ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 3, pp: (386-393), Month: July - September 2014, Available at: www.researchpublish.com

IMPLEMENTING A NEW SEMI-SUPERVISED APPROACH FOR INTERNET TRAFFIC CLASSIFICATION USING NSL-KDD DATASET

[1]Vajihe Abdi, [2]Marzieh AhmadZadeh

[1,2]Shiraz University of Technology, Shiraz, Iran

Abstract: **Network traffic classification is a process of finding type of end user applications toward network planning and bandwidth management, diagnostic monitoring, traffic analysis, prediction and engineering, anomalous traffic detection and QoS provisioning. Today with the improvement in field of information security, traditional network traffic classification such as payload based and port based classification are useless. Supervised, unsupervised and semi-supervised are three machine learning algorithms suggested to tackle traditional techniques. In this paper a semi-supervised approach including clustering (EM clustering, DBSCAN and k-Means), mapping and J48 classification is proposed assuming random 20, 50 and 80 percent of NSL-KDD dataset as unlabeled class attributes. Weka 3.7.11 is used for this implementation and overall precision, recall and F-Measure are the metric of performance evaluation comparing the results with 100 percent labeled training dataset. The results showed that the overall recall, precision and F-Measure of 20 and 80 percent of unlabeled dataset are more than 95.6% and for 50 percent unknown traffic flows is 90.1%. This measurement for full labeled training dataset is 98.8%.**

Keywords: **Traffic classification, Machine Learning, EM clustering, DBSCAN, K-Means, J48 classification, NSL-KDD dataset.**

I. INTRODUCTION

In recent years with the improvement in the field of programming and networks, networks are become widespread. Due to this occurrence, many data are produced hourly and this huge amount of data needs to be processed in order to extract information. This information is used in some arena, for example network management, misuse and anomaly detection. Data mining is a way to acquire this information. According to [1] data mining has many definitions and one of them is the non-intuitive extraction of useful information and meaningful patterns from enormous amount of data by automatic or semi-automatic means. Classification, clustering, association rule discovery, sequential pattern discovery, regression and deviation detection are data mining tasks. Classification [2] is a supervised data mining technique that finds a model for a training set. Indeed this model is based on a function of independent variables for class attribute (dependent variable). A testing dataset is used as verification and validation. Classification has many applications for example direct marketing, fraud detection, customer attrition/churn and sky survey cataloging. Clustering [3] is an unsupervised data mining technique that divides input dataset, containing data points, into groups. Similarity between intra data points (separate clusters) must be maximized; the same as dissimilarity for inter clusters. The similarity is measured based on Euclidean distance, Minkowski distance, Mahalanobis distance, Cosine similarity and so on. Clustering applications has a wide range from market segmentation to document clustering.

Network traffic classification is a process of finding type of end user applications toward network planning and bandwidth management, diagnostic monitoring, traffic analysis (shaping/policing), prediction and engineering, anomalous traffic detection and QoS provisioning. Payload based analysis and port based classification [4] are two common traditional network traffic classification techniques. In port based IP traffic classification, each TCP/UDP packet would be inspected for its port number. But there are some limitations including unregistered port number with Internet Assigned Numbers Authority[1], using other ports to avoid control restriction of operating system access and utilization of dynamic port number [4]. In order to solve the port based classification, payload based IP traffic classification was introduced. In this kind of classification contents of packets would be inspected to investigate applications signatures. A combination of port and payload based classification is introduced by Moore and Papagiannaki [5] that it first acts on port numbers and if there is no popular port number, payload inspection will be done. Entire flow payload inspection is next step to overcome the remaining unclassified flows. But payload based analysis cannot work well against packet encryption techniques, concurrent analysis for huge amount of flows and also in situations with privacy protection laws.

Thereafter Machine Learning[2] approaches were proposed. ML uses statistical features of flows and in this case there is no need to determine port and payload of packets. Three major groups of ML are supervised, unsupervised and semi-supervised approaches [4]. Supervised approach or classification, as mentioned before, builds a model based on attributes and labeled class. Decision tree, rule based classifier, nearest neighbor classifier and naïve Bayes are some supervised algorithms. Although supervised algorithm can works well but having a class attribute (labeled flows) are not always possible and also this approach suffers from detecting new classes [4]. Unsupervised approach or clustering, as mentioned before, is the process of grouping similar dataset into same clusters. K-Means, DBSCAN, hierarchical clustering and EM clustering are some instances of unsupervised algorithms. Unsupervised approach doesn’t need class attribute and can determine new classes but its constraint is related to its mapping between class labels and clusters [4]. Semi-supervised approach [6] is the combination of supervised and unsupervised approach. It means that semi-supervised uses first clustering and then classification in order to solve Internet traffic classification.

In this paper a semi-supervised approach, using 20, 50 an 80 percent of NSL-KDD dataset as unlabeled, is proposed. In the proposed model the first step is clustering using EM clustering, DBSCAN and k-Means, the one with higher correctly clustered instances will be chosen. Second step is mapping between clusters and class values with the help of clustering algorithm and maximum labeled flows exist in a cluster. Third step is classification using J48.

The rest of this paper is organized as follows. Section 2 is about some related works related to Internet traffic classification. The proposed semi-Supervised approach for Internet traffic classification is proposed in Section 3. Section 4 is about the implementation of proposed method and discussion results and finally in section 5 a conclusion is made.

II. RELATED WORKS

ML approaches differs from one another, so different clusters and classification measurements have various results. In this Section some approaches using real data or KDD dataset are explained. In [7] two models including misuse detection and anomaly detection using NSL-KDD are proposed. Misuse detection model used Principal Component Analysis[3] as a dimensionality reduction algorithm and four classification algorithms: Naïve Bayes, Decision Tree, Rule Induction and Nearest Neighbor. Anomaly detection model utilized from five clustering algorithms inclusive k-Means, Improved k-Means, k-Medoids, and EM clustering and Distance-based outlier detection. In order to evaluate each model Accuracy, false positive and execution time for clustering were measured. According to I. Syarif et al.’s results, decision tree classification had higher accuracy (=99.56%) and lowest false positive (=0.40%) with 10 fold validation. Also distance-based outlier detection and EM clustering had respectively 80.15% and 78.06% accuracy and 21.14% and 20.47% false positive. But the execution time of distance-based outlier was relatively higher than EM clustering.

In [8] an unsupervised approach for Internet traffic classification comparing k-Means, DBSCAN and AutoClass algorithm for its clustering part is proposed. In order to analyze algorithms two empirical packet traces were used as dataset. Their result showed that AutoClass obtains more accuracy than k-Means and DBSCAN. It is mentioned that although DBSCAN can distinguish noise and as a result the accuracy of this clustering algorithm is reduced, it made few number of clusters that includes more connections. The precision of k-Means with K=190, DBSCAN with Eps=0.02 and MinPts=3 and AutoClass are concluded as their results.

[1] IANA
[2] ML
[3] PCA

In [9] J. Erman et al. proposed a semi supervised traffic classification technique including clustering and classification. Dataset (campus, residential and wireless LAN) used to train and test gathered from the Internet link with 29 applications. At first a clustering was done in 64000 unlabeled flows and then labeled fix numbers of random flows in each cluster. Erman et al.'s proposal model achieved 94% accuracy using two labeled flows in each cluster with K=400. Also 80,800 and 8000 labeled flows were mixed with random number of unlabeled flows for the input dataset. They showed that with five or more labeled flows exists in each cluster, accuracy increased.

In [10], V. Kumar et al. used k-Means clustering with Euclidean distance to overcome the analysis of NSL-KDD intrusion detection dataset. Four categories of attacks: DOS[4], Probe, R2L[5] and U2R[6] with different size are besides Normal data with the highest number of instances in training dataset. Kumar et al. showed that their approach could deal with new type of attacks.

In [11] which uses real data as a training dataset and NSL-KDD as a testing dataset, S. Shaikh et al. proposed a semi-supervised method using DBSCAN clustering with Euclidean distance function. After clustering execution each cluster mapped to a label with maximum priority existed in that region and then classification was performed. Accuracy, precision, recall and F-Measure are used to evaluate the performance of presented architecture. Classifier overall accuracy achieved by this approach at 18 number of cluster was 97.76%. S. Shaikh et al. also indicated that classifier accuracy depends on MinPts and Eps of DBSCAN and also was related to the number of clusters.

In [12] proposed by J. Zhang et al. at first presented a new method to cope with small supervised training set and unknown applications and then a theoretical analysis proved performance of this method. Two real-world network dataset, named *isp* and *wide* were considered as its dataset. The most important components of System model consists of flow label propagation, clustering, cluster-application mapping, Nearest Neighbor classifier and Bag of Flow construction. In flow label propagation algorithm, flows that were not labeled would be labeled as a pre-labeled flow, sharing same 3-tuple (destination port number, destination IP address and transport protocol) in order to extend the number of pre-labeled flows and simultaneously a k-Means algorithm was executed on combination of labeled an un-labeled flows. After those, Zhang et al. used a cluster-application mapping function between the results of label propagation and k-Means clustering and finally nearest neighbor classification was performed on training data. In testing stage Bag of Flows[7] construction put flows sharing same 3-tuple in one Bag and eventually classification was done for Bag of Flows. This approach was compared with C4.5, KNN, Naïve Bayes, Bayesian Network and Erman's semi-supervised method from two perspective; accuracy and F-Measure. The accuracy and F-Measure of Zhang et al.'s methods on both datasets were more than five other approaches. It was also showed that proposed method had robust ability and good unknown detection performance on false detection.

III. PROPOSED MODEL

Semi-supervised approach as it mentioned before consists of two major steps, clustering and classification. Fig. 1 shows the components of proposed semi-supervised approach. Preprocessing is the first step in this approach, including normalization and PCA for dimensionality reduction. PCA provides a linear map for N dimensional feature space [13]. After clusters are generated, data points assign to a cluster have more similarities between their attributes ignoring class attribute (labeled or unlabeled flows). If data points assign to a cluster have also same labels, correctly clustered instances (True Positive) will be increased. In this paper the main factor to choose a clustering algorithm is the number of correctly clustering instances. According to this consideration a semi-supervised approach is proposed including, clustering, mapping and classification.

[4] Denial Of Service attack
[5] Remote to Local attack
[6] User to Root attack
[7] BoFs

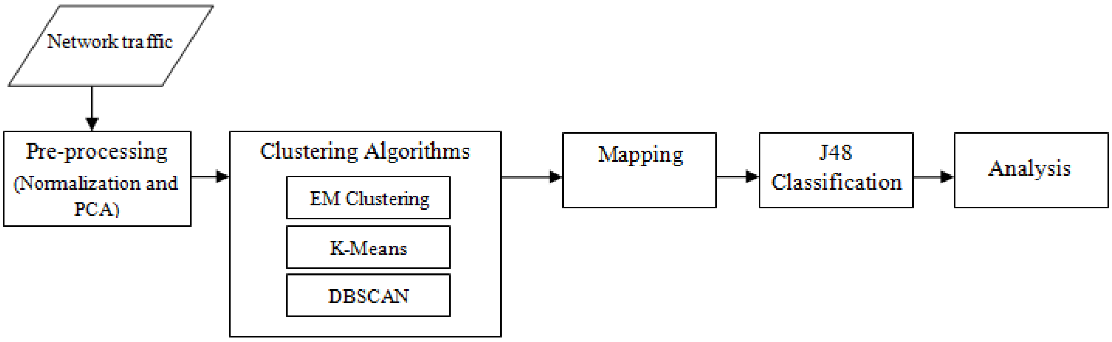

Fig. 1: A new semi-supervised approach for Internet traffic classification

A. Clustering

Three basic clustering algorithms including EM clustering, k-Means and DBSCAN are performed on preprocessed flows and the result of an algorithm with the higher number of correctly clustered instances will be chosen. Hierarchical clustering also has good results but because of huge amount of time and space it requires, it is not possible to run it on each system. Here three basic clusters are described.

K-Means

K-Means is one kind of partition-based clustering approaches that a centroid (center point) is considered for each cluster. Number of clusters (K points) must be selected before beginning of clustering and this is one limitation of k-Means for users without dataset domain knowledge. It then assigns data points to their closest centroid. After this assignment, from the data points assign to one cluster, a point that represents all data points (typically mean) is considered as centroid. As long as centroid doesn't change, closeness calculation and afterwards steps will be continued. Determining initial centroids are random; therefore clusters generated by k-Means differ from one another. Also k-Means has problems with different size, densities, non-globular shapes and outliers so one solution is usually to increase K. Besides these limitations, k-Means is simple and if initial centroids are chosen correctly it has good results [1].

DBSCAN

DBSCAN is a density-based clustering algorithm. There are some definitions used in this algorithm.

- Density is the number of data points assign in pre-defined radius (Eps).
- MinPts is minimum number of points that must be assigned in Eps.
- Core points are points that have more than MinPts in their Eps.
- Border points are in the neighborhood of a core point and they do not have MinPts in their Eps.
- Noise points are neither core points nor border points.

In DBSCAN algorithm at first, all cores, border and noise points are determined and labeled. Noise points will be eliminated and all core points that are within Eps of each other will be connected by an edge, each of them make a cluster. Each border point assigns to its associated core point cluster. DBSCAN is resistant to noise and different shape and size of data points but it is not against high dimensionality and various densities. Also there is no need to define number of clusters by user but determining Eps and MinPts are other challenges of this clustering algorithm [1].

EM Clustering

Expectation Maximization[8], explained by A. Dempster et al. [14], [15], is one type of k-Means that estimates density of data points generated from K normal distributions. EM clustering has two steps, Expectation (E-step) and Maximization (M-step). In E-step, this clustering calculates the instance cluster probabilities, which is known as expectation of the likelihood, and then re-labels this expectation of the likelihood to each instance. In M-step, parameters used to calculate E-step will be re-estimated by mean and variance, and the results of M-step are used for E-step. This iteration continues until convergence of results happens. EM clustering can find the number of clusters by cross validation or determining this number by user. Against this advantage, it cannot identify individual applications of interest. In [15] EM clustering algorithm is described with more details.

[8] EM

B. Mapping

The main function in mapping component is to map a class label to a cluster. Mapping is done between clusters and class values with the help of clustering algorithm and maximum labeled flows exist in a cluster in order to label unlabeled flows. Chosen clustering algorithm that has the most number of correctly clustered instances determined a label for some of clusters. Other clusters map to maximum number of class labels exist in a cluster.

C. J48 Classification

J48 classification applies fast statistical detection. In this paper it is used for classification step. J48, a Java implementation of C4.5, uses depth first construction and information gain. This algorithm selects an independent attribute with highest information gain to create decision tree until all records belongs to a class. In [1] details of J48 are presented.

IV. EXPERIMENTAL AND CONCLUSION RESULTS

A. NSL-KDD Dataset

According to [16], this benchmark dataset presented for network-based IDSs, is suggested to tackle the four problems of KDD'99 dataset: Absence of redundant records and no biased classifier in training set, existence a test set with no duplicate records, accurate evaluation of different learning techniques and reasonable number of records in train and test set resulting consistent and comparable results. 20% subset of the NSL-KDD, with 42 attributes and 25192 instances, is used in this paper for training dataset. 20, 50 and 80 percent of this dataset in 3 steps are considered as unlabeled flows. This assumption differs in different situations and this is because of encryption and other problems exist in determining labeled flows. Fig. 2 shows Number of instances in KDD training dataset.

Four types of attacks are including DOS[9], Probe, R2L[10] and U2R[11]. DOS attack makes memory and computing resources too busy or declines legitimate users to access machine. Probing attack meshes security controls by gathering network information. R2L attack obtains local access to a machine which is not permitted to by exploiting some vulnerability. U2R attack obtains root access to a system by exploiting some vulnerability.

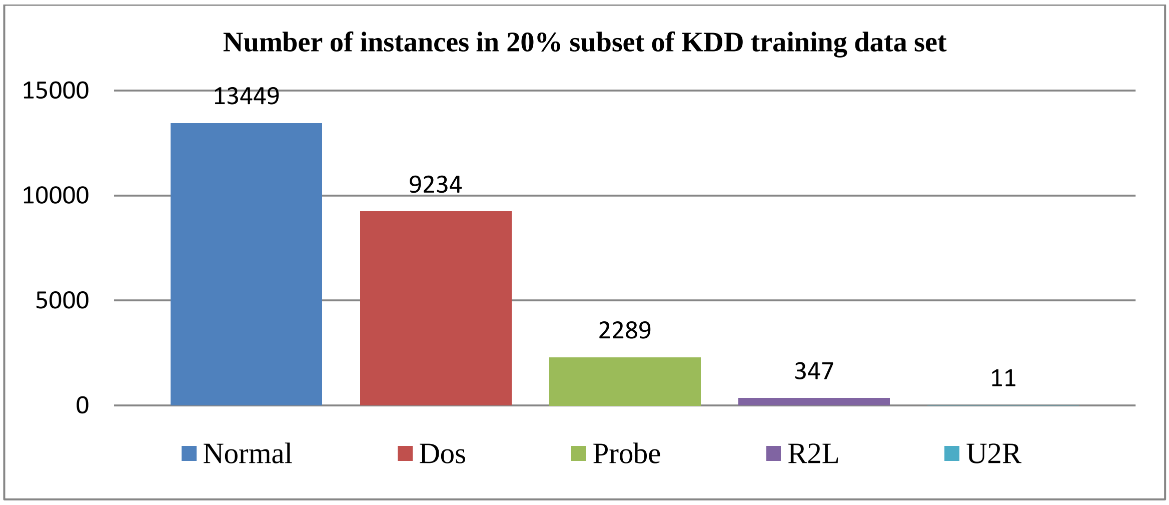

Fig. 2: Number of instances in dataset

B. Setup and Results

Weka[17] version 3.7.11 is used for experiment on a single machine (core-i5 processor, 4.00 Gigabyte RAM and extended 2 Gigabyte heap size for Weka). Weka, a Java based program, is a collection of open source of data mining and machine learning algorithms including preprocessing, attribute selection, classification, clustering, visualization and so

[9] Denial Of Service attack
[10] Remote to Local attack
[11] User to Root attack

on. 23 feature vectors are filtered using PCA, from 41 attributes exists in KDD. In [10] with NSL-KDD dataset, 4 clusters are used in k-Means but here because of 5 class labels exist in KDD, number of pre-determined clusters in k-Means is considered as 5. Determining the precise values of Eps and MinPts is difficult in DBSCAN, however with the help of experience, most likely range for values of Eps are considered 0.8, 0.9, 1 and 2, and for values of MinPts are 4, 6, 8, 10 and 12. Table I shows the results of this experience. Although the results of incorrectly clustered instances do not differ a lot with these ranges of Eps and MinPts, it is considerable that as Eps values decreases, number of noise points in same value of MinPts increases. Also as the MinPts increases in a particular value of Eps, the number of clusters and incorrectly clustered instances decrease but number of noise points increases. So here we choose Eps=0.9 and MinPts=12 for DBSCAN.

TABLE I: Determining Eps and MinPts in DBSCAN

		Number of noise points	Number of clusters	Incorrectly clustered instances in		
				Random 20% unlabeled dataset (%)	Random 50% unlabeled dataset (%)	Random 80% unlabeled dataset (%)
Eps=0.8	MinPts=4	143	141	63.52	62.23	65.96
	MinPts=6	186	133	63.35	62.06	65.79
	MinPts=8	236	126	63.16	61.86	65.60
	MinPts=10	312	117	62.75	61.56	65.29
	MinPts=12	353	113	62.70	61.41	65.13
Eps=0.9	MinPts=4	130	137	63.56	62.28	66.01
	MinPts=6	159	131	63.44	62.16	65.89
	MinPts=8	210	124	63.24	61.96	65.69
	MinPts=10	289	115	62.93	61.65	65.37
	MinPts=12	**332**	**111**	**62.77**	**61.48**	**65.21**
Eps=1	MinPts=4	96	131	63.69	61.98	65.46
	MinPts=6	119	126	63.60	61.89	65.37
	MinPts=8	145	123	63.49	61.87	65.26
	MinPts=10	229	114	63.16	61.45	64.93
	MinPts=12	267	111	63.01	61.30	64.78
Eps=2	MinPts=4	-	1	57.20	23.16	9.38

According to Table II, number of clusters in k-Means, EM clustering and DBSCAN are respectively 5, 8 and 111. Because of close results achieved by running k-Means and EM clustering 10 reputations of both, k-Means and EM clustering, is considered. Ranges of 95% confidence interval in random 20, 50 and 80 percent of dataset as unlabeled considered for comparison of k-Means and EM clustering are respectively (-7.19,1.85), (-0.20,1.164) and (-11.98,-2.05). In random 20 and 50 percent of dataset as unlabeled confidence interval includes zero so there is no difference in the results of EM clustering and k-Means, hence EM clustering is chosen. However in 80 percent unlabeled dataset k-means has less incorrectly clustered instances. Also as number of unlabeled flows increased, the results of three algorithms seem to be converged.

TABLE II: Incorrectly clustered instances for random 20, 50 and 80% unlabeled dataset

	Number of clusters	Incorrectly clustered instances in		
		Random 20% of dataset as unlabeled (%)	Random 50% of dataset as unlabeled (%)	Random 80% of dataset as unlabelled (%)
EM clustering	8	**46.37±2.46**	**48.75±3.85**	56.45±4.25
K-Means	5	43.70±3.79	42.57±3.78	**48.51±3.78**
DBSCAN	111	62.77	61.48	65.21

To indicate the performance of implemented semi-supervised approach, following metrics are applied.
In a class, True Positive[12] is number of correctly classified objects; False Positive[13] is number of falsely indicated as a class objects and False Negative[14] is number of class objects labeled as other classes. Recall, precision and F-Measure are respectively shown in Eq. (1), (2) and (3). Recall determines number of misclassified objects in a class. Precision determines number of correct classified objects and F-Measure is a balance between precision and recall.

$$Recall = \frac{TP}{TP+FP} \qquad (1)$$

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

$$F\text{-}Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (3)$$

Table III shows overall recall, precision and F-Measure achieved in different percent of unlabeled flows. As it is showed in Table III, overall precision, recall and F-Measure are more than 90% in all different percentage of unlabeled flows and this number in full labeled class attribute is 98.8%. These measurement were on evaluation matrix computed by 10 cross validation.

TABLE III: Overall recall, precision and F-Measure

	Overall TP rate	Overall FP rate	Overall Recall	Overall Precision	Overall F-Measure
0% unlabelled dataset	**98.8**	**1.0**	**98.8**	**98.8**	**98.8**
20% unlabelled dataset	95.8	2.6	95.8	95.6	95.6
50% unlabelled dataset	90.2	8.1	90.2	90.1	90.1
80% unlabelled dataset	95.8	2.6	95.8	96.0	95.8

Fig. 3 shows overall F-Measure achieved by different percentage of unknown flows in different classes. According to Fig. 2 and 3, more amounts of labeled classes exist in dataset; more F-Measure is achievable in different unlabeled flows. So Normal and DOS class in 20, 50 and 80 percent unknown flows have more than 90% F-Measure and this number for Probe is 80%, this is because of more amount of Normal, DOS and Probe labels exist in dataset. Because of small amount of U2R labels, k-Means can't determine this label in 80 percent unknown flows hence number of R2L labels are exceeded in this clustering.

V. CONCLUSION

DBSCAN, k-Means and EM clustering are three major clustering algorithms used recently. Each of these clustering algorithms generates different results and is suitable in different situations. In this paper a semi-supervised approach for Internet traffic classification including normalization, dimensionality reduction, DBSCAN, k-Means, EM clustering, mapping and J48 classification is presented. Metric used to determine one of the clustering algorithms is the number of correctly clustered instances. 20, 50 and 80 percent of 20% subset of NSL-KDD are used as input dataset and all results compared to 100 percent labeled NSL-KDD dataset. For 20 and 50 percent unlabeled flows EM clustering has more correctly clustered and for 80 percent, k-Means has better result. Overall recall, precision and F-Measure for all input dataset are more than 95.6% except for 50 percent unknown flows that is 90.1% and for a full labeled input dataset, J48 classification achieves more than 98.8% precision, recall and F-Measure. Datasets with 80 percent unknown flows cannot determine especial small amount of class label and this is because of its clustering algorithm. DBSCAN can determine all five label classes in different percentage of unlabeled flows so different policy in choosing one of clustering algorithms may be considered, for example time used for clustering execution, handling real time traffic classification and so on. Different policies will be used in traffic classification in future.

[12] TP
[13] FP
[14] FN

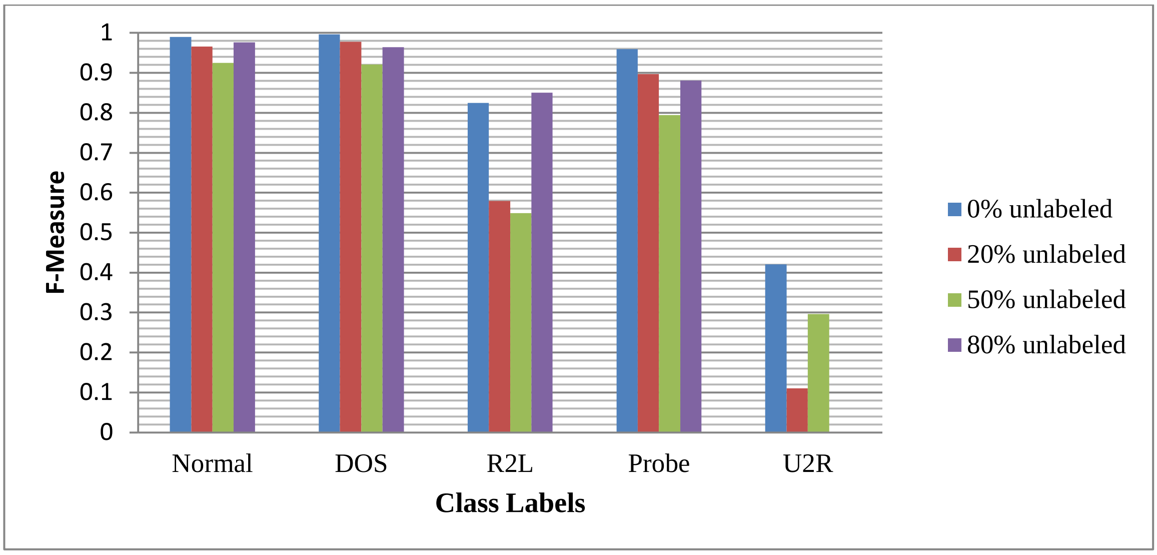

Fig. 3: Overall F-Measure in 5 different classes

REFERENCES

[1]. T. Pang-Ning, M. Steinbach, and V. Kumar, "Introduction to data mining," in Library of Congress, 2006.

[2]. K. D. Bailey, Typologies and taxonomies: an introduction to classification techniques vol. 102: Sage, 1994.

[3]. J. Kogan, Introduction to clustering large and high-dimensional data: Cambridge University Press, 2007.

[4]. T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," Communications Surveys & Tutorials, IEEE, vol. 10, pp. 56-76, 2008.

[5]. A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in Passive and Active Network Measurement, ed: Springer, 2005, pp. 41-54.

[6]. J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Offline/realtime traffic classification using semi-supervised learning," Performance Evaluation, vol. 64, pp. 1194-1213, 2007.

[7]. I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised clustering approach for network anomaly detection," in Networked Digital Technologies, ed: Springer, 2012, pp. 135-145.

[8]. J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in Proceedings of the 2006 SIGCOMM workshop on Mining network data, 2006, pp. 281-286.

[9]. J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Semi-supervised network traffic classification," in ACM SIGMETRICS Performance Evaluation Review, 2007, pp. 369-370.

[10]. V. Kumar, H. Chauhan, and D. Panwar, "K-Means Clustering Approach to Analyze NSL-KDD Intrusion Detection Dataset," International Journal of Soft Computing and Engineering (IJSCE) ISSN, pp. 2231-2307.

[11]. A. P. K. Shezad Shaikh, Vinod S. Mahajan, "Implementation of DBSCAN Algorithm for Internet Traffic Classification," International Journal of Computer Science and Information Technology Research (IJCSITR), vol. 1, pp. 25-32, October-December 2013.

[12]. J. Zhang, C. Chen, Y. Xiang, W. Zhou, and A. V. Vasilakos, "An effective network traffic classification method with unknown flow detection," Network and Service Management, IEEE Transactions on, vol. 10, pp. 133-147, 2013.

[13]. K. L. a. V. R. Vemuri, "Detecting and visualizing denial ofservice and network probe attacks using principal component analysis," in the Proc. of the 3rd Conference on Security Architectures, 2004.

[14]. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," Journal of the Royal Statistical Society. Series B (Methodological), pp. 1-38, 1977.

[15]. W. Lu and H. Tong, "Detecting Network Anomalies Using CUSUM and EM Clustering," in Advances in Computation and Intelligence, ed: Springer, 2009, pp. 297-308.

[16]. (10th July). The NSL-KDD Data Set. Available: http://nsl.cs.unb.ca/NSL-KDD/

[17]. E. Frank, M. Hall, G. Holmes, R. Kirkby, B. Pfahringer, I. H. Witten, et al., "Weka," in Data Mining and Knowledge Discovery Handbook, ed: Springer, 2005, pp. 1305-1314.