Flow Classification Using Clustering And Association Rule Mining

Umang K Chaudhary, Ioannis Papapanagiotou and Michael Devetsikiotis
Electrical and Computer Engineering, North Carolina State University, Raleigh, USA
Emails:{ukchaudh,ipapapa,mdevets}@ncsu.edu

***Abstract*—Traffic classification has become a crucial domain of research due to the rise in applications that are either encrypted or tend to change port consecutively. The challenge of flow classification is to determine the applications involved without any information on the payload. In this paper, our goal is to achieve a robust and reliable flow classification using data mining techniques. We propose a classification model which not only classifies flow traffic, but also performs behavior pattern profiling. The classification is implemented by using clustering algorithms, and association rules are derived by using the "Apriori" algorithms. We are able to find an association between flow parameters for various applications, therefore making the algorithm independent of the characterized applications. The rule mining helps us to depict various behavior patterns for an application, and those behavior patterns are then fed back to refine the classification model.**

I. Introduction

New applications create a vast diversity of Internet traffic. As the Internet continues to expand, optimum provisioning and managing has become a severe challenge. These applications are finding innovative ways to tensely use network resources. The network operators need a methodology in which applications can be profiled and classified by taking into account their recent behavior patterns. Traffic classification and analysis will help Internet Service Providers (ISPs) to plan resources for applications and optimize their business goals [12]. Moreover, pattern recognition and behavior profiling has become a necessity in the discovery of malicious traffic, and Denial of Service attacks [13].

Traffic classification at the payload level has been limited due to the inclusion of sensitive information [1], storage and processing overhead, and payload encryption [10]. New applications signature patterns will need to have vigorous string and regular expression search and optimization. In order to address those issues Netflow has been extensively used by vendors, because it is able to capture flow information and requires less storage overhead. Flow based classification techniques use header information and connection patterns, in order to deduce classification models. However, the conventional method of port based classification cannot accurately identify applications, because several applications (e.g. P2P) are based on dynamic TCP/UDP port assignment or use of unregistered ports. Transport-layer heuristics [8], statistical-based classification [4] and Graph-base classification [7] have also been investigated for netflow classification. The main drawback of those techniques is that they cannot profile applications with changing behavior, as well as there is a need of constant profiler updates for any future application.

Several other works focus on application classification through the use of clustering algorithms in Netflow traces or identifying user behavior patterns [11], but according to the authors knowledge, there is no work that combines the advantages of both worlds. Based on this, we have developed a two step Netflow classification algorithm. The classification model provides interaction information of flow parameters. With interaction information, various traffic patterns for a particular application class can be identified. The proposed process, first utilizes a clustering algorithm to cluster and classify the data, and secondly implements association rule mining technique for labeling flow datasets. More specifically, our contribution can be summarized as follows:

- The classification precision, recall and overall accuracy of the proposed methodology reaches very high values due to the use of Model-based clustering and rule base classification (overall accuracy 94% and perfect HTTP precision).
- We are able to *profile behavior patterns for each network applications* using the Apriori[1] association rule algorithm [2]. With rule evaluation parameters such as lift, confidence and support, we are able to model and characterize interaction of the flow attributes.
- Finally, due to the creation of the association rules after clustering the data, the process is *independent of the dataset*. Therefore, the algorithm is able to identify, only from flow information, possible unknown applications.

The paper is structured as follows: In the next section, we describe various network traffic classification technique. In the third section, a model is proposed using the Apriori algorithm and clustering. In the fourth section, we analyze K-Mean and Model based clustering driven by Apriori classification and discuss their results. Finally, conclusions are provided at the last section.

II. Proposed Model

A. Classification Model

The main part of the algorithm consists of unsupervised learning techniques with associative rule mining, to achieve finer accurate flow classification. The methodology is depicted

[1] The algorithm Apriori is written as one word, while the Latin "a priori" is two words.

in Fig.1 and the format was developed such that it matches the specifications of [11]. More specifically the model is comprised of the following units.

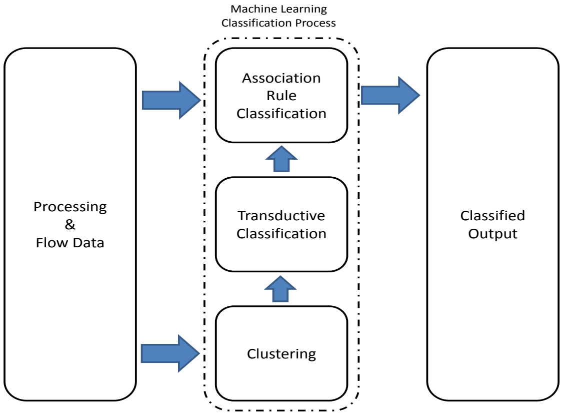

Fig. 1. Classification Model

The *Processing and Flow Data* unit converts the traces into flow data. The test traces are converted into flows and the attributes are stored in a database.

The *Clustering* unit applies clustering algorithms to the flow datasets. We compare both the K-Mean and Model based clustering algorithms, in order to determine the optimum performance. The unsupervised clustering techniques causes datasets with similar characteristics to be grouped together and to proceed for classification. In K-Mean, the data can be partitioned into k groups such that the Euclidean distance of the assigned cluster centers is minimized. K-mean requires the estimation of the number of clusters (k). In order to define the optimum k, the model was run for incremental values of k, and the optimum value was compared with Model-based Clustering algorithm. Model-based clustering assumes that data is produced by a data model. *Mclust package* is a normal mixture modeling and model-based hierarchical clustering with parameter estimation using Expectation-Maximization (EM). The algorithm includes a variety of covariance structures and functions for simulation from these models [6].

Model-based hierarchical clustering is used for clustering flow data. EM and Bayesian Information Criterion (BIC) provide comprehensive strategies for parametrization. Maximum Likelihood is used as a criterion for estimating the best fitted Model[2]. The prediction of the number of clusters is achieved by using BIC. Due to space limitations, the clustering techniques are not analyzed in detail [6].

Transductive Classification technique is used to label application classes (e.g., HTTP, SMTP) that constitute clusters. We apply the training and the test data to the clustering algorithms. The most common labels (applied during training phase) are defined as application class of the cluster. If there is a tie, or there are no labels, then classes are chosen randomly.

Association Rule Classification unit provides finer classification to the model. Association rules find regularities between flow parameters with different measures of interestingness for applications from transductive classifier output. The Apriori Algorithm is used for association rule learning. The derived rules are traced back to the main dataset and identified flows. Moreover, the rule association also helps predict IPs and ports used for servicing an application in the future. The rules heuristics applied to flow data causes accurate classification and thus making classification method finer due to association rule mining techniques.

B. *Classification Metrics*

The classification model is evaluated by the conventional machine learning metrics such as Precision, Recall and Accuracy [5]. The data set from a classification algorithm can be placed in a combination of the following categories - Positive and Negative. The following metrics are used as evaluation parameters for machine learning classification algorithms.

1) *True Positive* (TP): Total percentage of members (in our case flows) classified as Class A, which belongs to Class A.
2) *False Positive* (FP): Total percentage of members of Class A, which does not belong to Class A.
3) *False Negative* (FN): Total percentage of members of Class A, incorrectly classified as not belonging to Class A.
4) *True Negative* (TN): Total percentage of members which do not belong to Class A and are classified as not a part of Class A. It can also be given as (100% - FP).

These metrics will help us form the overall accuracy, precision and recall of the classification model. We define these terms as follows:

Precision is given as the ratio of True Positives to True and False Positives. This value helps us understand the classification capability to identify objects correctly.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Recall is given as the ratio of True Positives to True and False Negatives. This value helps us understand the classification capability to determine misclassified members are something else.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

Overall Accuracy is given as the ratio of all True Positives to all True and False Positives for all classes.

$$OverallAccuracy = \frac{\sum_{i=1}^{n}(TruePositive)}{\sum_{i=1}^{n}(TruePositive + FalsePositive)}$$

III. Evaluation and Results

We have utilized two clustering algorithms, K-mean and Model based clustering algorithms. For K-mean clustering, the number of clusters vary from 50 to 500 with a step of 50 and the average value of accuracy of each application is taken.

[2] In this case "Model" is defined as one of the Models of the Model-based clustering algorithm. Note that this Model is a subset of the proposed model

For Model based clustering, the number of clusters in the dataset are estimated by the BIC. The transductive classifier output is passed through rule-based classifier using the Apriori algorithm for classification.

A. Trace

For the evaluation we have used two data sets, one from a wired environment and one from a wireless. Waikato Internet Traffic Storage (WITS) traces were collected on 06-11-2001 at Auckland University. The data is anonymized with only TCP, UDP and ICMP and any payload within 64 bytes is zeroed [3]. Community Resource for Archiving Wireless Data At Dartmouth (CRAWDAD) provide wireless traces [9]. Packet headers from every wireless packet were sniffed form four of the campus buildings. The buildings were among the most popular wireless locations, and included libraries, dormitories, academic departments and social areas. The MAC address were sanitized by randomizing bottom six hex digits. IP address are anonymized using prefix-preserving IP sanitizer as described in [14]. For a given data set, we label the training data set by using port based classification.

B. Application Behavior Profiling

The association rule mining technique help us derive valuable relations within the datapoints. In our case, such association will help us recognize pattern in a particular application. The goal of the data mining and classification models is to build a heuristic for predictive recognizing the occurrence of application class from the datapoints.

The association rules composes of two item sets called an antecedent and consequent. *Antecedent* is the preceding event and *Consequent* is an event associated and followed after antecedent. In other words, an event occurring due to antecedent is followed by the consequent is depicted by the association rules for a particular class. Each rule is associated with three parameters, [2].

1) *Support* is the percentage of transactions that the rule can be applied to (the percentage of transactions, in which it is correct).
2) *Confidence* is the number of cases in which the rule is correct relative to the number of cases in which it is applicable (and thus is equivalent to an estimate of the conditional probability of the consequent of the rule given its antecedent).
3) *Lift* is the ratio of the probability that antecedent and consequent occur together, to the multiple of the two individual probabilities for antecedent and consequent. There are conditions where both support and confidence is high, and still result in to invalid rule. Therefore Lift indicates the strength of a rule over random occurrence of antecedent and consequent, given their individual support. It provides information about improvement and increase in probability of consequent for a given antecedent. In the other words, Lift is given as

$$Lift = \frac{RuleSupport}{Support(Antecedent) * Support(Consequent)}$$

TABLE I
ASSOCIATION RULES FROM APRIORI ASSOCIATION ALGORITHM

Traffic	Rules	C	Lift
DNS	dip=16 => dport=56322	1	21.8
	dport=4207 => dip=43	1	17.7
Mail	dip=8 => dport=53	1	6.7
	dip=16 => dport=53	1	6.7
HTTP	dip=1 & dport=1273 => srcip=233	1	81.5
	{} => dip=1	0.7	1
IRC	sip=162 => dport=37273	1	69
	{} => dip=16	0.7	1
SMTP	sip=262 & dport=4868 => dip=28	1	2.8
	dip=28 & dport=4868 => sip=262	1	15.3

Rules with lower lift and confidence values are filtered out. Confidence is measured for certainty of the rule. It measures event's itemset that matches the antecedent of the implication in the association rule and also matches consequent. Table I clearly mentions flow parameters in antecedent and consequent format for building association rules. The rules in the table are represented as

Antecedent => Consequent

The association rule indicates an affinity between antecedent and consequent with evaluation parameters such as Support, Confidence and Lift.

For example, for HTTP, destination IP with index 1 and destination port number equal to 1273 will have a flow from source IP of index 233 with confidence (noted as C) of 1 and lift of 81.5. When we trace back this flow, we are accurately able to classify this flow as HTTP. Moreover, in the future, if we observe any particular flow with this association then we would be certain to classify it as HTTP with confidence value of 1.

C. Classification

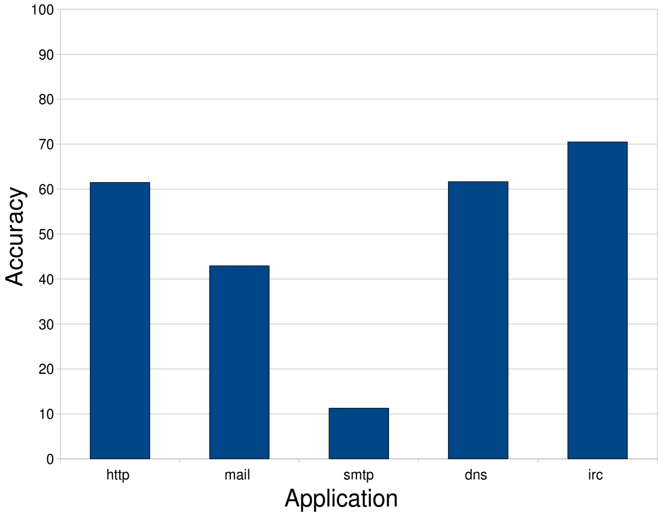

Fig. 2. Accuracy for K-mean clustering and rule-based classification

In Fig. 2, we show accuracy values of each application after clustering and rule-based classification. The number of cluster (k) increase causes growth in precision and recall for a particular application. This is because, the number of flows in each clusters decrease and mean square error reduces for each iteration. This causes flow data to be tightly clustered. With mean square error diminishing, misclassified flows are eliminated from the application class. We can observe an increase in precision for all the application classes. However, with the increase of the number of clusters the processing time also increases. After several runs, we noted diminishing returns when the number of clusters was greater than 300. In order to get the optimum accuracy, we have averaged it for all values of k between 300 and 500.

However, there is a need to estimate the number of clusters k for each clustering run. As we know same application class observes different behavior, therefore the value of k cannot simply be equal to the number of application classes. Hence it is difficult to estimate the k value for each new dataset as well as to maintain high accuracy and precision. One of the reasons is that K-mean clusters are partitioned by mean values iteratively. The mean values of random selected centers converge to the appropriate cluster. Finally, we observe that the overall accuracy is poor with a combination of K-mean with transductive and rule-based classification. Hence, we have used Model-based clustering that as will be shown, solves both issues.

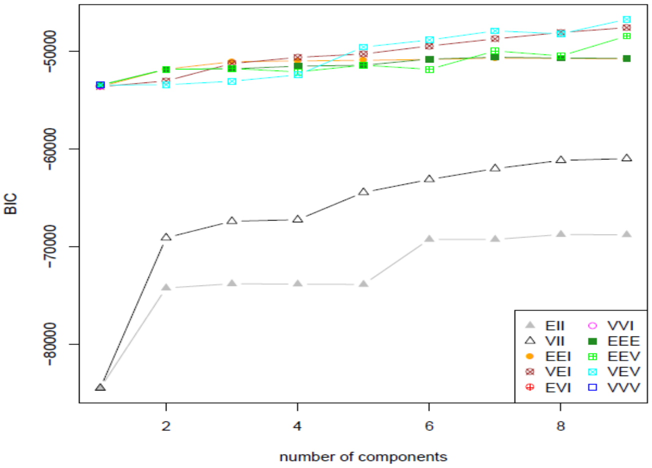

Fig. 3. Bayesian Information Criterion distribution

From Fig.3, we observe that Mclust uses BIC to estimate models and number of clusters. Model-based clustering estimates the cluster contour. For Mclust, geometric features are determined by covariances Σ_k. Each covariance matrix is parameterized by eigenvalue decomposition in the form [6]

$$\Sigma_k = \lambda_k d_k A_k D_k^T$$

where, D_k = orthogonal matrix of eigenvectors, A_k = diagonal matrix whose elements are proportional to the eigenvalues of Σ_k, λ_k = scalar quantity.

In the model, orientation for Σ_k depends on D_k. Also, the shape of the density contours is determined by A_k. The volume of the corresponding ellipsoid is proportional to $\lambda_k^d \mid A_k \mid$ where d represents data dimension. The characteristics of the geometric features of the cluster $-$ *orientation*, *volume* and *shape* are estimated from the data. These can vary between clusters, or be equal for all clusters. Hence we are able to estimate cluster contour for achieving precise model-based hierarchical clustering.

In one dimension dataset, equal variance and varying variance is represented by "E" and "V" respectively. For multiple dimensions, geometric characteristics of the model such as volume, shape and orientation is taken into consideration. The volume, shape and orientation represents clustering of flow data. In our case, VEV model represents volumes of all clusters as varying (V), shapes of all clusters as equal (E) and orientation of all clusters as varying (V). BIC is an approximation to the Bayes factor. It adds a penalty term to the log-likelihood estimation depending on the number of parameters.

$$BIC(k, \Sigma) \equiv 2 * p(x|\Theta, k, \Sigma) - v(k, \Sigma) * log(n)$$

where, k is the number of clusters, p is the conditional probability; π stands for different variance-covariance structures and $v(k, \Sigma)$ is the number of free parameters in a model with k clusters and covariance structure Σ; Θ represents maximum likelihood estimate of the parameters in the model (k, Θ) and n is the number of observations. BIC for each model is evaluated, and model with the highest BIC is the best model for the flow dataset.

As shown in Fig. 3, the best estimated model is VEV. VEV indicates volume and orientation is variable (V) and shape is equal (E) resulting into *Ellipsoidal distribution*. We can see in Fig.3 that BIC is strongest for VEV model for number of clusters equal to 9. The number of clusters is very close to the number of applications, as opposed to K-mean technique. Note that to get the highest accuracy from K-mean, more than 300 clusters were required.

In Fig.4, it is clearly shown that the model based clustering has performed better than K-mean classification for both data sets. Model-based clustering produces strong application clusters in the data. The Apriori based association rule classifier finds stronger association between flow parameters. The rules with high lift and confidence value, represent stronger correlation to the application. Hence, the rule set help us derive behavior pattern for a particular application class. We trace back the flows to the main trace file and we observe a strong probability that those flows belong to a particular application class. The overall accuracy of this method is around 94%, thus much higher than K-mean clustering method. We tested our model for CRAWDAD data and achieved similar accuracy of 93%.

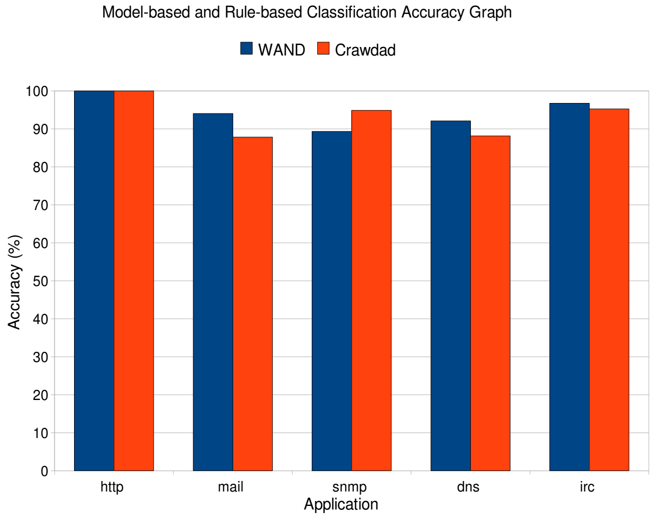

Fig. 4. Accuracy for Model-based clustering and rule-based classification

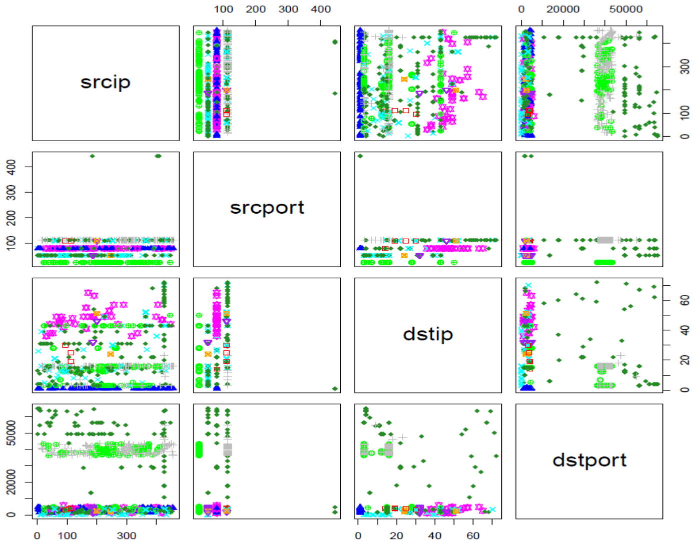

Fig. 5. Distribution of flow attributes in scatter plot matrix

In Fig.5, the distribution of IP and ports for source and destination is depicted for Model-based clustering method. For HTTP (blue) application, large number of source IPs direct traffic to few destination IPs. HTTP behavior is prominent because a number of source hosts are communicating with small number of destination hosts, i.e., HTTP servers. In DNS application, source hosts communicate with a set of DNS servers. Each host interacts with different domains such as www, .org and .net for resolving IP addresses. The DNS behavior is prominent because large group of source hosts communicate with small repeated set of destination IPs. In the IRC application, a group of hosts communicates with other group of hosts. There is roughly one-to-one correspondence between source and destination IP. This behavior is prominent because IRC (cross) traffic has set of source IPs communicate with approximately equal number of destination IPs. Moreover, source IP and destination port distribution show that destination port numbers are in the range of 30000 for SMTP (green) traffic. DNS destination ports are randomly distributed where as HTTP and MAIL have destination ports number in lower range of 1000. Model-based clustering with the Apriori algorithm provides reliable and accurate classification model as compared to K-mean method. The association rules helps us predict flows for particular application with high confidence and lift values.

IV. Conclusion

In conclusion, we have presented a classification model that achieves high flow classification accuracy with application behavior profiling. The use of the K-mean algorithm for Netflow classification was shown to be inefficient. On the other hand, Model based clustering with association rule mining techniques provided a much better accuracy. Moreover, the rule heuristics are produced automatically, making the algorithm modular and independent of the dataset. In addition, our model is able to detect new behavior patterns for next generation applications. As a future work, we are planning to analyze a bigger database of traces and define behavior patterns for a wider range of applications.

References

[1] F. Baker, B. Foster, and C. Sharp. Cisco architecture for lawful intercept in IP networks. *Internet Engineering Task Force, RFC*, 3924, 2004.

[2] C. Borgelt and R. Kruse. Induction of association rules: Apriori implementation. In *Compstat: Proceedings in Computational Statistics: 15th Symposium Held in Berlin, Germany, 2002*, page 395. Physica Verlag, 2002.

[3] WAND Trace Catalogue. *http://www.wand.net.nz/wits/catalogue.php*.

[4] C. Dewes, A. Wichmann, and A. Feldmann. An analysis of Internet chat systems. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 51–64. ACM, 2003.

[5] J. Erman, A. Mahanti, and M. Arlitt. Internet traffic identification using machine learning. In *Proceedings of IEEE GlobeCom*. Citeseer, 2006.

[6] C. Fraley and A.E. Raftery. MCLUST version 3 for R: Normal mixture modeling and model-based clustering. Technical report, Citeseer, 2006.

[7] M. Iliofotou, H. Kim, M. Faloutsos, M. Mitzenmacher, P. Pappu, and G. Varghese. Graph-based P2P traffic classification at the internet backbone. In *IEEE Global Internet Symposium*. Citeseer, 2009.

[8] T. Karagiannis, A. Broido, and M. Faloutsos. Transport layer identification of P2P traffic. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 121–134. ACM, 2004.

[9] David Kotz, Tristan Henderson, Ilya Abyzov, and Jihwang Yeo. CRAWDAD trace dartmouth/campus/tcpdump/fall01 (v. 2004-11-09). Downloaded from http://crawdad.cs.dartmouth.edu/dartmouth/campus/tcpdump/fall01, November 2004.

[10] A.W. Moore and K. Papagiannaki. Toward the accurate identification of network applications. *Passive and Active Network Measurement*, pages 41–54, 2005.

[11] TTT Nguyen and G. Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, 10(4):56–76, 2008.

[12] I. Papapanagiotou and M. Devetsikiotis. Aggregation Design Methodologies for Triple Play Services. In *IEEE CCNC 2010, Las Vegas, USA*, pages 1–5, 2010.

[13] V. Paxson. Bro: A system for detecting network intruders in real-time. *Comput. Networks*, 31(23):2435–2463, 1999.

[14] J. Xu, J. Fan, M. Ammar, and S.B. Moon. On the design and performance of prefix-preserving IP traffic trace anonymization. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, page 266. ACM, 2001.