Computer Networks 91 (2015) 752–771

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

# A low complexity real-time Internet traffic flows neuro-fuzzy classifier

Antonello Rizzi [a], Alfonso Iacovazzi [a,*], Andrea Baiocchi [a], Silvia Colabrese [b]

[a] DIET, Sapienza University of Rome, Via Eudossiana 18, Rome 00184, Italy
[b] Department of Pattern Analysis and Computer Vision, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy

ARTICLE INFO

*Article history:*
Received 27 January 2015
Revised 10 September 2015
Accepted 14 September 2015
Available online 21 September 2015

*Keywords:*
Traffic flow classification
Neurofuzzy networks
Features selection
Genetic algorithms
Classifier complexity
FPGA

ABSTRACT

Traffic flow classification to identify applications and activity of users is widely studied both to understand privacy threats and to support network functions such as usage policies and QoS. For those needs, real time classification is required and classifier's complexity is as important as accuracy, especially given the increasing link speeds also in the access section of the network. We propose the application of a highly efficient classification system, specifically Min–Max neuro-fuzzy networks trained by PARC algorithm, and compare it with popular classification systems, by considering traffic data sets collected in different epochs and places. We show that Min–Max networks achieve high accuracy, in line with the best performing algorithms on Weka (SVM, Random Tree, Random Forest). The required classification model complexity is much lower with Min–Max networks with respect to the other models, enabling the implementation of effective classification algorithms in real time on inexpensive platforms.



## 1. Introduction

Traffic analysis is the main technique used to exploit information leakage offered by observable features of packet traffic in a ciphered channel and infer as much as possible about the content of the traffic flow. Raymond [1] provides an overview of all possible attacks that can be carried out using traffic analysis. These attacks encompass both passive ones, aiming at breaking the user privacy and gaining knowledge of the type of exchanged information flows and possibly of part of their contents, and active attacks, aiming at selective disruption of specific types of information flows, or message delaying and message tagging.

A large body of literature has grown on the problem of application layer traffic classification by means of traffic analysis and several methods of classification based on statistical analysis of traffic patterns and machine-learning techniques have been proposed and analyzed. For general reviews see [2–8]. Besides being an obvious attack on privacy, traffic classification can have useful and legitimate goals, as pointed out in [4], such as: identification of user activities in order to apply traffic filtering and to support quality of service mechanisms; development of diagnostic tools for the detection of anomalous network behaviors, in order to identify possible worms or denial of service (DoS) attacks. In [9] traffic classification is deemed as a key component of automated QoS management. A reliable and easy to deploy traffic classification tool can also be a component of vulnerability assessment.

There is a vast literature presenting techniques that can identify traffic classes based solely on the use of traffic features that remain observable even after encryption, e.g., see [10,11] for general approaches to traffic flow classifications, [12,13] for the identification of encrypted Skype traffic within an aggregate traffic stream [14–16], for classification of flows

* Corresponding author. Tel.: +39 0644585365.
*E-mail addresses:* antonello.rizzi@uniroma1.it (A. Rizzi), alfonso.iacovazzi@uniroma1.it (A. Iacovazzi), andrea.baiocchi@uniroma1.it (A. Baiocchi), silvia.colabrese@iit.it (S. Colabrese).

http://dx.doi.org/10.1016/j.comnet.2015.09.011
1389-1286/

carried inside SSH connections [17,18], for classification of encrypted web pages among a set of pre-defined alternatives [19], for classification of Internet traffic, including peer-to-peer and content delivery traffic, that proves to be robust with respect to dynamic source port changing [20], for the identification of unknown applications.

Furthermore, as evidenced by Soysal and Schmidt in [19], a fundamental issue which has to be considered when doing performance evaluation is to use a test set for which the correct mapping between each flow and the class membership is known (ground truth), and, if a machine learning algorithm is used, previously and correctly classified data is demanded to characterize the traffic classes. Although an interesting issue in this regard consists in studying how well training data coming from a given networking environment can represent a useful data sampling for different contexts, this issue is not one of our goals in this work.

Among recent works on traffic classification, Zhang et al. [21] propose a nonparametric approach which exploits correlation of traffic flows in order to improve performance of the NN-based traffic classifier in case the training set is too small. Li et al. [22] develop a semi-supervised support vector machine (SVM) based on flow statistics, to identify and classify network application. They use a radial basis function (RBF) as the kernel function of the SVM and the co-training as a semi-supervised technique. The algorithm is implemented by procedures based on Weka 3.7 [23]. Wang et al. in [24] propose a token-based approach that uses machine learning techniques on statistical features of traffic. They first look for common substrings in the first $N$ bytes of the flow payload for each class, and then apply a feature selection algorithm to reduce the size of the token set. Their proposal achieves high classification accuracy with low computational complexity, but it requires payloads and it is not suitable for encrypted flows. In [25] Szabó et al. propose a novel framework that takes an incremental approach, whereby new features are exploited as packets of a flow are observed and also different flows are correlated, so adapting the abstraction level of the traffic analysis to different purposes. In case of strictly real time, single flow classification, no special advantage is brought about by this approach.

Overall, there is still ample room to investigate a real time traffic classification approach that maintains a high accuracy of classification, while lending itself to implementation on inexpensive platforms or within environments where computational burden must anyway be limited, such as low-end edge routers, packet filtering devices in customer networks, mobile terminals.

We propose the use of a neuro-fuzzy machine learning system, specifically Min–Max networks trained by PARC algorithm [26], for real time traffic flow classification relying only on simple features extracted from the first few packets of each flow. The proposed classification technique can be applied whenever it is possible to delineate individual flows, even if packet payloads are encrypted. As a matter of example, in case information flows are protected by means of SSL/TLS (secure socket layer/transport layer security), they can still be identified and it is possible to detect the length and direction of each packet making up the flow and to mark each captured packet with a timestamp (e.g., https is carried via an SSL/TLS connection inside a TCP connection, so IP and TCP headers are readable). On the other hand, with SSL/TLS, packet payloads beyond IP and transport headers are not accessible, thus defeating deep packet inspection tools. The same applies for traffic protected via SSH.

Key points shown in this paper are: (i) accuracy as high as 99% and anyway above 90% can be achieved even with only few initial packets and in any case by using no more than the first ten packets of a flow; (ii) complexity of the Min–Max based classification models is sensitively less than the complexity of other accurate classification models, notably the support vector machines (SVM). It is important to underline that a low structural complexity is a fundamental requirement in classification model synthesis, enabling the implementation of effective flow classification systems in real time on inexpensive platforms, such as FPGA based embedded systems. Under this perspective, our software comparative analysis of traffic flow classification algorithms encourages the adoption of the Min–Max PARC classification system in the design of a dedicated hardware solution.

On the other hand, when dealing with data links characterized by a very high number of flows, a low structural complexity of the classification model allows massive parallelization on high end FPGA systems, thus enabling to perform the classification of all concurrent flows in real time.

The remainder of the paper is organized as follows. In Section 2 a description of feature extraction procedure from traffic data is given. Section 3 gives a brief account of the Min–Max neurofuzzy classification network. Performance results are introduced and discussed in Section 4. A concise account of implementation aspects of the proposed classifier as a digital circuit is outlined in Section 5. Final remarks are given in Section 6.

## 2. Flow definition and feature extraction

In this work, a flow is the bi-directional, ordered sequence of packets exchanged within an application session. A flow can be made up of a single TCP connection or of a sequence of UDP datagrams. In general, the detection of a flow is done, as usual in routers, based on TCP/UDP/IP header fields and time gaps.

A number of selectors $S_1, \ldots, S_J$ are defined, as well as a time gap $G$. A table of active flows is maintained with the values of selectors of flows that have already being detected and are still active. Let $[s_1(i), \ldots, s_J(i)]$ be the selector values of the $i$th active flow at a given time, $i = 1, \ldots, F$, and let $t_i$ be the last time a packet of flow $i$ has been detected at the capture point. For a packet observed at time $t$ on the network interface where traffic capture takes place, the header fields $S_1, \ldots, S_J$ are checked: let their values be $[x_1, \ldots, x_J]$. Two cases can occur:

(i) if the matching conditions $x_k = s_k(i),\ k = 1, \ldots, J$, are jointly satisfied for some $i$, with $t \le t_i + G$, the packet is assigned to the $i$th flow, its features are appended to the flow feature array and $t_i$ is updated as $t_i = t$;
(ii) if $[x_1, \ldots, x_J] \ne [s_1(i), \ldots, s_J(i)]$ for all $i = 1, \ldots, F$, a new flow is instantiated, by letting $s_k(F+1) = x_k$ for $k = 1, \ldots, J$, and $t_{F+1} = t$.

Finally, if $t > t_i + G$ for some $i$, flow $i$ is deemed as ended, its entry is removed from the table of active flow, and $F$ is