Machine Learning based IP Network Traffic Classification using Feature Significance Analysis

Te-Shun CHOU
Department of Technology Systems, East Carolina University
Greenville, NC, U.S.A.

John PICKARD
Department of Technology Systems, East Carolina University
Greenville, NC, U.S.A.

Ciprian POPOVICIU
Nephos6 Inc.
Raleigh, NC, U.S.A.

ABSTRACT

After over 30-year deployment, IPv4 addresses are running short on supply with the growth of the Internet so dynamic. A new technology will take its place, IPv6, an evolution from IPv4 that includes virtually unlimited address space. However, it will take time to totally transit from IPv4 to IPv6. IPv6 will coexist with IPv4 for a period of time and then eventually replace IPv4.

This paper studied network traffic that included information of both IPv4 and IPv6. The traffic was collected from 600 US government websites that were all reported to have Domain Name Services (DNS) and Web services accessible over IPv4 and IPv6. Cloud based, Internet distributed monitoring agents were deployed in eight geographic locations to collect data. Both feature selection algorithms, filter and wrapper, were applied to the dataset and the classification accuracy was then studied. The results showed that feature selection algorithms effectively reduced the complexity of the classification model. The results also confirmed that the reduced feature set contributed a superior classification performance over full feature set.

Keywords: Feature selection, machine learning, classification, IPv4 and IPv6

1. INTRODUCTION

Currently there are two versions of the Internet Protocol (IP): IPv4 and IPv6. The version number refers to the version identification field in the IP header. IPv4 was designed as an experimental protocol as part of the Advanced Research Projects Agency Network (ARPANET) long before the Internet was commercialized. Over time as the Internet has grown in size and complexity, the inherent technological and security limitations of IPv4 have made it anachronistic and unsuitable for today's Internet requirements.

The chief technological limitation of IPv4 is the number of unique IP addresses that it is capable of supporting. With a 32-bit address field, IPv4 is only capable of supporting 4.3 billion unique addresses. Four of the five Regional Internet Registries (RIRs) have already exhausted their pools of IPv4 addresses, meaning that they can no longer accommodate requests from service providers or enterprise organizations for more IPv4 addresses.

In 1993, the Internet Engineering Task Force (IETF) began working on a successor to IPv4 and in 1995 standardized IPv6. The engineers of IPv6 took the opportunity to add new features and functionality based on the lessons of 30 years of experiences with IPv4. The address fields in IPv6 are increased to 128 bits which will support many more addresses than IPv4 and makes the addressing space much more scalable. The vast size of the IPv6 address space allows organizations to develop an addressing plan that is hierarchical and allows for flexibility and growth. Aggregation is simplified because there no need to conserve address space as is done with IPv4 Variable-Length Subnet Masking (VLSM). Additionally, the size and scalability of the multicast addressing space is greatly improved and a simpler mechanism for auto-configuration of addresses through Stateless Address Auto Configuration (SLAAC) is added.

In the paper, we analyzed a network traffic that included both IPv4 and IPv6 information. The network traffic was collected from forty-eight agents distributed in eight cities across three continents. The dataset included a significant amount of traffic records with a number of various features such as ping time and connection time. Having reduced the size of the dataset by using with the use of feature selection algorithms, a comparative study of decision tree based classifier was then studied in order to find the most informative features.

This paper is organized as follows: Section 2 demonstrated the experimental methodology. Section 3 presented the discussion of the experiment results. Finally, we concluded our work in the last section.

2. EXPERIMENTAL METHODOLOGY

A list of Federal Executive Agency Internet Domains was obtained from the 2016 listing of domains published by the General Services Administration (GSA) as .gov Domains dataset [1]. At the time of study, the dataset contained 1,315 domains which were downloaded as a .csv file.

Using a custom script, each of the 1,315 USGA domains was evaluated for the presence of a DNS AAAA record by sending an AAAA record query to Google's DNS resolver at 8.8.8.8. Of the 1,315 USGA domains, 600 domains (45.62%) returned AAAA records. Each of the 600 sites that returned a AAAA record was then polled at 15-minute intervals for a period of 30 days, from 48 network monitoring ITSonar [2] agents deployed

in 8 geographically distributed locations. The 48 ITSonar agents were deployed in Virtual Machines (VMs) hosted by Digital Ocean [3]. The number and location of the virtual machines are shown in Table 1.

Table 1. Agents by Geographic Location

Location	Continent	Number of Agents
Singapore (sgp)	Asia	6
Toronto (tor)	North America	6
Frankfurt (fra)	Europe	6
New York City (nyc)	North America	6
San Francisco (sfo)	North America	6
Amsterdam (ams)	Europe	6
Bangalore (bgl)	Asia	6
London (lon)	Europe	6

Each VM hosted by Digital Ocean ran on CentOS 7.3 with 1 CPU, 512MB of memory, 20GB of storage on a Solid-State Drive (SSD), 1TB of transfer data, and enabled for both IPv4 and IPv6. Each of the 6 agents at each geographic location were configured to gather IPv4 and IPv6 network data of 100 USGA web services domains so that data from all 600 USGA is collected from each geographic location.

IPv4 and IPv6 data is gathered from each agent for each USGA domain, which include: PING time, TCP/IP connect time (3-way handshake), DNS query time, HTTP download time for all elements of the domain web site, Traceroute path, and Autonomous System (AS) path.

The data set used for analysis consisted of data collected from July 31, 2017 to August 30, 2017. It included 4,800 network traffic connections and each was composed of 30 features. Table 2 illustrates the first three network traffic connections in the dataset and the features are shown in Table 3.

The type of feature is either discrete or continuous, i.e., the former is a qualitative scale and the latter is quantitative.

- Qualitative scales: The values were simply labels without any order involved. For example, the value of feature *Service_URL* is the website where the agents visited. The value of feature *Test_Type* was one of the symbolic set {HTTP, HTTP-PING}.
- Quantitative scales: The data was characterized by numeric values. *IPv4_Connect_Time_Min_ms* was an example which represented the time elapsed while the agent connected to a target website when using IPv4 protocol.

Table 2. The first three network traffic connections

Connection	Data
1	www.highperformancebuildings.gov,http://www.highperformancebuildings.gov,ecu-cet-sgp-01-02,100,100,1,8913,788.3,2,7454,382.16,0,7188.06,48.82,230.63,12033.44,431.38,0,16928.96,2531.33,0,18581.85,4107.33,-1,-1,-1,-1,-1,-1,HTTP
2	www.highperformancebuildings.gov,http://www.highperformancebuildings.gov,ecu-cet-tor-01-02,100,100,0,5027,142.84,0,2157,87.2,0,38.64,0.09,24.67,1726.81,75.91,144.18,11546.91,439.96,309.06,6767.55,741.98,-1,-1,-1,-1,-1,-1,HTTP
3	www.highperformancebuildings.gov,http://www.highperformancebuildings.gov,ecu-cet-fra-01-02,100,100,7,15016,749.99,8,8471,287.8,0,3916.75,5.67,87.11,15098.73,219.16,0,15866.33,1509.66,0,17365.19,1989.5,-1,-1,-1,-1,-1,-1,HTTP

Table 3. Features of the network traffic connections

Feature	Data type
Service_Name and Service_URL	Discrete
Agent_Name	Discrete
IPv4_Uptime_Percentage and IPv6_Uptime_Percentage	Continuous
IPv4_DNS_Time_Min_ms, IPv4_DNS_Time_Max_ms, and IPv4_DNS_Time_Average_ms	Continuous
IPv6_DNS_Time_Min_ms, IPv6_DNS_Time_Max_ms, and IPv6_DNS_Time_Average_ms	Continuous
IPv4_Connect_Time_Min_ms, IPv4_Connect_Time_Max_ms, and IPv4_Connect_Time_Average_ms	Continuous
IPv6_Connect_Time_Min_ms, IPv6_Connect_Time_Max_ms, and IPv6_Connect_Time_Average_ms	Continuous
IPv4_Load_Time_Min_ms, IPv4_Load_Time_Max_ms, and IPv4_Load_Time_Average_ms	Continuous
IPv6_Load_Time_Min_ms, IPv6_Load_Time_Max_ms, and IPv6_Load_Time_Average_ms	Continuous
IPv4_Ping_Time_Min_ms, IPv4_Ping_Time_Max_ms, and IPv4_Ping_Time_Average_ms	Continuous
IPv6_Ping_Time_Min_ms, IPv6_Ping_Time_Max_ms, and IPv6_Ping_Time_Average_ms	Continuous
Test_Type	Discrete

The entire work in this research included two stages of experiment: one was feature selection and the other was classification. Generally, the algorithms of feature selection are divided into two main categories, filter and wrapper, as defined in the work of John et al. [4]. Filter method operated without engaging in any information of induction algorithm. The prior knowledge, features, should have strong correlation with the target class, or should un-correlate to each other, and the filter method selects the best subset of features. Correlation based Feature Selection (CFS) [5] and Fast Correlation-Based Filter (FCBF) [6] were two examples of filter-based feature selection method.

On the other hand, wrapper method employed a predetermined induction algorithm to find a subset of features with the highest evaluation by searching through the space of feature subsets and evaluating quality of selected features. The process of feature selection acted like “wrapped around” an induction

algorithm. Machine learning algorithms such as ID3 [7] and C4.5 [8] were commonly used as the induction algorithm.

During the first stage, we applied both filter and wrapper feature selection algorithms to reduce the number of features in the dataset. Having acquired the reduced data set, the data was then fed into classifier in the second stage.

3. EXPERIMENTAL RESULTS

Weka [9] was employed in the research, which has been popularly adopted in the research of data mining and machine learning. In the first stage of the experiment, *Agent_Name* was selected as the target class; uptime percentage, DNS query and answer, TCP/IP connection, website reachable and download time, and test type were selected as features and each was normalized from 0 to 1. InfoGainAttributeEval with Ranker search method was chosen as the filter feature selection algorithm. It was a statistical measure to find a ranked list of the most predictive features by measuring the information gain with respect to the class.

For the wrapper feature selection, WrapperSubsetEval was selected that used J48 as a "wrapped around" induction algorithm and GreedyStepwise as the search method to find the most predictive features to class. The left of Table 4 shows the eight most discriminative features based on the values of information gain entropy in decreasing order after filter search. The right of Table 4 shows the optimal feature subset that was most relevant to the class after wrapper search. There were five features shown on both and they have been marked with an underline.

Table 4. Selected features

Filter: InfoGain/ Ranker	**Wrapper: J48/GreedyStepwise**
IPv6_Ping_Time_Min_ms	IPv6_Ping_Time_Min_ms
IPv6_Ping_Time_Average_ms	IPv6_Ping_Time_Average_ms
IPv6_DNS_Time_Min_ms	IPv6_DNS_Time_Min_ms
IPv6_Connect_Time_Min_ms	IPv6_Connect_Time_Min_ms
IPv4_DNS_Time_Min_ms	IPv4_DNS_Time_Min_ms
IPv4_Ping_Time_Min_ms	IPv4_Connect_Time_Max_ms
IPv4_DNS_Time_Max_ms	IPv4_Connect_Time_Average_ms
IPv4_Ping_Time_Average_ms	Test_Type

Having acquired the reduced data sets through the feature selection process, both full data set including 27 features and reduced data sets including 8 features were then fed into classifier in the second stage. Weka has implemented a number of classification algorithms. In the paper we used J48 as the classifier to categorize the network traffic connections to one of eight *Agent_Name*. The J48 algorithm implemented in Weka for building decision trees is based on C4.5. It is able to build decision trees from a set of training data using the concept of information entropy. By applying 10-fold cross-validation evaluation on each data set, standard measurements of classification accuracies are reported. Confusion matrix is shown in Table 5 and the denotations of *True Positive* (*TP*), *True Negative* (*TN*), *False Positive* (*FP*), and *False Negative* (*FN*) are defined. Table 6 summarizes the classification results. The highest *TPR*, the highest *PR*, and the lowest *FPR* were marked bold. The lowest *TPR*, the lowest *PR*, and the highest *FPR* were marked bold.

Table 5. Confusion matrix

		Actual Result	
		Positives	Negatives
Predicted Result	Positives	**TP** True Positive	**FP** False Positive
	Negatives	**FN** False Negative	**TN** True Negative

- *True Positive* (*TP*): Number of network traffic connections predicted positive that are actually positive
- *False Positive* (*FP*): Number of network traffic connections predicted positive that are actually negative
- *False Negative* (*FN*): Number of network traffic connections predicted negative that are actually positive
- *True Negatives* (*TN*): Number of network traffic connections predicted negative that are actually negative

$$TPR = \frac{TP}{TP + FN} \quad (1)$$

$$FPR = \frac{FP}{TN + FP} \quad (2)$$

$$PR = \frac{TP}{TP + FP} \quad (3)$$

Table 6. Classification results

	TPR			FPR			PR		
Location	Full	Filter	Wrapper	Full	Filter	Wrapper	Full	Filter	Wrapper
sgp	**0.897**	0.908	0.892	0.011	0.01	0.014	**0.918**	0.925	0.899
tor	0.848	0.852	0.845	0.029	0.021	0.021	0.805	0.850	0.852
fra	0.86	0.868	0.885	0.021	0.014	0.019	0.851	0.898	0.872
nyc	0.84	0.867	0.857	0.019	0.021	0.023	0.866	0.857	0.841
sfo	0.765	0.79	0.788	0.035	0.022	0.022	0.760	0.839	0.834
ams	0.88	0.883	0.875	0.018	0.026	0.025	0.874	0.829	0.832
bgl	0.892	**0.932**	**0.935**	**0.014**	**0.009**	**0.011**	0.899	**0.936**	**0.926**
lon	0.795	0.842	0.783	0.027	0.028	0.028	0.807	0.812	0.802
Average	0.847	0.868	0.858	0.022	0.019	0.020	0.848	0.868	0.857

The agents deployed in San Francisco (sfo) had the worst classification performances of *TPR*, *FPR*, and *PR* and Singapore (sgp) had the best overall classification accuracy among the eight cities. For both reduced datasets, after using filter and wrapper feature selection algorithms, the agents deployed in Bangalore had the best classification results, which had the highest *TPR* and *PR* and the lowest *FPR*. On the contrary, London had the worst classification performance except the *TPR*. As for the overall classification performance, both filter and wrapper feature selection algorithms showed higher accuracies in comparison with the outcomes that used full feature dataset.

When studying the decision tree structure, both classification models after filter and wrapper feature selection algorithms had the same topmost root node (*IPv6_Connect_Time_Min_ms*) and the same two child nodes (*IPv4_DNS_Time_Min_ms* and *IPv6_DNS_Time_Min_ms*) in the following branches, as shown in Figure 1. The root node of the model using full dataset was different, which was *IPv6_Connect_Time_Max_ms* and the node in the following layer was *IPv4_DNS_Time_Min_ms*, as shown in Figure 2.

Figure 1. The decision tree (top two layers) of classification models using reduced datasets after filter and wrapper feature selection algorithms

Figure 2. The decision tree (top two layers) of classification model using full feature dataset

4. CONCLUSIONS

Feature selection methods were used to identify and remove irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model. The utilization of both filter and wrapper feature selection algorithms proved to reduce the complexity of the model from 27 features to 8 features. The experimental results demonstrated that select 8 informative features contributed a superior classification performance over 27 features. The results also showed that *IPv6_Connect_Time* was the most important feature in the task of classification because it was the root node of the decision tree regardless using the reduced dataset or the full dataset. In the future, different feature selection and machine learning techniques will be applied for improving classification accuracy.

ACKNOWLEDGEMENT

The authors would like to thank Nephos6 for their support and assistance with this project and for their appreciation of the ITSonar platform. The authors would also like to thank the support of Department of Technology Systems in College of Engineering and Technology at East Carolina University.

REFERENCES

1. U.S. Government domain dataset. https://catalog.data.gov/dataset/gov-domains-api-c9856
2. ITsonar. https://www.nephos6.com/
3. DigitalOcean Cloud Computing - Servers, Storage & Hosting. http://www.digitalocean.com
4. G. John, R. Kohavi, and K. Pfleger, “Irrelevant Features and the Subset Selection Problem,” Proceedings ML-94, pp. 121-129, Morgan Kaufmann, 1994.
5. M. A. Hall, "Correlation-based Feature Selection for Machine Learning," Dissertation, University of Waikato, 1997.
6. L. Yu and H. Liu, “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution,” Proceedings of The Twentieth International Conference on Machine Leaning, pp. 856-863, Washington, D.C., August, 2003.
7. J. R. Quinlan, “Induction of Decision Trees,” Machine Learning, Volume 1, pp. 81-106, 1986.
8. J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
9. Weka. http://www.cs.waikato.ac.nz/ml/weka/