction. SBP 2011. Lecture Notes in Computer Science, Vol. 6589, pp. 317-324, Springer, Berlin, Heidelberg, 2011

[3] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Min. Knowl. Disc., Vol. 2, No. 2, pp. 121–167, 1998

[4] C. Schuldt, I. Laptev, B. Caputo, "Recognizing human actions: a local SVM approach", 17th International Conference on Pattern Recognition, (ICPR) 2004, Cambridge, UK, Vol. 3, pp. 32-36, 2004

[5] P. J. Tan, D. L. Dowe, "MML inference of decision graphs with multi-way joins", in: Advances in Artificial Intelligence. AI 2003. Lecture Notes in Computer Science, Vol. 2903, pp. 269-281, Springer, Berlin, Heidelberg, 2003

[6] J. L. Polo, F. Berzal, J. C. Cubero, "Weighted Classification Using Decision Trees for Binary Classification Problems", II Congreso Español de Informática, pp. 333-341, Zaragoza, Spain, September 11-14, 2007

[7] C. Chiu, Y. Ku, T. Lie, Y. Chen, "Internet auction fraud detection using social network analysis and classification tree approaches", International Journal of Electronic Commerce, Vol. 15, No. 3, pp. 123-147, 2011

[8] H. Neven, V. S. Denchev, G. Rose, W. G. Macready, "Training a binary classifier with the quantum adiabatic algorithm", arXiv preprint arXiv:0811.0416, 2008

[9] A. Quinn, A. Stranieri, J. Yearwood, "Classification for accuracy and insight: A weighted sum approach", Proceedings of the sixth Australasian conference on Data mining and analytics, Vol. 70, pp. 203-208, Australian Computer Society Inc., 2007

[10] L. Kuncheva, J. Bezdek, R. Duin. "Decision templates for multiple classifier fusion: an experimental comparison", Pattern Recognition, Vol. 24, No. 2, pp. 299–314, 2001

[11] D. Virmani, S. Taneja, G. Malhotra, "Normalization based K means Clustering Algorithm", arXiv preprint arXiv:1503.00900, 2015

[12] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, G. D. Tourassi, "Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance", Neural Networks, Vol. 21, No. 2-3, pp. 427-436, 2008

[13] J. S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition", In: Neurocomputing. NATO ASI Series (Series F: Computer and Systems Sciences), Vol. 68, pp. 227-236, Springer, Berlin, Heidelberg, 1990

[14] Machine Learning Depository, Wholesale customers Data Set, https://archive.ics.uci.edu/ml/datasets/wholesale+customers

[15] USArrests, http://stat.ethz.ch/R-manual/R-devel/library/datasets/html/USArrests.html

[16] "Datasets distributed with R Git Source Tree", https://forge.scilab.org/index.php/p/rdataset/source/tree/master/csv/datasets/attenu.csv

[17] https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/datasets/quakes.csv

[18] https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/datasets/volcano.csv

[19] https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/boot/channing.csv

[20] https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/boot/neuro.csv