International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Performance Evaluation of Clustering Algorithms for IP Traffic Recognition

Rupesh Jaiswal[1], Dr. Shashikant Lokhande[2], Aashiq Ahmed[3], Prateek Mahajan[4]

[1]Assistant Professor, Department of E&TC, Pune Institute of Computer Technology, Pune, India

[2] Professor, Department of E&TC, Sinhgad College of Engineering, Pune, India

[3]Student, Department of E&TC, Pune Institute of Computer Technology, Pune, India

[4]Student, Department of E&TC, Pune Institute of Computer Technology, Pune, India

Abstract: ***Literature reports the huge work of IP traffic recognition using machine learning (ML) Algorithms. Data is divided into groups of similar objects or Clustering process groups the data instances that have similar characteristics without any previous supervision or guidance. Clustering analysis can be used for identification of IP traffic protocols effectively by measuring the external statistical attributes like packet length and inter arrival time. Our research work shows the analysis using K-means and DBSCAN clustering algorithm. Our approach is evaluated using accuracy and execution time for clustering model.***

Keywords: Traffic, recognition, clustering, features.

1. Introduction and Related work

Analysis of traffic flows and associating them to different categories of Internet applications is done as part of IP traffic identification. Traditional Port based techniques and payload signature based techniques are absolute nowadays [1-5]. Newer approaches identify the internet traffic by understanding statistical behavior in externally observable features of the IP traffic. This method targets to cluster IP traffic flows into groups of similar characteristics. Machine learning is mostly used nowadays because of handling large traffic datasets and features needed to be handled are more. This is very well surveyed by Thuy T. T. Nguyen [6] and Arthur Callado [7]. Bernaille [8-10] perform traffic recognition using the first few packets of established flow based connection. Rest of our paper is arranged as follows. Section II describes the ML clustering algorithms. Section III shows the details of datasets developed and methodology used. Section IV-V explains implementation and result analysis. Section VI describes conclusion and future directions.

2. Clustering Techniques

Clustering techniques can be majorly classified as partitioning based, hierarchical based, density based, grid based and model based etc. Portioning based method can further be divided into user defined (K-Means) or data driven. Density based clustering can be DBSCAN, OPTICS or DENCLUE. Grid based clustering can be further divided as STING based or WAVELET-T type. Model based clustering techniques are Expectation Maximization (EM), Conceptual or Neural Network based. Clustering techniques used for IP traffic recognition are categorized and shown in fig.1.

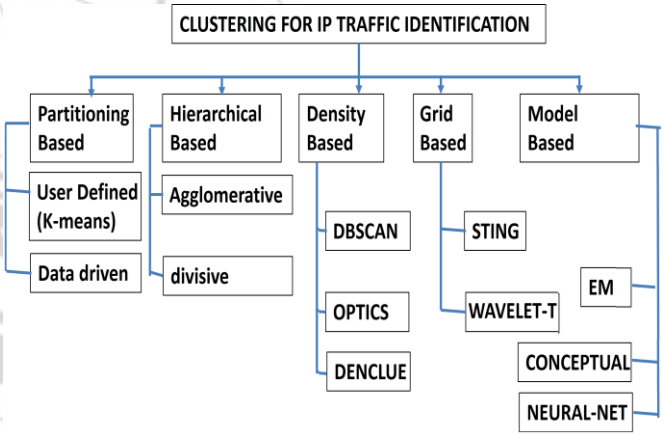

Figure 1: Different Clustering techniques

2.1 DBSCAN Technique

DBSCAN (Density-Based-Spatial-Clustering of Applications with Noise) needs two parameters: ε (eps) and the minimum number of points required to form a cluster (minPts). It finds a number of cluster groups starting from the estimated density distribution of corresponding application nodes in several applications. DBSCAN Clustering technique used for IP traffic recognition is explained with flowchart and shown in fig.2.

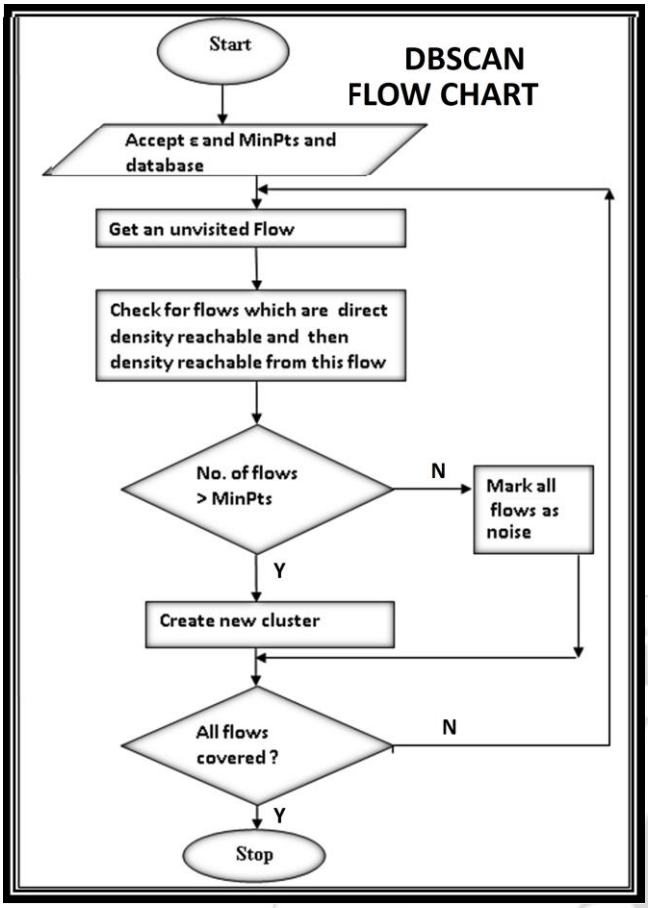

Figure 2: DBSCAN Clustering technique

2.2 K-MEANS Technique

K-means Partitions the given objects into *k* nonempty subsets. K-MEANS Clustering technique used for IP traffic recognition is explained with flowchart and shown in fig.3.

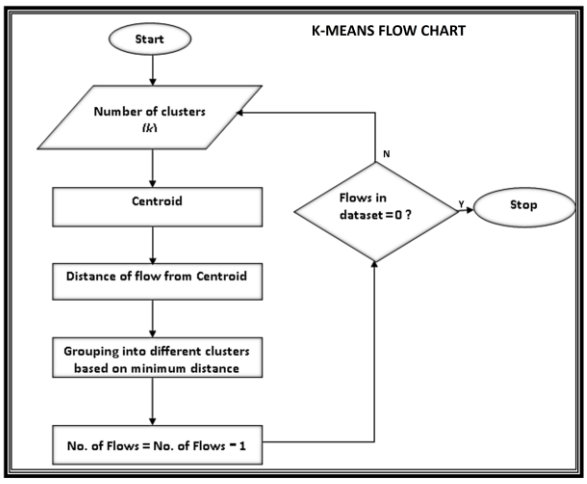

Figure 3: K-MEANS Clustering flowchart

3. Datasets development and Research Methodology used

We used following procedure to develop proprietary datasets for IP traffic recognition using clustering techniques.

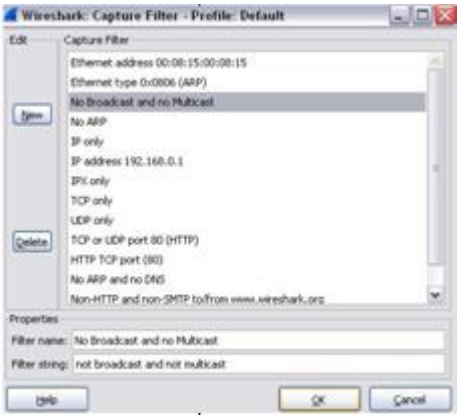

Figure 4: Capture Filter

We used Wireshark to store captured online packets in pcap format. HTTP, P2P and streaming traffic was captured using a 'No Broadcast and no Multicast' filter to filter out broadcast and multicast packets.

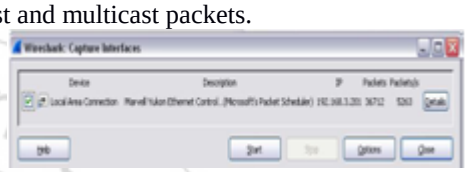

Figure 5: Capture Interface

After the filter, we select appropriate capture interface. Here- LAN over eth0.

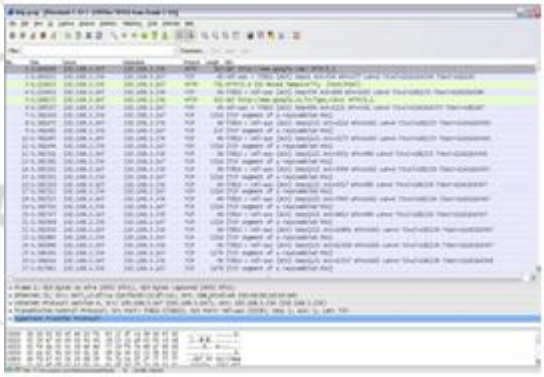

Figure 6: Traffic Captured

The figure shows the LAN traffic being captured along with the source address, destination address, protocol and the length of the packet.

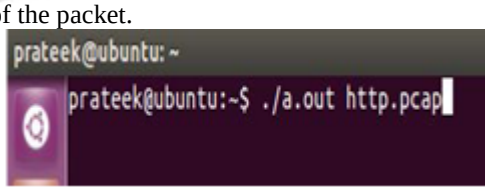

Figure 7: Generation of executable file

C program is used to convert the packets in the pcap file to flows. An executable file "a.out" is generated which will accept stored .pcap file as input.

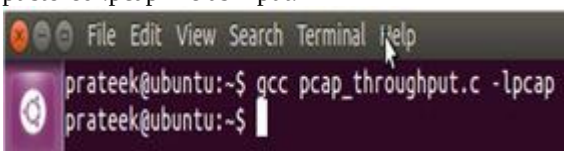

Figure 8: Operating executable on .pcap file

a. out is executed to generate flows of .pcap headers from http.pcap.

A flow is defined as a sequence of packets transmitted between two computers, which share same five-tuples: IP address (source & destination), Port address (source & destination) and transport layer protocol with a timeout of 64 seconds [8-10]. The statistical properties that can be used for distinguishing the exact source for the application are defined below.

- Mean Inter-Arrival Time

It is the time difference between the times of arrival of the two consecutive IP packets. Mean time of IAT of each two consecutive packets is considered as the feature for the classification of the traffic.

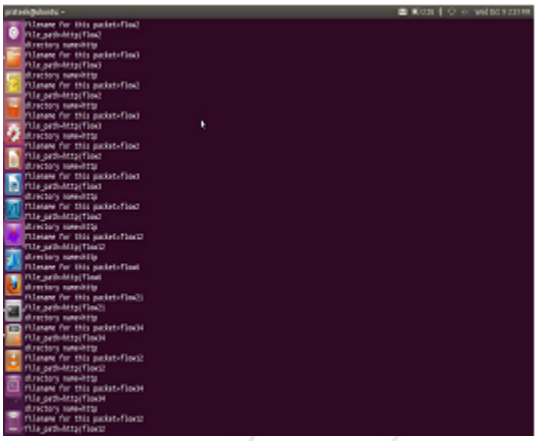

Figure 9: Flows of headers generated from .pcap file

- Packets per Second
  Number of packets being encountered by the host per second for a particular protocol is most of the times unique.
- Bytes per Second
  Number of bytes being received per second when each packet receives can be considered as a feature for the traffic classification.
- Total no of Bytes per Flow
  A flow can be considered as sequence of packets from source computer to the destination, which may be another host, multicast group or a broadcast domain. The number of bytes per flow i.e. per session time transaction between two machines.
- Total Idle Time
  It is the time for which the host can't accept the next packet of the same protocol i.e. it is in the idle state.
- Flow Duration
  It is the time for which the flow is alive in the communication using particular protocol. It is unique for each protocol enabling the ANN to distinguish the protocols.
- Mean IP Packet and Payload Length
  The IP packet header and payload length for each of the protocol is different. So it can be considered as the feature to train the ANN.
- Standard Deviation of IAT, IP packet and Payload Length
  Deviation of the group of the respective values from the mean value is termed as the standard deviation. The standard deviation of the IP packet length, IP payload Length and IAT is considered as the feature for classification of the protocols.
- Flags in the IP Packet Header
  URG, ACK, PSH, RST, SYN and FIN are the flags for each of the IP header. These flags are set differently for each protocol. Thus by acquiring the status of these flags, we can sniff which is the traffic being received by this particular host.

These features are common in almost all the standard research papers except coefficient of variation. We have tested our classifier models with different training data sets and also observed the effect of reduction in Feature set. The results obtained for accuracy using Co-efficient of variations as feature set is far better than other statistical features. COV = (Standard deviation of statistic / Mean value of statistic). Further in this research, we present classifier's performance on the basis of its accuracy, computational performance and other parameters. So, we have chosen the above mentioned as the main features for the classification process and as input to our ML algorithm.

The output of the C program i.e. flows files, is in a folder labeled same as the pcap filename. It contains the header details of all the received data packets in a particular flow. An octave code is used to calculate the features from the headers information extracted by the C program in flows. The program calculates the feature values which will be used for testing the clustering algorithms. We initially worked on 10 features namely-packets per sec, bytes per flow, bytes per sec, flow duration and Mean and Standard deviation of IP packet length, IP payload length and Inter arrival Time - as per previous work done in the research papers that we have referred.

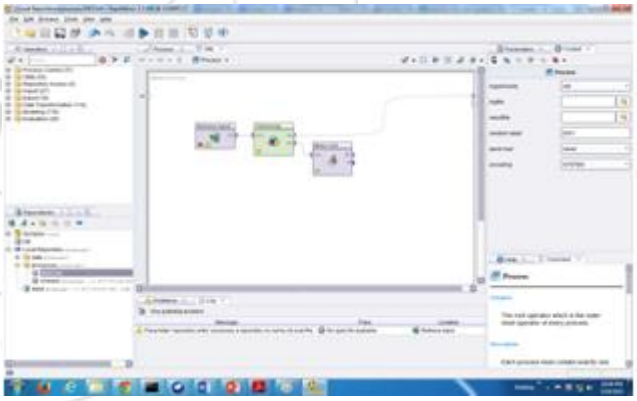

Figure 10: Process in RapidMiner for DBSCAN

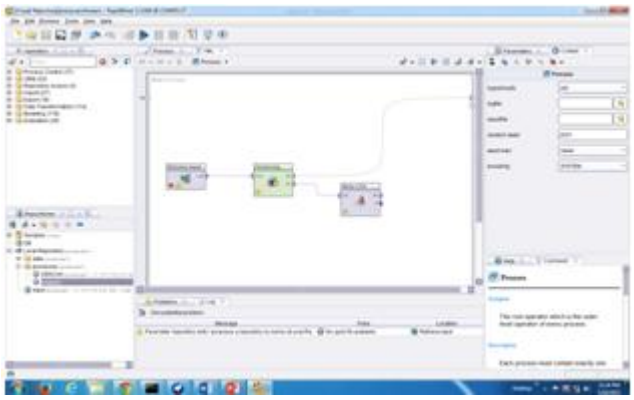

Figure 11: Process in RapidMiner for K-Means

4. Semi-Supervised Clustering and Its Implementation

The basic K-Means and DBSCAN Algorithms are not designed for a database being updated regularly. The clusters cannot be updated for each new flow introduced. The algorithms have to be re-run to generate new set of groups or clusters which include newer flows. Another disadvantage of Clustering Algorithms is their inability to classify results. Even though Clustering Algorithms separate the flows into adifferent clusters or groups, they don't identify whether they belong to HTTP or P2P traffic. To overcome these drawbacks and to enable faster online traffic identification, we implemented a Semi-Supervised Clustering based approach for Traffic Identification based on Classification Algorithms which require a Training Phase before Implementation. A known database of flows generated by HTTP and P2P packets is given to the Clustering algorithms. We then analyze the generated cluster groups for the accuracy with which they have been clustered and how this accuracy changes with change in value of DBSCAN and K-Means parameters. We evaluated accuracy of the algorithms as per the formula given.

Accuracy = (True positives / Total number of flows), where, True positives = Total number of correctly clustered flows. For both these algorithms the accuracy peaked at around 90 percent. The clusters generated by K-Means were 93 percent accurate for parameter values of k=2 and max iterations= 10. At this value we found the K-Means cluster centroids of the HTTP and P2P flows. Similarly the clusters generated by DBSCAN were 90 percent accurate for parameter values of epsilon= 0.025 and minPts= 3. The range of the cluster values of the DBSCAN clusters for HTTP and P2P flows at this value was observed. The designed Semi-Supervised Algorithm uses this range and the centroid values of K-Means to classify the flows. This process allows it to be faster than running the Clustering algorithms. Additionally, it also classifies the flow as HTTP or P2P whereas the Clustering algorithms only separate the flows into groups or clusters. However, its results having been derived from the Clustering algorithms can be said to be only around 90 percent accurate as the algorithms were. This can however be improved if a larger dataset is used and more work is done on choosing better features and parameters to improve the accuracy of the training phase.

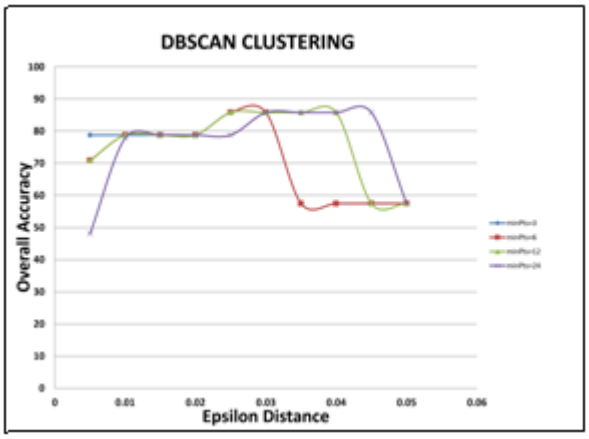

Figure 12: Parameterization and Accuracy of DBSCAN

Accuracy of the DBSCAN Algorithm increases as the value of epsilon is increased and minPts is kept fixed. After reaching its peak at 0.025, accuracy starts decreasing again. For the semi supervised algorithm we use the results for the peak accuracy. (Percentage of those instances that truly have class X, among all those classified as class X. as show in Fig, 13.) (Percentage of members of class X correctly classified as belonging to class X. as show in Fig, 14.)

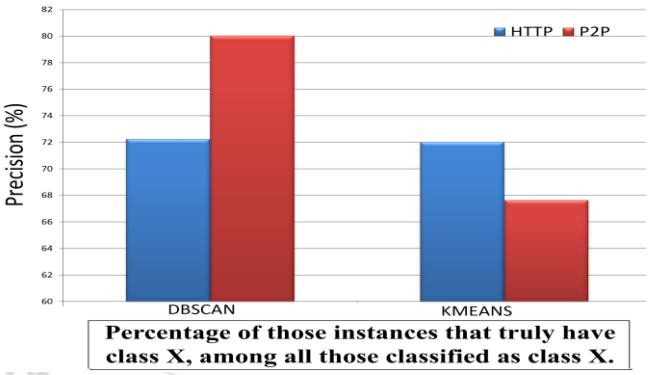

Figure 13: Precision Plot

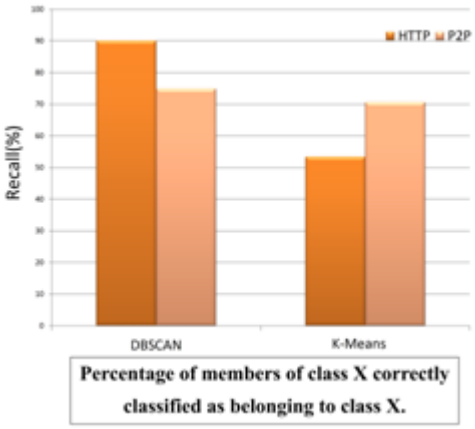

Figure 14: Recall Plot

5. Result Analysis of DBSCAN and KMEANS

Value of 'k' denotes the number of clusters the algorithm should divide the given dataset into. The K-Means algorithm separates the given dataset into 'k' clusters, even if they belong to a single cluster only. This property of the Algorithm can be useful to divide a cluster into further sub clusters. But this reduces the accuracy of the algorithm when the dataset is smaller in size and has flows of only one type.

5.1 Timing Analysis of Clustering Algorithms

Timing Table for 6456 flows are shown.

Table 1: Timing Analysis

ALGORITHM	EXECUTION TIMES (sec)
DBSCAN (Epsilon= 0.025, minPts=3)	42.22
K-Means (K=2, max Iterations =10)	9.55

The K-Means and DBSCAN algorithms take almost same execution time when the flows are less than 1000 in number. The DBSCAN Algorithm uses a function NgbrPts ( ) recursively calling itself and hence is slower than the K-

Means algorithm especially when the size of the dataset increases. The Semi-Supervised Clustering based approach takes the least time to execute, as only comparison takes place with no actual clustering.

5.2 Overall Performance Analysis of clustering algorithms

Table 2: Distribution of 6456 flows into clusters

Algo	*Traffic*	*Actual Flows*	*True Positives*	*False Negatives*	*Clustered Flows*
DB-SCAN	HTTP	3704	3338	366	4609
	P2P	2752	1471	1279	1838
K-Means	HTTP	3704	2775	929	3853
	P2P	2752	1944	808	2874

Table 3: Comparison of algorithms based on performance parameters

Parameters/Algos.	**DBSCAN**	**K-Means**
HTTP		
Precision (%)	72.4	72.02
Recall (%)	90.12	74.92
P2P		
Precision (%)	80.04	67.64
Recall (%)	53.45	70.64

Overall we observed that when program execution; DBSCAN provides more accurate results than K-Means when the numbers of flows are smaller. Accuracy of K-Means increases with increase in number of flows, while that of DBSCAN decreases. Also, while both DBSCAN and K-Means take the same amount of time for a smaller set of flows, DBSCAN takes a significantly longer time to execute for a larger set of flows. But the DBSCAN Algorithm has an ability to separate flows which are not close to the main clusters into noise, which may help us separate anomalies or random values.

The Semi-Supervised Clustering based approach takes the least time to execute, as only comparison takes place with no actual clustering, and gives an accuracy of about 90 percent. This accuracy can be improved if the size of the database is increased to find more accurate centroids and ranges. We also observed that for P2P traffic i.e. for Torrent traffic, number of flows generated are far greater than that for the number of flows generated by the same amount of HTTP traffic.

6. Conclusion and Future Work

Where port and payload based Internet Traffic Identification give an accuracy of about 50 to 70 percent, our work on Clustering algorithms have shown them to be accurate upto 90 percent which can be further improved by the use of better features and a larger dataset. We implemented both the DBSCAN and the K-Means algorithm to cluster flows generated by the online packet capture code.

While the K-Means algorithm gives a peak accuracy of about 93 percent and the DBSCAN a peak accuracy of about 90 percent for the captured pcap flows, when run online the DBSCAN algorithm gives slightly better clusters. The accuracy of K-Means algorithm decreases if there are flows of only one type as it divides it into sub clusters. However for the Online Internet Traffic Identification, we observed that the implementation of a Semi-supervised based approach is more efficient and gives a consistent accuracy of around 90 percent. It provides faster results, derived from the algorithms themselves, but its accuracy is decided by the size of the dataset used to derive the results.

As the size of data increases the better results can be obtained and this can be further used to provide a better semi-supervised solution. Also these clustering algorithms can be tried for other classes of application traffics like attack, https, gaming, streaming, VoIP, mail, dns and ftp. Also, other datasets like LBNL, UNIBS, CAIDA, AUCKLAND, WAIKATO and MAWI standard datasets can be tested in future.

References

[1] Moore, Andrew W. and Papagiannaki, Konstantina, Toward the Accurate Identification of Network Applications, Passive and Active Measurement Workshop (PAM 2005), March 2005.
[2] Karagiannis, Thomas; Broido, Andre; Faloutsos, Michalis and Claffy, K.C., Transport Layer Identification of P2P Traffic, Internet Measurement Conference (IMC '2004), October 2004.
[3] Madhukar, A. and Williamson, C., A Longitudinal Study of P2P Traffic Classification, 14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, September, 2006.
[4] Karagiannis, Thomas; Broido, Andre; Brownlee, Nevil; Claffy, K.C. and Faloutsos, Michalis, Is P2P dying or just hiding?, IEEE Global Telecommunications Conference, November 2004.
[5] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in network identification of P2P traffic using application signatures," in WWW2004, New York, NY, USA, May 2004.
[6] Thuy T. T. Nguyen, Grenville Armitage, Philip Branch, and Sebastian Zander, "Timely and Continuous Machine-Learning-Based Classification for Interactive IP Traffic", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 20, NO. 6, DECEMBER 2012.
[7] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A Survey on Internet Traffic Identification," *IEEE Commun. Surveys & Tutorials*, vol. 11, no. 3, pp. 37 –52, quarter 2009.
[8] Bernaille, Laurent and Teixeira, Renata, *Early Recognition of Encrypted Applications*, in Proc. 8th International Conference, Passive and Active Measurement Conference, Louvain-la-Neuve, Belgium, April 2007.
[9] Bernaille, Laurent; Teixeira, Renata and Salamantian, Kavé, *Early Application Identification*, Second Conference on Future Networking Technologies, December 2006.
[10] Bernaille, Laurent; Teixeira, Renata; Akodkenou, Ismael; Soule, Augustin and Salamantian, Kave, *Traffic Classification On The Fly*, ACM SIGCOMM Computer Communication Review, Volume 36, Number 2, April 2006, pp. 23-26.