Volume 4, No. 4, March-April 2013

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Survey of Traffic Classification using Machine Learning

Jamuna .A*
Department of Computer Science and Engineering
Karunya University Coimbatore, India
jamunaa@karunya.edu.in

Vinoth Ewards S.E
Department of Computer Science and Engineering
Karunya University Coimbatore, India
ewards@karunya.edu

Abstract: Traffic classification based on their generation applications has a very important role to play in network security and management. Traditional methods include the port-based prediction methods and payload-based deep inspection methods. Within the current network environment, the standard strategies suffer from variety of privacy issues, dynamic ports and encrypted applications. Recent research efforts are focused traffic classification supported flow statistical options and Machine Learning Techniques. This paper conducts a survey on the various Machine Learning (ML) techniques for IP traffic classification. Recent research tends to use machine learning techniques for classification.

Keywords: Traffic classification, Machine Learning (ML), Payload based-deep inspection methods.

I. INTRODUCTION

A substantial interest is shown on network traffic classification for the past few years. Grouping of traffic flows by their generation applications plays a vital role in network security and management, such as, lawful interception, intrusion detection and Quality of Service (QoS) control. The payload-based deep inspection methods and the port-based prediction methods are the conventional traffic classification methods. In the present scenario, the conventional classification methods undergo many practical problems such as dynamic ports and encrypted applications. The application of machine learning techniques based on flow statistical features used to traffic classification and it has been the source of interest for research in recent times. To intelligently conduct traffic classification, the Machine Learning would be supportive, as it can automatically search and describe useful structural patterns in a supplied traffic dataset. The application of Machine Learning techniques (a subset of the wider Artificial Intelligence discipline) to IP traffic classification has been closely concentrated by researchers.

Initially the unknown IP traffic may be identified and differentiated by defining its features. Features are properties of flows which are calculated over multiple packets (maximum or minimum packet lengths in each direction flow durations or inter-packet arrival times). Then the ML classifier is skilled to correlate the sets of features with known traffic classes (creating rules). By using previously learned rules the ML algorithm is used to classify unknown traffic. Every ML algorithm has a different approach to sorting and prioritizing sets of features. Every ML algorithm has a different approach to classify and order the set of features, which runs to different dynamic behaviors during training and classification. This paper provides a foundation for IP traffic classification in IP networks, evaluating the state-of-the-art approaches to traffic classification, a critique emerging ML-based techniques for IP traffic classification. The input of ML is taken in the form of a dataset of instances or examples. An instance denotes an individual, unconstrained example of the dataset. Every instance is depicted by the values of its features also known as attributes or discriminators that calculate the different aspect of instance. ML has two categories, namely Unsupervised and Supervised Learning**.** The supervised traffic classification is classified into two different types: parametric classifiers, such as C4.5 decision tree [1], SVM [2], Naïve Bayes, Bayesian network [4], Naïve bayes Tree [3] and non-parametric classifiers such as Nearest Neighbor (k-NN) [5]. The Unsupervised clustering techniques include basic K-Means, DBSCAN, and EM. By using supervised classification algorithms or unsupervised cluster classification algorithms, the flow statistical feature based traffic classification can be done.

II. TYPES OF MACHINE LEARNING

A. Supervised Learning:

Supervised learning is the method in which the training data is labeled is prior. Supervised learning induces knowledge structures that support the task of classifying new instances into pre-defined classes. The supervised training data is examined by the supervised traffic classification methods and produce an indirect function which can predict the output class for any testing flow.

B. Unsupervised Learning (Clustering):

The use of clustering Algorithms for traffic classification is normally done in two phases. The first phase consists of training the model with a relatively small set of data (training data), and the second phase consists of using the trained model to classify unknown traffic. During the training phase, the training data is used to build clusters based on some criteria of similarity, which will ideally separate the data into similar clusters (groups). The second phase consists of assigning a class to the flows to be identified, depending upon the label of the cluster similar to each flow. The Unsupervised Machine learning approach is based on a

classifier built from clusters which are thus found and labeled in a training set of data. Once the classifier has been built, the classification process consist of the classifier calculating as to which cluster a connection is nearest to, and thereby using the label from the calculated cluster in order to recognize the connection. The Three clustering algorithms selected for this work are K-Means, DBSCAN, AutoClass and EM. The K- Means algorithm produces clusters that are spherical in shape whereas DBSCAN algorithm has the ability to produce clusters that are non-spherical.

III. TYPES OF SUPERVISED LEARNING

The supervised traffic classification in turn can be divided into two types: parametric classifiers and non-parametric classifiers. Parametric classifiers, such as C4.5 decision tree [1], Bayesian network [4], SVM [2], Naïve Bayes [3], Naïve bayes Tree [3]. Non-parametric classifiers, E.g Nearest Neighbor (k-NN) [5].

A. C4.5 Decision Tree:

C4.5 is an ML algorithm which is a decision tree based classification algorithm and is an extension of Iterative Dichotomiser 3 (ID3) algorithm. It is primarily used to generate Univariate decision tree [6].Since its decision trees can be used for classification C4.5 is also called Statistical Classifier. A decision tree can be used for implementing a divide-and-conquer strategy. In a decision tree, the local region is iteratively split and each region is identified by a sequence number. The presence of internal decision nodes and terminal leaves makes a decision tree hierarchical data structure. The nodes deployed in the tree symbolize the features and the branches corresponding to possible values connecting features. The end of a series of nodes and branches is a leaf of the representing class. To determine the class of an instance, one just has to trace the path of nodes and branches to the terminating leaf. At each node of the tree, C4.5 selects any one feature of the data that divides its sample set and forms subsets that are augmented in one class or the other. The main idea behind this technique is that the normalized information gain is chosen to make the decision. Then the algorithm is repeated in the smaller sub lists.

B. Bayesian Classification:

Bays' Net (Bayesian Network), [4] is a probabilistic graphical model which is used to represent knowledge about an uncertain domain using a combination of acyclic graph with nodes and links, and some conditional probability tables [7].Each node represents a random variable and the edges between the nodes represent probabilistic dependencies among the corresponding random variables. The conditional dependencies are acquired by using known statistical and computational methods. The nodes represent features or classes, while the links between nodes represent the relationship between them. Conditional probability tables determine the strength of the links. There is one probability table for each node (feature) that defines the probability distribution for the node given its parent nodes. The probability distribution, whose feature value depends on the values of the parents, is a conditional distribution, if a node has one or more parents. This is also known as Belief Network. Auld *et al.* [4] stretched the work with the application of Bayesian neural networks for accurate traffic classification.

C. Naive Bayes (NBK, NBD):

Naïve Bayes is a classification method based on the Bayesian theorem [8]. It calculates and analyses the relationship between each attribute and the class of the sample. From the computing results, it can derive a conditional probability of an attribute and the class. In the classification process, the classifier must estimate the probabilities of the unknown sample instance as a class, by combining the prior knowledge with the actual value of the unknown sample instance. Moreover the classifier must estimate the probabilities of the feature having a certain value. The continuous feature can have a large number of values, thus the probability cannot be estimated from the frequency distribution. Nowadays there are two solutions for this problem: by fitting the continuous probability distribution, or by using the discretization techniques. Because the latter method transforms the continuous features into the discrete ones and does not require the distribution model. Moore and Zuev [9] used a supervised Naive Bayes classifier and 248 flow features to differentiate between different application types. In addition to copious number of TCP header resultant features, there were packet length and inter arrival times, Correlation-based feature selection was used to identify 'stronger' features, and showed that only a small subset fewer than 20 features is required for accurate classification. In order to concentrate on the difficulties tolerated by payload-based traffic classification like the encrypted applications and user data privacy, another technique namely the supervised naive Bayes technique has been applied by the author of [4] that categorizes the network traffic based on flow statistical features. A Naïve Bayesian decision method produces an accuracy of 65.6% in [10].

D. SVM:

For a pattern recognition method based on the statistical learning theory (STL) one can use the Support Vector Machine (SVM). Even though the classification of the low dimensional space is transferred into the higher dimensional one, the classification of the higher dimensional space becomes relativistic. As such, the result brings about the computation overhead; the best solution is designed on selecting the appropriate kernel functions. Instinctively, this model is a classification algorithm for the classification of the sample space, and it requires the samples of the different categories which are divided widely as possible by the optimal hyper-plane in the sample space and make it have maximal distance with the other different classes, thus achieving the maximal generalization capability. Ruixi Yuvan, Zhu Li & Xiaolong [11] classified the network traffic into broad categories of application using SVM based machine learning method. Este et al. [21] presented a simple optimization algorithm for each set of SVM working

parameters and applied one class SVMs to traffic classification.

E. Nearest Neighbor:

In 1968, Cover and Hart proposed the Nearest Neighbor (NN) algorithm. It is a basic and simple ML classification algorithm in the pattern recognition field. Actually, the generalization of Nearest Neighbor algorithms, namely the k-NN algorithm, is often used, because the K-NN algorithm can enhance the robustness of the models. Especially, on the lower dimensional classification, K-NN is a higher good extensively used method. K–NN is very simple to implement. Compared with some other ML methods, it has low time complexity and space complexity, yet powerful capability of discrimination. However, KNN is a lazy learning method. Using SVM methods and optimal discriminator selection, an accuracy of 96.9% is obtained in [12]. In 2004 Roughan et al. [13] proposed to use the nearest neighbours (NN), linear discriminate analysis (LDA) and Quadratic Discriminant Analysis (QDA) ML algorithms to map different network applications to predetermined QoS traffic classes.

F. Naïve Bayes Tree:

The Naïve Bayes (NB) Tree is a hybrid of a decision tree classifier and a Naïve Bayes classifier combining the reliability and robustness of the Naive Bayes algorithm and the swiftness of decision tree algorithms [3]. The NB Tree model is best defined as a decision tree of nodes and branches with Bayes classifiers on the leaf nodes. As with other tree-based classifiers, NBTree spans out with branches and nodes. the algorithm evaluates the 'utility' of a split for each attribute when Given a node with a set of instances. If the highest utility among all attributes is notably best compared to the utility of the current node the instances will be separated based on that attribute. If there is no split that provides a notably improved utility a Naïve Bayes classifier will be created for the current node. The utility of a node is calculated by discretising the data and performing 5-fold cross validation to evaluate the accuracy using Naïve Bayes. The utility of a split is the weighted sum of the utility of the nodes, where the weights are proportional to the number of instances in each node [1].

IV. TYPES OF UNSUPERVISED LEARNING

The Unsupervised clustering techniques are basic K-Means, DBSCAN, and EM.

A. K-Means:

K-Means clustering is a technique of unsupervised learning which is used to partition n observations into K clusters, in which each observation belongs to the cluster with the nearest average value. There are varieties of partition-based clustering algorithms obtainable. There are varieties of partition-based clustering algorithms obtainable [7]. The K-Means algorithm is chosen as a result of, one among the fastest and simplest. The K-means partitions objects during a data set into a fixed range of K disjoint subsets. Bernaille et al. [17] applied the K-Means algorithm to traffic clustering and labeled the clusters to applications using a payload analysis tool.

B. DBSCAN:

The k-means, DBSCAN and AutoClass algorithms were estimated by Erman *et al.* in [Traffic class using clustering algo] for traffic clustering on two empirical data traces. The empirical examination showed that traffic clustering generates high-purity clusters when the number of clusters is assigned a value greater than the number of real applications. DBSCAN [10] being a density based algorithm, considers the clusters as being the dense areas of objects which are differentiated by less dense areas. Unlike K-Means algorithm, this algorithm does not only work with spherical shaped clusters but can also find clusters of arbitrary shapes [14]. The DBSCAN algorithm takes two input parameters: epsilon *(*eps*)* which is the distance between two objects that are eps neighbors and the number of minimum points (minPts*),* which is the minimum required point to form a core object. DBSCAN finds the optimum number of clusters based on the minPts and eps instead of taking the number of clusters to generate as the input. Moreover, unlike K-means and EM, an object that is not part of an existing cluster is well thought out as noise.

C. Expectation Maximization:

The Expectation Maximization algorithm discussed in [15] determines the maximum likelihood estimation of parameters. This algorithm is a simple, practical and an iterative algorithm which is not a direct maximization or the simulation of complex posterior distribution. To simplify the computation, some potential data which is based on observing data is included and executes a series of simple maximization or simulation. This algorithm was mainly framed in order to collect the multiple Internet traffic traces in [16]. It works with the probabilities of each instance which belongs to each cluster. The algorithm works in two phases, an expectation phase during which the parameters used by the algorithm that govern the distinct probability distribution of each cluster are estimated and a maximization phase when they are continually re-estimated.

D. Auto Class:

AutoClass is a perfect clustering method [25] which not only determines the number of the cluster but also the estimate of the cluster by itself. In order to prevent introducing many parameters in addition to the presence of over fitting, the algorithm uses the finite mixture model which helps in calculating the number of clusters. It uses each of the parameters with a prior distribution. When new parameters are introduced, their prior probabilities are involved in operation. It is useful in avoiding the likelihood value from increasing on the basis of parameter numbers, and consequent over fitting problem may vanish. Because the Expectation Maximization Algorithm cannot be ensured to converge on the global optimal point, iterative operations with different initial value are required. The AutoClass algorithm sets a predetermined time limit, in the actual implementation process. The AutoClass algorithm with the Expectation Maximization algorithm selects the cluster that

best suits from the training set and achieves the local optimum. McGregor et al. introduced another evaluation method of clustering results in [6] which uses intra-class homogeneity as the evaluation criteria. Intra-class homogeneity refers to the maximum amount of network application flow in one cluster. The intra-class homogeneity of the final clustering result is the average value of each of the clusters' intra-class homogeneity. The authors of suggest that the network application is separable and with the increasing feature number, intra-class homogeneity also increases.

Table I. Comparison between different ML Techniques and Features

TITLE	ML ALGORITHM	FEATURES	DATA TRACES	TRAFFIC CONSIDERED
Roughan et al.[13][15]	Nearest Neighbour, Linear Discriminate Analysis and Quadratic Discriminant Analysis	• Packet Level • Flow Level • Connection Level • Intra-flow/Connection features • Muli-flow features Calculated on full flows	Waikato trace and section logs from a commercial streaming services	Telnet, FTP (data), Kazaa, Real Media Streaming, DNS, HTTPS
Moore and Zuev [9] [15]	Baysian Techniques	Total of 248 features, among them are • Flow duration • TCP port • Packet inter-arrival time statistics • Payload size statistics • Effective bandwidth based upon entropy • Fourier transform of packet inter-arrival time • Calculated on full flows	Proprietary Hand Classified Traces	A large range of Database, P2P, Buck, Mail, Services.
Park et al. [20][15]	Naive Bayes with Kernel Estimation, Decision Tree J48 and Reduced Error , Prunning Tree	• Flow duration • Initial Advertised • Window bytes • Number of actual data packets • Number of packets with the option of PUSH • Packet lengths • Advertised window bytes • Packet inter-arrival time • Size of total burst packets	NLANR, USC/ISI,CAIDA	WWW, Telnet, Chat (Messenger), FTP, P2P (Kazaa, Gnutella), Multimedia, SMTP, POP, IMAP, NDS, Oracle, X11
Auld et al.[4]	Bayesian Neural Network	246 features in total, including: • Flow metrics (duration, packet-count, to-tal bytes) • Packet inter-arrival time statistics • Size of TCP/IP control fields • Total packets in each direction and total for bi-directional flow • Payload size • Effective bandwidth based upon entropy • Top-ten Fourier transform components of packet inter-arrival times for each direction • Numerous TCP-specific values derived from tcptrace (e.g. total payload bytes transmitted, total number of PUSHED packets, total number of ACK packets carrying SACK information etc.)	Proprietary hand classified traces	A large range of Database, P2P, Buck, Mail, Services, Multimedia, Web ... traffic
Williams et al. [1]	C4.5 DecisionTree , Naive Bayes with Discretisation, Naive Bayes with Kernel Estimation, Bayesian Network and Naive Bayes Tree	• Protocol • Flow duration • Flow volume in bytes and packets • Packet length (minimum, mean, maxi-mum and standard deviation) • Inter-arrival time between packets (mini-mum, mean, maximum and standard de-viation)	NLANR	FTP(data), Telnet, SMTP, DNS, HTTP
Erman et al.[21]	Naive Bayes and AutoClass	• Total number of packets • Mean packet length (in each direction and combined) • Flow duration • Mean data packet length • Mean packet inter-arrival time	NLANR	HTTP, SMTP, DNS, SOCKS, FTP(control), FTP (data), POP3, Limewire
Jun Zhang et.al [22]	Nearest Neighbor (NN) AVG – NN MVT – NN MIN – NN	• Packets - Number of packets transferred in unidirection • Bytes - Volume of bytes transferred in unidirection • Packet Size - Min., Max., Mean and Std Dev. Of packet size in unidirection • Inter-Packet Time - Min., Max., Mean and	sigcomm lbnl keio wide isp	P2P, DNS, FTP, WWW, CHAT, and MAIL, BT and HTTP

		Std Dev. of Inter Packet Time in unidirection		
T.T Nguyen et.al [23]	Naïve Bayes	• Inter-packet arrival interval (min,max,mean,standard deviation) • Inter-packet length vaiation (min,max,standard deviation) • IP packets length (min,max,mean and standard deviation)	Wolfenstein Enemy Territory	HTTPS,HTTP.DNS,NTP,SMTP,IMAP,POP3, Telnet, SSH,Halflife, Kazaa, Bittorrent, Ginutella, eDonkey
Erman et al.[7][15]	K-Means	• Total number of packets • Mean packet length , mean payload length excluding headers • Number of bytes transferred • Flow duration • Mean inter-arrival time	Self – collected 8 1-hour compus traces between April 6-9, 2006	Web, P2P, FTP, Others
Jeffery Erman et .al [24]	K – Means	• Total number of packets • Average packet size • total bytes • total header bytes • number of caller to callee packets • number of callee to caller bytes • total caller to callee payload bytes • total caller to callee header bytes • number of callee to caller packets • total callee to caller payload bytes • total callee to caller header bytes.	Campus traces, Residential traces, WLAN traces.	BB, BitTorrent, DirectConnect, eDonkey, FTP, Gnutella-based P2P programs,GoToMyPC, HTTP, ICQ, IDENT, IMAP, IMAP SSL, JetDirect, KaZaA, MySQL, MSSQL, MSN Messenger, MSN Web Cam, NNTP, POP3, POP3 SSL, RTSP, Samba, SIP, SMTP, SOAP, SpamAssassin, SSH, SSL, VNC, Z3950 Client.
McGregor et al. [15][16]	Expectation Maximization	• Packet length statistics (min, max, qualitiles, ...) • Inter-arrival statistics • Byte counts • Connection duration • Number of transitions between transaction mode and bulk transfer mode • Idle time Calculated on full flows	NLANR and Waikato trace	A mixture of HTTP, SMTP, FTP (control), NTP, IMAP, DNS
Zander et al. [25]	AutoClass	• Packet length statistics (mean and variance in forward and backward directions) • Inter-arrival time statistics (mean and variance in forward and backward directions) • Flow size (bytes) • Flow duration Calculated on full-flow	Auckland – VI, NZIX – II and Leipzig – II from NLANR	Half-Life, Napster, AOL, HTTP, DNS, SMTP, Telnet, FTP (data)
Erman et al.[14]	K-Means, DBSCAN and AutoClass	• Total number of packets • Mean packet length • Mean payload length excluding headers • Number of bytes transfered (in each direction and combined) • Mean packet inter-arrival time • Total number of packets • Mean packet length (in each direction and combined) • Flow duration • Mean data packet length • Mean packet inter-arrival time • Message size (the length of the message encapsulated into the transport layer protocol segment) • Average inter packet gap	NLANR and a self-collected 1-hour trace from the University of Calgary	HTTP, P2P, SMTP, IMAP, POP3, MSSQL, Other
Hyunchul Kim et .al [3]	SVM	• Protocol • source port • Destination port • number of packets • Transferred bytes • the number of packets without Layer 4 payload, • start time , end time, duration • average packet throughput • byte throughput • max/min/average /standard deviation of packet size	PAIX backbone trace, KAIST, WIDE,Keio.	WWW, MAIL,CHAT, DNS,FTP,GAME,P2P

		• inter-arrival time • number of TCP packets with FIN,SYN,RSTS,PUSH,ACK,URG(Urgent), CWE(Congestion Window Reduced), and ECE (Explicit Congestion Notification Echo) flags set (all zero for UDP packet) • Size of the first ten packets		

V. CONCLUSION

This paper surveys significant works in the field of Machine learning based traffic classification, motivated by a desire to move away from port-based or payload-based traffic classification. It is obvious that ML can be applied well in the task of traffic classification. The use of a number of different ML algorithms for offline analysis, such as AutoClass, Expectation Maximization, Decision Tree, Naïve Bayes etc. has demonstrated high accuracy for a various range of Internet applications traffic. Early, ML techniques relied on static offline analysis of previously captured traffic. More recent work begins to address the requirements for practical ML-based real-time IP traffic classification in operational networks. It shows the various Data Traces and Features. The Table 1 shows how ML techniques out performs the previously used methods. In this survey paper, we have outlined a number of ML techniques and algorithms for traffic classification.

VI. REFERENCES

[1] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification," *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 5–16, October 2006.

[2] J. C H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: myths, caveats, and the best practices," in Proceedings of the ACM CoNEXT Con*ference*, New York, NY, USA, 2008, pp. 1–12.

[3] 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

[4] T. AULD, A.W.Moore, and S.F.Gull, "Bayesian neural networks for internet traffic classification, " IEEE trans.Neural netw., vol. 18, n0.1, pp.223-239, January 2007.

[5] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-ofservice mapping for QoS: a statistical signature-based approach to IP traffic classification," in Proceedings of the 4th ACM SIGCOMM *conference on Internet measurement*, New York, NY, USA, 2004, pp. 135–148

[6] R. Kohavi and J. R. Quinlan, Will Klosgen and Jan M. Zytkow, editors, "Decision-tree discovery", in Handbook of Data Mining and Knowledge Discovery, pp. 267-276, Oxford University Press, 2002

[7] Nguyen and G. Armitage, "Training on multiple sub-flows to optimise the use of machine learning classifiers in real-world ip networks," in Local Computer Networks, Annual IEEE Conference on, Los Alamitos, CA, USA, 2006, pp. 369–376

[8] G. H. John, P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers", in Proceedings of 11th Conference on Uncertainty in Artificial Intelligence, pp. 338-345, Morgan Kaufman, San Mateo, 1995.

[9] W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, pp. 50–60, June 2005.

[10] Andrew Moore, Denis Zuev and Michel Crogan , "Discriminators for use in flow-based classification", ISSN 1470-5559. Eval Rev., vol. 33

[11] Ruixi Yuvan, Zhu Li& Xiaolong, "An efficient SVM-based Method for Multi-Class Network Traffic Classification" in vol 1-8., 17-19 Nov. 2011

[12] Zhu Li , Ruixi Yuan, "Accurate Classification of the Internet Traffic Based on SVM method" ICC07- IEEE Conference on 2007.

[13] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of- service mapping for QoS: A statistical signature-based approach to IP traffic classification," in Proceedings of ACM/SIGCOMM Internet Measurement Conference (IMC) 2004, Taormina, Sicily, Italy, October 2004.

[14] J. Erman, M. Arlitt, and A. Mahanti. Traffic Classification using Clustering algorithms. In Proceedings of the SIGCOMM workshop on Mining network data, pages 281-286. ACM,2006.

[15] A T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, Fourth Quarter 2008.

[16] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, "Flow clustering using machine learning techniques," in *Proceedings of Passive and Active Measurement Workshop*, Antibes Juan-les-Pins, France, April 2004, pp. 205–214.

[17] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 23–26, April 2006.

[18] P.Cheeseman. ,J.Stutz, Bayesian Classification (Autoclass): Theory and results, Advances in knowledge discovery and data mining, America Association for Artificial Intelligence, 1996, 153-180

[19] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in IEEE 30th Conference on Local Computer Networks (LCN 2005), Sydney, Australia, November 2005.

[20] J. Park, H.-R. Tyan, and K. C.-C.J., "GA-Based Internet Traffic Classification Technique for QoS Provisioning," in Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Pasadena, California, December 2006.

[21] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "ACAS: automated construction of application signatures," in *Proceedings of the ACM SIGCOMM workshop on Mining network data*. New York, NY, USA: ACM, 2005, pp. 197–202.

[22] Jun Zhang, Yang Xiang, "Network Traffic Classification Using Correlation Information" in IEEE Transactions on Parallel and Distributed systems vol 33,2012

[23] T. Nguyen and G. Armitage, "Training on multiple sub-flows to optimize the use of Machine Learning classifiers in real-world IP networks," in Proc. IEEE 31st Conference on Local Computer Networks, Tampa, Florida, USA, November 2006.

[24] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Offline/ realtime traffic classification using semi-supervised learning," *Performance Evaluation*, vol. 64, no. 9-12, pp. 1194–1213, October 2007.

[25] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in IEEE 30th Conference on Local Computer Networks (LCN 2005), Sydney, Australia, November 2005.

