International Journal of Database Theory and Application
Vol.9, No.12 (2016), pp.45-54
http://dx.doi.org/10.14257/ijdta.2016.9.12.05

Internet Traffic Classification Using Machine Learning

M. P. Singh, Gargi Srivastava and Prabhat Kumar

NIT Patna, Bihar, India
mps@nitp.ac.in, gargisri68@gmail.com, prabhat@nitp.ac.in

Abstract

Internet traffic classification is one of the popular research interest area because of its benefits for many applications like intrusion detection system, congestion avoidance, traffic prediction etc. Internet traffic is classified on the basis of statistical features because port and payload based techniques have their limitations. For statistics based techniques machine learning is used. The statistical feature set is large. Hence, it is a challenge to reduce the large feature set to an optimal feature set. This will reduce the time complexity of the machine learning algorithm. This paper tries to obtain an optimal feature set by using a hybrid approach -An unsupervised clustering algorithm (K-Means) with a supervised feature selection algorithm (Best Feature Selection).

***Keywords**: Internet traffic, Classification, Management, Machine Learning (ML)*

1. Introduction

The process of interception and examination of messages so as to deduce relevant information from communication patterns observed is called internet traffic classification. The amount of information inferred from the internet traffic is directly dependent on the number of messages observed (plain or encrypted), intercepted and stored. Traffic monitoring and analysis is the basic requirement to adequately troubleshoot and solve problems that bring network services to a halt. There are many issues related to the internet traffic classification [13]. Some of them are discussed as follows. In general, ML classifier requires very large dataset having large number of features/ attributes which introduces organization problem. Though it increases accuracy of classifier, but classifier speed get reduced, if irrelevant features got selected for classifier. Multi-class imbalance problem is another main issue. In this case, the classifier is so overwhelmed by the majority traffic classes that it ignores the minority traffic classes. So techniques must be evolved to help ML algorithms to prevent multi-class imbalance. Another important issue is that a single classifier may not be suitable for all classification purposes. So the performance of ML algorithm is application dependent. The continued proliferation of different Internet applications behaviors and growing incentives to mask some applications to avoid filtering and/or blocking has become a very big obstacle in the path of traffic analysis. Major techniques did not classifies the network traffic of larger portion. UDP traffic is often ignored and bi-direction traffic flow is not considered in many cases. Traffic classification is an automated process to categorize traffic of computer network based on various parameters of networks' traffic. It is also the basis of automated intrusion detection systems; used to catch patterns that suggest a denial of service attack; automatically re-allocate network resources for customers according to priority; determines which customer's usage conflicts with the operator's terms of service. After the classification of network's traffic, pre-defined policy may be applied on classified traffic which will guarantee a certain level of quality. The same process is also applicable in case of access point to isolate traffic into individual flows and then apply the policy. Packets belong to a flow if they have identical 5-tuples of protocol type, source address, source port, destination port and destination address. Classification is achieved by various means

[29] like Port Numbers, Deep Packet Inspection, and Statistical Classification. Port number and deep packet inspection have a lot of disadvantages and limitations. For example: Port Number is useful only for those which uses fixed port numbers. Deep Packet Inspection is slow and requires significant amount of processing power. Machine learning (ML) techniques are one of the alternative to classify applications of different networks based on statistics of per flow, inter-arrival time and data wired. ML algorithms, for internet traffic classification, are broadly divided into two types: (i) supervised: the class of traffic flow must be known beforehand and (ii) unsupervised: accumulates traffic flows into different clusters according to similarities in the feature values. The organization of this paper is as follows: Section 2 describes related work done in the field of traffic classification. Section 3 explains the problem. Section 4 explains the proposed solution for the problem. Section 5 shows the result and finally, Section 6 concludes the paper.

2. Related Work

Various work [1-8 16-19] related to traffic classification has been carried out. This section explains the previous work of feature selection algorithms which are used before classification. Feature selection algorithm improves the performance of the classifier.

2.1. Multi-class Imbalance

The paper [14] states that computational cost of classifier is very high if categorization uses 248 statistical features of network flow. There are some methods namely Filter method (RELIEF and FOCUS filter), Wrapper method, and embedded selection method (CART (Classification and Regression Tree), ID3 (Iterative Dichotomizer 3), C4.5), to reduce total number of statistical features. Authors [9] claim for significant improvement in computational performance by using filter feature selection methods that maintain accuracy in classification significantly. Authors [20] presents a logistic regression model which provides a solution for class imbalance problem. The model maps multi-class classification into two-class classification. Internet traffic traces fall in either multi-minority (classes possess few of flows) or multi-majority (classes possess a lot of flows). Authors [10] present three methods, namely (i) Random under-sampling (ii) Random over-sampling, and (iii) Cost-sensitive learning. Cost-sensitive learning is best suited for large dataset. Authors [11] use information theory to check biasness of one feature in intra-class. Authors propose Best Feature Subset (BFS), feature selection method to ease the multi-class imbalance problem. Authors also compare BFS with fast correlation-based filter (FCBF) and Full-set using Naïve Bayes. This paper tries to modify this BFS algorithm so that it can be used with unsupervised algorithms. The unsupervised algorithm selected is K-Means. K-Means is a partition based clustering algorithm. It is simple and has better performance than DBSCAN and EM [21].

There are several issues in port-based and packet payload-based classification techniques such as port hopping, encryption, privacy issues, port masquerading *etc.* Machine learning is a promising alternative as it uses statistical features. The goal is to reduce number of features and improve overall accuracy in classification. ML improves classification accuracy in case of majority classes. But drawback of ML is that it reduces the classification accuracy of minority classes significantly. The problem is that we cannot ignore minority classes such as ATTACK.

Many work has been done based on Decision Tree for Feature Selection [9,12,22-28]. There are different methods/tools for bandwidth management like Congestion avoidance, Traffic shaping, Traffic classification, Scheduling algorithms, Bandwidth reservation protocols/algorithms, [15].

There are different performance metrics like g-mean, accuracy, mauc, and recall used in classification. BFS reduces g-mean only by 8-9% whereas FCBF reduces g-mean by

50%. Mauc and accuracy are also higher using BFS. Classification accuracy of 90% can be achieved. Biased relationship between feature and class is many to many. And hence, BFS chooses balanced with great discriminating ability optimal feature subset.

3. Problem Description

Based on the related work, this paper describes the problem as "Selecting an optimized feature set for internet traffic classification based on machine learning".

Different feature selection methods have been described in Section 2. Filter methods are very slow and wrapper methods are algorithm dependent. So, an algorithm is proposed which is fast and can work with any supervised or unsupervised algorithms**.**

4. Proposed Solution

This section proposes the solution for the proposed problem using ML. This paper presents the solution with the help of Flow Chart and then presents functional model. After that, it presents the steps of working of the proposed solution. The flow of the complete procedure is depicted by flowchart as shown in FIGURE 1, 2, 3. FIGURE 1: First step of algorithm is to generate clusters $C = C_1, C_2, \ldots\ldots, C_m$. A subset of the dataset with 248 features is given as an input to K-Means algorithm. FIGURE 2: Next step is to find set of optimal features. For this, 248 features and clusters generated in previous step are given as an input to BFS algorithm. The relevant feature set is obtained as the output of this step.

FIGURE 3: In the final step to obtain new clusters $C = C_1, C_2, \ldots\ldots, C_i$, the complete dataset with only relevant features is given as input to K-Means algorithm.

Functional Model: This model checks whether BFS algorithm can be used recursively to provide improved result after each iteration as shown in FIGURE 4. The statistical feature set is given as input to K-Means. K-means generate clusters which are given to the BFS algorithm. The output of BFS algorithm is then again given to K-Means algorithm to generate the final clusters. This paper checks whether entropy (E) obtained after first application of K-Means is greater than what is obtained after second application of K-Means. Also, whether any further iteration of BFS algorithm can produce better results in terms of accuracy.

Collecting traffic traces: First step is to collect internet traffic traces. It can be collected either on basis of time duration or on number of packets. This paper uses number of packets as unit of collecting traffic traces. The traffic traces have to be collected on different days and on different time of the day to get an appropriate mixture of the traffic.

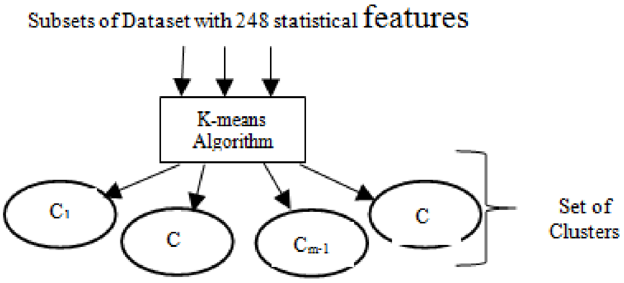


Figure 1. Generating Clusters

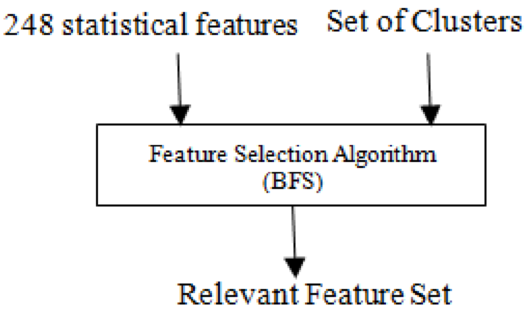


Figure 2. Finding Optimal Features

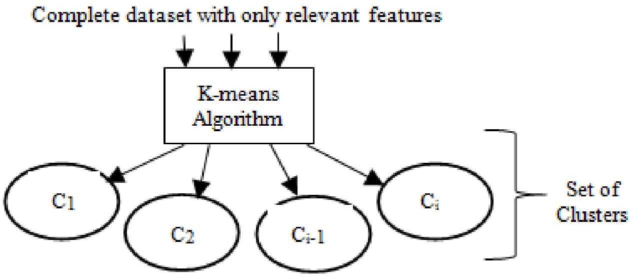


Figure 3. Obtaining New Clusters

Filtering TCP flows: Traffic traces collected are filtered to have TCP flows. In a particular traffic trace, there can be any number of TCP flows. Each TCP flow consists of chronicle history of messages exchanged between same end hosts. Each TCP flow is saved separately.

Generating statistics: Statistical features are generated for each traffic flow separately. All the 248 features are generated and saved corresponding to each traffic flow.

Generating dataset: Statistical features corresponding to each traffic flow are assembled together to form the database. The database can be described as N X 248 matrix, where N is the number of TCP traffic flows collected.

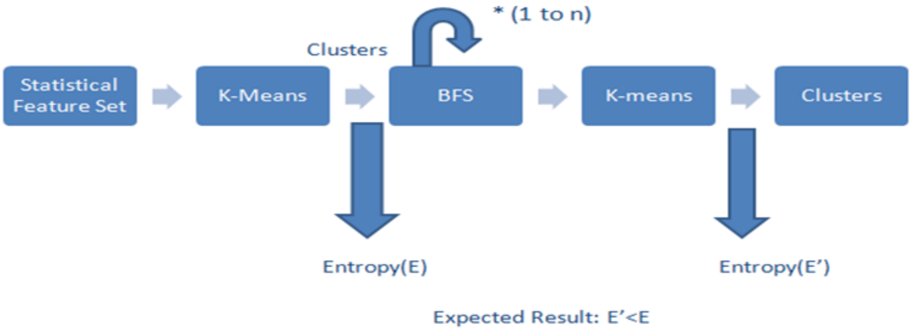


Figure 4. Functional Model

Generate clusters from subset of dataset: The K-Means algorithm is applied on a subset of the database. Clusters and their members are obtained as the result of this phase.

Optimal feature set: Algorithm in Sub-Section 4.9 is used to obtain optimal feature set.

Modified Best Feature Subset Algorithm: The work in this paper uses an unsupervised environment for execution of algorithm. For this purpose, algorithm has been modified from its original version represented in [11] by adding step no. 8 & 9.

Algorithm 1 Modified BFS algorithm

```
procedure Optimal Feature Set Selection
    1. $A = A_1, A_2, ....., A_m$: Statistical feature set; m: number of features
    2. $C = C_1, C_2, ...., C_q$: Set of classes; q: number of classes
    3. Calculate Relative Uncertainty (RU) for each feature of every class using the equation
       $$RU\left(A_i/C_j\right) = \frac{\sum_{k=1}^{N_{A_i}}\left(-p\left(A_{ik}/C_j\right) * \log_2\left(p\left(A_{ik}/C_j\right)\right)\right)}{\log_2(\min(N_{C_j}, N_{A_i}))} \quad (1)$$
       where $p\left(A_{ik}/C_j\right) = \frac{N_{i_{jk}}}{N_{C_j}}$
    4. Calculate Bias coefficient (B) using the equation:
       $$B\left(A_i/C_j\right) = 1 - RU\left(A_i/C_j\right) \quad (2)$$
    5. Select features whose Bias coefficient lies between 0.6 and 0.8. Let this set be $F = F_1, F_2, \ldots\ldots F_f$
    6. Calculate Symmetric Uncertainty (SU) for each feature in F using the equation and arrange in descending order:
       $SU(A_i, C) = 2 * \frac{IG(A_i|C)}{H(A_i)+H(C)}$    (3)
    7. By selecting different ranges of SU obtained select the corresponding features from F.
    8. Apply K-Means algorithm on all TCP flows using features only from F.
    9. Optimal feature set selected on basis of evaluation parameters which shows best result.
end procedure
```

Evaluation Parameters: There are two sets of evaluation parameter namely: (1) To select the optimal feature subset. (2) To compare the approach of this paper with other existing algorithms.

Selecting optimal feature set: The optimal feature set is selected on the basis of two parameters: (1) *Percentage of incorrectly classified instances:* This means the percentage of instances that were mapped to incorrect clusters. It changes with the number of features chosen in optimal set. Lower the percentage, better the optimal feature set. (2) *Sum of squared error:* It is the sum of square of distance between instance and centroid chosen for a particular cluster in K-Means algorithm. From two clusters, one having lower sum of squared error is selected.

Comparison with other approaches: The hybrid approach of this paper is compared to existing solutions namely Simple K-Means, Naive Bayes, C4.5, DBSCAN, and Expected Maximization. The parameters on which they are compared are:

True Positive Rate (TPR): It is the ratio of actual positives that are identified correctly.

$$TPR = \frac{TP}{TP+FN} \quad (4)$$

False Positive Rate (FPR): It is the ratio of actual negatives that are identified correctly.

$$FPR = \frac{FP}{FP + TN} \quad (5)$$

Precision: The numbers of instances which belong to class A, among all those mapped to class A.

$$Precision = \frac{TP}{FP+TP} \quad (6)$$

F-Measure: This parameter is used to rank and compare the per-application performance of ML algorithms.

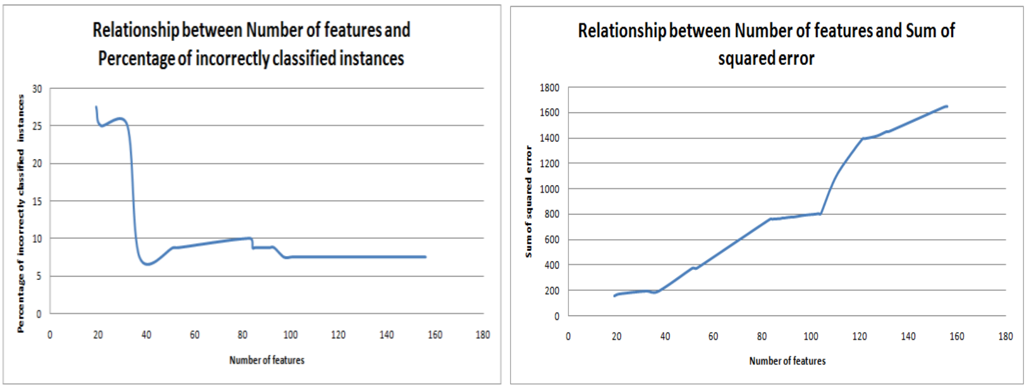
$$F - Measure = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (7)$$

5. Results and Discussions

Results are shown to justify the obtained optimal feature set and that the given approach is better than the existing algorithms.

5.1. The optimal feature set

A total of 156 features are obtained which act as discriminators that is they have a bias coefficient value between 0.6 and 0.8. By selecting value in different ranges a certain number of features are selected and two graphs are plotted: (1) between number of features and percentage of incorrectly classified instances (2) between number of features and sum of squared errors. Figure 5 a) demonstrations that the minimum percentage of incorrectly classified instances is obtained when the number of features is 37 or more than 97. Figure 5 b) demonstrations that the sum of squared error obtained when number of features is 37 is less than that obtained when the number of features is greater than 97. Given two clusters, we should always choose the cluster having minimum sum of squared error. From the above statements, it is concluded that the optimum number of features is 37. They are shown in Table 1.



(a) (b)

Figure 5. (a) Relationship between Number of Features and Percentage of Incorrectly Classified (b) Relationship between Number of Features and Sum of Squared Error

Table 1. Optimal Feature Set

max_data_wire	med_data_wire_a b	max_segm_size_b a
q1_data_wire_b a	med_data_wire_b a	total_packets_b a
ack_pkts_sent_b a	pure_acks_sent_a b	unique_bytes_sent_a b
actual_data_pkts_b a	SYN_pkts_sent_a b	min_segm_size_b a
avg_segm_size_a b	avg_segm_size_b a	q3_data_wire
q3_data_ip	max_data_control_a b	max_data_control
SYN_pkts_sent_b a	req_1323_ws_a b	req_1323_ts_a b
req_1323_ws_b a	req_1323_ts_b a	adv_wind_scale_a b
req sack_a b	req sack_b a	sacks_sent_b a
mss_requested_a b	initial_window-bytes_b a	data_xmit_time_a b
data_xmit_time_b a	RTT_stdv_a b	RTT_stdv_b a
segs_cum_acked_b a	pushed_data_pkts_b a	max_data_wire_b a
max_data_ip_b a		

5.2. Comparison with other Machine Learning Algorithms

The same data set were used for Simple K-means, Naive Bayes, C4.5, DBSCAN and Expected Maximization algorithms. There are results are tabulated in Table 2. DBSCAN is not mentioned in the table because it did not result in any clusters. Thereby proving it inappropriate for the purpose of the classification. It can be seen that the hybrid approach used in this paper is giving better results than any other classification algorithm.

Naive Bayes and J48 (Java implementation of C4.5) were used as the basis of comparison because they are the best supervised machine learning algorithms used for classification. Expected maximization and DBSCAN have been considered to show that K-Means is better than other popular clustering algorithms used for classification [5].

Table 2. Comparison with other Machine Learning Algorithms

Machine Learning Algorithm	TPR	FPR	Precision	F-Measure
Naive Bayes*	0.925	0.062	0.931	0.925
J48*	0.905	0.071	0.927	0.925
Naive Bayes**	0.863	0.051	0.878	0.863
J48**	0.9	0.045	0.895	0.906
K-Means + BFS	0.914	0.060	0.955	0.934

The training set was made by clusters generated using K-Means

The training set was made by clusters generated using Expected maximization

The functional model that was proposed to evaluate the linear or recursive nature of BFS algorithm is as follows. After the reduced feature subset is obtained, the dataset now has N rows and 37 columns. On the second iteration of the BFS algorithm, the information gain cannot be better than obtained for the original dataset. The reason is the values of the features are not going to change. And if we try to reduce the feature set, we have already seen the number of incorrectly classified instances will increase. Hence, the recursive use of BFS algorithm will not provide any enhancement to the result obtained by linear method.

6. Conclusion and Future Work

Internet traffic classification has become a very important research area due to its application in fields like traffic prediction, QoS assessment, bandwidth management, congestion avoidance and intrusion detection systems. ML algorithms have proved to provide better results than any other method. This paper proposed a hybrid approach - using an unsupervised clustering algorithm with a supervised feature selection algorithm. The results have shown to provide an optimum feature set that gives better results than other existing approaches like Naive Bayes, C4.5, DBSCAN and EM. The result of this paper can be used to check its suitability for real time applications like congestion avoidance, QoS assessment, bandwidth management *etc.*

In future work, this work may be checked on much larger dataset to improve amount of information that can be gained from traffic trace; Bidirectional flow; and larger network and a subnet of network to get better understanding of user behavior in network. Further, in this paper TCP flows are considered. The approach can be tested on UDP flows. A lot of traffic belongs to UDP. Considering UDP flows can help identify traffic belonging to other applications not identified by TCP. Thus, it can help to determine the specific applications more accurately.

References

[1] V. P. G. João, P. R. M. Inácio, B. Lakic, M. M. Freire, H. J. A. D. Silva and P. P. Monteiro, “Source traffic analysis”, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 6, no. 3, **(2010)**, pp. 21.

[2] E. L. Will, M. S. Taqqu, W. Willinger and D. V. Wilson, “On the self-similar nature of Ethernet traffic (extended version)”, IEEE/ACM Transactions on networking, vol. 2, no. 1, **(1994)**, pp. 1-15.

[3] W. Walter, V. Paxson and M. S. Taqqu, “Self-similarity and heavy tails: Structural modeling of network traffic”, A practical guide to heavy tails: statistical techniques and applications, vol. 23, **(1998)**, pp. 27-53.

[4] K. Faisal, N. Hosein, S. Ghiasi, C. N. Chuah and P. Sharma, “Streaming solutions for fine-grained network traffic measurements and analysis”, IEEE/ACM transactions on networking, vol. 22, no. 2, **(2014)**, pp. 377-390.

[5] B. Tania, S. Sahni, and G. Seetharaman, “PC-DUOS+: A TCAM architecture for packet classifiers”, IEEE Transactions on Computers, vol. 63, no. 6, **(2014)**, pp. 1527-1540.

[6] G. Sun, “Network traffic prediction based on the wavelet analysis and Hopfield neural network”, International Journal of Future Computer and Communication, vol. 2, no. 2, **(2013)**, pp. 101.

[7] D. Chunjiao, S. H. Richards, Q. Yang, and C. Shao, “Combining the statistical model and heuristic model to predict flow rate”, Journal of Transportation Engineering, vol. 140, no. 7, **(2014)**.

[8] C. Madalena, A. L. Goldberger and C. K. Peng, “Multiscale entropy analysis of biological signals”, Physical review E, vol. 71, no. 2, **(2005)**.

[9] W. Nigel, S. Zander, and G. Armitage, “A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification”, ACM SIGCOMM Computer Communication Review, vol. 36, no. 5, **(2006)**, pp. 5-16.

[10] L. Qiong and Z. Liu, “A comparison of improving multi-class imbalance for internet traffic classification”, Information Systems Frontiers, vol. 16, no. 3, **(2014)**, pp. 509-521.

[11] Z. Liu and L. Qiong, “A new feature selection method for internet traffic classification using ml”, Physics Procedia, vol. 33, **(2012)**, pp. 1338-1345.

[12] L. Mark and O. Maimon, “A compact and accurate model for classification”, IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 2, **(2004)**, pp. 203-215.

[13] D. Alberto, A. Pescape and K. C. Claffy, “Issues and future directions in traffic classification”, IEEE network, vol. 26, no. 1, **(2012)**, pp. 35-40.

[14] M. Andrew, D. Zuev and M. Crogan, “Discriminators for use in flow-based classification”, Queen Mary and Westfield College, Department of Computer Science, **(2005)**.

[15] I. O. S. Cisco, “Quality of Service Solutions Configuration Guide”, Congestion Avoidance Overview. Cisco, Accessed, vol. 18, **(2014)**.

[16] C. Jin, W. S. Cleveland, D. Lin and D. X. Sun, “Internet traffic tends toward Poisson and independent as the load increases”, In Nonlinear estimation and classification, Springer New York, **(2003)**, pp. 83-109.

[17] L. Karthik, A. Rangarajan and S. Venkatachary, “Algorithms for advanced packet classification with ternary CAMs”, In ACM SIGCOMM Computer Communication Review, ACM, vol. 35, no. 4, **(2005)**, pp. 193-204.

[18] M. Jianning, C. N. Chuah, A. Sridharan, T. Ye and H. Zang, “Is sampled data sufficient for anomaly detection”, In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, ACM, **(2006)**, pp. 165-176.

[19] X. Zhengtao, Y. Li and L. Xiong, “Using Multiscale Entropy Method to Analyze the Complexity of Traffic Flow”, In 2012 Second International Conference on Business Computing and Global Informatization, IEEE, **(2012)**, pp. 785-788.

[20] E. Najjary, T. G. U. Keller, M. Pietrzyk and J. L. Costeux, “Application-based feature selection for internet traffic classification”, In Teletraffic Congress (ITC), 2010 22nd International, **(2010)**, pp. 1-8.

[21] E. Jeffrey, M. Arlitt, and A. Mahanti, “Traffic classification using clustering algorithms”, In Proceedings of the 2006 SIGCOMM workshop on Mining network data, **(2006)**, pp. 281-286.

[22] K. Grabczewski and N. Jankowski, “Feature selection with decision tree criterion”, Fifth International Conference on Hybrid Intelligent Systems, **(2005)**, pp. 1-6.

[23] C. A. Ratanamahatana and D. Gunopulos, “Scaling up the naïve bayesian classifier: Using decision trees for feature selection”, Proceedings of Workshop on Data Cleaning and Preprocessing, **(2002)**, pp. 1-10.

[24] P. Langley and S. Sage, “Induction of selective Bayesian classifiers”, Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, **(1994)**, pp. 399-406.

[25] C. Claire, “Using decision trees to improve case-based learning”, In Proceedings of the tenth international conference on machine learning, **(1993)**, pp. 25-32.

[26] K. Miroslav, D. Flotzinger and G. Pfurtscheller, “Discovering patterns in EEG-signals: Comparative study of a few methods”, In European Conference on Machine Learning, Springer Berlin Heidelberg, **(1993)**, pp. 366-371.

[27] B. Tomasz, T. Riaz and J. M. Pedersen, “A method for classification of network traffic based on C5. 0 Machine Learning Algorithm”, In Computing, Networking and Communications (ICNC), 2012 International Conference on, **(2012)**, pp. 237-241.

[28] P. Nilima, R. Lathi, and V. Chitre, “Comparison of C5. 0 & CART classification algorithms using pruning technique”, In International Journal of Engineering Research and Technology, ESRSA Publications, vol. 1, no. 4, **(2012)**.

[29] URL http://en.wikipedia.org/wiki/Traffic_classification.