Predicting Network Traffic Using TCP Anomalies

Alina Lazar
Department of CSIS
Youngstown State University
Youngstown, OH
alazar@ysu.edu

Kesheng Wu, Alex Sim
Lawrence Berkeley National Laboratory
Berkeley, CA
kwu, asim@lbl.gov

***Abstract*—Accurately predicting network traffic volume is beneficial for congestion control, improving routing, allocating network resources and network optimization. Traffic congestion happens when a network device is receiving more data packets than its processing capability. The number of retransmissions per flow, packet duplication and synthetic reordering can seriously degrade the overall TCP performance. An unsupervised/supervised technique to accurately identify TCP anomalies occurring during file transfers based on passive measurements of TCP traffic collected using `Tstat` is proposed. This method will be validated on real large datasets collected from several data transfer nodes. The preliminary results indicate that the percentage of TCP anomalies correlate well with the average throughput in any given time window.**

***Keywords*-Network traffic; TCP performance; k-means; classification; `Tstat`**

I. Introduction

Large scientific facilities use Science DMZ, which includes several dedicated data transfer nodes, and high performance data movement tools, to attain high network transfers for high performance scientific applications. Network traffic prediction plays a vital role in maintaining healthy operations within all varieties of complex and diverse computer networks. Online traffic monitoring information, collected over time, can be used to predict future traffic volume and unexpected events in real-time.

Predicting future traffic has been addressed in the past mostly via time series forecasting by building regression models capable of drawing accurate correlation between future traffic volume and previously observed traffic volumes. In contrast to time series methods, other machine learning methods have been proposed to identify bottlenecks and explain the status of network traffic using features from passive network measurements.

`Tstat` is one available tool for monitoring the network traffic. It computes over one hundred different performance statistics at both the IP and TCP layers. At the large scientific facility 90K of TCP flows are collected per node daily and a total of 10GB of compressed data logs yearly. Recently, Hidden Markov Model and Recurrent Neural Networks have been proposed [1] to predict network traffic volume from some flow statistics, such as flow counts per time interval. These flow statistics are easier to compute compared to network throughput.

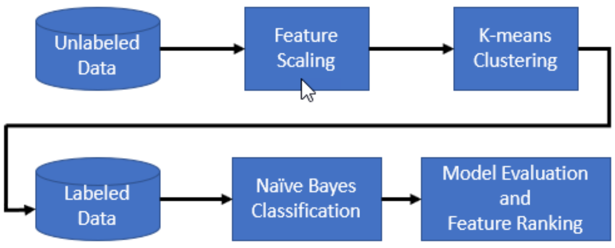
Unlabeled Data
Feature Scaling
K-means Clustering
Labeled Data
Naïve Bayes Classification
Model Evaluation and Feature Ranking

Fig. 1. Proposed machine learning guided methodology for identifying and categorizing anomalous flows.

It is well-known that TCP anomalies such as packet loss contributes to the variance of network throughout [2]. Therefore, it is essential to be able to correctly identify all these the anomalies. Previous research [3], [4] reported statistical correlation between multiple variables collected in the `Tstat` logs and the network traffic throughput.

II. Methods

Traffic flows collected in the `Tstat` logs have no feature or variable to designate them as anomalies. However, applying a combination of machine learning algorithms can predict which flows are normal and which are not. Specifically, first a clustering approach [4] automatically identifies two separate groups of homogeneous traffic flows with similar characteristics in terms of their features. Second a fast reliable supervised approach is used to classify flows on the fly and assign them to one of the two clusters. Valuable information about the main characteristics of each class can be derived from the most important features. The system computes the percentage of the normal class to check whether it models the average throughput. All the steps of this approach are highlighted in Fig. 1.

The proposed method employs a k-means clustering algorithm with two clusters to identify the number of TCP anomalies in any given time window and to compute the percentage of anomalies. K-means is an iterative clustering algorithm that takes as parameter the required number of clusters. At each step the cluster centroids are computed first and then all the data points are assigned to one of the clusters. The process stops when there is no significant change in the

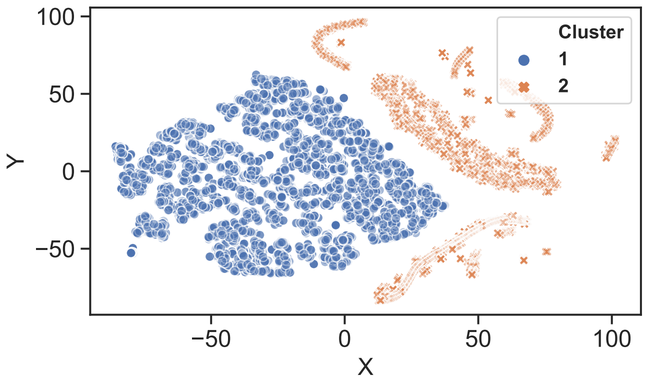

Fig. 2. t-SNE 2-dimensional visual representation of the network traffic flows collected during one day.

computed cluster centroids.

To understand how well, clustering works for this particular dataset a two-dimensional representation of data collected during one day is presented in Fig. 2. T-SNE [5] is a dimensionality reduction and visualization method that works well for non-linear datasets. Fig. 2 shows a clear delimitation between the two classes of network traffic flows computed by k-means.

As the next step an online classification algorithms from Spark.ML [6] is used to build a model using the clustering labels to predict the TCP anomalies computed in the next time windows. Naive Bayes is a simple but efficient algorithm based on conditional probability distributions. This algorithms have the robust ability to capture and model the nonlinearity of large time dependent datasets.

Feature ranking is completed using a nonlinear dimensionality reduction method based on an ensemble of randomized decision trees. These extra-trees are built on various subsamples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. This method can be used to rank the features in terms of their importance in the classification task. When applied to our labels dataset the first 5 most important features are shown in Table I.

To measure the correlation between the throughput and the percentage of TCP anomalies the root-mean-square error (RMSE) and the Kolmogorov-Smirnov test were used.

III. Experiments and Results

For this project we used `Tstat` data containing 104 features and almost 2 million flows, collected from four data trasfer nodes, during one calendar month, from May 27, 2017 until June 28, 2017. The flows are first ordered by the time of the first packet sent in each transfer and then the data is divided in 1-hour time intervals. From the original 104 features, only the 100 numeric features are included in the analysis. Clustering is applied on all the numeric normalized features in 96, 1-hour time windows to generate binary labels for all the traffic flows in these time windows. The percent of transfers in cluster 1 and each time window's average throughput are calculated.

TABLE I
Cluster Characterization

Cluster	Flows	Top Features		
		Feature	Avg.	Std. Dev.
1	297,891	s_mss_max	1446.02	1104.28
		s_cwin_max	4265.63	11654.51
		s_pkts_dup	0	0
		s_syn_cnt	1	0
		s_rst_cnt	0.0049	0.07
2	51,272	s_mss_max	10812.01	15891.24
		s_cwin_max	51272.01	158739.51
		s_pkts_dup	1.97	2.44
		s_syn_cnt	2.97	2.44
		s_rst_cnt	0.3947	0.4888

When comparing the percentage of flows in cluster 1 with the average throughput for each 1-hour time window, we see that the two time series overlap and therefore are highly correlated. Figure 3 shows this overlap for data that spans over 4 days. Out of around forty low throughput points only two were not correctly identify by the cluster percentage.

To select the most prominent five features from each subset, a procedure based on forest ensemble was used. The prominent features were identified using the highest variance in the first primary component of the forest ensemble result. Table I reports the main characteristics of the extracted clusters and their top-5 characterizing features. We immediately notice a cluster with approximately 85% of the flows (cluster 1 has 297,891 flows) that is significantly larger than the other cluster. Cluster 1 represents standard or ”normal” flows which are characterized by lower average values of four out of five best features with only values of 0 for the s_pkts_dup feature.

Quantitative measurements of the correlation are presented in Table II. The RMSE between the throughput and the percentage of TCP anomalies is 0.0979 and the KT test is 0.1294, with a p-val of 3.5381 for data node 5. These measurements show how strong the correlation between these two time series is. Results for three other data nodes presented in Table II show similar results in terms of correlation.

TABLE II
Model Evaluation

Node	Number of Flows	RMSE	KS
5	363,487	0.097901	0.129482
6	506,511	0.164298	0.382051
7	480,738	0.099288	0.203846
8	451,811	0.115041	0.119230

Fig. 4 and Fig. 5 report the cumulative distributions (CDF) of average throughput and maximum segment size (s_mss_max) for each of the two clusters. Figure 4 shows that cluster 2 is characterized by low performance in terms of throughput, and it probably represents flows with possible performance issues. In figure 5, cluster 1 shows a completely different distribution of the s_mss_max values. The flows in cluster 1, are characterized by generally low values of

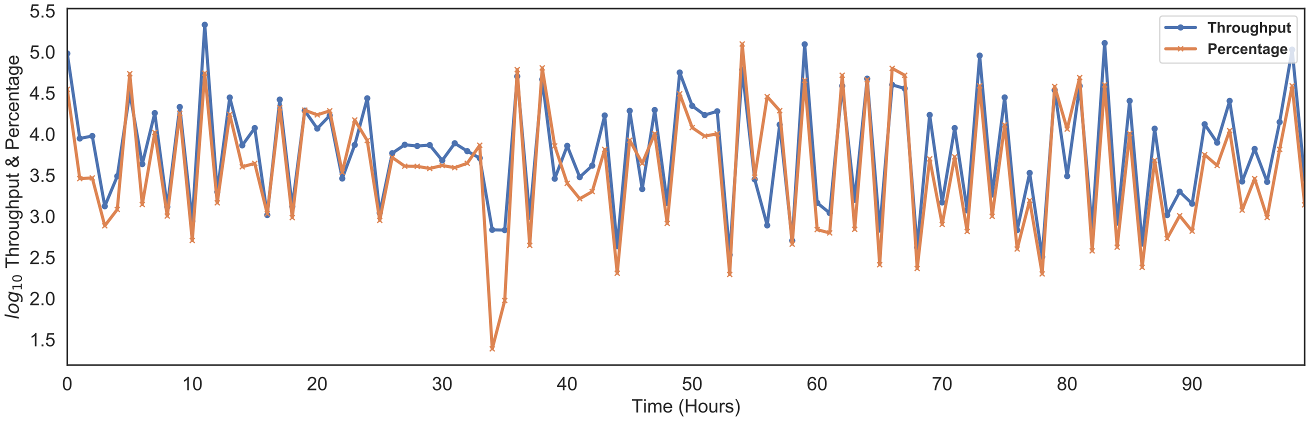

Fig. 3. Throughput and percentage of normal flows for 100, 1-hour time windows.

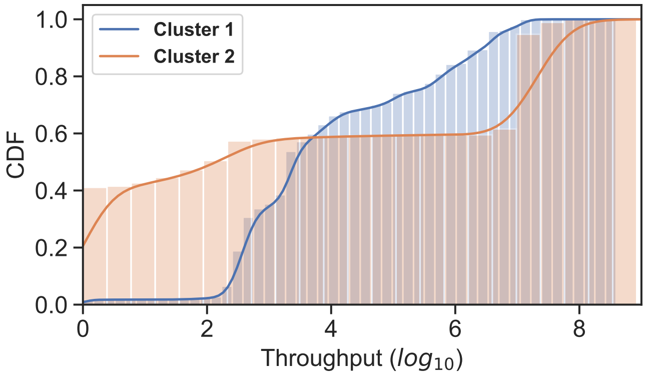

Fig. 4. Throughput cumulative distribution function plot for the two clusters

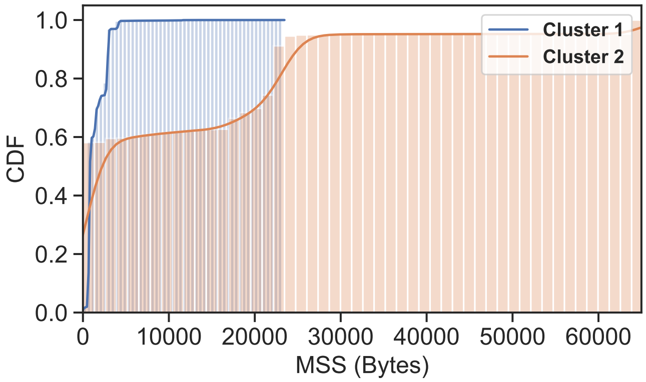

Fig. 5. MSS cumulative distribution function plot for the two clusters

s_mss_max. On the contrary, the flows of cluster 2 are associated with a large range of values for the same feature.

IV. Conclusion

Reliable network transfers are essential for successful operations at large scientific facilities where petabytes of large files need to be transferred daily. To identify possible problems and low throughput a method using clustering combined with classification algorithms to analyze `Tstat` logs has been proposed. This prototype uses online learning to handle streaming data. Therefore, the classification model only needs to be updated and not rebuilt from the ground up. This new method to detect network data transfer performance behavior can accurately and consistently cluster normal and anomalous network transfers and detect abnormally low throughput.

In future we plan to apply the same approach to divide the data into more clusters to identify not only anomalies, but specific problems such as packet duplication, retransmissions and synthetic reordering.

Acknowledgment

This work was supported by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Visiting Faculty Program (VFP) and by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

References

[1] Z. Chen, J. Wen, and Y. Geng, “Predicting future traffic using hidden markov models,” in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. IEEE, 2016, pp. 1–6.

[2] Z. Liu, M. Veeraraghavan, J. Zhou, J. Hick, and Y.-T. Li, “On causes of gridftp transfer throughput variance,” in *Proceedings of the Third International Workshop on Network-Aware Data Management*. ACM, 2013, p. 5.

[3] J. Kim, A. Sim, B. Tierney, S. Suh, and I. Kim, “Multivariate network traffic analysis using clustered patterns,” *Computing*, pp. 1–23, 2018.

[4] D. Apiletti, E. Baralis, T. Cerquitelli, P. Garza, D. Giordano, M. Mellia, and L. Venturini, “Selina: a self-learning insightful network analyzer,” *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 696–710, 2016.

[5] L. V. D. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[6] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen *et al.*, “Mllib: Machine learning in apache spark,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1235–1241, 2016.