Comparing Unsupervised Learning Approaches to Detect Network Intrusion using NetFlow Data

Julina Zhang, Kerry Jones, Tianye Song, Hyojung Kang, and Donald E. Brown
University of Virginia, jyz3uh, kmj2ge, ts7fx, hkang, deb@virginia.edu

Abstract - **Networks are vulnerable to costly attacks. Thus, the ability to detect these intrusions early on and minimize their impact is imperative to the financial security and reputation of an institution. There are two mainstream systems of intrusion detection (IDS), signature-based and anomaly-based IDS. Signature-based IDS identify intrusions by referencing a database of known identity, or signature, for each of the previous intrusion events. Anomaly-based IDS attempt to identify intrusions by referencing a baseline or learned patterns of normal behavior. Under this approach, deviations from the baseline are considered intrusions. We assume this type of behavior is rare and distinguishable from normal activity. Our research investigates unsupervised techniques for anomaly-based network intrusion detection. For this research, we use real-time traffic data from University of Virginia network. We evaluate the performance between Local Outlier Factor (LOF) and Isolation Forest (iForest) by probing the similarities and differences between the result of each approach. Distribution plots show there is greater variation of attributes in anomalies identified by iForest than those anomalies identified by LOF. Furthermore, iForest results are more distinctive from all data than the LOF results. With the assumptions that anomalies are points that are rare and distinctive, we find that iForest performs well in identifying anomalies compared to LOF.**

Index Terms - Anomaly Detection, Machine Learning, Network Security, Unsupervised Learning

INTRODUCTION

Many institutions depend on networks to facilitate the exchange of private and public information. The wide range of users (e.g., students, professors, and IT professionals) and user activity (e.g., browsing web pages, Facebook messaging, and video streaming) make investigating network traffic nearly insurmountable. As a result, sensitive and valuable information is left vulnerable to hackers. In February 2016, a group of hackers penetrated into, at the time, was considered the most secure financial messaging system in the world, the SWIFT Bank network. Exploiting loopholes in bank security system, the actors acquired login credentials, enabling their ability to initiate fraudulent money transfers. The heist resulted in a financial loss of $81 million dollars from the central bank of Bangladesh to Philippines accounts [1]. In November of the same year, a data breach incident at the Michigan State University led to a financial loss of two million dollars [2]. These two incidents demonstrate how critical this area of research is. Failing to detect cyber-criminal activities in a network highly damages the reputation of an institution, and the consequences are costly. Thus, it is imperative for institutions to detect intrusions within a network to minimize the impact.

Anomaly detection methods aim to identify events or observations that do not conform to patterns in a given dataset, often referred to as outliers. In a statistical sense, outliers refer to rare events or observations that typically distort patterns in data. These patterns are typically sensitive to ouliers so the traditional soluton is to simply remove these observations from a dataset [3]. In the context of network intrusion, we are interested in what anomalies are, not neccesarily looking for odd events but more change in activity that deviates from the norm. Such behavior can be indicative of suscpicious or malicious behavior thus common statistical approaches to outlier detection are ill-advised.

There are two mainstream intrusion detection systems (IDS), signature-based and anomaly-based IDS. Signature-based IDS identifies intrusion attempts by referencing a database of known identity, or signature, for each of the previous intrusion events. It uses predefined patterns considered malicious to search for attacks. A major limitation of this approach is it fails to detect novel attacks that are unique to the ones stored in the database. Attackers can devise new attack vectors or disguise the attack to evade the IDS[4]. Anomaly-based IDS attempt to identify intrusion by referencing a baseline or learned patterns of normal behavior. Under this approach, deviations from the pattern can be considered intrusions. A disadvantage of anomaly-based IDS is that it has higher false alarm rate than signature-based IDS [5].

Lack of publicly available data sources is a major challenge in network intrusion anomaly detection. Two of the most widely-used datasets, KDDCUP ’99 and DARPA are both nearly 20 years old. The KDDCUP ’99 dataset, characterized by flow level attacks, is flawed by redundancy. Approximately 75% records in the training and testing set are duplicated each set, respectively. This creates biased results leading to high-classification rates that are not representative of real-life network intrusions [6]. Additionally, the DARPA dataset was found to underestimate the number of novel and unusual events, leading to false alarms [7].

In this paper, we evaluate how well, using unsupervised anomaly detection approaches, our ability is in detecting observations significantly different and rare within the University of Virginia's network. We do not have access to ground truth to which the performance of these detection models can be assessed. To address this challenge, we use statistical visualizations to provide evidence of how anomalous our outliers are relative to normal behavior in netFlow traffic.

LITERATURE REVIEW

At enterprise level, there are several available intrusion detection systems. Snort, a widely used lightweight network IDS, detects intrusions by matching the observed packets to its extensive set of network intrusion detection rules [8]. Suricata, another signature-based IDS, performs network traffic inspection analysis in a similar fashion [9]. Bro, a powerful network security monitoring system, on the other hand, can perform both signature-based and anomaly-based intrusion detection. It has the capability to read Snort signatures and perform search for activity that resembles an intrusion [8]. Some of the well-developed commercial network IDSs, such as Lancope's StealthWatch, use anomaly-based approach to detect intrusions. StealthWatch is a netFlow-based network security monitoring system that employs sophisticated behavioral analytics and heuristics to analyze 90+ attributes of network flow data to establish baseline normal behavior of a host and alerts the security administrators if the security concern index reaches over a predefined threshold [10].

Intrusion detection approaches can be classified as either supervised, semi-supervised, or unsupervised. Unsupervised is the most challenging of the three – without having labeled information it is extremely difficult to validate the performance of the model [3]. Given that our approach is unsupervised, we must assume that the majority of instances in our dataset are normal, whereas anomalies are the minority and distinguishable [3].

Density-based approaches are one of several unsupervised approaches that identify anomalies without target values. Density based approaches rely on the assumption that outliers are found in sparse while normal instances are found in highly-dense regions [11]. Local Outlier Factor (LOF) is one density based technique used to emphasize locality. In this approach, the local density of an object is compared to that of its neighbors. Thus, outliers are considered areas that have lower density relative to their neighbors. Although there have been several extensions of these methods, LOF is still considered state of the art [12].

Another approach called Isolation Forest (iForest) uses a data-induced random binary tree structure, to effectively isolate anomalies as they tend to fall closer to the root of the tree since fewer conditions are required to differentiate them from the other instances. Normal instances will be found at the deeper end of the tree. An anomaly score is generated based on average path length from the root node [13].

This study aims to detect anomalies in unique IP and port number pair through the combination use of iForest and LOF.

METHOD

In this section, we discuss the procedures of our approach. Section I describes the data we use for the study. We then discuss data preprocessing in section II, and finally go into the details of our approach in section III.

I. Description of the Data

For this study, UVa's Information Security, Policy, Records Office (ISPRO) provides unidirectional netFlow flow records in version nine format. This data is collected everyday in real-time by a Gigamon server in which mirrors incoming and outgoing network traffic.

NetFlow provides an aggregated summary of the unidirectional traffic flows between two IP addresses. Table I shows features and their description used in this study. NetFlow is similar to the concept of a phone bill. The bill does not reveal details of each conversation but rather the meta information associated with it(e.g. phone numbers, the duration of the conversation, and, whether it is incoming or outgoing). Thus, an IP address is analogous to a phone number. It identifies the parties involved in the conversation. A specific port number identifies number the specific sub-branch the conversation should belongs to. The field Packets, records how many individual packets are transmitted in a specific netFlow. The relationship between individual packets and the entire netFlow is similar to that of an Amazon.com order. Within this order, there are a set of packages. In each package there are a set of components or items. The whole order is essentially an aggregation of these components. Similarly, a packet carries a portion of the data transmitted. The aggregation of those packets is equivalent to the amount of data transferred from the sender to the receiver.

TABLE I
NETFLOW FEATURES USED IN APPROACH

Net Flow Features	Description
Source Address	IP address sending packets
Destination Address	IP address receiving packets
Source Port	Port used associated with Source Address
Destination Port	Port used associated with Destination Address
Bytes	Total amount of data sent
Packets	Total count of packets
Duration	Total time of the connection
Rate	Speed at which the data was transmitted

II. Data Preprocessing

By mirroring data using a Gigamon server mounted to the outgoing network router of UVa, we obtain data of the entire University. NetfFlow files are generated every five minutes in every hour of every day. In total, there are 288 files produced daily. Each file is concatenated so we maintain connections longer than five minutes.

Since we want to observe the continuous behavior of a specific IP address during specific time, instead of looking at every IP address in the network, we filter by UVA's static IP addresses and a specific time range. We perform this step because dynamic IP are not indicative of the same user or user behavior. By using static IP, we can make assumptions about specific user types and behavior. For example, students submit assignments in UVa's learning collaboration environment, Collab. Due to nature of assignment deadlines, we know there are more assignment submissions during daytime relative to nighttime.

For this study, we used samples of netFlow data for five days between February 13th and 17th of 2017, between 9:00 and 10:00 am. The dataset is approximately five gigabytes. Additionally, since we could not investigate every static IP address, we chose to look specifically at the following: Electrical and Computing Engineering Department (ECE) and SHANTI Pages, a WordPress environment for the UVA community. We choose these webpages because they are associated with different types of user activity. Furthermore, we choose to investigate netFlow traffic through port number 80 for both webpages. Port 80 is commonly used to handle HTTP requests. We want our data to include as broad a range of user activities as possible. Under port 80, both UVa and non-UVa affiliated users can visit the pages freely.

After filtering our data, we are left with four attributes for two datasets per each of the five days. For example, for Monday, we have two datasets for ECE and SHANTI respectively. There are four attributes for each of the two datasets: bytes, packets, duration and rate. As the last step of our preparation before applying unsupervised learning approaches, we normalize all attributes for each of the ten resulting data sets so the attributes' values are the same scale and are weighted equally.

III. Approach

In the context of anomaly detection, there are two main assumptions: anomalies are rare, and they are distinctive from normal instances [3,13]. Through iForest and LOF, we aim to identify anomalies that are rare and distinctive from the majority, assuming the majority consists of normal activities. First, we apply iForest and LOF on each day's data to extract sets of outliers for each day, respectively. This results in a total of ten sets of identified anomalies: five sets identified by iForest and five sets identified by LOF across five days. Next, we extract netFlow flow records that are identified as anomalies by both iForest and LOF. Therefore, we have three sets of anomalies for comparison. We then generate frequency distribution plots for each set for comparison.

Results

This section provides visualizations and statistics on iForest and LOF for five days of NetFlow data at UVA. The statistics shown in Tables II – VI, show the statistics for the anomalies detected by each of the methods. The tables also show the statistics for the NetFlow on each of the days, and the characteristics of the flows that were in the intersection between the two methods.

Similarly, the graphical displays in Figures I- IV show the frequency distributions for each of the methods. Again, these figures show frequencies for each of the days and the frequencies for the intersections.

Figure I displays distribution plots for each attribute. In each of the four distribution plots, each day of the week is represented by a distinctive color. Figure II and III each illustrates the distribution of anomalous instances identified by iForest and LOF, respectively. Figure IV shows the intersection of the anomalous instances displayed in Figure II and III. In other words, all attributes of the intersection between the set of anomalies identified by iForest and LOF are plotted here.

At a first glance, Figure I and III, and II and IV appear to be similar, respectively. However, Figure I and II, as well as Figure I and IV show drastically different distributions. Looking more closely at these figures, we can see the distribution plots for duration for all four figures are similar. This tells us that the anomalous instances have similar distributions in duration attributes. Thus, we decide to remove duration from our investigation for anomalies. Next we investigate the attribute Bytes. The distribution plots for Bytes for Figure I and III are similar. This shows the distribution for Bytes in anomalies detected by LOF are similar to the entire dataset. In contrast, the distribution plots for Bytes in Figure I and II are drastically different. This suggest iForest performs better in capturing anomalous instances that are distinctive in their distribution, compared with the distribution of all other instances.

The next attribute we probe is Packets (Pkts). Comparing the distribution plots for Figure I and III, with that for Figure I and II, we find that again, iForest results display a distribution plot that is distinctive from that of Figure I and III. We suggest that this finding reinforce our assertion earlier, that iForest performs better than LOF in capturing anomalous instances that are distinctive.

The last attribute to consider is Rate. Since rate is simply Bytes divided by duration (Dur), we expect the situation for its distribution plots to be similar to that of Bytes. In accordance with our expectations, Figure I and III show similar distribution for rate, whereas Figure I and II show difference in the distribution plots for rate.

TABLE II
AVERAGE BYTES
(STANDARD DEVIATION IN PARENTHESIS)

Days	All	IForest	LOF	IForest ∩ LOF
Monday	0.12 (.22)	0.80 (.17)	0.30 (.38)	0.83 (.17)
Tuesday	0.098 (.19)	0.79 (.20)	0.19 (.33)	0.78 (.17)
Wednesday	0.12 (.19)	0.81 (.19)	0.26 (.31)	0.79 (.18)
Thursday	0.16 (.21)	0.84 (.17)	0.18 (.28)	0.85 (.23)
Friday	0.13 (.23)	0.86 (.15)	0.25 (.40)	0.96 (.071)

TABLE III
AVERAGE PACKETS
(STANDARD DEVIATION IN PARENTHESIS)

Days	All	IForest	LOF	IForest ∩ LOF
Monday	1.5e-3 (4.1e-3)	0.014 (7.0e-3)	4.0e-3 (6.5e-3)	0.012 (6.6e-3)
Tuesday	1.2e-3 (3.7e-3)	0.015 (7.0e-3)	2.6e-3 (5.4e-3)	0.011 (6.5e-3)
Wednesday	1.5e-3 (3.9e-3)	0.016 (7.0e-3)	3.9e-3 (5.4e-3)	0.012 (6.7e-3)
Thursday	1.6e-3 (4.0e-3)	0.015 (8.0e-3)	2.1e-3 (4.8e-3)	0.012 (8.6e-3)
Friday	1.7e-3 (4.1e-3)	0.017 (7.0e-3)	3.9e-3 (7.5e-3)	0.016 (7.9e-3)

TABLE IV
AVERAGE DURATION
(STANDARD DEVIATION IN PARENTHESIS)

Days	All	IForest	LOF	IForest ∩ LOF
Monday	3.5e-4 (2.3e-3)	6.6e-3 (8.2e-3)	1.4e-3 (4.9e-3)	4.7e-3 (8.1e-3)
Tuesday	2.3e-4 (1.4e-3)	3.7e-3 (5.0e-3)	5.1e-4 (1.9e-3)	2.2e-3 (3.6e-3)
Wednesday	1.8e-4 (1.2e-3)	2.9e-3 (4.3e-3)	5.4e-4 (1.9e-3)	2.4e-3 (3.5e-3)
Thursday	9.3e-5 (8.2e-4)	1.7e-3 (3.3e-3)	6.0e-5 (3.1e-4)	4.4e-4 (8.4e-4)
Friday	1.8e-4 (1.2e-3)	3.5e-3 (4.4e-3)	5.7e-4 (2.5e-3)	2.4e-3 (4.8e-3)

TABLE V
AVERAGE RATE
(STANDARD DEVIATION IN PARENTHESIS)

Days	All	IForest	LOF	IForest ∩ LOF
Monday	0.95 (.17)	0.50 (.29)	0.81 (.32)	0.42 (.33)
Tuesday	0.97 (.14)	0.48 (.33)	0.89 (.25)	0.51 (.31)
Wednesday	0.96 (.14)	0.44 (.34)	0.88 (.24)	0.50 (.30)
Thursday	0.95 (.15)	0.39 (.34)	0.91 (.25)	0.29 (.38)
Friday	0.95 (.18)	0.39 (.28)	0.81 (.35)	0.19 (.18)

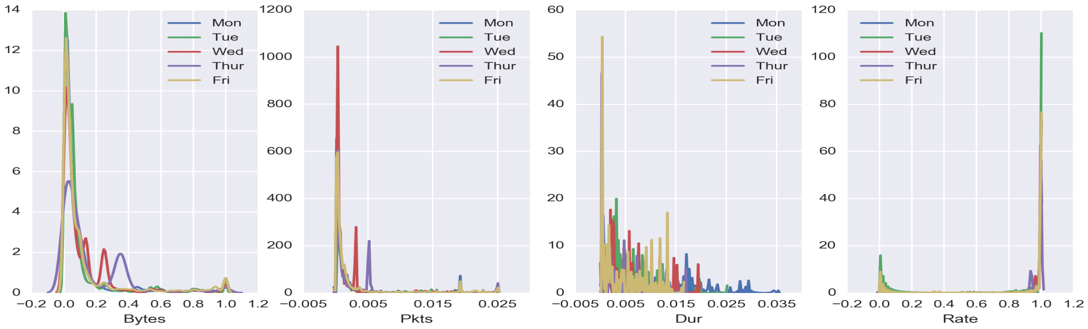

FIGURE I
DISTRIBUTION OF ATTRIBUTES ACROSS DAYS FOR ALL DATA

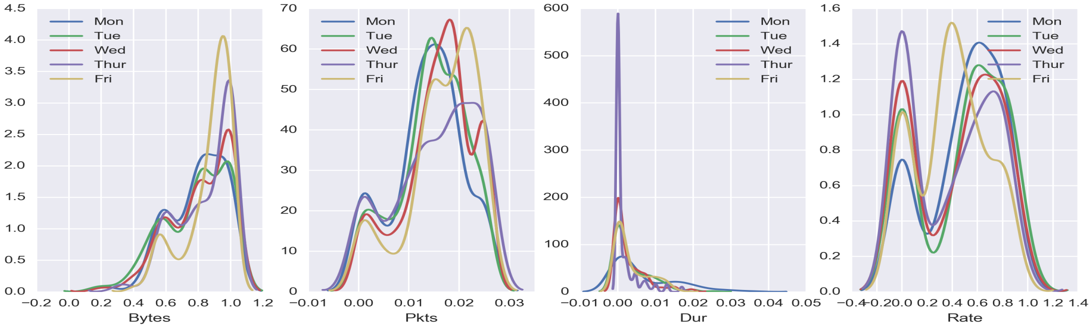

FIGURE II

DISTRIBUTION OF ATTRIBUTES ACROSS DAYS FOR ANOMALIES IDENTIFIED BY IFOREST

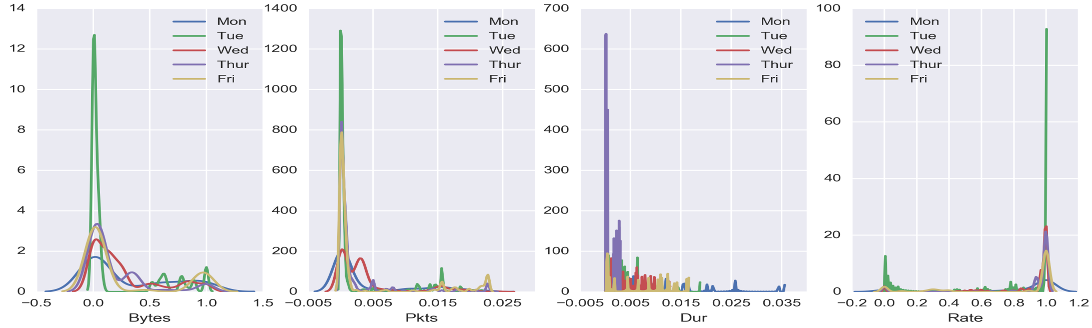

FIGURE III

DISTRIBUTION OF ATTRIBUTES ACROSS DAYS FOR ANOMALIES IDENTIFIED BY LOF

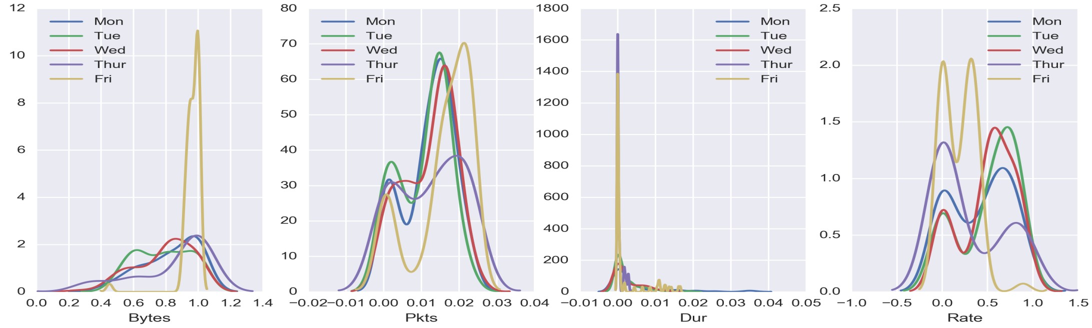

FIGURE IV

DISTRIBUTION OF ATTRIBUTES ACROSS DAYS FOR ANOMALIES IDENTIFIED BY IFOREST AND LOF

When comparing Figure IV with the rest of the figures, we find that it displays similar distribution plots with Figure II. Thus we deduce the intersection of the anomalous data captured by iForest and LOF agrees more with the results yielded by iForest.

CONCLUSION AND FUTURE WORK

With the above analysis, we conclude that comparing with LOF, iForest has better performance in capturing anomalous instances that are distinctive. However, there are limitations in our investigation. First, our model only looked at four numeric attributes. Due to this, iForest classifies the most extreme values as outliers. In other words, it would classify flows that have relatively higher byte, rate, packet, or duration as outliers, given that those flows do not occur as often. Our study only investigates http (port 80) traffic for two specific static IP addresses. For future work, the same approach can be tested on different traffic type such as File Transfer Protocol (FTP) or Simple Mail Transfer Protocol (SMTP) and/or on multiple IP addresses to further validate the implications from the results. Due to the lack of labelled data, we cannot evaluate our approach using traditional methods such as drawing and comparing ROC curves or computing AUC scores, i.e. we cannot compute the percentage our approach classified correctly and the percentage our approach classified incorrectly.

Instead, it would require a domain expert with years of experience with the network to determine whether the identified anomalies do in fact seem suspicious. In addition, we would need a domain expert's insight on choosing what attributes to include in the model and the best parameter values for our model. For example, a domain expert would be able to tell us what is the average probability of finding an intrusion in a given set of data, and we would use this value as the value for the amount of contamination parameter in the iForest algorithm to further improve our ability to detect anomalies.

ACKNOWLEDGMENT

We would like to thank our advisor Dr. Malathi Veeraraghavan from UVa Electrical and Computer Engineering Department for her constructive comments and guidance and our sponsor Leidos. We would also like to thank Sourav Maji from UVa Computer Engineering Department for providing valuable assistance. We would like to thank Jason Belford and Jeff Collyer from UVa ISPRO for providing data and insights for the project. We would also like to thank Ronald Hutchins from UVa Information Technology Services for his support.

REFERENCES

[1] M. Corkery, "Once Again, Thieves Enter Swift Financial Network and Steal," *The New York Times,* 12 Mar 2016. [Online]. Available: www.nytimes.com/2016/05/13/business/dealbook/swift-global-bank-network-attack.html

[2] R. Wolcott, "Data breach could cost $3 million, Michigan State says," Lansing State Journal, 30 November 2016. [Online]. Available: www.lansingstatejournal.com/story/news/local/2016/11/30/msu-estimates-spending-3-million-responding-data-breach/94541962/

[3] M. Goldstein and S. Uchida, "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data," PLOS one, pp. 1 - 31, 19 April 2016.

[4] Jyothsna, V. V. Rama Prasad and K. Munivara Prasad, "A Review of Anomaly based Intrusion Detection System," International Journal of Computer Applications , vol. 28, no. 7, pp. 26-35, August 2011.

[5] P. Garcia-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, et al., "Anomaly-based network intrusion detection: Techniques, systems and challenges," Computers & Security, vol. 28, no. 1-2, pp. 18-28, Feburary 2009.

[6] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A Detailed Analysis of the KDD Cup 99 Data Set," in IEE Symposium on Computational Intelligence in Security and Defense Applications, pp. 53-58, Ottawa, 2009.

[7] M. V. Mahoney and P. K. Chan, "An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection," in RAID: International Workshop on Recent Advances in Intrusion Detection, Berlin.[Online]. Available: cs.fit.edu/~pkc/papers/raid03.pdf

[8] Sans Insitute, "Web Application Attack Analysis using BRO IDS," [Online]. Available: https://www.sans.org/reading-room/whitepapers/detection/web-application-attack-analysis-bro-ids-34042. [Accessed 27 March 2017]

[9] "Suricata," [Online]. Available: suricata-ids.org.

[10] R. A. Grimes, "Detect network anomalies with StealthWatch," InfoWorld, 18 November 2014. [Online]. Available: www.infoworld.com/article/2848768/security/detect-network-anomalies-with-stealthwatch.html

[11] T. Jirachan and K. Piromsopa, "Applying KSE-test and K-means clustering towards Scalable Unsupervised Intrusion Detection," in *International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 82-87, Songkhla, 2015.

[12] C. O. Guilherme, J. Sander, A. Zimek, et al., "On the Evaluation of Unsupervised Outlier Detection: Measures, Datasets, and an Empirical Study," *Data Mining and Knowledge Discovery,* vol. 30, no. 4, pp. 891-927, 2016.

[13] F. T. Liu, K. M. Ting and Z.-H. Zhou, "Isolation Forest," in 2008 Eighth IEEE International Conference on Data Mining, Pisa, 2008.

AUTHOR INFORMATION

Julina Zhang, Kerry Jones, Tianye Song, Master Candidates, Data Science Institute, University of Virginia.