A Network Traffic Classification Technique using Clustering on Semi-Supervised Data

Ms. Sheetal S. Shinde Dr. Sandeep. P. Abhang

Abstract— **Network traffic classification is extensively required mainly for many network management tasks such as flow prioritization, traffic shaping/policing, and diagnostic monitoring Many approaches have been evolved till date like unsupervised approach, supervised approach and semi-supervised approach. This paper presents a network traffic classification based on semi-supervised approach. Flow statistics are used to classify the traffic. Few labeled and many unlabeled flows constitutes a training data set which is used to make the classifier. Here we develop the framework which consists of two processes the clustering which divides the training data into different groups and classification in which the labeling to the groups is done. To test the build model a well known test data set KDD CUP 1999[12] which include attack data and normal data , as well as iris dataset and GLASS data sets are used. This paper helps us to find out the appropriate value of K used to build no of groups as well as accuracy of the classifier build.**

Index Terms— **Component, Supervised, Unsupervised, Semi-supervised, Labeled, Unlabeled, Instance (records), Flow statistics/attributes/features, Training, Testing Classification, Clustering.**

I. Introduction

There are many applications of network traffic classification such as work load characterization and modeling, capacity planning and route provisioning. With real time constraints network operators always give high priority to the traffic interested in tools to manage traffic, such that traffic critical to business. This task of mapping flows to the network applications that generate the traffic is called traffic classification.

To identify traffic on internet the re-known method called "Port-based Classification"[1] , uses linking a well-known port number with a specific application. This method is ineffective because many recently developed applications do not communicate on standardized ports. Other method is "Deep Packet Inspection" [1]. In this approach, the packet payloads are analyzed to see whether or not they contain characteristic signatures of known applications. There are certain limitations. First, these techniques only identify traffic for which signatures are available. Maintaining an up-to-date list of signatures is a daunting task. Second, these techniques typically employ "deep" packet inspection because solutions such as capturing only a few payload bytes are insufficient or easily defeated. This technique can be extremely accurate when the payload is not encrypted. packet inspection techniques fail if the application uses encryption.

Machine learning [2,3], is one of the promising approach for traffic classification. There are two categories unsupervised and supervised in ML. The method in which the training data is labeled before is called as supervised learning. Labeled data means the input set for which the class to which it belong is known. The methodology in which the training data is unlabeled is called as unsupervised method. Unlabeled dataset is one for which class to which it belongs is unknown and is to be properly classified.

Another machine learning category is Semi-supervised methodology[4,5]. A learner and a classifier are two components of it. The learner is to distinguish a mapping between flows and traffic class from a training data set. consequently, the classifier is obtained using this learned mapping .

Fully labeled training data set is required to design the learner. It is very difficult as well as time consuming to obtain a fully labeled training data set. Quite the opposite, obtaining unlabeled training flows is reasonably priced. We build up and estimate a technique that allow us to design a traffic classifier using flow statistics using both labeled and unlabeled flows. Purposely, the learner is build using both labeled and unlabeled flows to show that unlabeled flows can help to make the traffic classification problem handy. Semi-supervised approach is advantageneous in the some situations. It is used to build fast and accurate classifier. This approach is vigorous and can lever formerly unseen flows. To improve the performance of the classifier it allows to add unlabeled flows. It classifies the given data set into appropriate classes using the k-means clustering algorithm[1,6,7,8].

II. Survey analysis

Traffic classification aims at identifying the traffic mixture in the network. With the fast evolution of Internet applications the efficiency of the port-based or payload-based identification approaches has been greatly diminished in current years. The machine learning (ML) based approach

Manuscript received Feb 10, 2012.

Ms. Sheetal S. Shinde, Computer Science and Engineering , Dr. B.A.M. University of Aurangabad/ M.I.T. College of Engineering/ G.S. Mandal's, Aurangabad, India , (e-mail: sheetal_only@rediffmail.com).

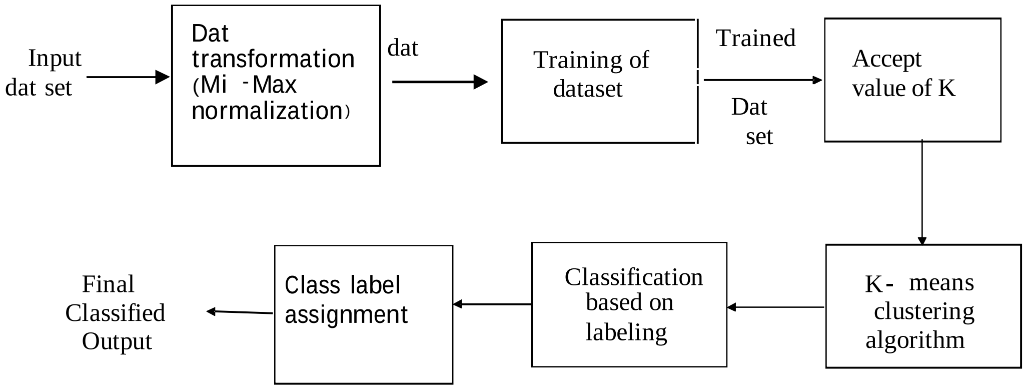
Dr. Sandeep P. Ahang , Computer Science and Engineering , Dr. B.A.M. University of Aurangabad/ M.I.T. College of Engineering/ G.S. Mandal's, Aurangabad, India, (e-mail: spabhang.india@gmail.com).

gets some port-independent and payload-independent statistical attributes of the traffic data and then use ML algorithms to build traffic classifier so that it can avoid the disadvantages of the earlier two methods.

The dynamic classification and identification of network applications[3], conscientious for network traffic flows offers considerable benefits to a number of key areas in IP network engineering, management and surveillance. Having number of shortfalls with these methods, many applications can use irregular port numbers and protocol decoding requires a large computing resources or simply not feasible if protocols are not known or encrypted.

Semi-supervised approach uses one of the technique called Self-training. In which, by using a small amount of labeled data a classifier is first trained. The unlabeled data is classified using the classifier. In general, the most convinced unlabelled points, together with their predicted labels, are added to the training set. The classifier is re-trained and the procedure repeated. Its own predictions are used by the classifier to teach itself. This procedure is also known as self-teaching or bootstrapping. Co-training is a technique assumes that (i) features can be split into two sets; (ii) each sub-set is sufficient to train a good classifier;(iii) the two sets are conditionally independent given the class.

III. SYSTEM ARCHITECTURE



A. The input data set

The input data set is the real data which captured in the real network. It includes many kinds of attack data, also includes the normal data.

B. Data Preprocessing

Real world data tend to be unclean, imperfect and conflicting. To increase the accuracy of the mining process as well as to improve the quality of data, data preprocessing techniques are used.

C. Data transformation

A appropriate normalization on data set is performed. This normalization technique is Min-max normalization. The data which satisfies the demands of K- Means algorithm should be numerical in nature. Therefore we have to convert the symbolic data into numerical and make them under the same evaluation standard initially.

1. Normalization:
   - Scaling attribute values to fall within a specified range.

     Example: To transform V in [min, max] to V' in [0,1],

     apply V'=(V-Min)/(Max-Min)
   - Scaling by using mean and standard deviation (useful when min and max are unknown or when there are outliers)
2. Aggregation: Moving up in the concept hierarchy on numeric attributes.
3. Generalization: Moving up in the concept hierarchy on nominal attributes.
4. Attribute construction: replacing or adding new attributes *D. Data reduction*

1. Reducing the number of attributes
   - Data cube aggregation: applying roll-up, slice or dice operations.
   - Removing irrelevant attributes: attribute selection (filtering and wrapper methods), searching the attribute space.
   - Principle component analysis (numeric attributes only): searching for a lower dimensional space that can best represent the data.
2. Reducing the number of attribute values
   - Binning (histograms): reducing the number of attributes by grouping them into intervals (bins).
   - Clustering: grouping values in clusters.
   - Aggregation or generalization
3. Reducing the number of tuples
   - Sampling : Data preparation and filtering steps can take considerable amount of processing time. Data pre-processing includes cleaning, normalization, transformation, feature ex- traction and selection, etc. The product of data pre-processing is the final training set

E. Clustering

K-means clustering algorithm gives details of cluster centroids along with number of components in each cluster. The mean of the parameter values of all the ponts in the cluster is nothing but the center of the cluster.

F. Assigning Cluster to Label

Once training data is clustered available labeled flows i.e. clusters are mapped to different known classes. In this semi-supervised learning process, some clusters are mapped to different flow types . The collection of number of records are treated as input for classification process. The instance of a dataset is a record or a tuple (x,y), where attribute set is denoted by x and class attribute by y. A set of flows is suppose $X = \{X_1,……..,X_N\}$. Xi is a flow instance, which is a vector of attribute values, $X_i= \{Xij \mid 1 \leq j \leq m\}$, where m is the number of attributes, and X is the value of the j^{th} attribute of the i^{th} flow. The set of traffic classes are denoted by Y , $Y =\{Y_1,……….,Y_q\}$, where the number of classes are denoted by q. The Y_i's can be classes.The mapping from m-dimensional variable X to Y forms a base for classification. The training is performed and the system is tested later on. The system is required to test on out of sample data. In the training phase center of the cluster is obtained. As well as in testing phase minimum distance of each record from centroid is compared , if found data is assigned to the same cluster.

G . *Cluster Labeling*

By using K-means clustering algorithm number of clusters are determined. A mapping from clusters to labels is done using probabilistic assignment technique. $P(Y = Yj |C_k)$, where j = 1,…,q where q is number of class types and k = 1,……….,K where K is the number of clusters. The set of flows which are labeled to different applications of training data are used to find out probabilities, (xi; yi), i =1,……,L, where L = the total number of different labeled applications. $P(Y = y_j \mid C_k)$ is then estimated by the maximum likelihood estimate, n_{jk} n_k , where n_{jk} is the number of flows that were assigned to cluster k with label j, and n_k is the total number of (labeled) flows that were assigned to cluster k. To complete probability of data samples belong. On the basis of this probability the class label is assigned to the clusters.k.

IV EXPERIMENTAL WORK

A. Dataset Description

An evaluation data set i.e KDD Cup 1999, Iris , Glass Data Set is used to train as well as to test the sysyem. Many machine learning algorithm were applied directly on the data which is a binary tcp dump data processed into connection records. Each connection record corresponds to a normal connection or to a specified attack. The data set is the

B. Experimenting with KDD CUP 99 dataset

The KDD Cup 1999 Intrusion detection data is used in our experiments. This data was prepared by the 1998 DARPA Intrusion Detection Evaluation program by MIT Lincoln Labs. Lincoln labs acquired nine weeks of raw TCP dump data. The raw data was processed into connection records, which consist of about 5 million connection records. The data set for our experiments contained 10000 records. This data set is divided

the mapping, now, within each cluster an evaluation is performed to determine to which class has maximum Classification model build is based on semi-supervised learning approach, thus both labeled and unlabeled data records are present. The output will basically would be the classification that will specify the class to which the dataset belong irrespective of the data input is labeled or unlabeled. The outcomes/results parameter after each phase can be viewed by the user. This classification technique helps in classifying data and also making the system to learn how to classify a new coming data. K is a positive integer number specifying the number of clusters, and has to be given in advance. The entire classification task terminate once the class are assigned to all the data samples.

Data is the basic input for any system. This system takes real world data set. Here attributes from dataset are real values which system analyst may encounter while dealing with real world applications. In this classification system where semi-supervised classification method is used require both labeled and unlabeld data. The Classification model is tested on KDD cup 99 dataset and it is also tested on IRIS dataset and GLASS dataset. Table I shows results obtained for the datasets specified. The cluster field specifies the value of the k and the accuracy is the percentage of accuracy of the classifier.

TABLE I: Results obtained for various values of K:

Glass data set		Iris data set		KDD CUP 1999	
clusters	Accuracy	Clusters	Accuracy	clusters	Accuracy
30	42.85%	30	97.22%	30	79.9%
35	41.07%	35	100%	35	83.3%
40	42.85%	40		40	66.95%
45	33.92%	45		45	79.45%
50	42.85%	50		50	80%
55	35.71%	55		55	58.6%
60		60		60	66.9%

real data which captured in the real network. It includes many kinds of attack data, also includes the normal data. The record includes TCP connection features such as duration, protocol and bytes, etc. The data set has 41 attributes for each connection record plus one class label, including 7 symbolic features and 34

continuous features. Each connection was described by features. The 39 different attack types present in the datasets. Each attack type falls exactly into one of the following four categories: Probing, DOS, U2R and R2L. So the classes included in these traces are: “Normal", “Probe”, “Dos”, “U2R” and “R2L”.

into training dataset which contained 8000 records and test dataset which contained 2000 records. Training dataset consists of 2400 labeled records and 5600 unlabeled records. As the data set has five different classes like normal data belongs to class1, probe belongs to class2, denial of service (DOS) belongs to class3, user to root (U2R) belongs to class4 and remote to local (R2L) belongs to class5. This dataset has been experimented on the designed classifier. The results obtained are displayed in next section, followed by discussion on the obtained results.

Following is the detailed description of the dataset:

- Number of Instances: 8000
- Number of Attributes: 41 plus the class attribute
- Attribute Information: duration, protocol_type, service, src_bytes, dst_bytes, urgent, land, wrong_fragment, flag, hot, num_failed_logins, logged_in,etc. Training dataset which contains 3200 labeled and 4800 unlabeled records gives 83.3% accuracy at K=30 (shown in table 1). Network Traffic Classifier gives higher accuracy on appropriate value of K.

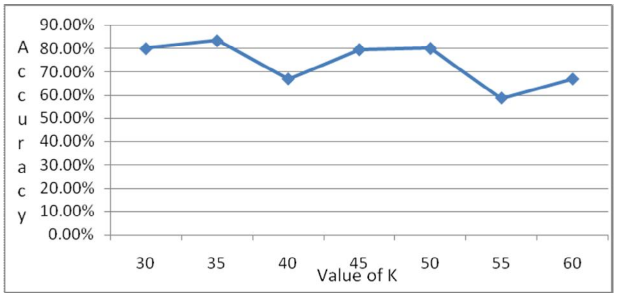


Fig. KDD Cup 1999

C. Experimenting with Iris Dataset

Iris Dataset is perhaps the best known dataset found in pattern recognition literature. This dataset have been experimented on the designed classifier. Training dataset which contains 36 labeled and 84 unlabeled records gives 100% accuracy at K=35 (shown in table 1).

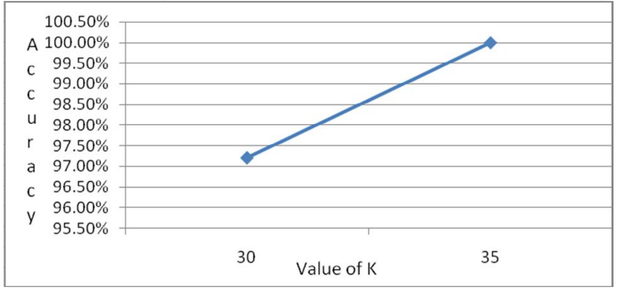


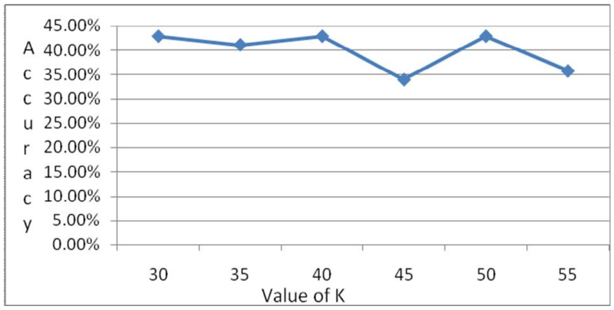
Fig Iris Data Set

For obtaining the classifier for IRIS dataset, the system constructed is trained with the number of training samples and rest of the samples are kept for testing. This is the Best Classifier obtained by this method.

D. Experimenting with GLASS Dataset

Fig. Glass Data Set

This dataset have been experimented on the designed classifier.



GLASS dataset have been experimented on the designed classifier. Training dataset which contains 54 labeled and 104 unlabeled records gives 42.85% accuracy at K=30 (shown in table 1).

V. Performance evaluation

The performance accuracy of the system is evaluated based on the generalization error. Generalization error can be obtained by testing the classifier using testing samples from the dataset. Testing samples are the remaining samples of the dataset after selecting samples for training. These samples are being tested for correct classification, thus find out that how much the system is generalized. The number of samples misclassified out of the total testing samples gives the generalization error. It is necessary to evaluate the performance of the system being designed. To do so generalization accuracy of the system is computed. The generalization accuracy is evaluated on the testing samples. Amongst all the testing samples it is determined how many samples are wrongly classified. Then accuracy percentage is given by

$$\frac{\text{Total Testing Sample- Misclassified Sample}}{\text{Total Testing Samples}} *100$$

Classifier gives 83.3 % accuracy respectively for KDD cup 99 dataset, for IRIS 100%, and for GLASS 42.85%.

VI. Conclusion

The aim of the project is to design and implement a semi-supervised learning approach for network traffic classification to find out the appropriate value of K. A semi-supervised approach to design a Network Traffic Classifiers is implemented successfully. Algorithm permits both labeled and unlabeled data to be used in training the network. It is observed that the range of classifiers accuracy lies between 42% to 100 % for various datasets. The results are shown in the table II. The value of the K for different data sets is ranges from 30 to 35.

TABLE II: Results obtained for Accuracy of Classifier:

Data Set	Training Dataset	Number of Features	labeled samples	Unlabeled Samples	No. Of Cluster (K)	Accuracy of Network Traffic Classifier
KDD CUP 1999	8000	41	3200	4800	30	83.3%
GLASS	158	09	54	104	30	42.85%
IRIS	120	4	36	84	35	100%

REFERENCES

[1] L. Yingqiu, Li Wei, L. Yunchun,” Network Traffic Classification Using K-means Clustering”, School of Computer Science and Engineering, Beihang University,Beijing 100083, China, 2007 IEEE

[2] Erman, A. Mahanti, and M. Arlitt. "Internet Traffic Identification using Machine Learning". in Proc.GLOBECOM'06, San Francisco,USA, November 2006.

[3] Automated Traffic Classification and Application Identification using Machine Learning Sebastian Zander, Thuy Nguyen, Grenville Armitage Centre form Advanced Internet Architectures Swinburne University of Technology ,Melbourne, Australia.

[4] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, “Semi-Supervised Network Traffic Classification”, SIGMETRICS'07, June 12.16, 2007, San Diego, California, USA. ACM 978-1-59593-639-4/07/0006.

[5] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, “Offline/Online Traffic Classification Using Semi-Supervised Learning”, Technical report, University of Calgary, 2007.

[6] Williamson,” Traffic Classification Using Clustering Algorithms”, University of Calgary, SIGCOMM’06 Workshops September 1115, 2006, Pisa, Italy. Copyright 2006 ACM 1595934170/06/0009.

[7] Munz, Sa Li, G. Carle , ” .Traffic Anomaly Detection Using K-Means Clustering “ , Computer Networks and Internet, Wilhelm Schickard Institute for Computer Science,University of Tuebingen, Germany .

[8] Amita Shrivastav Aruna Tiwari , Network Traffic Classification using Semi-Supervised Approach, 2010 Second International Conference on Machine Learning and Computing, © 2010 IEEEDOI 10.1109/ICMLC.2010.79

[9] E. Spafford, D. Zamboni, “Data collection mechanisms for intrusion detection systems”, Center for Education and Research in Information Assurance and Security , CERIAS Technical Report (2000)

[10] (Basic Book) H.Margaret , S. S. Dynham , “Data Mining Introductory and Advanced topics”.

[11] (Basic Book) M.Kamber, J.Han, “Data Mining Concepts and Techniques”, (2 nd ed.)

[12] KDD Cup 99 Intrusion Detection Datasets. Available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Sheetal Shinde received the B.E. degree in computer engineering from Pune University , Maharashtra, India and pursuing M.E. degree in Computer Science and Engineering from Dr. B.A.M. UNIVERSITY of Aurangabad Her research interests include machine learning , pattern classification and clustering.

Dr. Sandeep P. Abhang, Education : B.E., M.Tech. , Ph.D.
BE (CSE) M.Tech. (Computer engg.) Ph.D. (Computer Engg.) working as an Associate Professor in Computer science and engineering at MIT Aurangabad. Working in cloud computing domain, More than 10 papers are published in International journals.
Life Member of professional bodies like Computer society of India (CSI), ACEEE, ISTE.