Generalization and Optimization of Feature Set for Accurate Identification of P2P Traffic in the Internet using Neural Network

S. AGRAWAL, B.S. SOHI
Department of Electronics & Communication Engineering
Panjab University
Sector-14, Chandigarh
INDIA
s.agrawal@hotmail.com, bssohi@yahoo.com

Abstract - P2P applications supposedly constitute a substantial proportion of today's Internet traffic. The ability to accurately identify different P2P applications in internet traffic is important to a broad range of network operations including application-specific traffic engineering, capacity planning, resource provisioning, service differentiation, etc. In this paper, we present a Neural Network approach that precisely identifies the P2P traffic using Multi-Layer Perceptron (MLP) neural network. It is general practice to reduce the cost of classification by reducing the number of features, utilizing some feature selection algorithm. The reduced feature set produced by such algorithms are highly data-dependent and are different for different data sets. Further the feature set produced from one data set does not yield good results when tried upon other data sets. We propose an optimum and universal set of features which is independent of training and test data sets. The proposed feature set has enabled us to achieve significant improvement in performance of the MLP classifier. The few features in the proposed feature set results in a significant reduction in training time, while maintaining the performance, thereby making this approach suitable for real-time implementation.

Keywords – Peer-to-Peer (P2P), Traffic classification, Flow features, Multi-Layer Perceptron (MLP) Neural Network.

1 Introduction

Over the last few years, peer-to-peer (P2P) file-sharing has dramatically grown to represent a significant component of Internet traffic. P2P volume is sufficiently dominant on some links, thereby causing an increased local peering among Internet Service Providers. This is observable, but its effect could not be quantified on the global Internet topology and routing system not to mention competitive market dynamics. Despite this dramatic growth, reliable profiling of P2P traffic remains elusive, as the newer generation P2P applications are incorporating various strategies to avoid detection. We no longer enjoy the benefit of first generation P2P traffic, which was relatively easily classified due to its use of well-defined port numbers. Current P2P applications tend to intentionally disguise their generated traffic to circumvent both filtering firewalls as well as legal issues. Not only do most P2P applications now operate on top of nonstandard, custom designed proprietary protocols, but also current P2P clients can easily operate on any port number. Internet service providers as well as enterprise networks require the ability to accurately identify the different P2P applications, for a range of uses, including network operations and management, application-specific traffic engineering, capacity planning, resource provisioning, service differentiation and cost reduction.

These circumstances lead to a conclusion that accurate identification of P2P traffic is only possible by examining user payload. Yet user payload capture and analysis is not possible because of issues like legal, privacy, technical, logistic, and financial. Further P2P applications tend to support payload encryption, obfuscating payload characterization attempts. Therefore, the frequency with which P2P protocols are being introduced and/or upgraded renders user payload analysis impractical as well as inefficient.

The research community has responded by exploring classification algorithms capable of inferring target application without deep packet inspection. A number of researchers are investigating the use of Machine Learning (ML) techniques for IP traffic classification. The working of these techniques depends on the externally observable attributes of the traffic (such as

maximum/minimum packet length in each direction, flow duration, inter-packet arrival time etc.). Each traffic class is characterized by the same set of features but with different feature values. An ML classifier is trained to associate set of features with known traffic classes (mapping), and apply the trained ML classifier to classify unknown traffic.

At present, much of the existing research focuses on the achievable accuracy of different ML algorithms. Selecting few most relevant features out of a number of features is very important aspect to build ML classifier, as to improve its accuracy as well as its computational performance. Currently, the researchers are using some feature selection algorithms, which produce different feature sets for different data sets, and are claiming good accuracy. But the effect of using same set of features on the different data sets and devising a common feature set has attracted almost no investigation. In this paper, we mainly focus on devising a common feature set applicable to every data set, its optimization, and their effect on the performance of Neural Network for identification of P2P traffic. The main contributions are as follows:

- We propose a common and optimized feature set for the first time, which is independent of the data set to be classified.
- The performance of MLP (Multi-Layer Perceptron) algorithm is evaluated when using proposed feature set and traditional feature set selected by a popular feature selection algorithm. As the results show, the MLP algorithm achieves a significant improvement in the performance when proposed feature set is used.
- The proposed feature set is further optimized without degradation in performance of MLP classifier, but with significant reduction in model build time and classification speed.
- We observe that in the training data set, only few percent of the total flows are labeled as P2P flows. This means that the distribution of these data is class imbalanced, where P2P class is heavily under-represented (the minority class) in comparison with the other classes (the majority class). This is of particular importance because trained Neural Network classifier always bias the majority classes, thereby giving low prediction accuracy for under-represented class. We use resampling methods to overcome the class imbalance problem, under-sampling the majority classes and over-sampling the minority class (P2P). We do not perceive any impact of class imbalanced training data on the performance of MLP classifier when the proposed feature set is used.

Our paper is organized as follows. Section 2 briefly summaries key ML concepts and related work. Section 3 describes our problem and approach for devising proposed feature set. Section 4 presents the experimental results and discussion. We discuss conclusion and future work in section 5.

2 Machine Learning and Related Work

Since the Neural Network falls under the category of Machine Learning (ML) algorithms, so first we summarize the basic concepts of ML and review related work applying ML to IP traffic classification.

2.1 Machine Learning Concepts

Machine Learning algorithms are used to map instances of network traffic flows into different traffic classes, where each flow is represented by a set of statistical features and associated feature values. These features can be calculated from packets of a traffic flow – such as flow duration, inter-packet arrival time. Each traffic flow is described by the same set features, though each will exhibit different feature values depending on the traffic class to which it belongs.

ML algorithms can utilize either unsupervised (clustering) or supervised learning (classification) approaches. Supervised learning involves learning from a set of pre-classified examples to classify unseen examples. It consists of two stages - *training* the ML algorithm to associate sets of features with known traffic classes, and *testing* - applying the learning to classify unknown traffic; whereas, unsupervised learning approach discovers natural groups in the data using some internal heuristics. Unsupervised approaches group instances with similar properties (e.g. Euclidean space for distance-based learning) into clusters.

2.2 Feature Selection

Features are defined as any statistics that can be calculated from the packets within a flow. Maximum segment size, number of packets with

acknowledgement, packet length are some valid features. For bi-directional flows, the features are calculated for both directions of the flow. In this way, a number (few hundreds) of features can be calculated for each flow in the internet traffic.

In practical IP traffic classification work, the quality and size of a feature set greatly influence the effectiveness of ML algorithms. A large number of features can be defined and extracted from the internet traffic, but most of the obtainable features are redundant or irrelevant, that can increase the computational cost of ML algorithm and can also negatively influence algorithm performance.

The feature selection process removes irrelevant and redundant [1] features, i.e., those that can be excluded from the feature set without loss of classification accuracy. At the same time the removal of some features may also improve the performance of the ML algorithm. It is a preprocessing step to machine learning, and is the process of choosing a subset of original features that will optimize for higher learning accuracy with lower computational complexity and higher classification speed.

2.3 Evaluation Methods

For evaluating the performance of supervised ML algorithms, it is necessary to define training and testing datasets. The training set consists of known network traffic and is used to build the classification model. The testing set represents the unknown network traffic that we wish to classify. Both the training and testing sets are labeled with the appropriate class beforehand. By knowing the class of each flow, we can compare the predicted class against the known class to evaluate the performance of the classifier.

We use following standard metrics to evaluate the performance of ML classifiers. If a classifier is trained to identify members of class X:

• *Recall*: Defined as percentage of members of class X correctly classified as belonging to class X.

• *Precision*: Defined as percentage of those instances that truly have class X, among all those classified as class X.

These metrics range from 0 (poor) to 100% (optimal). It is important to note here that high precision is meaningful only when the classifier attains good recall, because recall tells about the ability of the classifier to correctly identify the instances of the target class. It is seen that most of the ML classifiers give a low value of recall.

2.4 Related Work

In this section, we discuss the payload-based and payload-independent methods reported in the literature and relevant to our work. Most protocols contain a protocol-specific string (called application signature) in the payload that can be used for application identification. These strings are often public information and can also be obtained by examining a number of network traffic traces.

Sen et al. [2] proposed an approach for detection of P2P traffic through application-layer signatures. They examined available documentation and packet-level traces to identify application layer signatures, and then utilized these signatures to develop filters that could track P2P traffic. They analyzed TCP packets in the download phase of file transfer. They decomposed P2P signatures into fixed pattern matches with fixed offsets and variable pattern matches with variable offset within a TCP payload. Those authors evaluated the accuracy and scalability of the application-layer signature technique. Their results showed that the proposed technique had < 5% for both false positives and false negatives, indicating that the technique was accurate most of the time.

Application signature is obtained by analyzing the user payload. Though payload-based classifiers show good results but not suitable for newer generation of P2P traffic, because of two major limitations: (a) they cannot be used if payload information is not available and (b) they cannot identify unknown classes of traffic.

Karagiannis et al. [3] proposed an approach to identify P2P flows at the transport layer. This approach was based on connection patterns, without relying on user's packet payloads. Their transport-layer approach relies primarily on two heuristics. The first one identifies source-destination IP pairs that concurrently use both TCP and UDP. If such IP pairs are found, not using specific well-known ports, then these flows are considered P2P flows. The second one considers the structural pattern of transport-layer connections between hosts.

Hu et al. [4] proposed a profile-based approach to identify traffic flows belonging to the P2P application. Depending on the patterns dominant in the application, they built behavioral profiles of the target application. Based on this behavioral profile, they used a two-level matching method to identify new traffic. At first level matching, they determined whether a host participates in the target application by comparing its behavior with the profiles. At the second level

matching, they compared each flow of the host with those patterns in the application profiles to determine which flows belong to this application. The results showed that popular P2P applications could be identified with high accuracy.

The preceding techniques are highly dependent on the information gathered through deep inspection of packet content (payload and port numbers), thereby limiting their usefulness. Newer techniques rely on traffic's statistical characteristics to identify the target application. Such techniques assume that traffic at the network layer has statistical properties (such as the distribution of flow duration, flow idle time, packet inter-arrival time and packet lengths) that are unique for certain classes of applications and enable different applications to be distinguished from each other.

Williams et al. [5] evaluated the efficiency and performance of different feature selection and machine learning techniques. They discussed the discriminative power of different flow features for the purpose of traffic classification and also investigated the influence of flow timeout and size of training data set. With 22 features they were able to achieve classification accuracies of over 99% for some ML algorithms, but again when the same data set was used for training and testing.

Zuev et al. [6] proposed a supervised machine learning approach to classify network traffic. They started by allocating flows of traffic to one of several predefined categories: Bulk, DataBase, Interactive, Mail, WWW, P2P, Service, Attack, Games and Multimedia. They then utilized 248 per-flow discriminators (characteristics) to build their model using Naive Bayes analysis. They evaluated the performance of the model in terms of accuracy (the raw count of flows that were classified correctly divided by the total number of flows) and trust (the probability that a flow that has been classified into a class, is in fact from that class). Although this approach is promising, there is a question about the scalability of the approach as it involves too many discriminators, and it takes much time to prepare the data (with many attributes) and assign the traffic flows to predefined categories only. To overcome these limitations, Moore and Zuev used Fast Correlation-Based Filter and a variation of a wrapper method to reduce the number of discriminators [7].

Auld et al. [8] applied Bayesian Neural Network to classify the internet traffic. They used Fast Correlation Based Filter (FCBF) to select ten features out of total 246 features. Then using Bayesian neural network, they achieved an average classification accuracy of 99.3%, where P2P traffic could be identified with only 62% accuracy. To maximize the average accuracy of each class, they equalized the proportion of flows for each class in the training data to overcome the class imbalance and reported 97.2% accuracy for P2P traffic. But all this performance evaluation was done using the same data set both for training and testing. Any ML algorithm is expected to exhibit good performance, if the algorithm is trained and tested on the same data set. The actual classification accuracy of a Neural Network classifier can be evaluated only if the classifier is trained and tested on different data sets, using the same set of features.

Li et al. [9] compared the effective and efficient classification of network-based applications using behavioral observations of network-traffic and those using deep-packet inspections. They demonstrated the accurate training of models from data with a high-confidence ground-truth and the use of an efficient and small feature set derived from the reduction of a large flow feature list using a sophisticated feature selection mechanism.

Raahemi et al. in [10] applied supervised machine learning techniques, namely Neural Networks and decision trees, to classify P2P traffic. They pre-processed and labeled the data, and built several models using a combination of different attributes for various ratios of P2P/nonP2P in the training data set. They reported a significant increase in their classifier accuracies when source and destination IP addresses are taken into account. This limitation dictates that the classifier can not be implemented outside the administrative domain of the individual service provider's networks.

In this paper, we use Multi-Layer Perceptron (MLP) Neural Network, which provides a promising alternative in classifying flows, based on payload independent statistical features derived from packet streams consisting of one or more packet headers. Each traffic flow is characterized by the same set of features but with different feature values. An MLP classifier is built by training on a pre-classified set of flow instances where the network applications are known. Then the built MLP classifier is used to predict the class of an unknown flow.

3 Experimental Approach

As the main focus of this work is to show the superiority of our proposed feature set over the feature sets calculated from the traditional feature selection algorithms, we use ten data sets and one ML algorithm. Using first data set, we build one classifier model for each feature set, and then those classifier models are tested with all of the ten data sets for their prediction accuracy for the target application (P2P).

In the following sections, we detail the Multi-Layer Perceptron (MLP) algorithm and feature selection procedures used to devise a common and reduced feature set. The details of IP traffic data sets and selected features are also given.

3.1 Data Sets

We illustrate our method with pre-classified data described originally in [11]. This data consists of description of internet traffic that has been classified manually, to provide the input sets for the training and testing phases. To construct the sets of data, the day trace was split into ten blocks (approximately 28 minutes each) of transport control protocol (TCP) traffic flows, with each instance described by its membership class and a set of 248 features. This feature set includes flow duration statistics, TCP Port information, payload size statistics, Fourier transform of the packet inter-arrival time and more. In order to provide a wider sample of mixing across the day, the start of each sample was selected randomly (uniformly distributed over the whole day trace).

While each of the data sets represent approximately the same period of time, the number of instances per data set fluctuates as a result of the variation in the activity throughout the course of the day. Further details of the original hand-classification are given in [12], and the data sets themselves are described at length in [11].

Our central object for classification is the traffic flow and for the work presented, we have limited our definition of a traffic flow to being a complete TCP flow, those that start and end correctly, e.g., with the first SYN, and the last FIN ACK.

3.2 Feature Selection

We use the Correlation-based Filter (CFS), which is computationally practical and outperforms the other filter method (e.g. Consistency based Filter) in terms of classification accuracy and efficiency [4, 13]. The Correlation-based Filter examines the relevance [14] of each feature, i.e., those highly correlated to specific class but with minimal correlation to each other [13]. We use a Best First search algorithm to generate candidate subsets of features from the full feature set, since it provides higher classification accuracy than Greedy search algorithm.

These feature selection algorithms are highly data-dependent, i.e. for every data-set, these algorithms result in different set of features, which are unique to each data-set. It is found that for a given data-set, the classification algorithm gives the highest classification accuracy, when its own unique feature set is used. This means that before we submit the data set to such classifier, it is necessary to record the data set first and then run the feature selection algorithm to select the sub-set of features, build the classifier as per that feature set and finally submit the data set for classification. This tedious and lengthy procedure can not be carried out for each and every data set to be classified, and also not suitable in real-time applications. We want a common feature set, on the basis of which, the ML algorithm is trained once, then all of the data-sets could be classified, using the same features, by trained ML algorithm, without loss of classification accuracy. For this purpose, we propose a common reduced feature set.

We use the WEKA machine learning software suite [15], often used in traffic classification efforts, to evaluate the Multi-layer Perceptron (most commonly used) supervised machine learning algorithms. Evaluation of our proposed feature set with other ML algorithms is the subject of current ongoing work.

3.3 Multilayer Perceptron (MLP)

Multilayer perceptron [16] is a multi-layer feed-forward kind of neural network. Each layer is composed of multiple numbers of neurons, where a neuron (or simply node) is a basic processing unit. The output of a neuron is a combination of the multiple inputs from other neurons. Each input is weighted by a weight factor. A neuron outputs if the sum of the weighted inputs exceeds a pre-defined threshold function of the neuron. The architecture of the multilayer perceptron consists of a single input layer of neurons, one or multiple hidden layers and a single output layer of neurons (Fig. 1).

In order to learn, the perceptron must adjust it weights. The learning algorithm compares the

actual output to the desired output to determine the new weights repetitively for all training instances. The network trains with the standard backpropagation algorithm, which is a two-step procedure. The activity from the input pattern flows forward through the network, and the error signal flows backward to adjust the weights. The generalized delta rule adjusts the weights leading into the hidden layer neurons and the weights leading into the output layer neurons. Our multiplayer perceptron uses sigmoid functions, which is a continuous activation function.

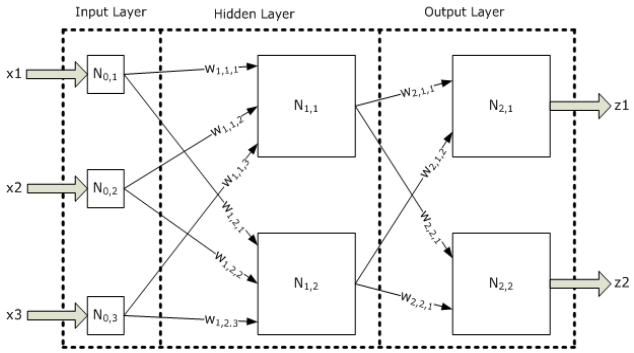


Fig. 1 Architecture of MLP Neural Network

In our MLP neural network, the number of input nodes is equal to the number of features and the number of output nodes is equal to the number of classes, in which the IP traffic is to be classified. We take only one hidden layer, which has as many nodes as the average of the number of features and the number of classes.

In this algorithm, we perform normalization of all features including the class feature (all values are between -1 and +1 after the normalization). We set the learning rate (weight change according to network error) to 0.3, the momentum (proportion of weight change from the last training step used in the next step) to 0.2 and we run the training for 500 epochs (an epoch is the number of times training data is shown to the network).

4 Results and Discussion

Our ultimate goal is to show the impact of the feature selection and the number of features in the feature set on the performance of our chosen ML algorithm, for detection of P2P application in the internet traffic. First, 'Best First' search algorithm is used to generate candidate subsets of features from the full feature set. Further using correlation-based feature subset evaluation, we identify the best subset of features. This procedure of generating best subset of features is carried out for first data set. We refer to this as the 'CFS-BEST FIRST' feature set, the detail of which is given in the Table 1.

Feature Number	Feature Description
1	Port Number at server
2	The total number of ack packets seen carrying TCP SACK blocks (Server to Client)
3	Total packets seen with the PUSH bit set in the TCP Header (Server to Client)
4	Maximum Segment Size requested (Client to Server)
5	The total number of bytes sent in the initial window (Server to Client)
6	The missed Data, calculated as the difference between the ttl stream length and unique bytes sent (Client to Server)
7	The maximum number of retransmission (Client to Server)
8	Minimum of control bytes in packet (Server to Client)

Table 1 Description of 'CFS-BEST FIRST' feature set

Then, we devise a common reduced feature set, using the following procedure:

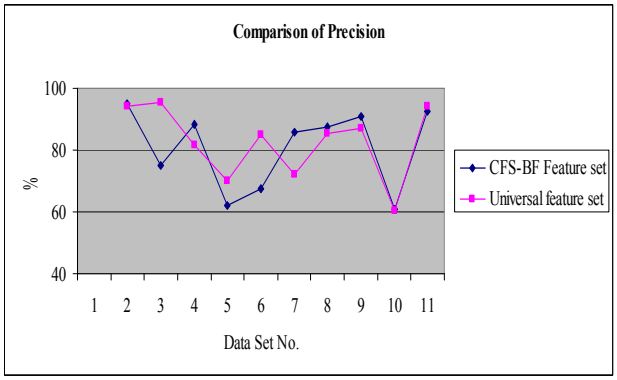
- Run the CFS-BEST FIRST feature selection algorithm on the ten data-sets, and get the ten different sets of features, referred as 'CFS-BEST FIRST' feature sets.
- The feature, which occurs more frequently in these ten feature sets, has more discriminative power. On the basis of this heuristic, we construct a new feature set, which includes all those features occurring in more than two out of ten feature sets.
- By selecting features in this way, we could be able to keep the number of feature in our new feature set to a minimum, so as to ensure lower computational complexity.
- The new feature set is given the name 'Universal' feature set, the description of which is given in Table 2:

Feature Number	Frequency of Occurrence	Feature Description

1	10	Port Number at server
2	7	The total number of full-size RTT samples (Client to Server)
3	6	Time Stamp requested (Client to Server)
4	6	The missed Data, calculated as the difference between the ttl stream length and unique bytes sent (Client to Server)
5	4	Total packets seen with the PUSH bit set in the TCP Header (Server to Client)
6	3	Maximum Segment Size requested (Client to Server)
7	3	The total number of ACK packets received after losses were detected and a retransmission occurred. (Server to Client)

Table 2 Description of 'Universal' feature set

The proposed 'universal' feature set consists of 7 features, the frequencies of occurrence of which are also shown in the second column of the Table 1. On the basis of these two feature sets (from Table 1 &2), we build two separate MLP classifier models using first data set, and evaluate these models for their prediction accuracy, in terms of precision and recall, on all of the ten data sets. A comparison of their performances is shown in the Fig. 2.



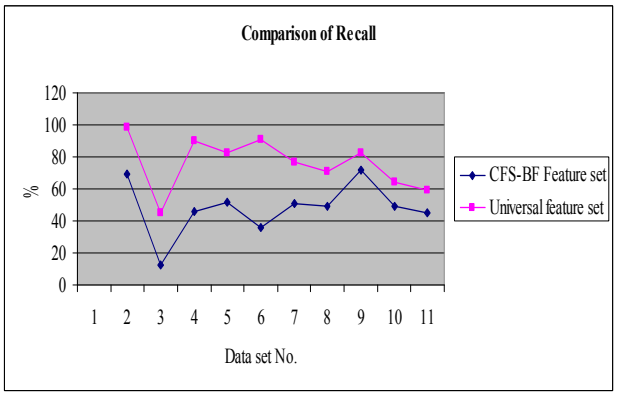
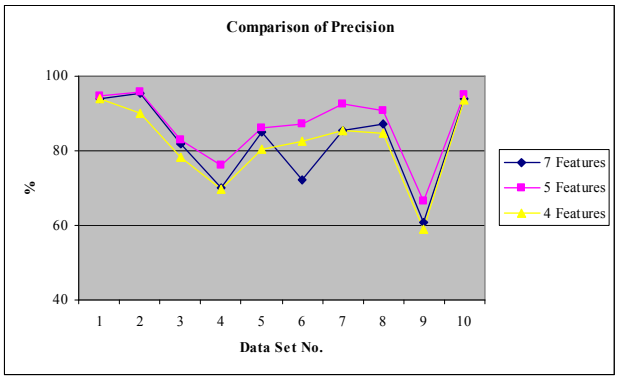


Fig. 2 Performace comparison of MLP classifiers built using CFS-BF and Universal feature sets

It can be easily seen that our proposed feature set is more effective because we could achieve an improvement of 1.98% in mean precision and 27.81% in mean recall. A very large increase in Recall is noteworthy since high precision is meaningful only when the classifier achieves high value of recall. So the results have established the superiority of our proposed feature set for accurate prediction of P2P traffic.

We refer this proposed feature set as 'Universal' feature set because it is now independent of data set to be evaluated. In order to reduce the feature extraction time, the classifier build time and the classification time, it is necessary to further reduce the number of features in the proposed feature set without making any compromise with the prediction accuracy of the built classifier. So first, we drop the last two features (feature number 6 & 7, with Frequency of occurrence = 3) from the proposed feature set, thereby creating a set of 5 features and then dropping the feature number 5, thereby creating a set of 4 features. Using these two reduced feature sets (5 feature set and 4 feature set), we evaluate the performance of the MLP classifier again. Fig. 3 shows the performance comparison of the three classifiers built using feature sets of 7, 5, and 4 features respectively.



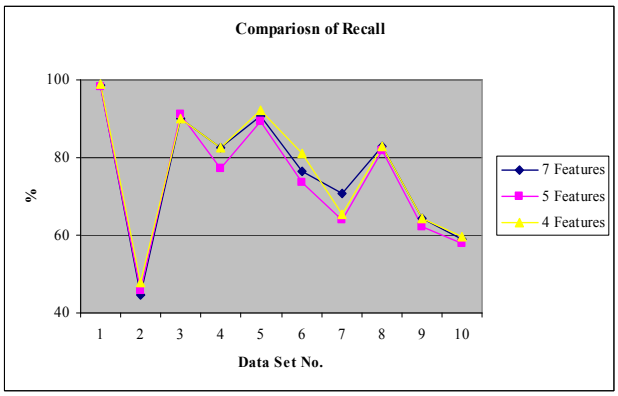


Fig. 3 Performance comparison of MLP classifiers built using feature sets of 7, 5, and 4 features

It is obvious from the figure that the performance of classifier could be maintained while the number of features in the 'universal' feature set is reduced from 7 to 4. When the feature set of 5 features is used, the MLP classifier shows significant improvement in the precision and a marginal decrease in the recall. But with the feature set of 4 features, an improvement in recall is achieved, while maintaining the precision.

Further, it is anticipated that the classification accuracy of the MLP classifier declines over time as the composition of internet traffic changes. Therefore, to get the idea of re-training interval and to test the suitability of the proposed 'Universal' feature sets with time, we use a test data set (taken 1 year later) [11] from different site, and evaluate the performance of the above-mentioned classifier models, a comparison of which is shown in the Fig. 4.

Fig. 4 Performance comparison of MLP classifiers on Test data set

This figure also shows the mean values of precision and recall of the built classifiers for previously used ten data sets. It can be seen that the value of precision is still around the mean value, while the value of recall is much above the mean value, irrespective of the feature set used for building the classifier model. So the classifier models built on the basis of 'Universal' feature sets are still performing excellently even after one year, without retraining of built MLP classifier models, suggesting less frequent re-training of the built classifier. At the same time, the results suggest that 4 features are sufficient to train the MLP classifier with reasonable performance.

We expect a reduction in the training time and classification speed when the number of features is reduced from 7 to 4 in the proposed feature set. A comparison of training time and classification speed of these three classifiers is shown in the Fig. 5.

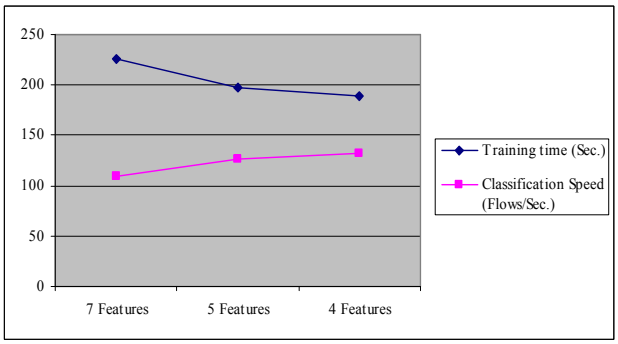
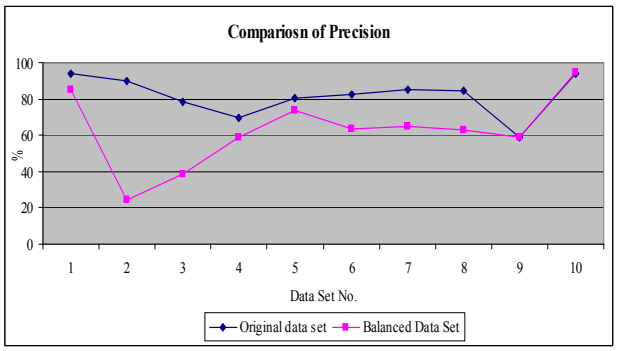


Fig. 5: Computational performance of classifiers

The result shows a decrease in training time and an increase in classification speed, when the number of features in the feature set is reduced from 7 to 4. This further justifies the use of 4 features in the 'universal' feature set, and can be referred as

'optimized' feature set. The very high classification speed of the MLP classifier makes it a suitable candidate for on-line classification.

Inspection of training data set reveals that P2P instances account for only 1.36%, although the bandwidth consumed is as more as 70%. This means that the distribution of the flow instances in a data set is class imbalanced, where P2P class is heavily under-represented (the minority class) in comparison with the other classes (the majority class). This is of particular importance because built ML classifier model always bias the majority classes, thereby giving low prediction accuracy for under-represented class [8]. To overcome this class imbalance problem, we use under-sampling for majority classes and over-sampling for minority class (P2P). This procedure is applied on the training data set, which originally consists of 339 (1.36%) P2P instances out of total 24863 instances. The new training data set (Balances Data Set) thus created consists of 612 (12.8%) P2P instances out of total 4769 instances. We build another MLP classifier model on the basis of the balanced data set, using our proposed optimized feature set. Then the built model is tested against all of the ten data sets for prediction accuracy of P2P traffic. Fig. 6 shows the comparison of performances of two classifier models, where one is trained with the original data set and the other with balanced data set.



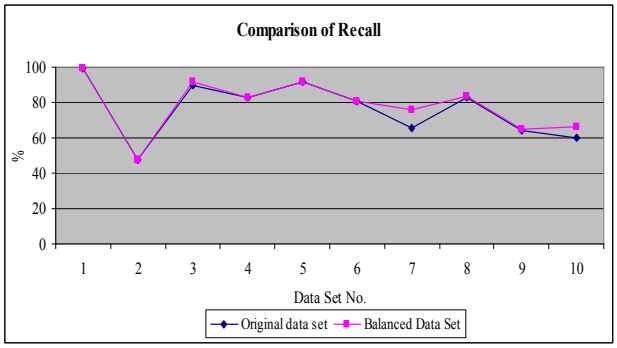


Fig. 6 Performance comparison of MLP classifiers using different training data

It is clear from the figure that the performance of the classifier, trained on the balanced data set, does not improve, rather the precision decreases significantly. That means the class imbalanced training data doest not affect the performance of the MLP classifier when the proposed optimized feature set is used.

Performance metrics were measured using a 3 GHz, Intel Core 2 Duo CPU workstation with 3GB of RAM.

5 Conclusion

P2P application is becoming more prominent and consuming as more as 70% bandwidth of IP network. Accurate identification of P2P applications facilitates in a broad range of network operations like capacity planning, network expansion, resource provisioning and lawful interception. To overcome the limitations of traditional approaches like port, payload and host behavior based analysis; researchers have shown an increased interest in the machine learning techniques for IP traffic classification.

This paper has demonstrated the selection of features and successful application of Multi Layer Perceptron (MLP) neural network for P2P traffic identification. Our main findings are as follows.

- Our proposed 'universal' feature set is more effective because we could achieve an improvement of 1.98% in mean precision and 27.81% in mean recall over the feature set selected from traditional method. A very large increase in Recall is noteworthy since high precision is meaningful only when the classifier achieves high value of recall.
- An MLP neural network, trained on the proposed 'Universal' and 'optimized' feature set, is able to identify P2P traffic, with up to 93.9% precision and 99.1% recall for data trained and tested on the same day, and 80.3% precision and 94.9% recall for data tested twelve months apart, without retraining the built classifier. By providing high performance without calculating all of the possible features and sophisticated

traffic processing, this approach offers good results.

- 'Universal' and 'optimized' feature set consists of only 4 features, which are to be calculated in the one direction only i.e. from Client to Server. This feature set results in significant reduction in training time and an improvement in classification speed without any degradation in the performance, making this approach a low-overhead method with potential for real-time implementation for identification of P2P traffic in the internet.
- By under-sampling of majority class and over-sampling of minority class in the training data set and building the MLP classifier with optimized feature set results in no improvement in the performance of the MLP classifier. The experiments here indicate that the class imbalanced training data has no negative impact on the performance of the MLP classifier for prediction of P2P traffic. This suggests that the class imbalanced training data can be used for training of MLP classifier, thereby reducing pre-processing time used in re-sampling.

To confirm the potential and suitability of the proposed approach, our future work will include the following areas.

- An evaluation of our approach on further sources of classified data from other sites will give insight into the stability and robustness of this approach.
- Testing the classifier on data from later times, to get an idea of the retraining interval.
- For the proposed feature sets, the performance of other machine learning algorithms will be evaluated and compared, so that the versatility of these feature sets can be established.
- Evaluation of this approach against traffic flows which are incomplete, such as incompletely observed TCP flows, and partially observed flows which include user datagram packet (UDP) services.
- Specific implementation issues and algorithmic optimization need to be explored further.

References:

[1] A. Appice, M. Ceci, S. Rawles, and P. Flach, Redundant feature elimination for multi-class problems, *Proceedings of the 21st International Conference in Machine Learning*, Banff, Canada, July 2004.

[2] S. Sen, O. Spatscheck, and D. Wang, Accurate, Scalable In-Network Identification of P2P Traffic using Application Signatures, *Proceedings of the 13th International World Wide Web Conference*, NY, USA, May 2004, pp. 512-521.

[3] T. Karagiannis, A. Broido, M. Faloutsos, and K. Klaffy, Transport Layer Identification of P2P Traffic, *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC 2004)*, Italy, October 2004, pp. 121-134.

[4] Y. Hu, D. Chiu, and J.C.S. Lui, Profiling and Identification of P2P Traffic, *Computer Networks,* Volume 53, issue 6, April 2009, pp. 849-863.

[5] N. Williams, S. Zander, and G. Armitage, Evaluating machine learning algorithms for automated network application identification, *Technical Report 060401B*, CAIA, Swinburne Univ., April 2006.

[6] D. Zuev, and A W. Moore, Traffic classification using a statistical approach, *Springer-Verlag Lecture Notes in Computer Science*, Vol. 3431, Springer Berlin, 2005, pp.321-324.

[7] W. Moore and D. Zuev, Internet traffic classification using Bayesian analysis techniques, *Proceedings of ACM Sigmetrics*, Alberta, Canada, June 2005, pp.50-59.

[8] T. Auld, A.W. Moore, and S.F. Gull, Bayesian Neural Network for Internet Traffic Classification, *IEEE Transaction on Neural Networks,* vol. 18, No. 1, January 2007, pp 223-239.

[9] W. Li, M. Canini, A.W. Moore and R. Bolla, Efficient Application Identification and the Temporal and Spatial Stability of Classification Schema, *Computer Networks*, Volume 53, issue 6, April 2009, pp. 790-809.

[10] B. Raahemi, A. Hayajneh, and P. Rabinovitch, Peer-to-peer IP traffic classification using decision tree and IP layer attributes, *International Journal of Business Data Communications and Networks,* vol. 3, issue 4, pp.60-74, 2007.

[11] A. Moore, D. Zuev, Discriminators for use in flow-based classification, *Technical report, Intel Research*, Cambridge, 2005.
[12] A. W. Moore and D. Papagiannaki, Toward the accurate identification of network applications, *Proceedings of 6th Passive Active Meassurement Workshop (PAM)*, vol. 3431, Mar. 2005, pp. 41–54.
[13] 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*, Vol. 36, No. 5, October 2006, pp. 7-15.
[14] A. Blum and P. Langley, Selection of relevant features and examples in machine learning, *Artificial Intelligence*, 97(1-2), 1997, pp. 245–271.
[15] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, *The WEKA Data Mining Software: An Update*, SIGKDD Explorations, Volume 11, Issue 1, 2009.
http://www.cs.waikato.ac.nz/ml/weka/.
[16] Simon Haykin, *Neural Networks*, Pearson Education, 2006.