A Framework & System for Classification of Encrypted Network Traffic using Machine Learning

Nabil Seddigh, Biswajit Nandy, Don Bennett, Yonglin Ren, Serge Dolgikh, Colin Zeidler, Juhandre Knoetze and Naveen Sai Muthyala
Solana Networks, Ottawa, Canada
{cnsm2019}@solananetworks.com

Abstract— **Traffic classification solutions are widely used by network operators and law enforcement agencies (LEA) for application identification. Widespread use of encryption reduces the accuracy of traditional traffic classification solutions such as DPI (Deep Packet Inspection). Machine Learning based solutions offer promise to fill the gap. However, enabling such systems to operate accurately in high speed networks remains a challenge. This paper makes multiple contributions. First, we report on the development of MLTAT, a high speed network classification platform which integrates DPI and machine learning and which supports flexible deployment of binary or multi-class classification solutions. Second, we identify a set of robust features which fulfill a dual-constraint - support 10Gbps computation rates and sufficient accuracy in the supervised machine learning models proposed for network traffic classification. Third, we develop a set of labeled data suitable for training the system and a framework for larger scale ground truth generation using co-training. Our findings indicate detection rates around 90% across 8 traffic classes, benchmarked in the system at 10Gbps rates.**

Keywords—Traffic Classification, Encrypted Traffic, Machine Learning, Bid Data, Ground Truth

I. INTRODUCTION

Recent trends to protect privacy have led to the encryption of many popular applications. Currently, 50-70% of Internet traffic [6] and 60% of mobile traffic is encrypted [13], with forecasts estimating a 90% encryption rate in the near future. Existing traffic analysis tools, security monitoring solutions and network probes are less accurate and effective as they fail to classify and filter encrypted traffic. DPI methods, for example, which are highly accurate when classifying unencrypted traffic, are rendered all but useless because pattern matching algorithms are unable to operate on encrypted data.

Multiple studies have examined the efficacy of Machine Learning for classification of encrypted traffic, with promising results. The work carried out in academic environments now requires further research to address issues hindering its transition to real-world usage. The practical application of machine learning techniques to the classification of encrypted network traffic in real world scenarios is a major challenge. Much of the reviewed literature claims high level of accuracy for various methods in different scenarios, but the authors generally do not make their raw data available so that their results may be replicated. Where data is available, solutions tend not to generalize well when applied in a different network or environment. The dearth of such data highlights a second but important need - the challenge of acquiring sufficient, well labeled network classification data with which to train classifiers [1]. Finally, we note that unlike academic environments or offline tools which can base solutions on 100-200 features, real-time requirements for high speed networks, demand a identification of a limited set of features as the basis of classification.

To address the above issues, this paper reports on efforts to build the MLTAT (Machine Learning Traffic Analytics Tool) platform which uses machine learning to classify encrypted network traffic for high speed networks of 10Gbps and beyond. We study whether a limited number of easy-to-compute features can be used as the basis of machine learning classification, while achieving high classification accuracy for multiple different applications. A key part of the work involved creation of labeled network traffic datasets which were used to train the classification models. While some of the data sets were created using scripts and manual effort, several semi-supervised machine learning techniques were investigated to assist with expediting the labeling effort.

This paper is organized as follows. Section 2 discusses related work. Sections 3 and 4 present the MLTAT system and architecture as well as the study carried out using the system. Section 5 provides a description of the research into creating labeled datasets for MLTAT. Section 6 concludes this paper.

II. RELATED WORK

Prior research has studied the use of machine learning for classification of encrypted traffic. Using Weka, the authors [4] classified encrypted applications - SSH and Skype - using C4.5, AdaBoost and Genetic Programming. They concluded that different feature sets were required for different applications to improve the detection accuracy. In [2] the authors used K-means and K-NN clustering with 17 features for real-time classification of encrypted Bit Torrent P2P and Skype traffic on a Cisco platform. They achieved reasonable accuracy with no technical, memory or performance implementation limitations.

In [5] Weka was used to study 5 algorithms (J48, NaiveBayes, NBTree, AdaBoost and LibSVM) applied against 90 features to classify encrypted applications including Gmail, Facebook, iCloud, and Microsoft Update. A key contribution was analysis of the minimum number of packets to be observed for a flow in order to achieve desired accuracy levels. In [8], the authors implement a C5.0 classifier and compare its classification accuracy to 5 different DPI tools when applied to Netflow. In [9] the authors describe TIE, an open traffic classification platform which uses an ensemble of ML methods to combine results from multiple algorithms for improved

978-3-903176-24-9

accuracy. In [11], the authors study the efficacy of PCA feature selection and pre-classification clustering to improve the accuracy of K-NN based classification. In [12] the authors studied the use of semi-supervised classification of encrypted applications with the proposal of a new algorithm to map clusters to target classes. A key highlight of their work is the requirement for a small amount of labeled training data. In [14], using Adaboost and C5.0, the authors studied the efficacy of features such as flow burstiness and periods of inactivity (idle_time) for accurate classification.

III. MLTAT System & Architecture

The MLTAT system depicted in Fig 1 ingests packet capture files, then processes and classifies the data to produce output files. The system integrates open source including Apache Spark, HDFS (Hadoop Distributed File System), HBase and Nginx. MLTAT has two primary functions: it is an experimental tool allowing users to train, tune, and validate machine learning models for network flow classification, and a powerful engine to use those models in production environments. Individual components were benchmarked and most scaled to 10Gbps and some beyond 100Gbps. The system was developed in Python with File Parsing and Feature Computation rewritten in C to achieve desired performance.

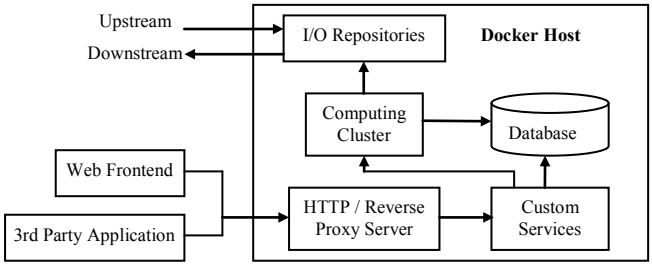

Fig. 1. MLTAT System Architecture.

Convenient user interfaces and APIs are provided to train new classification models using training data (labeled network traffic). Users are able to select a machine learning algorithm, specify search criteria for hyper-parameters, and selectively include or exclude descriptive features. Given these inputs, the system performs grid-search cross-validation to find the best set of hyper-parameters and evaluates training using held-out test data. Accuracy and a confusion matrix for the trained classifier are presented, allowing one to evaluate performance. Once satisfied with performance, classifiers can be enabled to make predictions on new data.

Many classifiers, using different feature sets/algorithms, and trained on different datasets, can be handled by the system. This flexibility allows creation of classifiers specifically targeted for particular applications. Combined with the options for ensemble learning, one may deploy a single binary or multiclass classifier, many binary classifiers using the one-vs-one or one-vs-all methods, or some other useful combination.

A. Machine Learning Algorithms for Classification

Based on findings in previously published studies on network traffic classification [5][10] and our own experiments, a number of supervised machine learning algorithms were integrated into MLTAT including Logistic Regression, Support Vector Machines, Decision Trees, Adaboost, Neural Networks and Naive Bayes. Hyper-parameters for each of these algorithms are exposed by MLTAT, providing a convenient mechanism for experimentation with different models. The system performs grid-search cross-validation to find the best set of hyper-parameters among those suggested by the user.

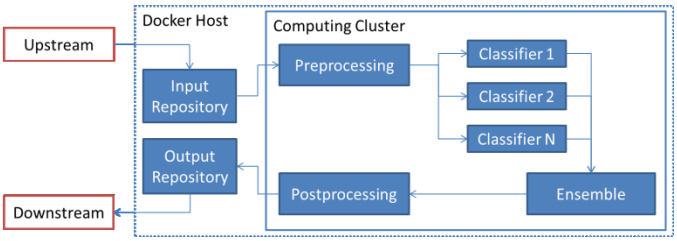

Fig. 2. Classification Pipeline

As illustrated in Fig. 2, MLTAT uses a form of bagging to combine the predictions of several independent classifiers into a single result. Two forms were implemented: simple majority vote and a weighted vote based on the confidence of each individual classifier.

B. Flow Determination & Direction

In MLTAT, flows are considered to be bidirectional and identified by the unique 5-tuple (source IP, destination IP, source port, destination port and protocol). Features are computed for both the forward and reverse direction of flows. For TCP flows, the start of the flow is identified when the SYN packet is observed while flow end is detected via observation of a FIN. An MLTAT configurable timeout enables end of flow detection for UDP flows and when the TCP FIN packet never arrives. Feature computation required identification of forward vs reverse directions of a flow with heuristics developed to handle various situations which are encountered.

C. Features

Determining which features to support in MLTAT required research. Requirements to support 10Gbps and beyond in semi real-time meant that it was not possible to compute a list of 200+ features and use feature selection to short-list the features of interest. We sought to identify a minimal set of features that can be computed at high speed during classification and used for prediction. Flow features implemented by MLTAT include: (a) Minimum, maximum, mean, and variance of packet size in both directions (b) Minimum, maximum, mean, and variance of packet inter-arrival time in both directions (c) Total flow duration (d) Protocol (e) Total packet, byte, and payload count in both directions (f) Entropy of packet size and inter-arrival time in the "backward" direction.

IV. MLTAT Study & Evaluation

A. Experimental Setup & Dataset

Studies were undertaken using MLTAT to classify traffic collected using the dataset generated during this project - described in section V. The dataset included 2170 PCAP files containing 354,808 flows representing 8 traffic classes (9 classes if counting DNS). The data was divided into training and testing flows with the former used for model training and the latter used for prediction. In total, this included audio chat

(12,666 flows), audio stream (71716 flows), file transfer (3867 flows), P2P (14,758 flows), text chat (9,097 flows), video chat (3170), video stream (36,648 flows), web traffic (8,230 flows), and DNS (194,656 flows).

We used weighted precision as a metric due to the imbalanced data set with TP, FP, and FN as the total True Positives, False Positives and False Negatives respectively:

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$WeightedPrecision = \frac{TP}{TP + \sum_{i=1}^{n} FP_i \times \frac{Total\ Flows_{main}}{Total\ Flows_i}}$$

B. Binary Classification

In this approach, binary classification was used to classify the main application with all other applications labeled as a background class. There was one such model configured for each different application using the One-vs-Rest methodology. For example, for Video Stream, the first configured training model consisted of video stream traffic with all other traffic labeled as background/other traffic. The results of the individual binary classifiers were combined into a final result. The final recall score presented for each class of traffic is thus MLTAT's fused result of the individual 5 algorithms listed earlier (C4.5, SVM, MLP, Adaboost, Logistic Regression).

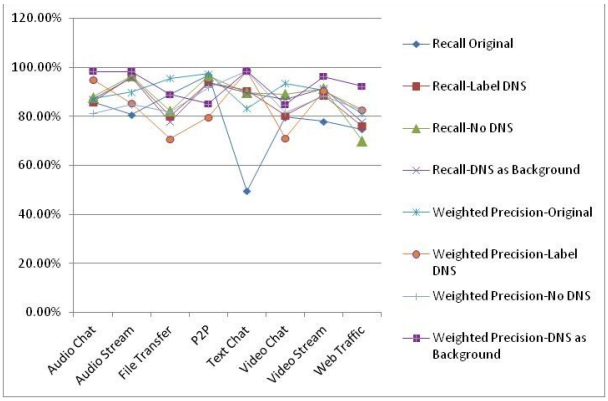

Fig. 3. Binary Classification

The training dataset originally included DNS flows generated by the applications. We wish to understand whether their exclusion improves results. Multiple scenarios were run: (i) The initial approach included DNS as part of the primary class of traffic; (ii) All the DNS flows were labeled separately to create a separate DNS traffic class; (iii) DNS flows were removed; (iv) DNS flows were extracted from the main class and labeled as part of the background class of traffic.

Fig. 3 presents the recall and weighted precision scores for the 4 scenarios listed earlier. In terms of overall weighted accuracy, the original, label-DNS, no-DNS and DNS-as-background tests yielded 79.3%, 86.4%, 87.9% and 87.5%. The DNS-as-background scenario (4th case) yields the best overall results when considering recall and weighted precision.

C. Multi-Class Classification

Another set of experiments used MLTAT to carry out multi-class classification using C4.5, where the models classified multiple classes of traffic at once. Recall and weighted precision scores were: Audio Chat (84.35% and 86.16%), Audio Stream (91.09% and 90.72%), DNS (99.82% and 98.17%), File Transfer (72.55% and 86.77%), P2P (96.57% and 87.16%), Text Chat (92.29% and 96.59%), Video Stream (91.93% and 85.71%). Most traffic classes have recall above 90% and minimum weighted precision of 85.7%. We ascertain also an overall accuracy of 90% and a weighted accuracy closer to 88% (when excluding DNS) - the latter is examined due to the imbalanced data set.

V. Groundtruth / Training Data

Currently, there are very few publicly available labeled datasets which include different network traffic types and applications. This limitation hampers research and development efforts, specifically related to traffic classification algorithms. In this work, we developed a mechanism that allows for labeling of collected (encrypted and non-encrypted) network traffic into different traffic classes (e.g., video, audio) and applications (e.g., Netflix, Youtube, Skype, DropBox).

Labeled traffic is essential for the training of Supervised Machine Learning based Classifiers. In addition, such data can serve as ground truth to be used for the evaluation and verification of traffic classifier systems and algorithms. Recently Amazon launched SageMaker Ground Truth [15], a service that allows labeling of Texts and Images. However, it is not suitable for labeling network traffic traces.

Two categories of labeled datasets are created: (1) Type-1 datasets include traffic from a single application (2) Type-2 datasets include traffic representing multiple applications collected from an enterprise or campus network. Type-2 datasets are labeled based on semi-supervised learning techniques. A small number of a priori labeled flows are required for the semi-supervised algorithms. The Type-1 labeled flows serve as the a priori labeled flows for this purpose. It is assumed that the characteristics of Type-1 flows are similar to the Type-2 traffic that is being labeled.

A. Type-1 Dataset Creation

In this study, Type1 traffic was generated from a desktop for a particular application and captured at the same location using tcpdump. The collected data was automatically labeled with a given traffic class or application category, as listed in TABLE I. The 10 selected classes constitute some of the dominant traffic classes currently observed in networks.

TABLE I. Traffic Classes - Type 1 Data

	Traffic Classes	Applications
1	Video Streaming	YouTube, Netflix
2	Video Chat	Skype, Messenger
3	Audio Streaming	Spotify, SoundCloud
4	VoIP	Skype, Messenger
5	File Transfer	Dropbox, Google Drive
6	Mail	Gmail, Yahoo
7	Web browsing	Firefox, Chrome
8	P2P	BitTorrent, eDonkey
9	Chat Messages	Facebook, Telegram
10	ToR Traffic	Video streaming, Web browsing

Many issues are encountered during large scale data capture for labeling. The primary objective is to keep the data as clean as possible (i.e., avoid background noise generated by various applications including multicast and broadcast traffic). Data was captured on a Linux platform which proved to be a better option than Windows since disabling of background processes is easier on Linux. If Virtual Machines (VM) are used it is important to configure the VM's network settings in bridged mode. We addressed the problem of buffering at the NIC card by modifying the NIC parameters to disable TSO and LRO such that the Ethernet NIC always sends packets it receives from the wire to the TCP stack without buffering.

We note that it is extremely difficult to capture payload-only traffic even after the steps outlined were taken. There are various advertisements and signalling packets that inevitably become part of the data capture. The difficulty arises in particular because traffic is often encrypted.

B. Type-2 Dataset Creation

Semi-supervised learning has been used in many areas where labeled data is hard to obtain, but abundant unlabeled data exist. Our proposed approach to generate labeled data relies on a semi-supervised approach referred to as Co-training.

The co-training approach is built on self-learning with improved performance as presented in [3] and [7]. First, we conducted significant feature analysis, based on domain knowledge and characteristics of a variety of features. In the feature selection phase, we chose two fully independent subsets from the available features: (*i*) packet-related features, (*ii*) time-related features. Subsequently, co-training follows the concept of self-learning, to add the most trustworthy unlabeled data with their predicted labels into the training dataset. Thus, the training dataset grows with high confidence, and increasingly larger volumes of unlabeled data is converted to labeled data.

We built two classifiers for classification. Multiple machine learning algorithms and classification methods were studied and *Random Forest* (*RF*) which utilizes a bagging approach was selected for the classifier due to its accuracy and performance. *RF* randomly selects a sample from the training set and also selects a subset from the features. The process is repeated and then the classification results aggregated.

There are multiple phases in the co-training approach. Experimentally, we established five phases for co-training. In each phase, different criteria are utilized based on different confidence levels. The higher confidence of classified results at in earlier phases ensures that the most accurate classification results are included in the training set.

Within each phase, it trains two classifiers (C_1 and C_2), obtains the corresponding predicted labels (L_1 and L_2), and takes advantages of different confidence levels (P_1 and P_2) as the thresholds to absorb the unlabeled data into the training set. For instance, in the first phase, if the predicted labels from two classifiers are identical ($L_1 = L_2$) and both have high confidence (eg $P_1 \geq 80\%$ and $P_2 \geq 80\%$) , such flows are added to the training set. Similarly, in the second phase, if the predicted labels are the same ($L_1 = L_2$) and either of them has high confidence ($P_1 \geq 80\%$ or $P_2 \geq 80\%$), such flows are added. In the third phase, for any label (i.e., $L1 \neq L2$) flows are selected with the same confidence level as the previous phase. In subsequent phases, flows are selected with lower confidence ($P_1 \geq 70\%$ or $P_2 \geq 70\%$). In the final phase, we make use of three different classifiers: *Random Forest*, *Neural Networks* and *AdaBoost*, with a majority voting mechanism to determine the labels for all remaining flows. Since the criteria and conditions are experimental, different users can choose their own criteria in a flexible way, based on their requirements and scenarios.

C. Test Results

We carried out a series of experiments to validate the effectiveness of the co-training (CO) approach. The selected dataset includes a mix of 6 traffic types (Video Streaming, Audio Streaming, VoIP, P2P, Chat Messages, Web Browsing) representing Type-1 data as listed in TABLE I. Three sets of experiments were performed with different volumes of training data. Training data constituted 20%, 10% and 5% of the total flow volume in the dataset. The precision and recall scores were evaluated per traffic class. We observe that all classes have a precision and recall score above 88%, except Web Browsing. We note that with decreased training data, the precision and recall score for each class decreases. In particular, the *Web Browsing* score dropped significantly. The overall accuracy of labeling for the six traffic types are 93.14%, 91.61% and 88.77%, with 20%, 10% and 5% training data respectively. Thus, the overall accuracy of this approach is high but there is room for improvement due to the requirement for high accuracy of labeled data.

For the experiment using co-training with 5% training data, 50.85%, 20.52%, 12.58% and 3.88% flows are labeled during Phase I, Phase II, Phase III, and Phase IV respectively. The remaining flows are labeled during the last phase (Phase V). The proposed co-training approach execution time is approximately 70 minutes on an Intel i5-8600 @ 3.1GHz system with 16 GB RAM (for the dataset consisting of 46,500 flows).

VI. Conclusion

The MLTAT platform was developed to support network traffic classification of encrypted traffic using machine learning. The system supports a restricted short-list of 30 features which are easily implemented in real-world systems and whose computation can scale to 10Gbps and beyond in high speed data planes. Using co-training based semi-supervised machine learning we created labeled data sets representing different encrypted applications, with which the machine learning network classification models can be trained. We evaluated the accuracy of the system using binary and multi-class classification models with the observation that the former provided better overall results when tested using the dataset from this project.

Acknowledgment

We acknowledge that some of this work was carried out under the CSSP program led by Defence Research and Development Canada's Centre for Security Science, in partnership with Public Safety Canada. Project partners also included the RCMP.

REFERENCES

[1] F. Gingoli et al, "GT: Picking up the Truth from the Ground for Internet Traffic", ACM Computer Communication Review, Vol 39 #5, Oct 2009

[2] R. Bar-Yanai, M. Langberg, D. Peleg, L. Roditty, "Realtime classification for encrypted traffic.", In proceedings of SEA 10 - 9th Int Conference on Experimental Algorithms, Naples, Italy, May 2010.

[3] X. Li, F. Qi, D. Xu and X. Qiu "An internet traffic classification method based on semi-supervised support vector machine",. IEEE International Conference on Communications, ICC 2011, pp. 1-5, 2011

[4] R. Alshammari and N. Zincir-Heywood, "Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?", Journal of Computer Networks, Volume 55, #6, April 2011

[5] WC Barto, "Classification of Encrypted Traffic Using Machine Learning Algorithms", Master of Science Thesis, Dept of Electrical & Computer Engineering, Air Force Institute of Technology, Ohio, June 2013

[6] "Shine a light on the darkening Internet: How to thrive despite Encryption - an exploration of the impact of encryption on ANI use cases", Sandvine Whitepaper, October 2018

[7] J. Yan, X. Yun, Z. Wu, H. Luo, S. Zhang, S. Jin and Z. Zhang, "Online traffic classification based on co-training method",. 13th Int PDCAT Conference, pp. 391-397. 2012.

[8] T. Bujlow, V. Carela-Español and P. Barlet-Ros, "Comparison of Deep Packet Inspection (DPI) Tools for Traffic Classification", Technical Report version 3, UPC-DAC-RR-CBA-2013-3 ed.) Universitat Politècnica de Catalunya, Barcelona, Spain, June 2013

[9] W. de Donato, A. Pescapé and Dainotti, "Traffic Identification Engine: An Open Platform for Traffic Classification", In Journal of IEEE Networks, Volume 28, #2, March 2014

[10] R. Alshammari. and N. Zincir-Heywood, "How Robust Can a Machine Learning Approach Be for Classifying Encrypted VoIP?', Journal of Network and Systems Management, Volume 23, Issue 4, October 2015,

[11] T. Wiradinata and A. Paramita, "Clustering and Feature Selection Technique for Improving Internet Traffic Classification Using K-NN", Journal of Advances in Computer Networks, Volume 4 #1, March 2016

[12] T. Glennan, C. Leckie and S. Erfani, "Improved Classification of Known and Unknown Network Traffic Flows Using Semi-supervised Machine Learning", ACISP 2016 Conference, Melbourne, Australia, July 2016

[13] "5 Traffic Management Trends that will Shape the Mobile Industry", Industry Report, OpenWave, January 2017

[14] H. Oudah, B. Ghita and T. Bakhshi, "A Novel Feature Set for Internet Traffic Classification using Burstiness", 5th Int Conf on Info Systems Security & Privacy, ICISSP, Prague, Czech Republic, Feb 2019

[15] "Amazon SageMaker Ground Truth", https://docs.aws.amazon.com/sagemaker/latest/dg/sms.html. Accessed on 12th June, 2019