User Traffic Classification for Proxy-Server based Internet Access Control

Saad Y. Sait*, M. Sandeep Kumar* and Hema A. Murthy*
*Dept. of Computer Science & Engineering
Indian Institute of Technology Madras, Chennai - 6000036
Email: {saad,hema}@cse.iitm.ac.in, sandeep.madanala25@gmail.com

***Abstract*—In a LAN, Internet access should be managed well for a better user experience. Those using a larger share of the bandwidth may be restricted during peak hours to enable others to use the Internet. This can be viewed as a problem of classifying the users based on their Internet usage into normal and high categories, following which control policies may be applied. For this purpose, a proxy-based mechanism has been proposed for classification of users according to the share of their Internet access. The advantage of this approach is that users sharing the same computer can be distinguished by the proxy server and appropriate control policies can be exercised. To understand user behaviour, data is collected at the proxy server in a campus LAN. Machine learning algorithms are then used to learn and characterise user behaviour. In particular, Naive Bayes' and Gaussian Mixture Model based classifiers are used. It is observed that the algorithms are able to scale in that users are clustered into two different groups. Performance evaluation on a held out data set indicates that users can be accurately distinguished 94.96% of the time. The algorithm is also practical since the time consuming task of model building need be done only once a month offline, while the daily task of classification may be accomplished in a period of 20 mins for GMMs. It has also been shown how the user behavior of the two groups of users may be characterized. This would be a useful aid in the design of policies and algorithms for Internet access control.**

***Index Terms*—Internet access control, Abuse reduction, Fairness, Network Traffic Modelling, Proxy server**

I. Introduction

Internet has become a vital source of information in today's world. It is a necessity in a broad spectrum of institutions right from schools and colleges and going on to huge commercial establishments. It is also in widespread use in our homes. A major contributing factor to the consumption of bandwidth (and consequent dissatisfaction on the part of many users) is the excessive use of bandwidth-hungry applications like P2P file sharing and videos [1] [2] by those addicted to Internet surfing. These constitute a minority of the users yet they consume most of the bandwidth [3] [4]. They need to be controlled by suitable bandwidth control policies during peak hours for a better overall user experience.

Traditional mechanisms for bandwidth management include scheduling, traffic shaping and congestion avoidance. None of these mechanisms maintain per-node or per-user state on their own, so they cannot be used to control a user with a history of abusive usage.

Many concepts have been proposed for fairness - examples are [5] [6] [7] [8]. Eventhough sharing Internet access seems to be an issue of fairness, we are really more interested in controlling abusive users, and overall user satisfaction. Further, even these approaches do nothing to penalize past abusive usage. [9] monitors the WAN access router periodically using SNMP, and uses TCP connection admission control for controlling usage. Their results indicate control of link usage and higher throughput after implementation of control; their main objective however is to control the link usage during peak times rather than controlling abusive usage. [3] maintains per-node state by obtaining (Netflow®) statistics for each node every 10 mins. They allocate a common quota for all nodes, and when this quota is exceeded, the packets are considered abusive traffic and stored on a low-priority queue which is served only when the regular queue becomes empty. In this way they are able to control abusive Internet access. The authors call this scheme quota-based priority control (QPC). [10] extends this work by combining QPC with time-of-day pricing (TDP), and obtains significant improvements in peak-hour usage and fairness, and load balancing across the time periods. In both these works, QPC uses a sampling interval of 10mins at the end of which usage statistics for each user are obtained. Also, these approaches do not distinguish between users sharing the same computer because they use Netflow® statistics.

We propose as an alternative a classifier-based mechanism that predicts the category of the user given his cumulative usage over the day. This eliminates the need for polling the routers at frequent intervals of time, preventing the consumption of local network bandwidth. Our solution operates at the level of the proxy server, so we have the added advantage of distinguishing between users sharing the same computer. We also characterize the behavior of both groups of users. These may then be used to apply control policies at the level of a proxy server or a router. In this paper, we restrict ourselves to the classification and characterization of users, and we leave the control policies for future work.

Several machine learning techniques are available to model and subsequently classify the data. In our work, we have used the Naive Bayes(NB) classifier and the Gaussian Mixture Model(GMM). These have been successfully used in the area of traffic analysis; while [11] and [12] use NB classifier to classify application traffic, [13] uses it to distinguish between chat and non-chat traffic and [14], [15] use it to distinguish between normal and anomalous emails. On the other hand,

978-1-4673-2393-2/12/$31.00

[16] and [17] use GMM to classify application traffic.

Classification models are much more promising in terms of classification accuracy when a suitable subset of features is selected. The idea is to select a subset of features that are highly correlated with the category, yet they are uncorrelated amongst themselves [18]. An example of this is the Max-Relevance Min-Redundancy (mRMR) approach for feature selection [19]. We make use of this approach to choose a subset of features which gives good classification accuracy.

This work represents a novel attempt on our part to predict the share of Internet usage based on features extracted from proxy server logs. Our analysis has been performed for the case of control of downstream bandwidth, as this is more often the bone of contention on campus LANs like ours. Section II describes our data set. Section IV discusses the bandwidth prediction and control framework that we have proposed. Section V and VI give a brief overview of the classifiers used (NB, GMM) and feature selection algorithm respectively. Section VII describes our experiments and results obtained. Finally Section V concludes the paper.

TABLE I: Features Extracted from Proxy Logs

Feature Type	Feature Name
URL Type	Social Networking, Video Hosting, Academics, Email, Web Search, Movies, News, Sports, Shopping, Travel, Sharing
Content Type	text, image, video, audio, compressed format, executables, javascript, real time audio/video, rss feeds, pdf, xml
HTTP Request Type	get, post, head, put, delete
Action	hit, miss, tcp denied
Size	different sizes
Statistical	total bytes downloaded, number of packets, number of hours of access, bytes per hour, average bytes per packet, variance of bytes per packet

II. Data Set Used

A proxy server acts as an intermediate node for requests from a client to servers across the Internet. The proxy server connects to the remote server on behalf of the client, and fetches the requested resource, which it then sends to the client. Typically, a proxy server on a LAN is used to perform the following tasks: (1) It keeps machines on the local network anonymous for security reasons. (2) It caches web pages, thereby improving response times, and reducing bandwidth consumption. (3) Performs content filtering through predefined rules. (4) Logs details of each request.

Proxy server logs contain much information about user access patterns, and thus may be used to train models for statistical classification. The complete Squid proxy log format can be found in [20]. The data set we used consists of one month of proxy server logs of a university campus which had a 90Mbps Internet access link, with 6906 users on the LAN. The following useful information pertaining to each request/response pair was contained in the logs

- Time Stamp "Unix time" (seconds since Jan 1, 1970)
- Client IP address

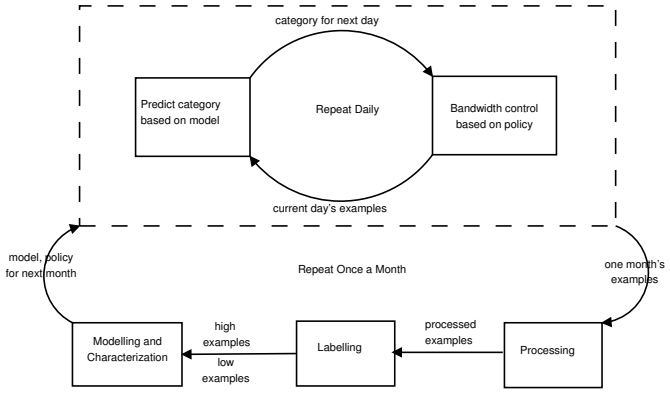


Fig. 1: Bandwidth Prediction and Control Framework

- Action/Code, describes how the request was treated locally
- Size. The amount of data sent back to the user in bytes
- Method, the HTTP request method
- URI, the requested URI
- Content, Content-type of the Object (from the HTTP reply header)

Each of the fields above can take on several different values which are listed in TABLE-I. Some of these values actually represent categories. For example 'social networking' includes all accesses to facebook, linkedin, twitter, and other social networking sites; 'academic' includes accesses to all academic institutions and 'mail' includes accesses to gmail, yahoo and other mail providers. Each of these has been considered as a feature for the purpose of classification. A bag of words model is used as a feature vector. Here each feature is an element of a vector, the number of occurences of the feature in a certain time period (day or week) corresponds to the value of the element. The last row of TABLE-I shows some statistical features recorded for each user during the day. Our proxy server disables P2P file sharing, which is why the torrent content type is missing from TABLE-I.

As a convention in this paper we refer to users who download from the Internet excessively as *high* users and those who download from the Internet moderately or rarely as *low* users. All our analysis has been performed for the downstream case as this is often the bone of contention on campus LANs like ours.

III. Bandwidth Prediction and Control Framework

The framework shown in Fig 1 has been proposed to predict and control Internet usage. At the end of every month, daily features are extracted from proxy logs, which are then processed, labelled, modelled and characterized using suitable mechanisms (discussed below). The prediction step is where the classifier is used in order to classify the users into high and low categories, and is done on a daily basis using cumulative daily or weekly statistics. The output user categories and policies are given as input to the bandwidth control algorithm to control the users. Each of these steps are discussed below.

As mentioned before, the bandwidth control algorithm is not the subject of the current research work.

A. Processing

As mentioned before, each feature F_i can take on values f_i which are basically numerical counts for feature i for a particular user during the day. These counts may be divided by the total number of packets to obtain ratios. The ratios r_i are computed as

$$r_i = \frac{f_i}{P}$$

where P is the number of packets downloaded by the user during the day. We will refer to these two types of data as 'counts' and 'ratios'. The advantage of using ratios is that they provide the percentage of use of each feature thus reflecting user behavior better. On the downside, because they factor out the number of packets, a user with low usage may also have the same pattern of usage as another with high usage.

For both counts as well as ratios we apply the natural logarithm, as the resulting data resembles the gaussian probability distribution more closely. Then, we convert the data to z-scores i.e. make the mean 0 and variance 1:

$$\frac{log(f_i) - E(log(F_i))}{\sqrt{V(log(F_i))}}$$

In the following discussion, 'examples' refers to a feature vector consisting of z-scores computed through the above steps.

B. Labelling

Labelling is the task of ground truthing i.e. which examples belong to the high category, and which ones belong to the low category. The models for the two categories are then trained and tested based on this division. As mentioned before, users are categorized by the share of bandwidth, and this is done by fixing a threshold on this value; those with share above this threshold would then fall into the high category, and those below would fall into the low category. Note should be made here that this share value is used only for labelling and not for classifying examples; the reason is that the computation of this share value requires polling of devices with a small sampling interval (as will be seen later in this section) or alternatively computation from proxy logs; while the problem with the former is the small sampling interval, the latter approach is time consuming and not suitable on a daily basis.

The easiest way to compute the share is

$$s = \frac{b_u}{tot}$$

where b_u is the number of bytes downloaded by a user u during a day, and tot is the total number of bytes downloaded by all users during that day. Note that a share value on a day with large number of users should not be treated on par with the same share value on a day with small number of users; on a day with small number of users, a user is eligible for a higher share value. So, the share s is not comparable across the days as such. The metric proposed for normalization across the days is called the usage fairness index φ_u:

$$\varphi_u = \frac{b_u}{\frac{tot}{N}} \tag{1}$$

where N is the number of Internet users during the day. In other words, this is the ratio between the bytes downloaded by a user and the number of bytes he is eligible to download under the case of equal sharing between the users. However, simply counting the total number of Internet users during the day and substituting in place of N would give an inaccurate estimate of φ_u. This is because, a user who accesses the Internet for only 5mins cannot be treated on par with the one who uses it for hours on end. Therefore, the number of users is counted by taking into account the time spent on the Internet; if T_{avg} represents the average time spent in a day by an average user on the Internet, a user i contributes an amount

$$\frac{T_i}{T_{avg}}$$

to the number of users. Thus, we compute effective number of users N_{eff} as

$$N_{eff} = \frac{\sum_i T_i}{T_{avg}} \tag{2}$$

Substituting N_{eff} in (2) in place of N in (1) we get

$$\varphi_u = \frac{b_u}{tot} * \frac{\sum_i T_i}{T_{avg}}$$

The value of T_i can be approximated by breaking up the time into 5-minute intervals. In that case T_i would be the number of slots during which user i accessed the Internet, and T_{avg} would represent the average number of slots during which an average user accessed the Internet. φ_u is calculated for each user u for each day. Based on this value, the data is divided into high and low usage classes by choosing a threshold. This shows how usage fairness index φ_u can be used to label daily data. It is evident that computation of φ_u requires knowledge of the users who accessed the Internet in each 5-minute time slot during the day, and if it were to be recorded by network management it would require a polling interval of 5 minutes. This we avoid, and compute the share values only at the end of the month from the proxy logs.

This same concept of fairness index may be extended for time durations of a week. Weekly data may be labelled using the weekly usage fairness index. Note that this labelling is purely based on usage fairness index for a particular time duration (day or week), and is indifferent to the identity of the user. We call this *usage-based labelling*.

The usage fairness index may also be computed for time durations of a month. The monthly usage fairness index helps us to categorize the users based on their usage over the entire month. The monthly usage fairness index is assumed to be indicative of user habits i.e. a user with high value of monthly usage fairness index is a habitual high user of the Internet, and similarly for a low user. So, this helps us to categorize the users. Subsequently all the data of high users are placed

TABLE II: Usage and User based labelling

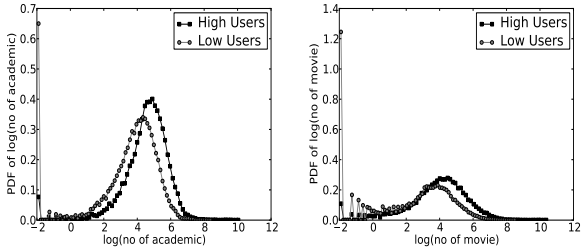
Labelling Type	Duration	Labelling value
Usage-Based	daily	That day's usage fairness index for that user
	weekly	That week's usage fairness index for that user
User-Based	daily	That month's usage fairness index for that user
	weekly	That month's usage fairness index for that user

in one class and all the data of low users are placed in another class. This is called *user-based labelling*, and it can be used to characterize the behavior of the high and low users as we shall see later. TABLE-II gives details of both these approaches to labelling.

In the ensuing discussion we will refer to examples categorized using usage-based labelling as *usage examples*, and the groups resulting as a result of this categorization as *usage groups*. Similarly, we will refer to examples categorized using user-based labelling as *user examples*, and the groups resulting as a result of this categorization as *user groups*.

C. Modelling and Prediction

Models may be built assuming that the data has a unimodal or a multi-modal probability distribution function. In either case Gaussian distributions have been assumed. Prediction has been performed using classifiers. Whereas an NB classifier has been used for the unimodal case, a GMM has been used to model and classify for the multi-modal case. Fig 2 shows the histograms for the academic hits and movie hits among the two categories of user examples. It shows two peaks, which means there may be different behaviors exhibited among the same category of user examples. We use this intuition as justification for trying a multi-modal model.



(a) Academic hits (b) Movie hits

Fig. 2: Histogram for some selected features

IV. Classifiers Used

Given a class variable C which can take values c_i, n feature variables F_i, which can take corresponding (continuous) feature values f_i, actual probabilities $P(C = c_i|F_j = f_j)$ can be represented as simply $P(c_i|f_j)$ and pdfs are represented as $p(.)$. It follows from Baye's rule that

$$P(c|f_1, f_2, ...f_n) = \frac{P(c)p(f_1, f_2, ..., f_n|c)}{p(f_1, f_2, ...f_n)}$$

If it is assumed that each of the features f_i are independent of the others, then

$$p(f_1, f_2, ..., f_n|c) = \prod_i p(f_i|c)$$

So,

$$P(c|f_1, f_2, ...f_n) = \frac{P(c)}{p(f_1, f_2, ...f_n)} \prod_i p(f_i|c)$$

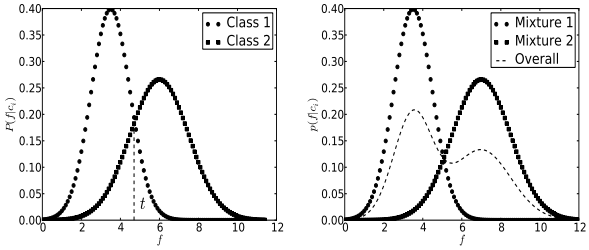
This is the *Naive-Bayes probabilistic model*. The denominator is constant for all the classes, and under the assumption that $P(c)$ is the same across all the classes, then feature vectors can be classified as

$$classify(f_1, f_2, ...f_n) = \underset{i}{argmax} \prod_j p(f_j|c_i)$$

This is called the *Naive-Bayes classifier* and has been found to be simple yet efficient as a classifier. If we assume a gaussian probability distribution for the features, then

$$p(f_j|c_i) = \frac{exp(-\frac{1}{2*\sigma_{ij}^2}(f_j - \mu_{ij})^2)}{\sqrt{2\pi}\sigma_{ij}}$$

where μ_{ij} and σ_{ij}^2 represent the mean and variance for feature j with respect to class i respectively. Fig 3a illustrates the classification for a univariate case. The two curves are the probability distributions of the feature with respect to two classes. So, on application of the above mechanism, all values above the threshold t are classified with one class, and all values below it are classified with the other.



(a) Naive Bayes (b) Mixture Modelling

Fig. 3: Classification Approaches

The efficacy of the above model is subject to the validity of the assumption that each feature follows a unimodal gaussian distribution. If the features exhibit multiple modes, it may be more suitable to use a *Gaussian Mixture Model*. Fig 3b shows (for a univariate case) a curve with two modes, and also the two modes it can be represented by. In this case, the likelihood of an example $\bar{x} = (f_1, f_2, ...f_n)$ belonging to a class c can be obtained by extension from Baye's rule as

$$p(\bar{x}|c) = \sum_{m=1}^{M} w_m \ p(\bar{x}|\Phi_m)$$

where Φ_m represents mixture m of class c, w_m represents the mixing weight corresponding to the mixture m which is

also the prior probability for the mixture m and M is the total number of mixtures for each class. As each mixture is represented by its mean vector and covariance matrix, $\Phi_m = (\bar{\mu}_m, \Sigma_m)$. Following from the assumption of a multi-variate gaussian for each mixture, $p(\bar{x}|\Phi_m)$ can be written as

$$p(\bar{x}|\Phi_m) = \frac{exp(-\frac{1}{2}(\bar{x} - \bar{\mu}_m)^t {\Sigma_m}^{-1}(\bar{x} - \bar{\mu}_m))}{(2\pi)^{D/2}\,|\Sigma_m|^{1/2}} \quad (3)$$

where $\bar{\mu}_m$ is the mean of mixture m, Σ_m represents the covariance matrix for mixture m and D is the dimension of the example $\bar{x}$. The parameters of the model are $\{\, w_m \;\; \bar{\mu}_m \;\; \Sigma_m \,\}, (1 \leq m \leq M)$, and are computed by an iterative process of Expectation Maximization (EM) which seeks to maximize the likelihood of the data given the model. The covariance matrices Σ_m are assumed to be diagonal matrices to reduce the number of parameters to be computed and thereby make for easier computation. In this work the Universal Background Model (UBM) approach has been used and the means $\bar{\mu}_m$ only adapted [21]. The UBM is a GMM for the entire data put together with examples from both the classes in the same proportion as the original data. This UBM is then adapted based on training examples from high class to obtain the GMM for high examples, and the GMM for low examples is obtained likewise by adapting the UBM using examples from low class.

V. Max-Relevance Min-Redundancy Feature Selection

In this paper mutual information has been used as a measure of predictability of one feature by another:

$$I(X;Y) = H(X) - H(X|Y)$$

where $H(X)$ is the entropy of feature X, $H(X|Y)$ is the entropy of feature X given Y and $I(X;Y)$ is the mutual information between features X and Y. The intuitive meaning of entropy $H(X)$ is uncertainty in the value of random variable X. So, $H(X|Y)$ is the uncertainty in the value of random variable X given the value of another (possibly related) random variable Y. Therefore, it follows that $I(X,Y)$ is the reduction in uncertainty of random variable X given another random variable Y, and can be regarded as a measure of predictability. We make use of the mechanism in [19] which uses mutual information for feature selection. The steps involved are as follows (explained later):

1) Use Max-Relevance Min-Redundancy(mRMR)to select n sequential feature sets $S_1 \subset S_2 \subset S_3 ... S_{n-1} \subset S_n$ where n is the number of features.
2) Select a candidate feature set S_m where $1 \leq m \leq n$ from the sets S_i.
3) Apply forward selection algorithm to obtain a set of features $S^* \subset S_m$ with better classification accuracy.

Each of the above is explained below.

A. Max-Relevance Min-Redundancy

The aim is to select features that can predict the class well, while at the same time have low redundancy amongst themselves.The latter condition is required in order to obtain better classification accuracy. Therefore a set of features S is selected so as to maximize

$$max\, \phi = D - R$$

where (relevance)

$$D = \frac{1}{|S|} \sum_{F_i \in S} I(F_i; C)$$

and (redundancy)

$$R = \frac{1}{|S|^2} \sum_{F_i, F_j \in S} I(F_i; F_j)$$

Features are added incrementally to the set S so as to maximize ϕ thus obtaining the sequence of subsets $S_1 \subset S_2 \subset S_3 ... S_{n-1} \subset S_n$.

B. Candidate Feature Set

Selection of candidate feature set involves the following two steps:

1) Find a range $[p, q] \subset [1, n]$ such that for all $i \in [p, q]$ the respective cross-validation classification error e_i corresponding to S_i is consistently small.
2) Within $[p, q]$ find the smallest classification error $e^* = min_{k \in [p,q]} e_k$. The optimal size of the candidate feature set, n^*, is chosen as the smallest k that corresponds to e^*.

Fig 4 shows how the candidate feature set is chosen. $[lb, ub]$ is the range in which the classification error is consistently small. n^* is chosen from this range as the cardinality with minimum error e^*.

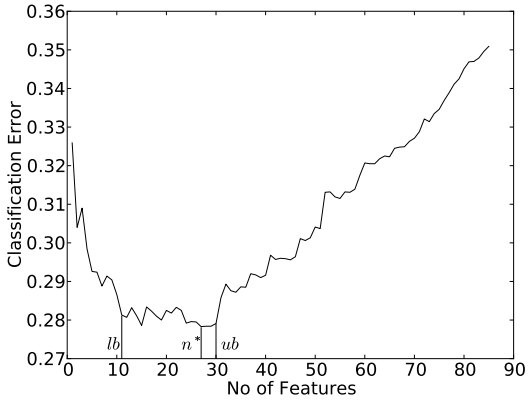


Fig. 4: Selection of Candidate Feature Set

C. Forward Selection Algorithm

The goal of the forward selection algorithm is to minimize the classification error of a particular classifier (in our case Naive Bayes). Therefore a wrapper approach has been used. The wrapper first searches for feature subsets with one feature, denoted as Z_1, that leads to the largest error reduction; This feature is denoted as F_1^*. Then, it selects a feature F_2^* from $S_{n^*} - Z_1$ so that the two feature set $\{F_1^*, F_2^*\}$ leads to the largest error reduction. This repeats until the classification error begins to increase, at which point we have found a subset S^* with better classification accuracy.

TABLE III: Execution Times for Various Steps

Action		Step no.	Execution Time
Extraction of features from proxy logs		1	1 day
Processing		2	26 secs
Labelling		3	1 min
GMM	PCA	4	8 secs
	Modelling	5	25 mins
	Classification	6	20 mins
NB+mRMR	MI Computation	4	1 day
	mRMR feature ranking	5	1 sec
	Candidate Feature Set	6	6 mins
	Forward Selection	7	a few hours
	Modelling	8	30 secs
	Classification	9	15 secs

VI. Experiments

Proxy server logs for the month of January 2012 were used for the experiments. Cumulative values - daily and weekly counts and ratios over the day were extracted from the logs for each user; weekly counts were averaged over a window of seven days. Simultaneously, the usage fairness index was computed for each user and stored. The log function was applied to this data, after which zscores were computed. The data was clustered using k-means clustering algorithm into 2 groups based on the usage fairness index. This division into groups constitutes the labelling for the data. Both usage-based as well as user-based labelling were performed on the data to obtain usage examples and user examples respectively. For the unimodal model mRMR feature selection was used to select features to obtain subsets with optimal classification accuracies; NB classifier was used in conjunction with mRMR. For the multi-modal model, after computing z-scores, PCA was performed in order to map the data to a space in which the dimensions are uncorrelated with one another. This is especially useful as our GMMs assume diagonal covariance matrices. TABLE-III shows approximate execution times for different steps in our framework on an octa-core 3.0 GHz 64-bit AMD CPU with 32GB RAM . Clearly GMMs take much less time because we have not used the feature selection algorithm in conjunction with GMMs. We also note that the time consuming steps like feature extraction, MI computation and forward selection are all done only once a month offline; while the classification step which is done every day may be performed swiftly. This makes our system practical.

In all our experiments, while estimating classification error, 4-fold cross validation was used. The classification error was determined as

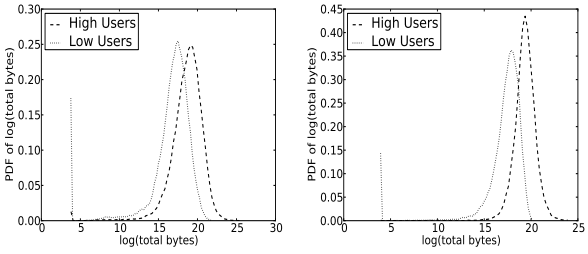
$$\frac{acc_h + acc_l}{2}$$

where acc_h is the accuracy for high examples and acc_l is the accuracy for low examples. This was done in order to give equal importance to classifying both the categories. The intuition is that while it is important not to classify a low example as a high example (so that the service for the former is not affected during bandwidth control), it is equally important to control the high examples (so that they can be kept within the limits).

TABLE-IV shows the classification accuracy obtained when we use a Naive Baye's classifier to classify usage examples or user examples based on the total bytes accumulated (assuming a unimodal probability distribution). Note that even though usage-based labelling has the smallest classification error, it is still substantial (around 13%); user-based labelling has a much higher classification error. This means that if labelling had been done using the total bytes downloaded, there would have been a substantial error in the labelling; in this way we justify our use of the usage fairness index instead to label the data. Also note that in the case of user-based labelling the daily data has a larger error than the weekly data. This is due to the larger degree of overlap between the two groups of users over the total bytes downloaded in a day (See Fig 5). This may also be understood intuitively - the weekly total of bytes is less prone to fluctuation (and thus more stable) for a particular user group than the daily value. This leads to lesser overlap between the two groups, which in turn leads to better classification.

TABLE IV: Class. Accuracy of Total Downloaded Bytes

Labelling	Duration	Classification Accuracy
Usage-Based	daily	86.85%
	weekly	84.38%
User-Based	daily	62.93%
	weekly	73.30%



(a) Daily data (b) Weekly data

Fig. 5: PDFs of total bytes downloaded among user categories

TABLE-V and TABLE-VI list the classification accuracies for NB classifier and GMM respectively. Results indicate that

TABLE V: Classification Accuracy for NB Classifier

Labelling	Duration	Type of Data	Class. Accuracy
Usage-Based	daily	counts	91.07%
		ratios	88.01%
	weekly	counts	87.6%
		ratios	84.43%
User-Based	daily	counts	70.42%
		ratios	69.57%
	weekly	counts	80.07%
		ratios	76.06%

TABLE VI: Classification Accuracy for GMM

Labelling	Duration	Type of Data	Modes	Class. Accuracy
Usage-Based	daily	counts	2	93.16%
			4	92.87%
		ratios	2	94.96%
			4	94.5%
	weekly	counts	2	85.7%
			4	85.37%
		ratios	2	86.36%
			4	84.87%
User-Based	daily	counts	2	65.44%
			4	65.98%
		ratios	2	66.67%
			4	66.48%
	weekly	counts	2	84.07%
			4	81.5%
		ratios	2	83.83%
			4	83.25%

in general, usage-based labelling gives better classification than user-based labelling. This is because in the latter case, examples belonging to the two groups are in many cases indistinguishable, and therefore, there exists a larger degree of overlap; although, as we have seen earlier for the total bytes feature (Fig 5), weekly data is more stable and well-separated, giving better accuracies than daily data.

Note must be made from the table that in general mixture models give better classification that unimodal models. This is due to the presence of different groups within each category which are better modelled using mixture models. TABLE-VIII and TABLE-IX show the behavior of groups of (daily) usage and (weekly) user examples respectively. These groups correspond to each mixture of the GMM. The mean of the examples belonging to the mixture gives us the average behavior of the group in terms of Internet accesses; the weights of each mixture tell us what percentage of examples in that category belong to that group. Note the same percentage-wise distribution among both categories into groups. This arises because of the use of the UBM coupled with the adapt-means-only approach [21]. The meanings of the codes in the table are listed in TABLE-VII. Since the z-score is obtained by subtracting the mean and dividing by the square root of variance, it is suitable to estimate the 'bigness' of a value and the other advantage is it can be compared across the features.

TABLE-VIII indicates that in the case of (daily) usage examples there are roughly 4 groups, 2 in each category. Within the high category, there are examples with a large number of accesses to most categories of web-sites; there are also examples with a small number of web accesses, but whose average packet size (as indicated by the 'AVGBPP' feature) is large. This large average packet size arises because of a relatively large number of accesses to video hosting sites, with relatively small number of packets downloaded during the day. Similarly, among the low usage examples there are cases of overall low number of web accesses (group 4) and others with large number of web accesses, but whose average packet size is small (group 3).

TABLE-IX shows the behavior of different groups of (weekly) user examples. It indicates different patterns of usage based on number of accesses to academic and entertainment sites. Both groups of high user examples have high access to video hosting, movie and social networking sites. Group 3 consists of low users who exhibit a preference for academic over entertainment sites; group 4 consists of low users who have overall low usage for all URL categories.

TABLE VII: Explanation of Extent of Usage

Usage Code	Expansion	Z-score Range
vl	very low	upto -0.5
l	low	-0.5 to 0.0
h	high	0 to 0.5
vh	very high	more than 0.5

TABLE VIII: Groups of Daily Usage

	High Usage Groups		Low Usage Groups	
Feature	1	2	3	4
	57%	43%	57%	43%
NUMACCHRS	vh	h	h	vl
NUMPACKETS	vh	l	h	l
MAIL	h	l	h	l
ACADEMIC	h	l	h	l
SHARING	h	l	h	l
VIDHOST	vh	h	h	l
MOVIE	vh	l	h	l
SHOPPING	h	l	h	l
AVGBPP	h	vh	l	vl
BYTES	vh	vh	h	vl

TABLE IX: Groups of Users

	High Users Groups		Low Users Groups	
Features	1	2	3	4
	66%	34%	66%	34%
MAIL	h	h	h	vl
ACADEMIC	h	l	h	vl
SHARING	h	l	h	vl
SEARCH	h	h	h	vl
VIDHOST	vh	h	l	vl
MOVIE	h	h	l	vl
SOCNET	h	h	h	vl
NEWS	h	l	h	vl
GAMING	h	l	l	l
SHOPPING	h	l	h	vl
SPORT	h	l	h	vl

Fig 6 shows the percentage of bandwidth consumed due to access to different URL categories. It indicates that accesses to video hosting sites consume the most bandwidth.

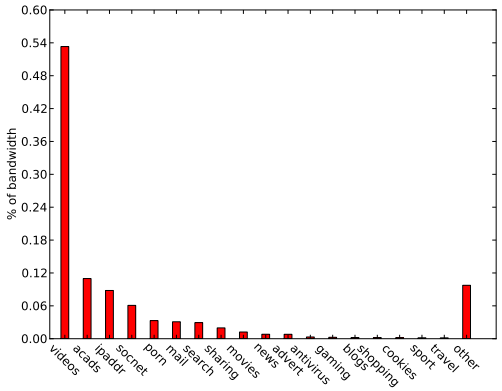


Fig. 6: Bytes downloaded from different URLs

Next, based on the categorization of users into high and low using the monthly usage fairness index (user based labelling), a set of 8 million examples randomly chosen from the month were profiled. The results as illustrated by Fig 7 shows the proportion of different features accessed by the high and low users. Fig 7a indicates that high users access leisure sites like movies, video hosting, pornography, shopping and gaming more than low users. This also agrees with the profiles of the high user examples we obtained using GMMs (TABLE-IX). Low users, on the other hand have a greater tendency to access the more serious sites like social networking, search, academics, sharing and sport. In the same vein, as shown in Fig 7b high users display a greater tendency for video, shockwave (gaming applications) and real-time streaming (xfcs) file types, whereas, low users are more inclined to request documents (xml, pdf), search related file-types (safbrw - google returns this content type which contains a list of web-sites safe for browsing), file-types for authentication of secure web-sites(ocsprsp). Fig 7e indicates a greater tendency on the part of low users to use a connect request; a connect request is normally made when authenticating oneself with a password, and uses SSL/HTTPS. As high users access a higher percentage of video content-type, they have a larger percentage of downloads of large HTTP packets (>1MB), as shown in Fig 7c.

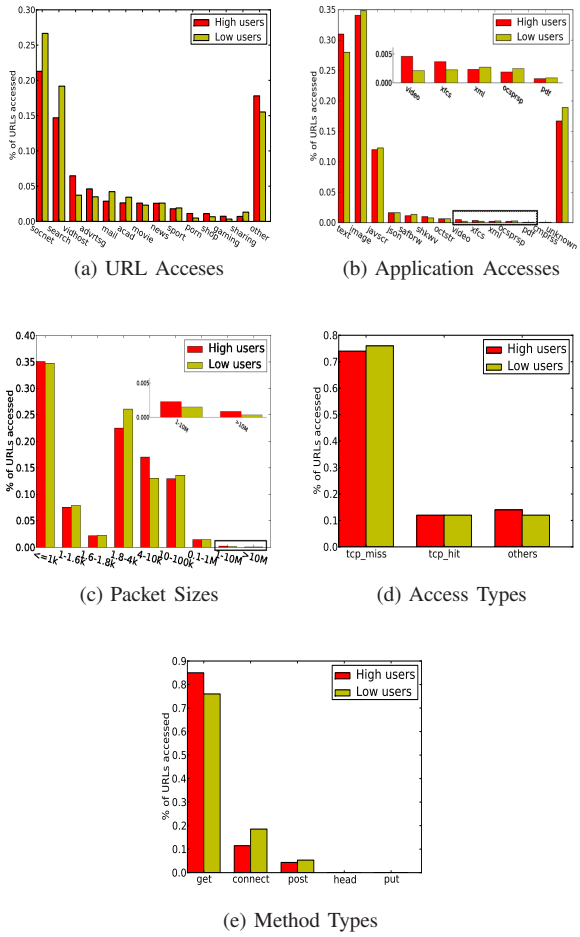


Fig. 7: Proportion of different types of accesses for High and Low users

VII. Conclusion

In this paper we proposed a framework for the control of high bandwidth users. Our solution is unique in that it is proxy-server based, so we have the capability of handling users who share the same computer. Our scheme is also unique in that we have used a classifier to predict a user's usage; so we need only cumulative daily statistics, thereby eliminating the need to poll devices frequently. We used two classification models - NB+mRMR and GMMs, of which GMM was found to be faster yet more accurate, with accuracies of upto 94.96% for the classification of usage examples. We found the weekly data to be more suitable for classifying user examples, with accuracies upto 84.07%. We were also able to make use of user-based labelling in order to characterize user behavior, and in our case we found that almost 50% of the bandwidth is consumed by accesses to video hosting sites. We also found that (in our campus network) high users access leisure sites like movies, video hosting, shopping and gaming more than low users. These results may in turn be used to set policies for users and devise algorithms to control Internet access. Therefore, we do believe that usage-based labelling and user-based labelling are complementary and useful approaches. While usage-based labelling may be used to build classification models for daily usage for the next month, user-based labelling may be used to characterize user behavior, and in this way devise control policies for Internet access for the next month. Our system is also practical as time consuming tasks are only performed once a month while daily tasks can be performed swiftly. Our results may be further improved by trying other models for classification.

Acknowledgment

This work was carried out under the IU-ATC project funded by the Department of Science and Technology (DST),

Government of India, and the UK EPSRC Digital Economy Programme.

The authors would like to acknowledge the use of computational resources at the High Performance Computing Facility at IIT Madras for this research work.

REFERENCES

[1] M. Kihl, C. Lagerstedt, A. Aurelius, and P. Odling, "Traffic analysis and characterization of internet user behavior," in *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Oct. 2010, pp. 224–231.

[2] R. Pries, F. Wamser, D. Staehle, K. Heck, and P. Tran-Gia, "Traffic measurement and analysis of a broadband wireless internet access," in *Proc. of IEEE 69th Vehicular Technology Conference*, Barcelona, Spain, Apr. 2009, pp. 1–5.

[3] T.-C. Lin, Y. Sun, S.-C. Chang, S.-I. Chu, Y.-T. Chou, and M.-W. Li, "Management of abusive and unfair internet access by quota-based priority control," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 44, pp. 441–462, Mar. 2004.

[4] K. Bommepally, T. Glisa, J. Prakash, S. Singh, and H. Murthy, "Internet activity analysis through proxy log," in *National Conference on Communications (NCC)*, Jan. 2010.

[5] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication: shadow price proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, pp. 237–252, 1998.

[6] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[7] L. Massoulie and J. Roberts, "Bandwidth sharing: objectives and algorithms," in *INFOCOM '99*, 1999.

[8] Y. Zheng and Z. Feng, "A new fairness criterion and its realization by using a new scheduling algorithm in the internet," in *Proc. of Sixth IEEE Symposium on Computers and Communications*, Nagoya, Japan, 2001, pp. 444–449.

[9] A. Kumar, M. Hegde, S. Anand, B. Bindu, D. Thirumurthy, and A. Kherani, "Nonintrusive TCP connection admission control for bandwidth management of an internet access link," *IEEE Commun. Mag.*, vol. 38, pp. 160–167, May 2000.

[10] S.-I. Chu and S.-C. Chang, "Time-of-day internet-access management by combining empirical data-based pricing with quota-based priority control," *IET Communications*, vol. 1, pp. 587–596, Aug. 2007.

[11] A. Moore and D. Zuev, "Internet traffc classifcation using bayesian analysis techniques," in *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, Banff, Canada, Jun. 2005, pp. 50–60.

[12] T. Nguyen, G. Armitage, P. Branch, and S. Zander, "Timely and continuous machine-learning-based classification for interactive IP traffic," *IEEE/ACM Trans. Netw.*, vol. PP, 2012.

[13] C. Livadas, R. Walsh, D. Lapsley, and W. Strayer, "Using machine learning technliques to identify botnet traffic," in *Proc. of 31st IEEE Conference on Local Computer Networks (LCN)*, Tampa, USA, Nov. 2006, pp. 967–974.

[14] M. Marsono, M. Kharashi, F. Gebali, and S. Ganti, "Distributed layer-3 email classification for spam control," in *Canadian Conference on Electrical and Computer Engineering(CCECE)*, May 2006, pp. 742–745.

[15] U. Premarathne, U. Premaratne, and K. Samarasinghe, "Network traffic self similarity measurements using classifier based hurst parameter estimation," in *5th International Conference on Information and Automation for Sustainability (ICIAFS)*, Dec. 2010, pp. 64–69.

[16] M. Dusi, A. Este, F. Gringoli, and L. Salgarelli, "Using gmm and svm-based techniques for the classification of ssh-encrypted traffic," in *Proc. of IEEE International Conference on Communications (ICC)*, Dresden, Germany, Jun. 2009.

[17] V. Panchamukhi and H. Murthy, "Port-based traffic verification as a paradigm for anomaly detection," in *National Conference on Communications (NCC)*, Feb. 2012.

[18] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003.

[19] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 1226–1238, Aug. 2005.

[20] (2012) Squid proxy server. [Online]. Available: http://www.squid-cache.org

[21] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, Jan. 2000.