An Efficient Method for Internet Traffic Classification and Identification using Statistical Features

Remya Raveendran
Computer Science and Engineering
AdiShankara Institute of Engineering and Technology
Kalady, India

Raghi Menon
Computer Science and Engineering
AdiShankara Institute of Engineering and Technology
Kalady, India

Abstract— **Traffic Classification is a method of categorizing the computer network traffic based on various features observed passively in the traffic into a number of traffic classes. Due to the rapid increase of different Internet application behaviors', raised the need to disguise the applications for filtering, accounting, advertising, network designing etc. Many traditional methods like port based, packets based and some alternate methods based on machine learning approaches have been used for the classification process. Proposed a new traffic classification scheme to utilize the information among the correlated traffic flows generated by an application. Discretized statistical features are extracted and are used to represent the traffic flows. The removal of irrelevant and redundant features from the feature set is done by Correlation based feature selection with high class-specific correlation and low inter correlation. For the classification process Naïve Bayes with Discretization is used. The proposed scheme is compared with three other Bayesian models. The experimental evaluation show that NBD outperforms the other methods even in the case of a small supervised training samples.**

Keywords— ***Traffic Classification; Traffic Flows; Naïve Bayes; Correlation Based Feature Selection; Feature Discretization***

I. INTRODUCTION

Application oriented traffic classification is a fundamental technology for modern network security. It is useful to tackle a number of network security problems including lawful interception and intrusion detection. For example, traffic classification can be used to detect patterns indicative of denial of service attacks, worm propagation, intrusions, and spam spread. In addition, traffic classification also plays an important role in modern network management, such as quality of service (QoS) control. Many open source and commercial tools, with traffic classification function have been deployed and there is an increasing demand on the development of modern traffic classification techniques. While traditional traffic classification techniques may rely on the port numbers specified by different applications or the signature strings in the payload of IP packets, modern techniques normally utilize host/network behavior analysis or flow level statistical features by taking emerging and encrypted applications into account. Recently, substantial attention has been paid on the application of machine learning techniques to statistical features based traffic classification. In the state-of-the-art traffic classification methods, Internet traffic is characterized by a set of flow statistical properties and machine learning techniques are applied to automatically search for structural patterns. These methods can address the problems suffered from by the traditional methods, such as dynamic port numbers and user privacy protection.

Recent research shows that flow statistical feature based traffic classification can be enhanced by feature discretization. Particularly, feature discretization is able to dramatically affect the performance of naive Bayes (NB). NB is one of the earliest classification methods applied in Internet traffic classification, which is a simple and effective probabilistic classifier employing the Bayes' theorem with naive feature independence assumptions. Since independent features are assumed, an advantage of the NB classifier is that it only requires a small amount of training data to estimate the parameters of a classification model. The main reason for the underperformance of a number of traditional classifiers is the lack of the feature discretization process. For example, feature discretization can effectively improve the accuracies of the support vector machine (SVM) and k-NN algorithms at the price of lower classification speed. More interestingly, NB with feature discretization demonstrates not only significantly higher accuracy but also much faster classification speed. Considering complex network situation, a difficult question is that how to obtain a high-performance statistical feature based traffic classifier using a small set of training data. The solutions to this question are essential to address a number of difficult problems in the field of network security and management. For instance, in practice, we may only manually label very few samples as supervised training data since traffic labelling is time-consuming, especially for new applications and encrypted applications.

Moreover, a big challenge for current network management is to handle a large number of emerging applications, where it is almost impossible to collect sufficient training samples in a limited time. These observations motivate our work. In this paper, we provide a solution to effectively improve the traffic classifier with a small set of training samples. We propose a new traffic classification scheme to utilize information among the correlated traffic flows generated by an application. In the proposed scheme, an extension of Naïve Bayes called Naïve Bayes with Discretization (NBD) is used. The empirical study shows that the proposed scheme can effectively improve the traffic classification performance with a small set of training data and it outperforms the existing state-of-the-art traffic classification methods.

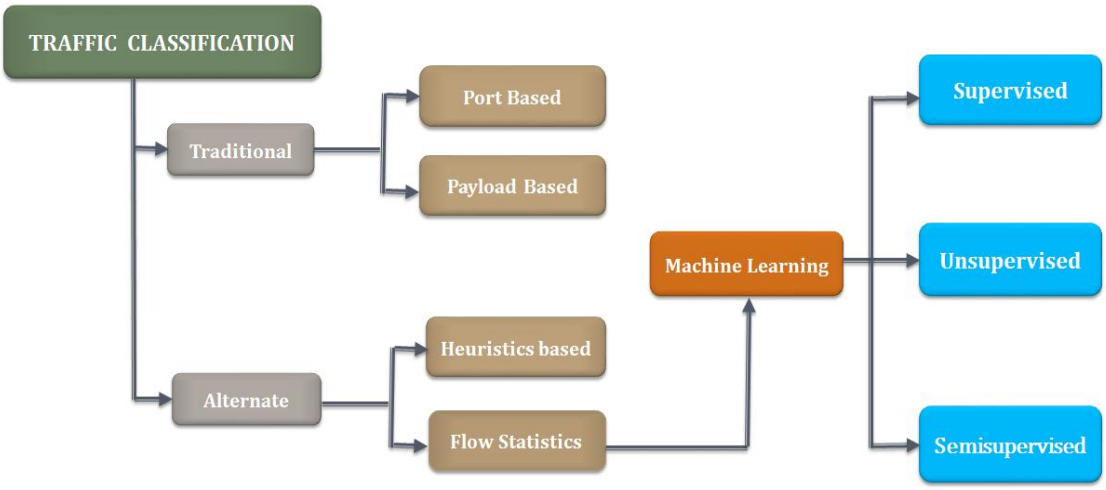

Fig. 1. Traffic Classification Approaches

II. Related Works

The phrase traffic classification refers to methods of classifying traffic data sets based on features passively observed in the traffic, according to specific classification goals. Although traffic classification is a rather specific research field, the goals of these research papers are not identical. Some only have coarse classification goals, i.e., whether it’s transaction-oriented, bulk-transfer, or peer-to-peer file sharing. Some have a finer-grained classification goal, i.e., the exact application generating the traffic. Selection of traffic features used for classification evolves with application development. Media-rich entertainment applications and associated attempts to discriminate against such applications.

Fifteen years ago, researchers could reasonably accurately classify traffic using TCP or UDP port numbers, but as applications began to use unpredictable ports, accurate classification requires payload examination. Examining payload is a controversial methodology due to privacy concerns, and is not even possible for encrypted payload, so researchers have also studied techniques that are independent of packet content, such as statistical features based on network flows or underlying social networks to identify per-host behavior.

Methods to classify traffic at an application level include exact matching, e.g., of port number and payload; heuristic methods, applied e.g. on connection patterns to infer social networks; or machine learning based on statistical features. We group machine learning methods into two categories: Supervised Learning and Unsupervised Learning. Naive Bayes, Decision Tree, NN, LDA, QDA, Bayesian Neural network are supervised learning algorithms; EM, AutoClass and K- Means are unsupervised learning algorithms. Traffic classification approaches are depicted in *Fig. 1*.

A. Port based IP traffic classification:

TCP and UDP provide multiplexing of multiple flows between IP endpoints with the help of port numbers. Historically many applications utilize a ‘well known’ port to which other hosts may initiate communication. The application is inferred by looking up the TCP SYN packet’s target port number in the Internet Assigned Numbers Authority (IANA)’s list of registered ports. However, this approach has limitations. Firstly, some applications may not have their ports registered with IANA (for example, peer to peer applications such as Napster and Kazaa). An application may use ports other than its well-known ports to avoid operating system access control restrictions. Also, in some cases server ports are dynamically allocated as needed. Although port-based traffic classification is the fastest and simple method, several studies have shown that it performs poorly, e.g., less than 70% accuracy in classifying flows [1] [2].

B. Payload based IP traffic classification:

This approach inspect the packet header to determine the applications. Packet payloads are examined bit by bit to locate the bit streams that contain signature. If such bit streams are found, then packets can be accurately labelled. This approach is commonly employed for P2P traffic detection and network intrusion detection. Major disadvantages of this approach is that the privacy laws may not allow administrators to inspect the payload*;* it also imposes significant complexity and processing load on traffic identification device; requires substantial computationally power and storage capacity since it analyses the full payload [2] [3] [4].

C. Protocol Behaviour or Heuristics Based Classification:

Transport-layer heuristics over a novel method that classifies traffic to their application types based on connection-level patterns or protocol behavior. This approach is based on observing and identifying patterns of host behavior at the transport layer. The main advantage of this method is that there is no need for packet payload access [3] [5].

D. Classification based on flow statistics traffic properties:

The preceding techniques are limited by their dependence on the inferred semantics of the information gathered through deep inspection of packet content (payload and port numbers). Newer approaches rely on traffic's statistical characteristics to identify the application [6][7][8][9]. An assumption underlying such methods is that traffic at the network layer has statistical properties that are unique for certain classes of applications and enable different source applications to be distinguished from each other. It uses network or transport layer which has statistical properties such as distribution of flow duration, flow idle time, packet interarrival time, packet lengths etc. These are unique for certain classes of applications and hence help to distinguish different applications from each other. This method is feasible to determine application type but not generally the specific client type. For example, it can't determine if flow belongs to Skype or MSN messenger voice traffic specifically. The advantage of this approach is that there is no packet payload inspection involved.

E. Machine Learning IP Traffic Classification Approaches:

Machine learning (ML) techniques [8] [11]provide a promising alternative in classifying flows based on application protocol (payload) independent statistical features such as packet length and inter-arrival times. Each traffic flow is characterized by the same set of features but with different feature values. A ML classifier is built by training on a representative set of flow instances where the network applications are known. The built classifier can be used to determine the class of unknown flows. Much of the existing research focuses on the achievable accuracy (classification accuracy) of different machine learning algorithms. The studies have shown that a number of different algorithms are able to achieve high classification accuracy. The effect of using different sets of statistical features on the same dataset has seen little investigation. ML approaches for traffic classification are supervised, unsupervised and semi-supervised. Review of some techniques are given as follows.

1) Traffic Classification Using Clustering Algorithms[10]:

Different types of clustering algorithms have been proposed such as K-Means, DBSCAN, AutoClass etc. The K-Means clustering algorithm is a partition-based algorithm, the DBSCAN algorithm is a density-based algorithm, and the AutoClass algorithm is a probabilistic model-based algorithm. . This work considers two unsupervised clustering algorithms, namely K-Means and DBSCAN, that have previously not been used for network traffic classification. The authors evaluate these two algorithms and compare them to the previously used Auto Class algorithm, using empirical Internet traces. The experimental results show that both K-Means and DBSCAN work very well and much more quickly then Auto Class. Our results indicate that although DBSCAN has lower accuracy compared to K-Means and Auto Class, DBSCAN produces better clusters.

2) Classification using Semi Supervised Technique:

Erman et al. [13] proposed to use a set of supervised training data in an unsupervised approach to address the problem of mapping from flow clusters to real applications. However, the mapping method will produce a large proportion of “unknown” clusters, especially when the supervised training data is very small. In this paper, we study the problem of supervised traffic classification using very few training samples. From the supervised learning point of view, several supervised samples are available for each class.

3) Classification using Supervised Techniques:

Bayesian Network (BayesNet) [8] [12] [18] is structured as a combination of a directed acyclic graph of nodes and links, and a set of conditional probability tables. Nodes represent features or classes, while links between nodes represent the relationship between them. Conditional probability tables determine the strength of the links. This paper has demonstrated the successful application of a Bayesian trained neural network to Internet classification on data from a single site for two days, eight months apart. The main findings are as follows. A sophisticated Bayesian trained neural network is able to classify flows, based on header-derived statistics and no port or host (IP address) information, with up to 99% accuracy for data trained and tested on the same day, and 95% accuracy for data trained and tested eight months apart. Further, the neural network produces a probability distribution over the classes for a given flow. The entropy of this distribution is (negatively) correlated with prediction accuracy, and can be used as a rejection criteria for the predictions. Accuracy is further improved when a proportion of predictions may be rejected. The accuracy values significantly improve upon those from a naive Bayesian method and compare favorably with the 50%–70% figure reported using the IANA port list. By providing high accuracies without access to packet payloads or sophisticated traffic processing this technique offers good results as a low-overhead method with potential for real-time implementation.

Naive Bayes [14] technique is a supervised machine learning technique. Flow contents such as port numbers, flow length and time between consecutive flows are used to train the classifier. Moreover, to train the classifier 248 full-flow based features were used. The chosen traffic for application was categorized into different groups such as database, mail services, games and multimedia, www, p2p, bulk data transfer and attack. Later the work is extended with use of neural network approach. The authors proposed a Bayesian framework which classifies traffic without the use of any port or host address. A multilayer perceptron classification network is used for assigning probabilities to flows. 246 flow features are used as input to the first layer of network.

Naïve Bayes Tree (NBTree) [13] is a hybrid of a decision tree classifier and a Naïve Bayes classifier. Designed to allow accuracy to scale up with increasingly large training datasets, the NBTree model is a decision tree of nodes and branches with Naïve Bayes Classifiers on the leaf nodes.

Naive-Bayes with kernel estimation (NBK) [8] is based on the Bayesian theorem. This classification technique analyses the relationship between each attribute and the class for each instance to derive a conditional probability for the relationships between the attribute values and the class. Naïve Bayesian classifiers must estimate the probabilities of a feature having a certain feature value. Continuous features can have a large (possibly infinite) number of values and the probability cannot be estimated from the frequency distribution. This can be addressed by modelling features with a continuous probability distribution. Here evaluate Naive Bayes in kernel density estimation (NBK). Kernel density estimation models features using multiple (Gaussian) distributions, and is generally more effective than using a single (Gaussian) distribution.

III. Methodology

In proposed system, a novel parametric approach is used to deal with the correlated flows in an effective way, which can significantly improve the classification performance. The classification process of our proposed scheme is focused on flow-level traffic classification. In pre-processing step we have to use Correlation Coefficient to provide high class specific data and the individual predictions of the correlated flows so as to conduct more accurate classification. Our research shows that the goal can be achieved by following the approach of classifier combination. Naive Bayes classifier has demonstrated high classification speed and good performance using the discretized statistical features in traffic classification. It is easy for Naive Bayes classifier to produce the posterior probability that a testing flow belongs to a traffic class. The scheme can achieve much better classification performance than the existing state of the art methods. Also it is time consuming and improves the accuracy. The classification process is given in *Fig 2.*

A. Pre-processing

Here the IP packets crossing across a network is collected and used for constructing the flows by examining the header of packets. A flow can be define as successive IP packets having the same 5-tuple: source IP, source port, destination IP, destination port, and transport layer protocol. Since we are focusing on a statistical approach for classification process, we need to extract the flow statistical features and is discretized for representing the traffic flows.

B. Correlation Based Feature Selection

Here statistical features are extracted and are used to represent traffic flows that is done by pre-processing to apply feature selection **[16]** to remove irrelevant and redundant features from the feature set. The correlation-based feature subset selection is used in the experiments, which searches for a subset of features with high class-specific correlation and low inter correlation. Correlation coefficient is denoted as 'r' where

$$r = \frac{[n(\sum xy) - (\sum x)(\sum y)]}{[\sqrt{[n(\sum x^2) - (\sum x)^2][n(\sum y^2) - (\sum y)^2]}]}$$

where n – is the number of instances

x – is the attributes to be tested or correlation

y – is the attributes to be tested against the x

Here the threshold value is selected as 0.75. The value above the threshold have high inter correlation and is considered to be redundant.

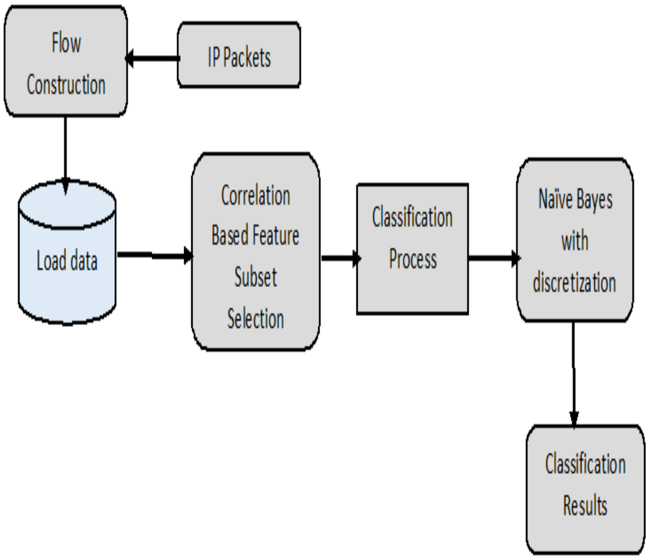

Fig. 2. System Design

C. Feature Discretization

Discretization **[17]** is a process of converting numeric values into intervals and associating them to a nominal symbol. These symbols are then used as new values instead of the original numeric values. The new dataset will be smaller than that of the previous one, i.e.) a discretized feature will be having a fewer possible values than that of non-discretized one. The key process in discretization is the selection of intervals which can be determined by an expertise in the field or by discretization algorithm. There are two approaches for discretization: One is to discretize each feature without the knowledge of the classes in the training set (unsupervised discretization). The other is to make use of the classes when discretizing (supervised discretization). Entropy Based Minimum Description Length (ENT-MDL) discretization is used here which is proposed by Fayyad and Irani [15].

D. Naïve Bayes Classification

A Naïve-Bayes (NB) ML algorithm [20] is a simple structure consisting of a class node as the parent node of all other nodes. The basic structure of Naïve Bayes Classifier is shown in *Fig 3* in which C represents main class and a, b, c and d represents other feature or attribute nodes of a particular sample. No other connections are allowed in a Naïve-Bayes structure. Naïve-Bayes has been used as an effective classifier. . It is easy to construct Naïve Bayes classifier as compared to other classifiers because the structure is given a priori and hence no structure learning procedure is required. Naïve-Bayes assumes that all the features are independent of each other. Naïve-Bayes works very well over a large number of datasets, especially where the features used to characterize each sample are not properly correlated.

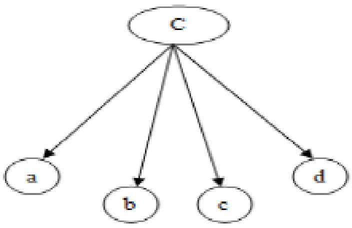

Fig 3: Naïve Bayes Classifier

IV. Experimental Evaluation

In this section, we evaluate the proposed HNB scheme on a real-world traffic dataset. The scheme is compared with three state-of-the-art traffic classification methods based on Bayes theorem.

A. *Dataset*

For the purse of classification, we are using the datasets that are created from the real world network traffic traces named as 'wide'. The wide dataset consists of traffic flows which are randomly selected from the wide trace and carefully recognized by the manual inspection. It consists of 3416 instances with 7 classes such as (bt, dns, ftp, http, smtp, yahoomsg, ssh) and 22 attributes. The statistical features extracted for the process [14] is listed in Table 1.

Correlation based feature selection with best first search is used for the creating the candidate set of features. Out of the 20 features, 10 features were selected as candidate features. Discretization based on MDL assigns the features to discrete nominal symbols, which reduces the feature values and increases the accuracy of the various algorithms.

TABLE1: STATISTICAL FEATURES

Types Of Features	Feature Description	Number
Packets	Number of packets transferred	2
Bytes	Volume of bytes transferred	2
Packet Size	Min, Max, Median and Standard. Deviation of packet size	8
Inter Packet Time	Min, Max, Median and Standard. Deviation of inter packet time	8
	Total	20

A. *Performance Evaluation*

Performance evaluation of the algorithm is done by using the following metrics: overall accuracy, precision, recall, F-measure, and classification speed.

- Overall accuracy: the ratio of the number of correctly classified traffic flows to the total number of all flows in a given trace.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

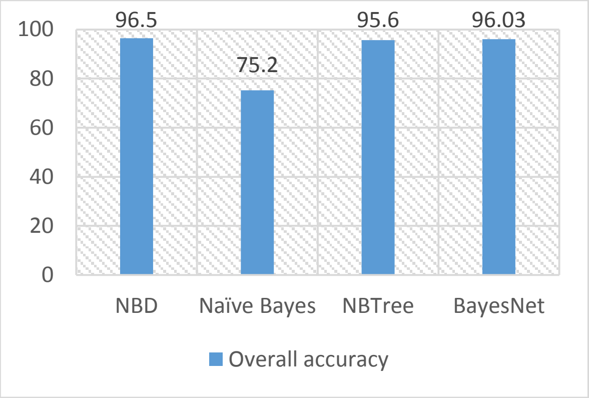

Fig. 3. Overall Accuracy

- Precision: the ratio of True Positives over the sum of True Positives and False Positives

$$Precision = \frac{TP}{TP + FP}$$

- Recall: the ratio of True Positives over the sum of True Positives and False Negatives

$$Recall = \frac{TP}{TP + FN}$$

- F-measure: widely-used metric in information retrieval and classification, it considers both precision and recall in a single metric by taking their harmonic mean.

$$F - measure = \frac{2 \times precision \times recall}{precision + recall}$$

For the experimental purposes the proposed scheme is compared with three other Bayesian models like Naïve Bayes, Bayes Net, and NBTtree. The analysis is done using the WEKA tool. Results shows that accuracy, precision, recall and F-Measure of the proposed scheme has increased as compared to the others as shown in the Table 2.

B. *Per-application performance*

Fig.5 shows the precision, recall and F-measure of seven applications like bt, dns, ftp, http, smtp, yahoomsg, ssh. From the figure it is clear that HNB provides high precision, high recall and high F-Measure for all of the 7 classes compared to the other Bayesian models.

C. *Overall Accuracy*

In Table 2 the accuracy of the algorithms is given. It shows that NBD outperforms the other by having an accuracy about 96.5%. *Fig.4* shows the graphical representation of the accuracy.

D. *Classification Speed*

The time taken or building the NBD model is only 0.12 sec but in the case of BayesNet and NBTree it is more. Among these NBTree consumes more time and Naïve Bayes consumes less time.

TABLE 2: PERFORMANCE EVALUATION

Algorithms	Correctly classified	Incorrectly Classified	Overall Accuracy (%)	Error (%)	Time (sec)
NBD	3407	9	96.48	3.59	0.12
Naïve Bayes	2570	846	75.232	24.76	0.08
BayesNet	3314	102	96.07	3.92	0.17
NBTree	3352	64	95.6	4.33	11.85

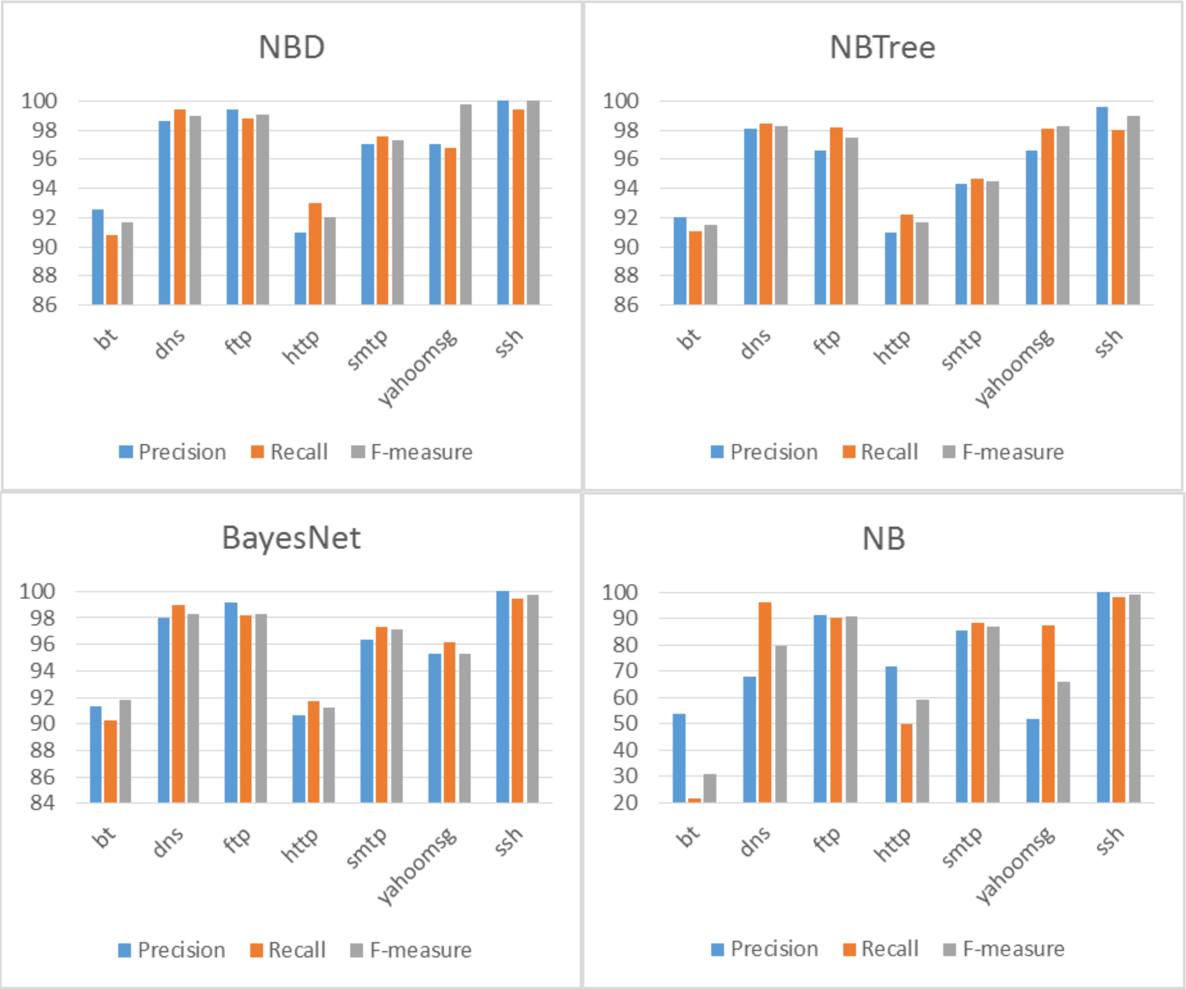

Fig. 4. Per Application Precision, Recall and F-Measure

V. CONCLUSION

Traffic classification plays an important role in the network security as the applications and their behavior are changing day to day. As a result there increased the need for accurate classification of the network flows. Here we have proposed a Naïve Bayes Discretization model with CFS feature selection for the accurate classification of internet traffic. We have compared the method with three other Bayesian models. Our experiment shows that it provide an accuracy of 96.5% which is better than that of the other state-of-the art methods. NBD is easy to build and is applicable to various real world applications.

REFERENCES

[1] J. Erman, A. Mahanti, and M. Arlitt. Byte Me: A Case for Byte Accuracy in Traffic Classification. In ACM SIGMETRICS MineNet Workshop, June 2007

[2] A. Moore and K. Papagiannaki. Toward the accurate identification of network applications. In P AM, April 2005.

[3] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. Blinc: Multilevel traffic classification in the dark. In ACM SIGCOMM, August 2005

[4] S. Sen, O. Spatscheck, and D. Wang. Accurate, scalable in-network identification of p2p traffic using application signatures. In WWW, May 2004

[5] T. Karagiannis, A. Broido, M. Faloutsos, and kc claffy . Transport layer identification of p2p traffic. In ACM IMC, October 2004.

[6] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli. Traffic classification through simple statistical fingerprinting. ACMSIGCOMM CCR, 37(1):7–16, January 2007

[7] L. Bernaille, R. Teixeira, and K. Salamatian. Early application identification. In ACM CoNEXT, December 2006.

[8] N. Williams, S. Zander, and G. Armitage. A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification. ACM SIGCOMM CCR, 36(5):7–15, October 2006.

[9] T. Auld, A. W. Moore, and S. F. Gull. Bayesian neural networks for internet traffic classification. IEEE Transactions on Neural Networks, 18(1):223–239, January 2007.

[10] J. Erman, M. Arlitt, and A. Mahanti. Traffic Classificaton Using Clustering Algorithms. In ACM SIGCOMM MineNet Workshop,September 2006.

[11] T. T. Nguyen and G. Armitage. A survey of techniques for internet traffic classification using machine learning. IEEE Communications Surveys and Tutorials, to appear, 2008.

[12] R. Bouckaert, "Bayesian Network Classifiers in Weka", Technical Report, Department of Computer Science, Waikato University, Hamilton, NZ 2005

[13] R. Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", in Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD), 1996.

[14] Jun Zhang, Chao Chen, Yang Xiang, Wanlei Zhou, and Yong Xiang, " Internet Traffic Classification by Aggregating Correlated Naive Bayes Predictions", IEEE

[15] Transactions on Information Forensics and Security, vol.8, no.1, pp.5-15, Jan. 2013.

[16] U.M.Fayyad and K.B.Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In Proceedings of the International Joint Conference on Uncertainty in Artificial Intelligence, 1993

[17] Guyon and A. Elisseeff, "An introduction to variable and feature selection,"J. Mach. Learn. Res., vol. 3, pp. 1157–1182, Mar. 2003

[18] H.Kim, K.Claffy, M.Fomenkov, D.Barman, M.Faloutsos, and K. Lee, "Internet traffic classification demystified: Myths, caveats, and the best practices," inProc. ACM CoNEXT Conf., New York, 2008, pp.1–12

[19] A. Moore and D. Zuev. Internet traffic classification using Bayesian analysis techniques. In ACM SIGMETRICS, June 2005.

[20] Liangxiao Jiang, Harry Zhang, Zhihua Cai, "A Novel Bayes Model: Hidden Naive Bayes", *IEEE Transactions on Knowledge & Data Engineering*, vol.21, no. 10, pp. 1361-1371, October 2009, doi:10.1109/TKDE.2008.234

[21] Ioan Pop, "An approach of the Naive Bayes classifier for the document classification," General Mathematics, Vol. 14, No. 4, pp.135-138, 2006