Traffic Classification Technique in Computer Networks

S. M. Parvat[1], Prof. Dr. S. D. Lokhande[2]

Department of E & TC, Sinhgad college of Engineering, Pune, India

Swati.parvat@gmail.com[1]

l_shashikant@yahoo.co.in[2]

***Abstract*- Traffic classification enables a variety of applications and topics, including Quality of Service, security, monitoring, and intrusion-detection that are of use to researchers, accountants, network operators and end users. Capitalizing on network traffic that had been previously hand-classified provides with training and testing data-sets. The classification of network traffic can be done using Machine Learning Method, for this the use of simulating tools like NS2 can be used. It requires network protocol headers and the properties of unknown traffic for a successful classification stage.**

***Keywords*— Machine Learning (ML), Internet Protocol (IP), Network Simulator version 2 (NS2).**

I. Introduction

Accurate traffic classification is of fundamental importance to numerous other network activities, from security monitoring to accounting, and from Quality of Service to providing operators with useful forecasts for long-term provisioning. A traffic classifier that can achieve a high accuracy across a range of application types without any source or destination host-address or port information is presented in this report. A supervised machine learning based on a Bayesian trained neural network is used here.

Traffic classification is a process of identifying network traffic based on the features that can be passively observed in the traffic. The features and classification results may vary according to specific classification requirements and analysis needs. In early days, traffic classification was performed as part of traffic characterization work, often motivated by the dominance of a certain protocol in a network. The previous studies have discussed various classification methodologies (e.g., well-known port number matching, payload contents analysis, machine learning, etc.). Many variants of such methodologies have been introduced continuously to improve the classification accuracy and efficiency. However, it is extremely difficult for any method to claim 100 percent accuracy due to fast-changing and dynamic nature of the Internet traffic.

There is a growing need for accurate and timely identification of networked applications based on direct observation of associated traffic flows. Also referred to as 'classification', application identification is used for trend analysis (estimating capacity demand trends for network planning), adaptive network based Quality of Service (QoS) marking of traffic, dynamic access control (adaptive firewalls that detect forbidden applications or attacks) or lawful interception.

Classification based on well-known TCP or UDP ports is becoming increasingly less effective – growing numbers of networked applications are port-agile (allocating dynamic ports as needed), end users are deliberately using non-standard ports to hide their traffic, and use of network address port translation (NAPT) is widespread (for example a large amount of peer-to peer file sharing traffic is using non-default ports).

Payload-based classification relies on some knowledge about the payload formats for every application of interest: protocol decoding requires knowing and decoding the payload format while signature matching relies on knowledge of at least some characteristic patterns in the payload. This approach is limited by the fact that classification rules must be updated whenever an application implements even a trivial protocol change, and privacy laws and encryption can effectively make the payload inaccessible.

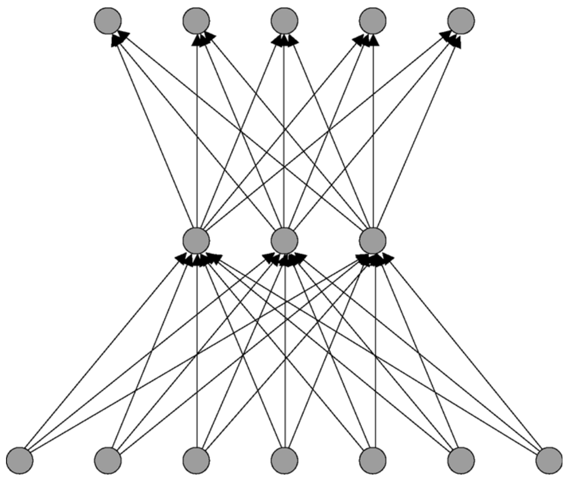
Fig. 1 Example of neural networks: Perceptron neural network with one hidden layer

Machine learning (ML) techniques provide a promising alternative in classifying flows based on application protocol (payload) independent statistical features such as packet length and inter-arrival times. Each traffic flow is characterized by the same set of features but with different feature values. A ML classifier is built by training on a representative set of flow instances where the network applications are known. The built classifier can be used to determine the class of unknown flows.

An assumption underlying such methods is that traffic at the network layer has statistical properties such as Packet lengths, packet inter-arrival time, Inter-Packet lengths, Packet arrival order, Fourier transform of packet inter-arrival time, Average inter packet gap that are unique for certain classes of applications and enable different source applications to be distinguished from each other [1]. These Statistical properties are also known as Features or Attributes or Discriminators.

The contributions of Neural Networks in this classification can be listed as following [2]:

- Illustration of the Bayesian framework using a neural network model that allows identification of traffic without using any port or host [Internet protocol (IP) address] information;
- A classification accuracy of over 99% when training and testing on homogeneous traffic from the same site on the same day;
- A classification accuracy of 95% in the situation of training on one day of traffic from one site, and testing on traffic from that site for a day eight months later.

Recent research in the area of traffic classification has brought many interesting methods that can identify applications without relying on "well-known" port numbers, or looking at the contents of packet payloads. Most of these newer schemes classify traffic by recognizing statistical patterns in externally observable characteristics, e.g., For a given flow identified by the (src IP, src port, dst IP, dst port, proto).

Machine learning algorithms map instances of network traffic flows into different network traffic classes. Each flow is described by a set of statistical features and associated feature values. A feature is a descriptive statistic that can be calculated from one or more packets – such as mean packet length or the standard deviation of inter-arrival times. Each traffic flow is characterized by the same set of features, though each will exhibit different feature values depending on the network traffic class to which it belongs. System memory usage increases with the number of instances in the training/testing set [3], as does CPU usage. As the public trace files used in this study contain millions of network flows, we perform flow sampling to limit the number of flows and therefore the memory and CPU time required for training and testing [7].

A number of data sets are described; each data set consists a number of objects, and each object is described by a group of features (also referred to as discriminators). Leveraged by a quantity of hand-classified data, each object within each data set represents a single flow of TCP packets between client and server. The features for each object consist of the (application-centric) classification derived elsewhere and a number of features derived as input to probabilistic classification techniques [4].

These data sets are available and can be observed at [8], [9].

II. Internet Traffic Classification

A. *The importance of IP traffic classification*

Understanding traffic behavior is an important part of computer network operations and management. A decade of research on traffic classification has provided various techniques to identify types of traffic information. As the Internet continuously evolves in scope and complexity, its traffic characteristics are also changing in terms of traffic composition and volume. Peer-to-peer (P2P) and multimedia traffic applications have rapidly grown in popularity, and their traffic occupies a great portion of the total Internet traffic volume these days. P2P applications generate a substantial volume in enterprise networks. In 2008, a study by a Japanese Internet service provider (ISP) observed that a significant portion of P2P traffic is recently being replaced by multimedia and web traffic. In particular, a newer generation of P2P applications is incorporated with various obfuscation strategies, such as ephemeral port allocation and proprietary protocols, to avoid detection and filtering. A popular communication application like Skype eludes detection by payload encryption or plain-text ciphers. The dynamic nature of Internet traffic adversely affects the accuracy of traffic classification and makes it a more challenging task. The previous studies have discussed various classification methodologies (e.g., well-known port number matching, payload contents analysis, machine learning, etc.). Many variants of such methodologies have been introduced continuously to improve the classification accuracy and efficiency. However, it is extremely difficult for any method to claim 100 percent accuracy due to fast-changing and dynamic nature of the Internet traffic. The classification accuracy is also questionable since there is often no ground truth dataset available. In another respect, each research aims at different levels of classification. Some only had a coarse classification goal such as classifying traffic protocol or application type; while others had more detailed classification goal such as identifying the exact application name. Therefore, it is often unfair to cross-compare each classification method in terms of accuracy. To overcome this issue, we need to investigate how we can provide more meaningful information with such limited traffic classification results rather than focusing on improving 1 or 2 percent of classification accuracy.

B. *Types of IP traffic*

Table I lists the classes of applications. Each traffic class contains a variety of different types of data. Internet traffic can be of many types out of which some are listed as below:

- P2P

- VoIP
- Multimedia
- Gaming
- Video sharing
- Bulk data (FTP) traffic
- WWW
- HTTP

Table I

Example of Network Traffic Allocated to each category

Classification	**Example Application**
Bulk	ftp
Database	Postgres, sqlnet oracle
Interactive	telnet
Mail	Pop2/3, smtp
Services	X11, dns, idap
Attack	Internet worm and virus attacks
Games	Online games, Microsoft direct play
Multimedia	Windows media player, real player

C. Traffic Recognition Techniques

There are following existing techniques which are commonly used for classifying internet traffic.

- Port Number based approach
- Payload Inspection based approach
- Statistical properties based approach

D. Limitations

The limitations of the first two Port Number based approach and Payload Inspection based approach are discussed in introduction chapter of this report.

E. Classification based on statistical traffic properties

The preceding techniques are limited by their dependence on the inferred semantics of the information gathered through deep inspection of packet content (payload and port numbers). Newer approaches rely on traffic's statistical characteristics to identify the application. An assumption underlying such methods is 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 source applications to be distinguished from each other such as models of connection characteristics - such as bytes, duration, arrival periodicity - for a number of specific TCP applications), and in (where the authors analyzed Internet chat systems by focusing on the characteristics of the traffic in terms of flow duration, packet inter-arrival time and packet size and byte profile). Later work also observed distinctive traffic characteristics, such as the distributions of packet lengths and packet inter-arrival times, for a number of Internet applications.

The results of these works have stimulated new classification techniques based on traffic flow statistical properties. The need to deal with traffic patterns, large datasets and multi-dimensional spaces of flow and packet attributes is one of the reasons for the introduction of ML techniques in this field.

ML has historically been known as a collection of powerful techniques for data mining and knowledge discovery, which search for and describe useful structural patterns in data. Machine learning is the study of making machines acquires new knowledge, new skills, and reorganize existing knowledge.' A learning machine has the ability to learn automatically from experience and refine and improve its knowledge base. ML has a wide range of applications, including search engines, medical diagnosis, text and handwriting recognition, image screening, load forecasting, marketing and sales diagnosis, and so on. Input and output of a ML process: Broadly speaking, ML is the process of finding and describing structural patterns in a supplied dataset. ML takes input in the form of a dataset of instances (also known as examples). An instance refers to an individual, independent example of the dataset. Each instance is characterized by the values of its features (also known as attributes or discriminators) that measure different aspects of the instance. (In the networking field consecutive packets from the same flow might form an instance, while the set of features might include median inter-packet arrival times or standard deviation of packet lengths over a number of consecutive packets in a flow.) The dataset is ultimately presented as a matrix of instances versus features.

The output is the description of the knowledge that has been learnt. How the specific outcome of the learning process is represented (the syntax and semantics) depends largely on the particular ML approach being used.

III. Machine Learning

A. Review of classification with Machine Learning

Much of the existing research focuses on the achievable accuracy (classification accuracy) of different machine learning algorithms. The studies have shown that a number of different algorithms are able to achieve high classification accuracy. The effect of using different sets of statistical features on the same dataset has seen little investigation. Additionally, as different (in some cases private) network traces have been used with different features, direct comparisons between studies are difficult.

B. Types of Machine Learning

ML algorithms that have been used for IP traffic classification generally fall into the categories of being supervised or unsupervised. Unsupervised (or clustering) algorithms group traffic flows into different clusters according to similarities in the feature values. These clusters are not pre-defined and the algorithm itself determines their number and statistical nature. For supervised algorithms the class of each traffic flow must be known before learning. A classification model is built using a training set of example instances that represent each class. The model is then able to predict class membership for new instances by examining the feature values of unknown flows. Most ML techniques used for IP traffic classification focus on the use of supervised and unsupervised learning.

Different types of machine learning can be categorized as below:

- Classification (or supervised learning)
- Clustering (or unsupervised learning)
- Association
- Numeric prediction

C. Unsupervised Learning

Supervised Learning classification techniques use pre-defined classes of training instances. In contrast, unsupervised learning or clustering methods are not provided with this guidance; instead, they discover natural clusters (groups) in the data using internalized heuristics. Clustering focuses on finding patterns in the input data. It clusters instances with similar properties (defined by a specific distance measuring approach, such as Euclidean space) into groups. The groups that are so identified may be exclusive, so that any instance belongs in only one group; or they may be overlapping, when one instance may fall into several groups; they may also be probabilistic, that is an instance belongs to a group with a certain probability. They may be hierarchical, where there is a division of instances into groups at the top level, and then each of these groups is refined further – even down to the level of individual instances.

D. Supervised Learning

Supervised learning creates knowledge structures that support the task of classifying new instances into pre-defined classes. The learning machine is provided with a collection of sample instances, pre-classified into classes. Output of the learning process is a classification model that is constructed by examining and generalizing from the provided instances. In effect, supervised learning focuses on modeling the input/output relationships. Its goal is to identify a mapping from input features to an output class. The knowledge learnt (e.g. commonalities among members of the same class and differences between competing ones) can be presented as a flowchart, a decision tree, classification rules, etc., that can be used later to classify a new unseen instance.

There are two major phases (steps) in supervised learning:

• Training: The learning phase that examines the provided data (called the training dataset) and constructs (builds) a classification model.

• Testing (also known as classifying): The model that has been built in the training phase is used to classify new unseen instances.

There exist a number of supervised learning classification algorithms, each differing mainly in the way the classification model is constructed and what optimization algorithm is used to search for a good model. Examples include the supervised Decision Tree and Naive Bayes classification algorithms [2].

IV. Methodology

A good ML classifier would optimize Recall and Precision. A number of general ML concepts take a specific meaning when applied to IP traffic classification.

A. Training and testing a supervised ML traffic classifier:

Fig. 2, 3 and 4 illustrate the steps involved in building a traffic classifier using a supervised learning algorithm. In this example, the traffic classifier is intended to recognize a particular class of applications (real-time online game traffic) in amongst the usual mix of traffic seen on an IP network [5]. Fig. 2 captures the overall training and testing process that results in a classification model. As noted earlier, the optimal approach to training a supervised ML algorithm is to provide previously classified examples of two types of IP traffic: traffic matching the class of traffic that one wishes later to identify in the network.

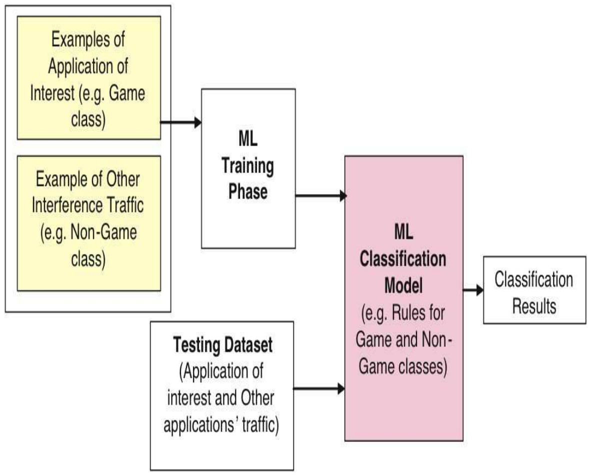
Examples of Application of Interest (e.g. Game class)
Example of Other Interference Traffic (e.g. Non-Game class)
ML Training Phase
ML Classification Model (e.g. Rules for Game and Non-Game classes)
Classification Results
Testing Dataset (Application of interest and Other applications' traffic)

Fig. 2 Training and testing for a two-class supervised ML traffic classifier

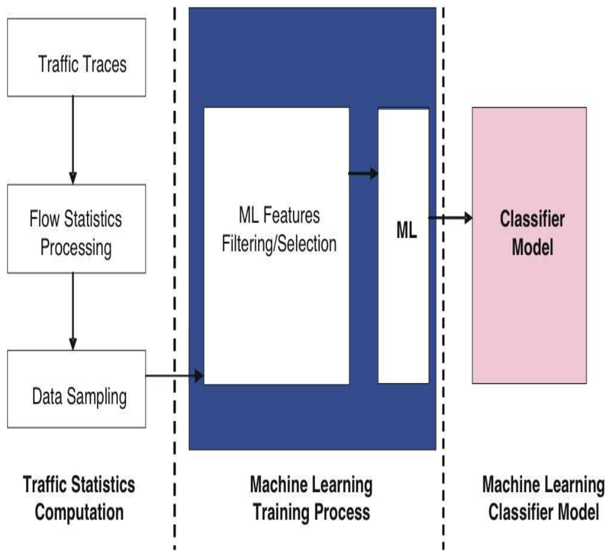
Traffic Traces
Flow Statistics Processing
Data Sampling
ML Features Filtering/Selection
ML
Classifier Model
Traffic Statistics Computation
Machine Learning Training Process
Machine Learning Classifier Model

Fig. 3 Training the supervised ML traffic classifier

Fig. 3 expands on the sequence of events involved in training a supervised ML traffic classifier. First a mix of ‘traffic traces’ are collected that contain both instances of the application

of interest (e.g. online game traffic) and instances of other interfering applications (such as HTTP, DNS, SSH and/or peer2peer file sharing). The ‘flow statistics processing’ step involves calculating the statistical properties of these flows (such as mean packet inter-arrival time, median packet length and/or flow duration) as a prelude to generating features. An optional next step is ‘data sampling’, designed to narrow down the search space for the ML algorithm when faced with extremely large training datasets (traffic traces). The sampling step extracts statistics from a subset of instances of various application classes, and passes these along to the classifier to be used in the training process. The output of Fig. 3 is a classification model.

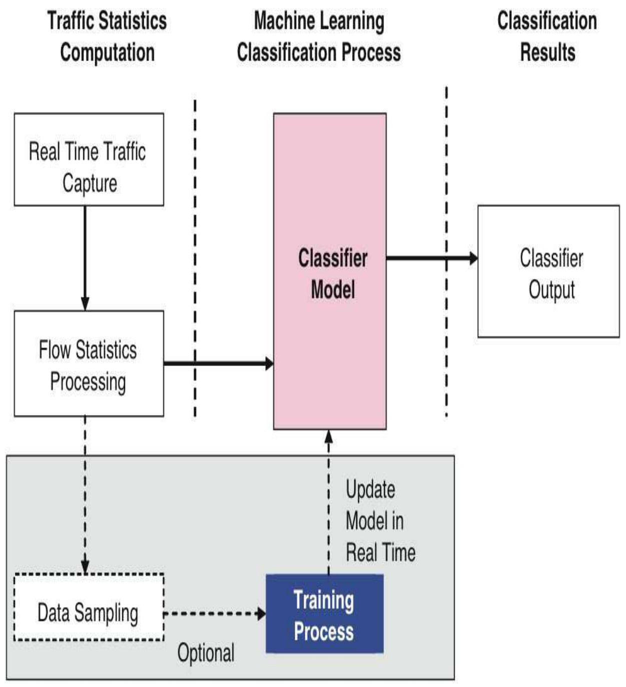
Traffic Statistics Computation
Machine Learning Classification Process
Classification Results
Real Time Traffic Capture
Flow Statistics Processing
Classifier Model
Classifier Output
Update Model in Real Time
Data Sampling
Optional
Training Process

Fig. 4 Data flow within an operational supervised ML traffic classifier

Fig. 4 illustrates data flow within an operational traffic classifier using the model built in Fig. 3. Traffic captured in real-time is used to construct flow statistics from which features are determined and then fed into the classification model. It is presumed that the set of features calculated from captured traffic is limited to the optimal feature set determined during training. The classifier’s output indicates which flows are deemed to be members of the class of interest.

B. Discriminators

Discriminators are the features which make the type of traffic different from one another. A summary of discriminators is provided [4] by using which the machine can be trained using algorithms and output can be calculated. The result will be based on the type of traffic and the amount of that particular type of traffic.

V. Design and Experimentation

The following supervised algorithms that have been implemented in the WEKA [10] ML suite:

- Bayesian Network
- C4.5 Decision Tree
- Naïve Bayes
- Naïve Bayes Tree

The algorithms used in this study are simple to implement and have either few or no parameters to be tuned. They also produce classifications models that can be more easily interpreted. Thus algorithms such as Support Vector Machines or Neural Networks were not included. Naive-Bayes (NBD, NBK) is based on the Bayesian theorem [2]. This classification technique analyses the relationship between each attribute and the class for each instance to derive a conditional probability for the relationships between the attribute values and the class. Naïve Bayesian classifiers must estimate the probabilities of a feature having a certain feature value. Continuous features can have a large (possibly infinite) number of values and the probability cannot be estimated from the frequency distribution. This can be addressed by modeling features with a continuous probability distribution or by using discretisation. We evaluate Naive Bayes using both discretisation (NBD) and kernel density estimation (NBK). Discretisation transforms the continuous features into discrete features, and a distribution model is not required. Kernel density estimation models features using multiple (Gaussian) distributions, and is generally more

effective than using a single (Gaussian) distribution. C4.5 Decision Tree (C4.5) creates a model based on a tree structure. Nodes in the tree represent features, with branches representing possible values connecting features. A leaf representing the class terminates a series of nodes and branches. Determining the class of an instance is a matter of tracing the path of nodes and branches to the terminating leaf. Bayesian Network (BayesNet) is structured as a combination of a directed acyclic graph of nodes and links, and a set of conditional probability tables. Nodes represent features or classes, while links between nodes represent the relationship between them. Conditional probability tables determine the strength of the links. There is one probability table for each node (feature) that defines the probability distribution for the node given its parent nodes. If a node has no parents the probability distribution is unconditional. If a node has one or more parents the probability distribution is a conditional distribution, where the probability of each feature value depends on the values of the parents.

Naïve Bayes Tree (NBTree) is a hybrid of a decision tree classifier and a Naïve Bayes classifier. Designed to allow accuracy to scale up with increasingly large training datasets, the NBTree model is a decision tree of nodes and branches with Naïve Bayes classifiers on the leaf nodes [7].

C. Flow and Feature Definitions

The main limitation in choosing features was that calculation should be realistically possible within a resource constrained IP network device. Thus potential features needed to fit the following criteria:

- Packet payload independent
- Transport layer independent
- Context limited to a single flow (i.e. no features spanning multiple flows)
- Simple to compute

The following features were found to match the above criteria and became the base feature set for experiments:

- Protocol
- Flow duration
- Flow volume in bytes and packets
- Packet length (minimum, mean, maximum and standard deviation)

- Inter-arrival time between packets (minimum, mean, maximum and standard deviation).

Packet lengths are based on the IP length excluding link layer overhead. Inter-arrival times have at least microsecond precision and accuracy. As the traces contained both directions of the flows, features were calculated in both directions.

The use of network simulator version 2 (NS2) can be implemented for determining the type of network traffic. The NS2 uses TCL (Tool Command Language) scripts for coding where one can give the parameters of network traffic and get the output in terms of statistics for classifying the network traffic.

VI. CONCLUSION AND FUTURE WORK

The design will be consisting of algorithms implemented using NS2 in TCL scripts. The features for training the algorithms will be some of the discriminators [4]. By using which the results can be drawn. The data traces are available [8], [9] studying which the discriminators or features can be decided and accordingly appropriate machine learning algorithm can be chosen to implement in NS2. The final result will be in the form of statistics which will be the outcome of the TCL program by running which the type of network traffic can be decided and can be classified accordingly.

REFERENCES

[1] Denis Zuev and Andrew W. Moore "Traffic Classification using a Statistical Approach"

[2] Tom Auld, Andrew W. Moore, Member, IEEE, and Stephen F. Gull "Bayesian Neural Networks for Internet Traffic Classification"

[3] Andrew W. Moore and Denis Zuev "Internet Traffic Classification Using Bayesian Analysis Techniques"

[4] Andrew W. Moore, Denis Zuev, Michael Crogan "Discriminators for use in flow based classification"

[5] Thuy T.T. Nguyen and Grenville Armitage "A Survey of Techniques for Internet Traffic Classification using Machine Learning" IEEE Communications Surveys & Tutorials, Vol. 10, No. 4, 2008

[6] Byungchul Park and James Won-Ki Hong, Young J. Won, "Toward Fine-Grained Traffic Classification"

[7] Nigel Williams, Sebastian Zander, Grenville Armitage "A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification" ACM SIGCOMM Computer Communication Review 5 Volume 36, Number 5, October 2006

[8]http://www.cl.cam.ac.uk/research/srg/netos/nprobe/data/papers/sigmetrics/index.html

[9]http://www.cl.cam.ac.uk/~awm22/publications/RR-05-13.pdf

[10] Waikato Environment for Knowledge Analysis (WEKA) 3.4.4, http://www.cs.waikato.ac.nz/ml/weka/ (viewed August 2006).