VOL. 11, NO. 9, MAY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ENSEMBLE FUZZY SUPPORT VECTOR MACHINE CLASSIFIER BASED ON MAXIMUM SPANNING TREE FOR BIG DATA ANALYTICS

B. Rajendran[1] and Saravanan Venkataraman[2]
[1]Research and Development Centre, Bharathiar University, India
[2]Department of Computer Science, College of Computer and Information Sciences, Majmaah University, Kingdom of Saudi Arabia
E-Mail: rajendran.bhojan@gmail.com

ABSTRACT

Today the buzz word in information technology is big data. Classification is one among the thrust research problem in such big data and its corresponding application scenarios. This research work makes use of an ensemble fuzzy support vector machine in order to perform the classification task. Maximum spanning tree is used for feature selection among the big data. KDD Cup 99 is multivariate dataset which consists of 40, 00,020 instances with 42 attributes chosen for evaluating the performance of the proposed work. Simulation results show the proposed ensemble classifier. This paper is organized as follows. Section 1 discusses on the introduction to big data along with the scope of research. Section 2 briefs the related works. Section 3 presents the proposed research work. Section 4 portrays experimental results. Section 5 concludes the thesis with future scope of research work.

Keywords: Big data, classifier, fuzzy inference system, support vector machine, KDD cup dataset.

1. INTRODUCTION

Big Data analytics engage processing assorted data from a variety of data sources producing complementary datasets [1]. For this reason, the data sets are not only pigeonholed by their enormously large volumes but also by their heterogeneity and the distributed attainment of data. Several data mining techniques have been proposed in the literature to process such data sets [1]. Laterally, from traditional centralized data mining systems where a single learner has full access to the global dataset [2], data mining systems typically use ensemble learning techniques consisting of a hierarchy of multiple local learners operating on subsets of the global dataset [3].

In [16] the authors mentioned a terminology called HACE theorem stating Big Data starts with large-volume, heterogeneous, autonomous sources with distributed and decentralized control, and seeks to explore complex and evolving relationships among data. The HACE theorem portrays that the above said characteristics make an extreme challenge for discovering useful knowledge from the Big Data.
The following are certain research challenges in mining big data.

- **Limited data access:** In distributed data mining, each local learner has only limited access to the entire dataset [3]. There are two types of data partition [4]. In the instance based mining, each local learner accesses a subset of instances (with all features) of the entire dataset; while in the feature-based mining, each local learner accesses a subset of the feature space of all instances. In this work, we focus on the scenario with feature data.
- **Limited communication capability:** Due to the large data volume and the limited communication capabilities of individual learners, it is costly to centralize unprocessed data within the system, which makes the centralized mining expensive if not infeasible [3].
- **Coping with large rates of non stationary data:** The data accessible to local learners also grows fast and the statistical properties of the data may change dynamically over time [4].

The primary research problem lies on the current definition of Big Data. In this research work the network traffic data gratifies the characteristics of big data classification which is the primary task for addressing big data analytics to be more cost effective. In recent days, plenty of applications suffer from the big data problem that includes network traffic risk investigation, geospatial classification and big business forecasting. Intrusion detection and prediction are considered to be time receptive applications and also it needs highly efficient big data techniques to embark upon the problem on the go.
Some of the recently emerging technologies also aid to perform big data analytics on several applications such as Hadoop Distributed File Systems (HDFS) and Hive database [5] are implemented to resolve research problems like big data classification. On the other hand the applications also in need of continuous expansion in big data domain probably suffer from the big data problems.

2. RELATED WORKS

Machine learning algorithms are proposed for the classification task of network intrusion traffic [6-10]. Support Vector Machine (SVM) is one among the machine learning algorithms that obtained greater attention by researchers. On the other hand the computational cost of the SVM is generally more than many other classification algorithms. To address this research issue more SVM mechanisms are also developed in the machine learning research [11], [12]. Also representation-learning algorithms [13] were proposed in machine learning research. In particular cross-domain representation-learning (CDRL) technique proposed by

Tuand Sun [14] is considered suitable for our chosen big data classification problem. CDRL has encountered certain challenges, including the difficulty in selecting relevant features, constructing geometric representation, fetching appropriate features and segmenting various types of data. Then, the notion of unit-circle algorithm (UCA) [15] also proposed which represents the intrusion traffic data by unit circles and allocates many related records to fewer unit-circles. This feature of UCA obtained big data classification to work effectively than that of CDRL.

3. PROPOSED WORK

MST is a weight edge graph with their weighted sum which must not be larger than any other MST weight. There are three algorithms for finding the Maximum spanning tree. One of them was developed in 1926 by a Czech scientist and two more which are mostly used is Prim's and KrusKal's Algorithm. These three algorithms are Greedy algorithms and run in polynomial time complexity. Hence in this work, KrusKal's algorithm is used to create Maximum spanning tree. Running time complexity of KrusKal's algorithm is O (E log V).

Before understanding the concept of Maximum spanning tree implemented in this paper, it is necessary to explain the idea behind a complete graph. We will create a complete graph in which every node is connected to its remaining nodes; by considering each feature representing a node. Vertices of a complete graph have the values of Fisher Score of each selected feature with respect to class. Edges between two features are assigned a weight that is Fisher Score of two different features. Vi is the set of vertices { (FS(f1,C), FS(f2,C), FS(f3,C) ....FS(fm,C) } and Eij is the set of edges {Q (Fi, Fj : C)}.

Kruskal Algorithm is used in proposed algorithm for MST construction that follows greedy approach, and produces a MST of a weighted connected graph. Hence this results in a Maximum spanning tree. Now each and every pair of nodes (Fi, Fj) are traversed until a pair of vertices is found, whose weight smaller than both the vertices of the edges. Then, this weighted edge is split into two trees, thus selecting the highest values in the feature.

Consider a dataset D with f features F={ F1, F2, F3,….Ff} with Class C and calculate Fisher Score of each feature and select a set of feature F′= {F1, F2, F3,….Fm} with their fisher score is greater than a particular threshold β. Vi= {FS(f1,C), FS(f2,C), FS (f3, C)….. FS (fm, C)} and Eij = {Q (Fi, Fj: C)} for each i = 0 to m and j = 0 to m. After the completion of Maximum spanning tree those edges removed which weight smaller then both Fisher values of features are removed. For instance, take FS (Fi, C) > Q (Fi, Fj: C) <FS (Fj, C). Then Q (Fi, Fj: C) edge shall have to be removed. After removing all insignificant edges we get a set of trees. From these the most relevant feature from each tree is selected on the basis of their Fisher score.

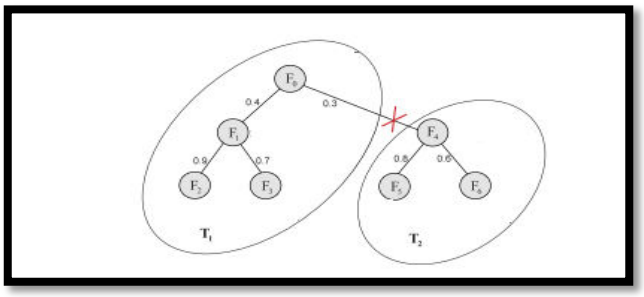

Figure-1. Representation of MST.

3.1. Maximum Spanning Tree (MST) based feature selection

In this proposed approach, we combine fisher score with MST to obtain the required result. The Fisher Score[6] method calculates a score for each attribute based on certain criteria and selects those attributes whose score is higher than the preset threshold value and the algorithm produces significantly better results.

The proposed algorithm is divided into four steps:
Input: D (F0, F1……Fn, C);
Output: F (F0, F1…..Ff);

Step-1: Irrelevant Feature Selection Using Fisher Score.
A: Score = FS (Fi, C)
B: if (Score>Threshold)
Selected feature)

Step-2: Maximum Spanning Tree Construction
A: Complete Graph (G) = Null;
B: Co-Relation = Q (Fi, Fj: C);
C: add Fi, Fj to Complete Graph (G) With Fisher Score as the weight of the corresponding edge;
D: Construct Maximum Spanning Tree = KrusKal's (G);
Step-3: Tree Partition and Cluster Formation
For each node(Ei) {
Select the node with max(FS(Fi, C))
Remove Edge (Q (Fi, Fj: C)) and associated node;
F'=Fi;
repeat step 3;
Step-4: Select Feature from Each Cluster and get a set of feature: F′ = {(F0, F1…..Ff)};

Flow diagram (Figure-4) shows the operation of the proposed algorithm. It is the last three steps which are our main focus area, as they implement redundant feature removal. Process starts with the selection of features from the total set of features using fisher score and then construction of the complete graph which results in a Maximum Spanning Tree with the help of KrusKal's algorithm. Finally, a cluster is made to get the set of selected features.

3.2. Ensemble fuzzy support vector machine classifier

Fuzzy support vector machine reduces the training time and improves the efficiency. The final step of pre-processing is scaling the training data, *i.e.* normalizing

all features so that they have zero mean and a standard deviation of 1. This avoids numerical instabilities during the SVM calculation. We then used the same scaling of the training data on the test set.After the important features are extracted in terms of the values of the parameters θj , the parsimonious fuzzy rules are applied based on the support vectors $S\{x_s^l\}$,which lies as l=1 and N_s discovered by the SVM.

The training process is performed as follows:

1) Each support vector corresponds to a fuzzy rule. The number of fuzzy rules equals to the number of support vectors;

2) Given the ith support vector x_s^i ; i=1,…, L

a) The premise part of the ith fuzzy rule is evaluated as follows: the MF of fuzzy set for the jth input variable in the ith rule is

$$A_i^j(x_j) = a^j(x_j - m_i^j) \quad (1)$$

Where m_i^j is the jth element of the ith support vector x_s^i.

b) The consequent part of the ith fuzzy rule is induced from α_0 and class labels, i.e., the consequent value of the ith rule is

$$b_i = \alpha_0^{(i)} y_s^{(i)} \quad (2)$$

where$\alpha_0^{(i)}$represents non-zero $\alpha_0^{(i)}$ and $y_s^{(i)}$ is the class label corresponding to the ith support vector x_s^i .Th class I membership function of x is defined using the minimum operator for

$$m_i(x) = \min_{j=1.n.} m_{ij}(x) \quad (3)$$

If x is satisfied

$$nD_k(x)\begin{cases} > 0 \; for k = i \\ \leq 0 \; for k \neq i, k = 1,..n \end{cases} \quad (4)$$

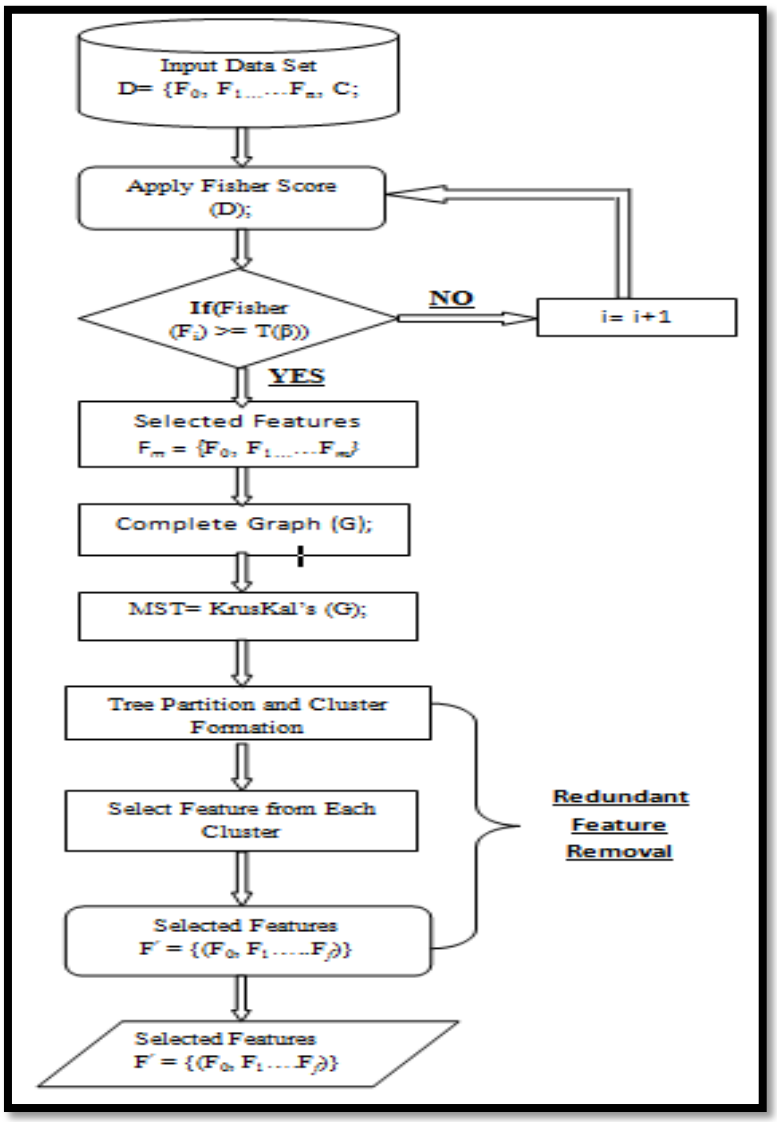

Figure-2. MST-FSVM algorithm's process flow diagram.

The fuzzy rule selection procedure is described by the following steps.

Step-1: Evaluate the misclassification rates (MRs) of the rules on the validation dataset and the test dataset separately.

Step-2: Set s=1 and assign a small value to threshold $h_s(h_s > 0)$

Step-3: Select the most influential fuzzy rules by $\{Rule_i | \alpha_0^{(i)} orw_i > h_s\}(5)$

Step-4: Construct a fuzzy classifier (FC) by using the influential fuzzy rules selected in Step-3.

Step-5: Apply FC to the validation dataset v and the test dataset t to obtain new MRs: Ev(s).

Step-6: If Ev(s) =Ev(0), stop the selection and use FC(s-1) as the final compact classifier and Et(s-1)as the measure of generalization performance for FC(s-1); Otherwise, increase s by 1, assign a higher value to threshold h_s , and go to Step-3.

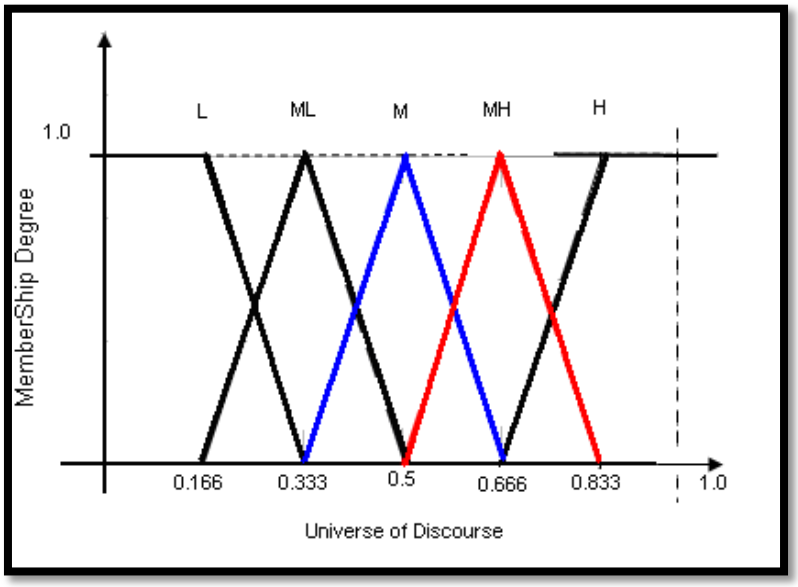

Figure-3. Membership degree.

4. EXPERIMENTAL RESULTS

This section presents experimental results to portray the classification efficiency on the KDD Cup dataset from UCI KDD archive. KDD Cup dataset has four gigabytes of compressed binary TCP dump data from seven weeks of network traffic, which was processed into about five million connection records, among which we randomly select 50000 records as the training dataset. Each connection record is labelled either as a "normal" connection or as an "attack". The performance of the algorithms such as SVM, FSVM and proposed ensemble MST-FSVM algorithms are evaluated using the metrics such as detection rate, false alarm rate and time taken for classification. The computer with 2.4 GHz processor, 2 GB RAM with L2 cache is used. MATLAB tool is used to write the source code for the above mentioned algorithms. The simulation results are presented in Table-1. From the results it is observed that the proposed ensemble MST-FSVM classifier better detection rate (Figure-4), lesser false alarm rate (Figure-5) with comparably reduced timed taken for classification (Figure-6).

Table-1.

Detection rate (%)			**False alarm rate** (%)			**Time taken for classification** (Seconds)		
SVM	FSVM	MST-FSVM	SVM	FSVM	MST-FSVM	SVM	FSVM	MST-FSVM
58.71	61.85	81.93	41.29	38.15	18.07	6392	5821	4093

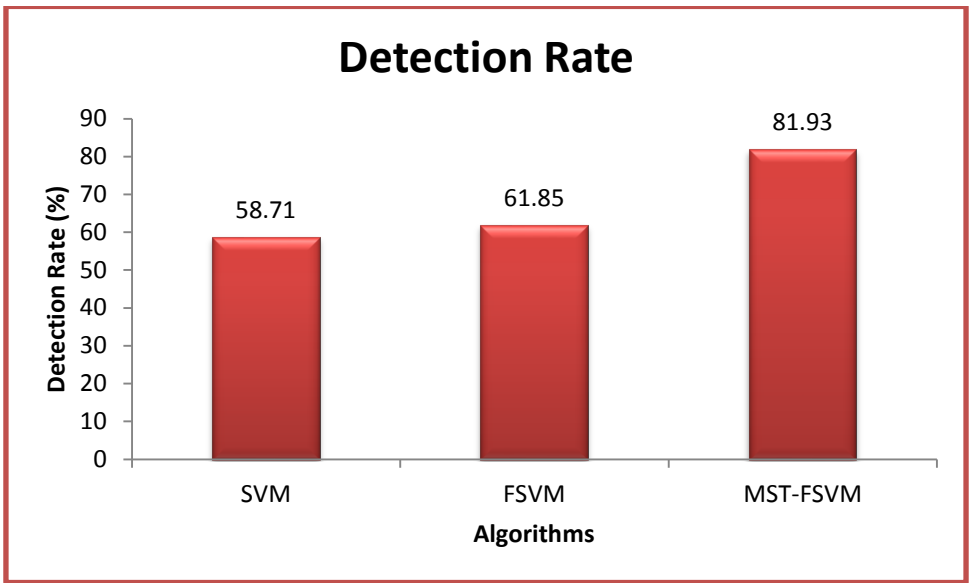

Figure-4. Detection rate.

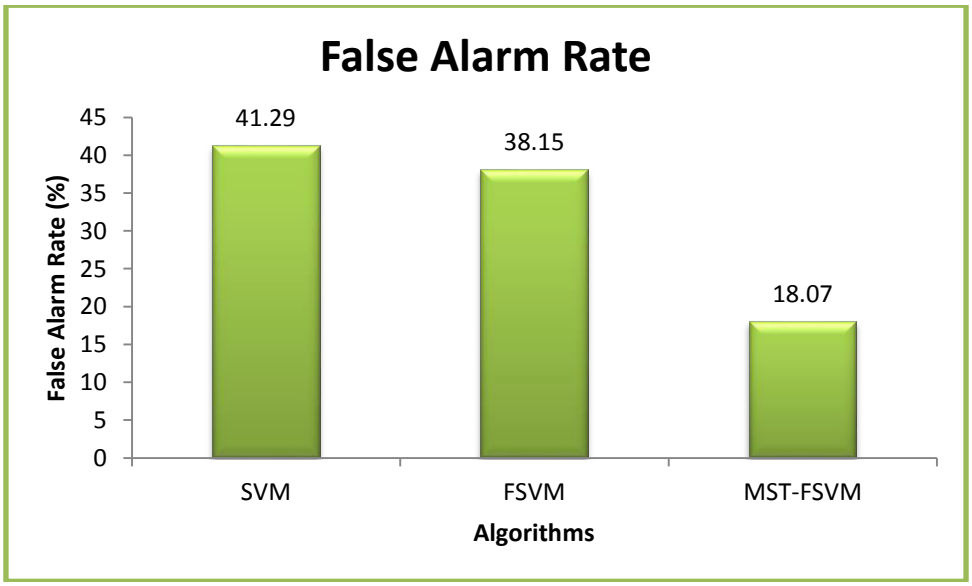

Figure-5. False alarm rate.

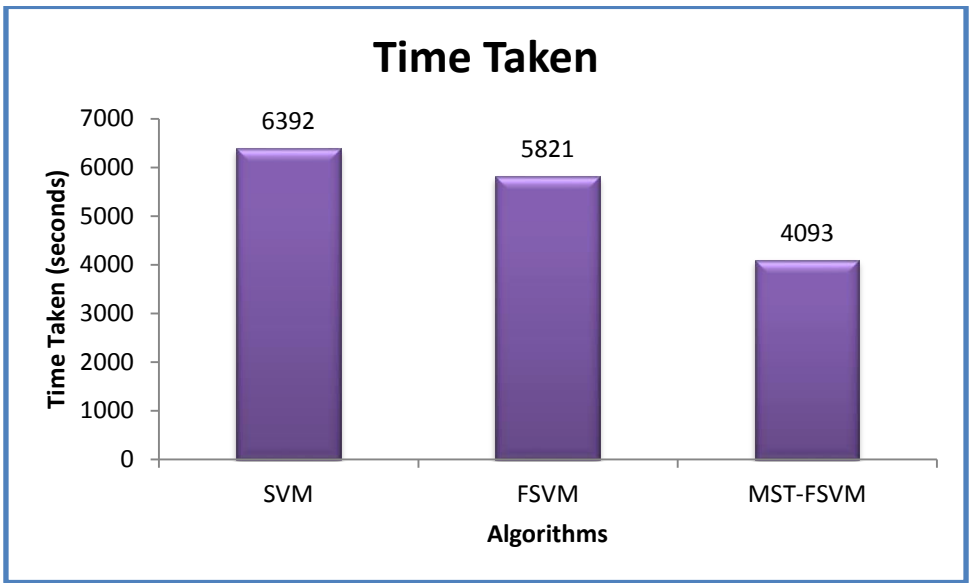

Figure-6. Time taken for classification.

5. CONCLUSIONS AND FUTURE SCOPE OF RESEARCH

The proposed work is named as ensemble fuzzy support vector machine classifier with maximum spanning tree. The proposed work makes use of the maximum spanning tree to perform feature selection task. Fuzzy support vector machine classifier is then trained using 50000 samples. The performance metrics such as detection rate, false alarm rate and time taken for classification are chosen and the results show that the proposed classifier obtains better results. This part of doctoral research contributed a maximum spanning tree based ensemble fuzzy support vector machine classifier. This research work maps one of the big data analytics problems with the network intrusion detection. The future work is planned to design a deep learning neural network classifier.

REFERENCES

[1] B. Park and H. Kargupta. 2002. Distributed data mining: Algorithms, systems, and applications. Distributed data mining handbook. pp. 341-358.

[2] M. Kantardzic. 2011. Data mining: Concepts, models, methods, and algorithms. Wiley-IEEE Press.

[3] P. Chan and S. Stolfo. 1993. Experiments on multistrategy learning by meta-learning.CIKM ’93.

[4] S. Agrawal, V. Narasayya and B. Yang. 2004. Integrating vertical and horizontal partitioning into automated physical database design. SIGMOD ’04.

[5] T. White. 2012. Hadoop: The definitive guide. O'Reilly Media.

[6] S. Carlin, K. Curran. 2012. Cloud computing technologies. International Journal of Cloud Computing and Services Science (IJ-CLOSER). 1(2): 59-65.

[7] S. B. Kotsiantis. 2007. Supervised machine learning: A review of classification techniques. Informatica. 31: 249-268.

[8] P. Laskov, C. Schafer and I. Kotenko. 2004. Intrusion detection in unlabeled data with quarter-sphere support vector machines. In: Proc. of the DIMVA Conference. pp. 71-82.

[9] G. Huang, H. Chen, Z. Zhou, F. Yin and K. Guo. 2011. Two-class support vector data description. Pattern Recognition. 44: 320-329.

[10] I. Corona, G. Giacinto and F. Roli. 2008. Intrusion detection in computer systems using multiple classifier systems. Studies in Computational Intelligence (SCI). 126: 91-113.

[11] G. Giacinto, R. Perdisci and F. Roli. 2005. Network intrusion detection by combining one-class classifier. In: F. Roli and S. Vitulano (Eds.) ICIAP 2005, LNCS 3617. pp. 58-65.

[12] O. L. Mangasarian and D. R. Musicant. 2000. Lagrangian support vector machine classification.TR 00-06, Data Mining Institute, Department of Computer Science, University of Wisconsin, USA - ftp://ftp.cs.wisc.edu/pub/dmi/techreports/00-06.pdf.

[13] V. Jeyakumar, G. Li and S. Suthaharan. 2012. Support vector machine classifiers with uncertain knowledge sets via robust convex optimization. Optimization - The Journal of Mathematical Programming and Operations Research. Taylor and Francis, DOI:10.1080/02331934.2012.703667. pp. 1-18.

[14] Y. Bengio, A. Courville and P. Vincentar. 2012. Representation Learning: A Review and New Perspectives. ArXiv: 1206.5538v2 [cs.LG].

[15] W. Tu and S. Sun. 2012. Cross-domain representation-learning framework with combination of class-separate and domainmerge objectives. In: Proc. of the CDKD'12 Conference. pp. 18-25.

[16] S. Suthaharan. 2012. A unit-circle classification algorithm to characterize back attack and normal traffic for intrusion detection. In: Proc. of the IEEE International Conference on Intelligence and Security Informatics.pp. 150-152.

[17] X.Wu, X. Zhu, G.Q. Wu, W. Ding. 2014. Data Mining with Big Data. IEEE Transactions on Knowledge and Data Engineering. 26(1).