

Network Machine Learning Research Group  
Internet-Draft  
Intended status: Informational  
Expires: December 5, 2016

S. Jiang, Ed.  
B. Liu  
Huawei Technologies Co., Ltd  
P. Demestichas  
University of Piraeus  
J. Francois  
Inria  
G. M. Moura  
SIDN Labs  
P. Barlet  
Network Polygraph  
June 3, 2016

Use Cases of Applying Machine Learning Mechanism with Network Traffic  
draft-jiang-nmlrg-traffic-machine-learning-00

## Abstract

This document introduces a set of use cases in which machine learning technologies are applied to network traffic relevant activities, including machine learning based traffic classification, traffic management, etc.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 5, 2016.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Terminology . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Methodology of Learning from Traffic . . . . .	<a href="#">4</a>
<a href="#">3.1.</a>	Data of the Network Traffic . . . . .	<a href="#">4</a>
<a href="#">3.2.</a>	Data Source and Storage . . . . .	<a href="#">5</a>
<a href="#">3.3.</a>	Architecture Considerations . . . . .	<a href="#">5</a>
<a href="#">3.4.</a>	Closed Control Loop . . . . .	<a href="#">6</a>
<a href="#">4.</a>	Use Cases Study of Applying Machine Learning in Network . . .	<a href="#">6</a>
<a href="#">4.1.</a>	HTTPS Traffic Classification . . . . .	<a href="#">6</a>
<a href="#">4.2.</a>	Malicious Domains: Automatic Detection with DNS Traffic Analysis . . . . .	<a href="#">9</a>
<a href="#">4.3.</a>	Machine-learning based Policy Derivation and Evaluation in Broadband Networks . . . . .	<a href="#">10</a>
<a href="#">4.4.</a>	Traffic Anomaly Detection in the Router . . . . .	<a href="#">11</a>
<a href="#">4.5.</a>	Applications of Machine Learning to Flow Monitoring . . .	<a href="#">12</a>
<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">15</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">15</a>
<a href="#">7.</a>	Acknowledgements . . . . .	<a href="#">15</a>
<a href="#">8.</a>	Change log [RFC Editor: Please remove] . . . . .	<a href="#">16</a>
<a href="#">9.</a>	Informative References . . . . .	<a href="#">16</a>
	Authors' Addresses . . . . .	<a href="#">17</a>

## [1.](#) Introduction

Machine learning technology has been successful in solving complicated issues. It helps to make predictions or decisions based on large datasets. It could also dynamically adapt to varying situations and response to real-time issues. Therefore, more and more research starts on applying machine learning in the network area.

Among many aspects of networks, the network traffic is one of the most complicated managed objectives. Its volume is rapidly growing

along with the Internet explosion. It is always dynamically changing. Most network traffic flows only last a few minutes, or even shorter. And the user contents within traffic is becoming more diverse due to the development of various network services, and increasing use of encryption. Consequently, it is more and more

challenging for administrators to get aware of the network's running status and efficiently manage the network traffic flows. Although more and more data regarding network traffics are generated, traditional mechanisms based on pre-designed network traffic patterns become less and less efficient.

It is natural to utilize powerful machine learning technology to analyze the large mount of data regarding network traffic, to understand the network's status, such as performance, failures, security, etc. It is a big advantage that machines can measure and analyse the network traffic, then report the results and predictions to humans for further decision. The machines could handle vast amounts of data which is almost impossible for humans to deal with, in close to real time. Even more, if the speed and accuracy of the prediction is high enough, it is possible that the subsequent action based on the prediction result could form a closed control loop to achieve autonomic management. However, the maturity of latter might be far in the future. Today, the traditional control programs still look more reliable than machine learning based control mechanisms.

This document firstly analyzes the data of the network traffic from various perspectives; and also discusses several important practical considerations, including the training data source, data storage and the learning system architecture. It then introduce a set of use cases, which have been shown to work well although there is large scope for improvements, including ML-based traffic classification, traffic management, interface failure prediction, etc.

Editor notice: this document is in the primary stage. It collects the use cases presented in the proposed Network Machine Learning Research Group (NMLRG) session in IETF95 meeting.

## [2.](#) Terminology

The terminology defined in this document.

**Machine Learning** A computational mechanism that analyzes and learns from data input, either historic data or real-time feedback data, following a set of designed features and algorithms. It can be used to make analysis, predictions or decisions, rather than following strictly static program instructions.

**Network Traffic** The amount of data moving across a network at a given point of time. They are mostly encapsulated in network packets.

**Traffic Flow** A sequence of packets from a source computer to a destination [[RFC6437](#)]. It is the unit of network traffic.

**Feature (machine learning)** In machine learning and pattern recognition, a feature is an individual measurable property of a phenomenon being observed. Choosing informative, discriminating and independent features is a crucial step for effective algorithms in pattern recognition, classification and regression.

**Algorithm (machine learning)** Machine learning algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions expressed as outputs, rather than following strictly static program instructions. A incomplete list of machine learning algorithms includes supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, deep learning, etc.

### [3.](#) Methodology of Learning from Traffic

#### [3.1.](#) Data of the Network Traffic

There is plenty of valuable data related to the network traffic. These data are raw features in learning process. Following is a simple classification of network traffic data.

**Measurable properties** There are many measurable properties of network traffic, such as latency, number of packets, duration, etc. These properties are also very essential features, especially for use cases relevant to performance, QoS (Quality of Service), etc.

**Data within communication protocols** The user contents are

encapsulated in layered communication protocols. Many information are contained within the protocol headers, for example the source and destination IP addresses in the IP header, the port numbers in the TCP/UDP header, etc. Transport layer protocols are often related to the type of applications, such as FTP (File Transfer Protocol) for file transfer, HTTP (Hyper Text Transfer Protocol) for web, etc; and many application-relevant data are embedded within these protocols. These could also be essential data for classification or application-oriented analysis. However, some traffic will not provide transport or application information, due to unknown protocols or encryption.

User content User contents are the payload of packets, which might be obtained by DPI (Deep Packet Inspection) within the transit network if the packets are unencrypted, or they could be analyzed by the source or destination nodes.

Data in network signaling protocols Traffic flows are managed or indirectly influenced by various network signaling protocols. For

example, the routing protocols determine the next hop of a specific network traffic flow, or even the traffic path (by some sophisticated routing protocol such as MPLS-TE (Multi-Protocol Label Switching - Traffic Engineering), segment routing, etc.); the P2P (Peer to Peer) protocol can even decide the destination of a specific content traffic. They are relevant and are potential features for traffic analysis. Furthermore, the traffic of these signaling protocols themselves may also be learning objectives.

### [3.2.](#) Data Source and Storage

Within networks, forwarding devices such as routers, switches, firewalls, etc., are the entities that directly handle the network traffic. Thus, they could collect network traffic data, such as measurable properties, protocol information, etc. Source nodes or destination nodes, particularly servers, could also be the source of network traffic data. They could either report the collected data to a central repository for storage and learning, or collect and store the data by themselves for local learning. This depends on the learning architecture, which is discussed in the following section.

### [3.3.](#) Architecture Considerations

## Global learning vs. local learning

- \* Global learning refers to the tasks that are mostly network-level, so that they need to be done in a global viewpoint. In this case, the learning entity is normally centralized and is different from the data source entities.
- \* Local learning is more applicable to the tasks that are only relevant to one or a limited group of devices, and they could be done directly within that one node or that limited group of nodes. In this case of grouped nodes, the data may also need to be transited from the data source entity to learning entity.

## Offline & online learning

- \* Co-located mode: training (offline, based on historic data) and prediction (online, based on real-time data) are both done within the same entity. The entity could be a central repository or a specific node.
- \* De-coupled mode: training is done in the central repository, and prediction is made by the routers/switches/firewalls or other devices that directly process the network traffic.

Central learning & distributed learning Central learning means the learning process is done at a single entity, which is either a central repository or a node. Distributed learning refer to ensemble learning that multiple entities do the learning simultaneously and ensemble the results together to sort out a final results. Since network devices are naturally distributed, it could be foreseen that ensemble learning is a good approach for a certain of use cases.

### [3.4.](#) Closed Control Loop

The prediction made by machine learning mechanism could be directly used on manipulating the network traffic, or other relevant actions, such as changing the device configuration, etc.

However, as the introduction section said, this kind of utilization might be suitable only for a small set of the use cases, due to the limited accuracy of machine learning technologies. Besides, some critical usages simply cannot tolerate any false decision.

#### 4. Use Cases Study of Applying Machine Learning in Network

Editor notes: This section is a collection of the work presented in the proposed NMLRG session in IETF95 meeting. More contributions on use cases are welcome.

##### 4.1. HTTPS Traffic Classification

Managing network traffic requires a good understanding of the content of traffic flows for various purposes. Indeed, enhancing the QoS by prioritizing or scheduling the flows or enforcing security policies by filtering some of them cannot solely on rely protocol headers like IP, TCP or UDP headers. Analyzing the user content with DPI is so necessary. However, this poses serious concerns regarding the user privacy. In addition, OTT (Over-the-Top) actors would prefer to fully control their network traffic rather than being subject to any intermediaries policies. As a result, encrypting the traffic has been widely adopted in last years.

In that context, traffic management is facing to severe difficulties since DPI is not efficient anymore. Using an intermediary service or proxy are the only ways to analyze the content of encrypted traffic but it requires a high trustfulness in the intermediaries and so not always guaranteed, for example with end-users of an operator networks.

Therefore, new techniques wit the ability to extract knowledge and insight from encrypted flows is necessary. Especially HTTPS

[RFC2818] is now a major protocol use over Internet because it provides secure Web communication while Web is now embracing various services which have been provided apart in the past: email, video streaming, chat, VoIP, file sharing, etc. It relies on TLS (Transport Layer Security) [[RFC5246](#)], [[RFC6066](#)] to encapsulate HTTP requests.

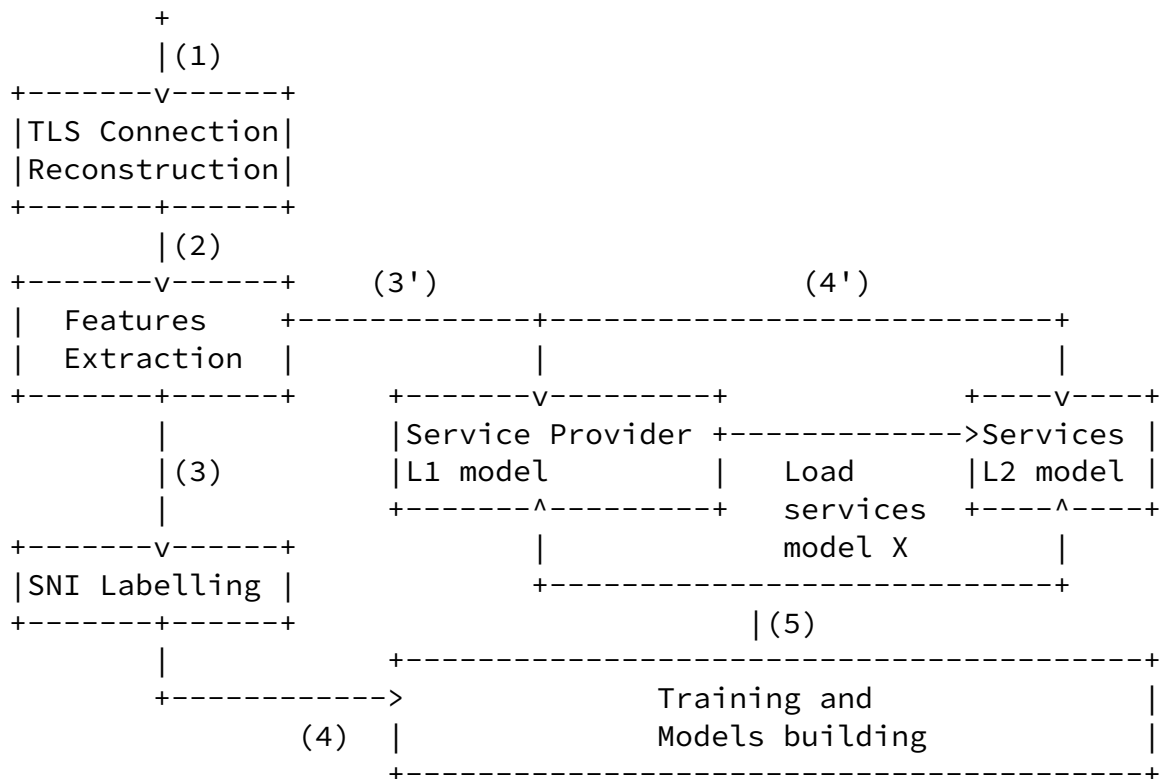
Being able to identify the service and the providers of an HTTPS

connection would help in applying different strategies for managing the corresponding flow. For instance, VoIP (Voice over IP) and email do not require the same QoS or some service use might be prohibited like file sharing to avoid data leakage in a company.

As a concrete example, Google, Facebook or Amazon are service providers while maps, drive, gmail are services of Google. To identify them when they are accessed by a user, IP addresses and DNS (Domain Name System) names based identification is not reliable as the users can rely on intermediates to respectively serve as proxy or resolve DNS requests. The SNI (Server Name Indication) [[RFC5246](#)] is an extension of HTTPS which is indicated by the user when initiating the TLS handshake (Client Hello). SNI actually contains the hostname to which the request is addressed. Such a hostname is significative of the service and service provider name. However, SNI is an optional field and can be easily forged to circumvent HTTPS filtering without impacting service use [[bypasssni](#)]. More advanced mechanisms are hence necessary to improve the robustness of identification even in the case of non collaborative users.

Because the objective is to automatically label an HTTPS connection by the service and service provider associated with. The TLS handshake is not encrypted but data exchanged during this phase (random number, selected ciphers,...) is not distinctive of the accessed service. However, the nature of accessed service directly impacts on user content transmitted through the secure channel especially on the type, size and way to transmit those data. Such metadata are still measurable properties.





## Two-levels HTTPS traffic classification

In figure above, step(1) consists in reconstructing the HTTPS connection and retrieving packets on top of which the following metrics are observed (2):

- o Inter Arrival Time
- o Packet size
- o Encrypted data size: this feature has the advantage to be strongly related to the service accessed instead of the packet size which is biased by other lower layer headers

Based on these values, aggregated features are computed: average, minimum, maximum, 25th percentile, median, 75th percentile.

Because different providers may offer a similar service, a single classifier could fail to distinguish them. A multi-level machine learning approach has been proposed. For learning, a dataset without forged SNI is used (3) to build the classifiers (4). The result is (5):

- o a first level model (L1 model) whose the goal is to identify the service provider,

- o a set of second level models (L2 models), one for each service provider to identify specific service of a service provider

Once all classifiers are trained, a new unknown HTTPS connection is first matched against the LV1 model (3'). The output is the predicted service provider but also leads to load the corresponding LV2 model (4') to determine the specific service of this service provider.

This framework is independent of the ML technique. being used. Each model could be also built with a different technique but our study have shown that best results are obtained with Random Forest.

The HTTPS classification framework has been tested over 288,901 connections from lab users. Standard evaluation procedure have been applied. Less representative features have been automatically discarded. Using a ten-fold cross-validation, each tested connection has been marked as perfect identification (both the service provider and the service name are rightly identified), partial identification (only the service provider is identified) or invalid (none of them). 93.1% falls in the first category, 2.9% in the second and the rest in the third. Full results are available in [[httpsframework](#)].

Although results are promising, the current method can only be applied at the end once the HTTPS connection, i.e. after being reconstructed. This avoids to apply any kind of policies to the corresponding traffic flow. Future challenge is thus to classify the connection before it ends in order to apply.

#### [4.2.](#) Malicious Domains: Automatic Detection with DNS Traffic Analysis

Since their inception, domain names have been used to provide a simple identification label for hosts, services, applications, and networks on the Internet [[RFC1034](#)]. In the same way, domains and the DNS infrastructure have also been misused in various types of abuses, such as phishing, spam, malware distribution, among others.

Newly registered malicious domain names are well-know to a very distinct initial DNS lookup pattern than legitimates ones: typically, they exhibit an abnormally higher number of lookups [[Hao2011](#)]. One of the reasons is that malicious domains tend to rely upon spam campaigns within the first ours after the registration of these domains in order to maximize the number of victims before the domain is detected and taken down.

In order to protect users from such domains, nDEWS (New Domains Early

Warning System) [[Moura2016](#)], a tool that classifies the newly registered domains based on their initial lookup pattern, has been

proposed. To perform that, it is required to have access to (i) a domains registration database and (ii) authoritative DNS server traffic data, which is typically the case for Top-Level Domains (TLD) registries. These domains are classified using k-means as a clustering method into two clusters using four features extracted from the analyzed DNS traffic: # DNS queries, # IP addresses, # Autonomous Systems (ASes), and # Countries, which were chosen empirically.

As a result, in an automated fashion, a large variety of suspicious domains can be detected, including phishing, malware, but also other types, such as fake pharmaceutical shops as well as counterfeit sneakers. In this particular case, the responsible registrars are notified in this pilot study about these websites. Ultimately, it allows these websites to be taken down, minimizing the potential number of victims.

#### [4.3.](#) Machine-learning based Policy Derivation and Evaluation in Broadband Networks

Service provisioning is becoming more complex. For instance, there are services having diverse quality requirements, there is variance of the requirements in time and space, and there is the need for utmost resource efficiency. Moreover, full agility in time and space (in order to accomplish resource efficient service provisioning) requires the solution of computationally intensive tasks. In this respect, policies can play a role: specify the network behaviour in time periods and service area regions.

In this direction, machine learning can have a fundamental role, e.g., for learning situations encountered and "good" ways (policies) for handling them. The contribution addresses the role that machine learning can play for policy derivation and evaluation. In more detail it addresses the requirements on the role of machine learning, including potential inputs and outputs.

Knowledge and machine learning can be an important aspect of wireless networks. Knowledge is created both regarding the contexts and their occurrence, as well as on the association of the context with

specific actions and its scoring. The latter encompasses development of knowledge on how to handle acquired contexts; this knowledge will include the contexts encountered, the corresponding handlings done (decisions applied), the potential alternative handlings, and the respective efficiency of each handling (actually applied or alternate).

Reinforcing "good" solutions per each encountered context (e.g. reinforcement learning) can be a vital and unique element of a

Jiang, et al.

Expires December 5, 2016

[Page 10]

---

Internet-Draft

Network Machine Learning

June 2016

knowledge-based management system. Machine learning can be realized through clustering to discover underlying structures in data, regression to identify patterns and predict values in cell and network usage, classification to classify first-seen unknown users, and density estimation to model complex user behavior and network usage. Several deep architectures and techniques (such as pre-training) can be utilized, in order to generalize better on complex data with underlying information and be able to make accurate predictions, even on unseen data.

As a result, depending on what we want to achieve, the proper machine learning approach can be used.

Through machine learning it will be possible to provide faster and targeted solutions to specific network problems. Moreover, it is possible cluster various usage profiles and prioritize the traffic according to the criticality level. For instance, mission critical services need special attention with respect to latency and prioritization, compared to plain services which may tolerate a bit of delay without jeopardizing the overall quality. In addition, machine learning can lead to improved results in KPIs (Key Performance Indicator) such as end-user throughput, latency, energy consumption and overall cost effectiveness. Moreover, reliability can be increased since certain problematic situations may be predicted before happening, hence it will be possible to act pro-actively and alleviate the negative impact of a problem in the network.

It is evident that machine learning can have significant importance in policy derivation and evaluation in broadband networks, especially towards in 5G infrastructures which will be complex, heterogeneous and need to accommodate multi-services ranging from mobile broadband

to massive machine type, mission critical and vehicular communications.

#### 4.4. Traffic Anomaly Detection in the Router

Modern routers usually have the capability that makes alarms of high bandwidth usage rate of a specific interface. When network traffic exceeds a certain threshold, the router will consider it as an anomaly event and report it to the NMS (Network Management System). For instance, in some routers/switches, there exists configuration such as "trap-threshold { input-rate | output-rate }" to trigger traffic alarms, which is statically configured by experienced administrators. However, network traffic is usually not static and even changes significantly due to the changes of carried services, residential situation, and etc. Thus, static configuration could not effectively identify the traffic anomaly events.

Jiang, et al.

Expires December 5, 2016

[Page 11]

---

Internet-Draft

Network Machine Learning

June 2016

To address above issue, machine learning technologies are applied for routers/switches to learn local traffic pattern and detect the traffic anomaly events based on the learning results.

Wavelets are employed to analyze time-series network traffic for anomaly detection. In some certain interval, the routers measure, record, and analyze the input and output traffic rates respectively, or in the form of rate sums. (The former is recommended for a finer granularity analysis.)

Running for some time, the router would get a set of "time-rate" data, collected as time-series waves for further wavelet analysis. Besides wavelets, this use case proposes other machine learning techniques such as outlier detection. For this way, features are to be extracted from wavelets for supervised or unsupervised learning.

After data collection, the router would sort up the data and figure out the alarm threshold statistically based on data distribution, to discriminate the normal and outlier traffic rates. When interface traffic exceeds the threshold, the router would make alarms to the NMS. The router could dynamically adjust the alarm threshold with new coming data, by periodical anomaly analysis. This approach helps devices detect traffic anomaly more efficiently and effectively, compared to traditional way of learning at the central repository that collects traffic information from various devices.

This use case could be extended from single interface to multiple ones, that is, device scope of multiple traffic waves, and even wider scope of multiple devices in a certain domain. Thus would make the analysis more comprehensive.

Besides wavelet analysis, there might be more techniques to explore, such as correlation analysis of traffic anomaly events among multiple devices.

#### 4.5. Applications of Machine Learning to Flow Monitoring

A commercial cloud-based flow monitoring service from Network Polygraph [[polygraph](#)] has used Machine Learning analysis as a cost-effective alternative to DPI for traffic classification, which identifies the application responsible for each network traffic flow.

Nowadays, DPI is considered as the standard technology for traffic classification. However, DPI is generally expensive as it requires the analysis of the payload of every single packet. This usually involves the use of powerful, specialized hardware appliances, which need to be deployed in every link to obtain full coverage of the network. In the case of Network Polygraph, the use of DPI is

impractical, because the volume of data to be exported to the cloud would be overwhelming (i.e., all traffic should be replicated). A more viable alternative is the use flow-based monitoring technologies, such as NetFlow [[RFC3954](#)] or IPFIX [[RFC7011](#)], where the volume of exported data is significantly lower. Flow-based monitoring technologies provide summarized information (e.g., duration, traffic volume) for every connection (or "traffic flow") handled by a router. The information available in flow records is more limited compared to DPI (e.g., packet payloads are not available). As a result, most flow-based monitoring tools base their classification on the port numbers or simple heuristics, which are known to be highly unreliable.

To address this problem, Network Polygraph uses a traffic classification approach based on ML. Several studies showed that supervised learning can achieve similar classification accuracy to DPI at a fraction of its cost. However, supervised methods suffer from some practical limitations that make them very difficult to

deploy and maintain in production environments. For example, they require a costly training phase prior to its deployment and need to be frequently retrained, every time there is a change in the network or in the network applications.

This section describes the ML approach used by Network Polygraph for online classification of NetFlow/IPFIX traffic. To solve the practical limitations of supervised learning, Network Polygraph incorporates an automatic retraining system. Figure 1 shows the components and data flow of the classification engine, which is divided in two parts:

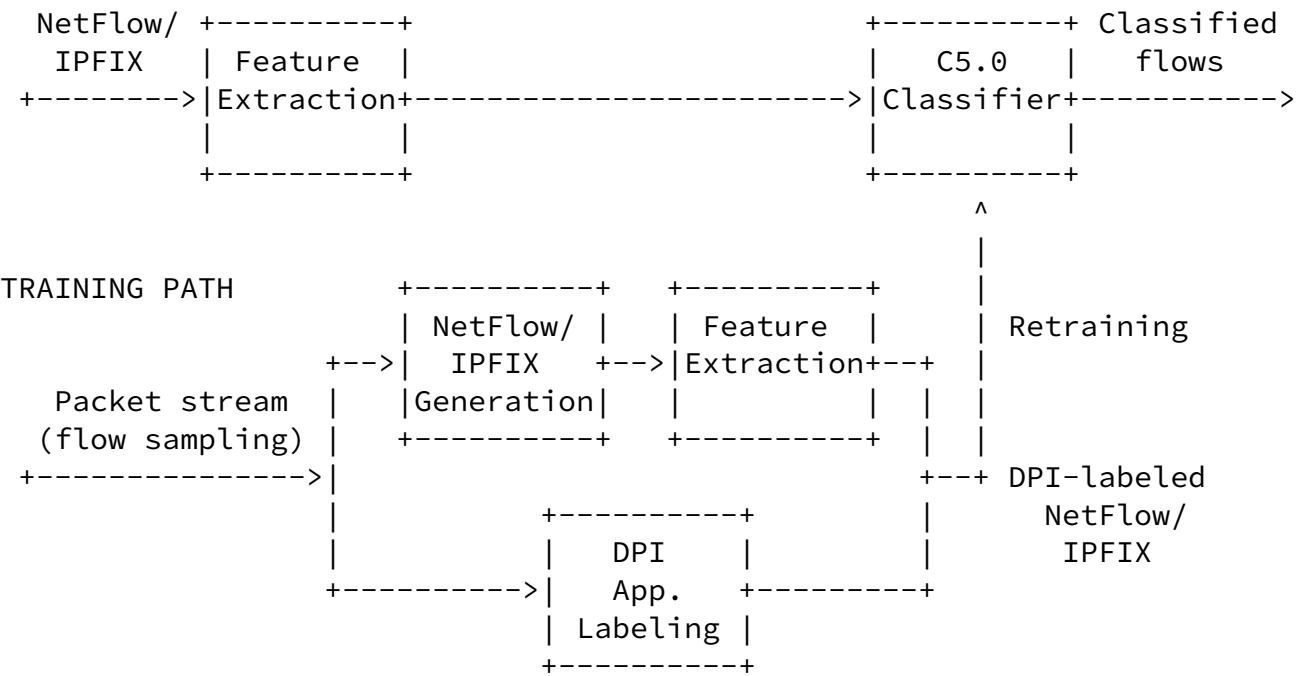
- o The classification path (Figure 1, top) is in charge of the classification of the traffic online using ML. The input of the classification path are the NetFlow/IPFIX flows exported by the routers, while the output are the classified flows. Several traffic features are extracted from each flow, including the information directly available in the flow records (e.g., addresses, ports, packet and byte counts) together with some features we construct (e.g., average packet size, rate and interarrival time). The traffic features are the input of the traffic classification algorithm, whose function is to identify the application that generated the flow. Among the different supervised algorithms, a C5.0 decision tree was selected, because it has been shown to present the best accuracy/cost ratio for traffic classification. Other supervised methods, e.g., Support Vector Machine (SVM) and Artificial Neural Network (ANN), obtain similar accuracy, but classification and training times are faster with decision trees. In Network Polygraph, training times are

critical as the training path is continuously updating the classification model in the background.

- o The training path (Figure 1, bottom) implements the automatic retraining system, which is responsible of automatically updating the classification model when it becomes obsolete. To that end, a random packet-level sample of the network traffic is continuously collected using flow-based sampling. Sampled flows are then labeled using DPI. It is possible to use DPI in the training path because training can be performed only with a small data sample (e.g., 1/1000 flows). This significantly reduces the

computational overhead and volume of data to be exported. The labeled sample is used to verify the accuracy of the classification model. The system accuracy is estimated by comparing the output of DPI (training path) and C5.0 (classification path) for those flows sampled in the training path. If the estimated accuracy falls below a configurable threshold, the labeled sample is used to generate an updated model using only those features available in NetFlow/IPFIX (IP Flow Information Export) records. This training process can also be performed in few vantage points, and use it for other networks where only NetFlow/IPFIX monitoring data is available.

CLASSIFICATION PATH



Network Polygraph classification engine data flow

Figure 1

In order to validate the performance of the described ML approach, the accuracy of Network Polygraph was measured using a complete 14-day trace from the 10-Gigabit link that connects the Catalan Research and Education Network (Anella Científica) to its Spanish



counterpart (RedIRIS). The trace contained about 70 million flows with a flow sampling rate of 1/400. The experimental results showed that, with a 96% retraining threshold, the system sustained an average classification accuracy of 97.5%, needing only 15 retrains during the 14 days, which were performed automatically without requiring any human intervention. When the retraining threshold was decreased to 94%, the accuracy was slightly reduced to 96.76% with only 5 retrains.

The target objective is to progressively reduce the dependence on DPI technologies, which are expensive, difficult to deploy, not scalable, and not robust against encryption, in favor of flow-based machine learning approaches that are more cost-effective and can be easily offered as a cloud service. In this direction, some research challenges include the classification of web services and CDN traffic from flow-based measurements, and the combination of multiple ground truths obtained from vantage points in different networks.

## 5. Security Considerations

This document is focused on applying machine learning in network, including of course applying machine learning in network security, on higher-layer concepts. Therefore, it does not itself create any new security issues.

## 6. IANA Considerations

This memo includes no request to IANA.

## 7. Acknowledgements

The authors would like to acknowledge Josep Sanjuas, Andreas Georgakopoulos, Kostas Tsagkaris, Valentin Carela, Wazen M. Shbair, Thibault Cholez, and Isabelle Chrisment for their contributions.

The author would like to acknowledge the valuable comments made by participants in the IRTF Network Machine Learning Research Group, particular thanks to Lars Eggert, Brian Carpenter, Albert Cabellos, Shufan Ji, Susan Hares, Rudra Saha, and Dacheng Zhang.

Jerome Francois was partly funded by Flamingo, a Network of Excellence project (ICT-318488) supported by the European Commission under its 7th Framework Programme.

This document was produced using the xml2rfc tool [[RFC7749](#)].

8. Change log [RFC Editor: Please remove]

[draft-jiang-nmlrg-traffic-machine-learning-00](#): original version, 2016-06-03.

9. Informative References

[bypassssni]

Shbair, W., Cholez, T., Goichot, A., and I. Chrisment, "Efficiently Bypassing SNI-based HTTPS Filtering", IFIP/IEEE International Symposium on Integrated Network Management (IM2015) , 2015.

[Hao2011] Hao, S., Feamster, N., and R. Pandrangi, "Monitoring the Initial DNS Behavior of Malicious Domains", Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference (IMC 2011) , Nov 2011.

[httpsframework]

Shbair, W., Cholez, T., Francois, J., and I. Chrisment, "A Multi-Level Framework to Identify HTTPS Services", IEEE/IFIP Network Operations and Management Symposium , 2016.

[Moura2016]

M. Moura, G., Mueller, M., Wullink, M., and C. Hesselman, "nDEWS: a New Domains Early Warning System for TLDs", IEEE/IFIP International Workshop on Analytics for Network and Service Management (AnNet 2016), co-located with IEEE/IFIP Network Operations and Management Symposium (NOMS 2016) , 04 2016.

[polygraph]

"Network Polygraph", <<https://polygraph.io>>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), DOI 10.17487/RFC1034, November 1987, <<http://www.rfc-editor.org/info/rfc1034>>.

[RFC2818] Rescorla, E., "HTTP Over TLS", [RFC 2818](#), DOI 10.17487/RFC2818, May 2000, <<http://www.rfc-editor.org/info/rfc2818>>.

[RFC3954] Claise, B., Ed., "Cisco Systems NetFlow Services Export Version 9", [RFC 3954](#), DOI 10.17487/RFC3954, October 2004, <<http://www.rfc-editor.org/info/rfc3954>>.

- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), DOI 10.17487/RFC5246, August 2008, <<http://www.rfc-editor.org/info/rfc5246>>.
- [RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", [RFC 6066](#), DOI 10.17487/RFC6066, January 2011, <<http://www.rfc-editor.org/info/rfc6066>>.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", [RFC 6437](#), DOI 10.17487/RFC6437, November 2011, <<http://www.rfc-editor.org/info/rfc6437>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, [RFC 7011](#), DOI 10.17487/RFC7011, September 2013, <<http://www.rfc-editor.org/info/rfc7011>>.
- [RFC7749] Reschke, J., "The "xml2rfc" Version 2 Vocabulary", [RFC 7749](#), DOI 10.17487/RFC7749, February 2016, <<http://www.rfc-editor.org/info/rfc7749>>.

#### Authors' Addresses

Sheng Jiang (editor)  
Huawei Technologies Co., Ltd  
Q 22, Huawei Campus, No.156 Beiqing Road  
Hai-Dian District, Beijing, 100095  
P.R. China

Email: [jiangsheng@huawei.com](mailto:jiangsheng@huawei.com)

Bing Liu  
Huawei Technologies Co., Ltd  
Q 22, Huawei Campus, No.156 Beiqing Road  
Hai-Dian District, Beijing, 100095

P.R. China

Email: leo.liubing@huawei.com

Jiang, et al.

Expires December 5, 2016

[Page 17]

---

Internet-Draft

Network Machine Learning

June 2016

Panagiotis Demestichas  
University of Piraeus  
Piraeus  
Greece

Email: pdemestichas@gmail.com

Jerome Francois  
Inria  
615 rue du jardin botanique  
54600 Villers-les-Nancy  
France

Email: jerome.francois@inria.fr

Giovane C. M. Moura  
SIDN Labs  
Meander 501  
Arnhem, 6825 MD  
The Netherlands

Email: giovane.moura@sidn.nl

Pere Barlet  
Network Polygraph  
Edifici K2M - Parc UPC  
Jordi Girona, 1-3, Barcelona 08034  
Spain

Email: pbarlet@polygraph.io

