A Comparison of Unsupervised Anomaly Detection Methods on a Unified Host and Network Dataset

Abstract— **Cybersecurity is a constant and growing threat, and systems administrators need to efficiently detect and respond to security breaches to protect their network. By effectively detecting anomalous activity in NetFlow data, systems administrators can limit the amount of packet capture they analyze to quickly evaluate and respond to threats. Machine learning methods are a valuable way to do so. Supervised machine learning methods have been shown to effectively detect malicious activity, but they fall short in detecting previously unseen types of malicious activity and they require labeled datasets. Conversely, unsupervised machine learning methods can detect previously unseen types of malicious activity and do not require labeled datasets. In this paper, we compare unsupervised anomaly detection methods to detect potentially malicious connections in NetFlow data using the NetFlow dataset in "The Unified Host and Network Dataset" from the Los Alamos National Laboratory (LANL). We evaluate a classical Autoencoder, Isolation Forest, Elliptic Envelope, Local Outlier Factor, and One Class Support Vector Machine. We show that all five methods effectively separate the data into connections with different network activity, but Isolation Forest, the classical Autoencoder, and Elliptic Envelope detect anomalies that are the most different from the non-anomalies. Our findings are useful for reducing the amount of packet capture systems administrators need to evaluate to respond to threats and for furthering unsupervised anomaly detection in cybersecurity.**

Keywords—unsupervised learning, anomaly detection, cybersecurity, NetFlow, packet capture

I. Introduction

Cybercrime poses as a constant and ever changing-threat to countries, institutions, and organizations alike [1]. As the threat of cybercrime grows and evolves, many cybersecurity researchers are analyzing network traffic and using machine learning methods to detect malicious activity within it [2]. In traditional cybersecurity, many researchers analyze two main data sources: NetFlow and packet capture. Packet capture describes granular information about the connection including packets and payload. It is used to evaluate data describing a security incident to ultimately detect and remove the security threat. NetFlow describes flow-level information like time, duration, devices, protocols, ports, packets, and bytes. NetFlow data is useful to analyze because it consists of sufficient information to characterize normal network traffic behavior [3]. Advanced statistical analysis of NetFlow data can find policy violations, botnets, or malware by finding flows that deviate from normal traffic behavior, often in terms of the number of packets and bytes sent [4].

Traditionally, systems administrators analyze packet capture, evaluating packets and their payloads to find the data describing the security incident. However, using NetFlow data, systems administrators can find suspicious flows first, then use the suspicious flows to filter what packets and payloads to evaluate in packet capture. Without filtering packet capture with NetFlow, the systems administrator would need to evaluate all packet capture. By analyzing NetFlow data first, systems administrators can limit the number of packets and payloads to analyze and respond to threats quicker [5]. As a result, it is important to successfully find anomalies in the NetFlow data to respond to cybersecurity threats more efficiently. In this paper, we use the "Unified Host and Network Dataset" from the Los Alamos National Laboratory (LANL) to detect anomalous traces in NetFlow data.

II. Literature Review

A number of researchers have analyzed unsupervised anomaly detection methods and have developed intrusion detection systems for enterprise networks. Zhang, Jones, Song, Kang, and Brown compare Local Outlier Factor and Isolation Forest for detecting anomalies on unidirectional NetFlow data for two IP addresses operating on Port 80. They find that Isolation Forest outperforms Local Outlier Factor, as Isolation Forest classifies flows such that the difference between anomalous and nonanomalous packet and byte distributions is greater [6].

Fernandes, Rodrigues, and Proença propose an unsupervised anomaly detection method using principal component analysis (PCA) and historical network data analysis to create a profile of normal network activity. They classify activity that deviates from this profile as anomalous [7]. Cao, Nicolau, and McDermott propose classical autoencoders and variational autoencoders for network anomaly detection. They also discuss the performance of principal component analysis, one class support vector machines, and local outlier factor. They find that their approaches to classical autoencoders and variational autoencoders perform well in conjunction with these well-known anomaly detection methods [8].

III. Data

In this paper, we evaluate unsupervised anomaly detection methods on the NetFlow dataset in "The Unified Host and Network Dataset" from the Los Alamos National Laboratory (LANL). The NetFlow dataset consists of netflow records, which are records of network activity between a client and server. LANL's NetFlow dataset is formatted differently from many NetFlow datasets, as it is formatted with records of

biflows, bidirectional connections with features separated by source and destination activity, and it is aggregated by five tuple, which is the set of source device, source port, destination device, destination port, and protocol. LANL formatted the data in this way using a process called network stitching that matches unidirectional flows into biflows. In cases when stitching was unsuccessful, the number of source or destination packets and bytes is 0. We removed those connections from the dataset and we assume that all the remaining connections with nonzero packets and bytes were collected and stitched correctly. By formatting the data in this manner, LANL's NetFlow dataset consists of traces, which are a higher level representation of the network connection. The features the data are the following:

- Time – The start time of the event in epoch time format.
- Duration – Duration of the event in seconds.
- SrcDevice – Anonymized ID of the device that likely initiated the event.
- DstDevice - Anonymized ID of the receiving device.
- Protocol – The protocol number.
- SrcPort – The port used by SrcDevice.
- DstPort – The port used by DstDevice.
- SrcPackets – The number of packets sent by SrcDevice during the event.
- DstPackets – The number of packets sent by DstDevice during the event.
- SrcBytes – The number of bytes sent by SrcDevice during the event.
- DstBytes – The number of bytes sent by DstDevice during the event.

In this paper we use a representative random sample of 1 million observations from one day of data in the NetFlow dataset. The representative random sample was taken such that each its summary statistics are within one standard deviation of those describing the complete day of data. As unsupervised anomaly detection methods find relative anomalies, since anomalies are observations that differ from the majority of observations in the dataset, the amount of anomalies and the difference between anomalies and non-anomalies may differ for different samples. However, given our sample is large and representative, we achieve consistent results with other samples and we assume our results extend to even larger samples of LANL's NetFlow data and the complete dataset.

IV. Methodology

In this paper, we build five unsupervised anomaly detection models on the 1 million point sample. We manually tune hyperparameters for each algorithm on multiple samples to ensure correctness and consistency. The five algorithms are Isolation Forest, One Class Support Vector Machine, Elliptic Envelope, Local Outlier Factor, and a classical Autoencoder. We detail each of them below.

A. Isolation Forest

Isolation Forest classifies anomalies as points that are remote from the majority of the data. It uses decision trees with random splits to partition the data. As outliers are infrequent, different from the majority of the data, and remote in feature space, random partitioning should cause the outliers to lie close to the root of the tree, meaning outliers need fewer random splits of the tree to be separated from the majority of the data. Isolation Forest calculates an anomaly score for each of the points using aspects of the tree and classifies points according the score, with scores close to 1 indicating anomalies and scores less than 0.5 indicating non-anomalies. The points with the greatest anomaly scores are ultimately classified as the anomalies.

B. One Class Support Vector Machine

In this paper, we use the Scholkopf approach for a One Class Support Vector Machine (One Class SVM). This approach maps all the points to a different feature space and places a hyperplane in the feature space such that it separates the majority of the points from the origin and lies maximally far from both the region surrounding the majority of the points and the origin. The hyperplane creates a binary function to classify the data points such that all points in the smaller region defined by a probability density function created by the hyperplane are classified as non-anomalies, and all points outside of that are classified as anomalies.

C. Elliptic Envelope

Elliptic Envelope assumes the data are Gaussian and fits an ellipse to surround most of the data. It classifies observations as non-anomalies when they are within the ellipse and as anomalies when they are outside of the ellipse. The model fits an ellipse to the data such that the percent of observations outside the ellipse is equal to the contamination, the specified percentage of points expected to be outliers.

D. Local Outlier Factor

Local Outlier Factor classifies points with low local density as anomalous. Points with low local density are those that are remote in feature space. The algorithm calculates each point's local density and compares the point's local density to the average of its neighbors' local densities. If the point is less dense that its neighbors are, then it is more remote than its neighbors. It classifies the most remote points as anomalies.

E. Autoencoder

Autoencoders learn a compressed representation of the inputted data and use it to recreate the original, inputted data as output. The more anomalous the input is, the more difficult it is for the autoencoder to recreate it, and thus the more different the output is from the input. In this paper, we use the Euclidean distance between the input vectors and the output vectors to calculate the difference between the input and the output. The observations with the greatest difference between their input and output are classified as anomalies

F. Evaluation

To evaluate our models, we compare ratios of the intracluster similarity of all the connections to the intracluster similarity of the nonanomalous connections for all of the models. We measure intracluster similarity as the average Euclidean distance

to the cluster centroid. The intracluster similarity for non-anomalies should be smaller than the intracluster similarity for the entire dataset, as the set of nonanomalous points should be similar and thus be close to each other in the feature space. The ratio tells us how many times more similar the nonanomalous observations are than the entire dataset is. As a result, the larger the ratio, the greater the difference between nonanomalous data and anomalous data, and the better the anomaly detection model.

We also evaluate our anomaly detection methods with chi-squared tests to show whether the anomalous and nonanomalous traffic have statistically different port distributions and protocol distributions. A difference in port distribution indicates a difference in network activity for anomalous and nonanomalous traffic. For instance, normal traffic is often on HTTP and HTTPS, whereas malicious traffic often occurs on lesser used ports. If the anomalous traces have a statistically different port distribution than the nonanomalous traces, then the anomalous and nonanomalous traces must be different as they are involved in different network activity. Likewise, a difference in protocol distribution also shows a difference in network activity between the anomalous and nonanomalous traces. This chi-squared test on the independence of anomalous or nonanomalous classification and port distribution, and of anomalous or nonanomalous classification and protocol distribution confirms that the anomalous traces are in fact different from the nonanomalous traces.

V. Results

A. Port Analysis

Fig. 1 shows the port distribution for nonanomalous traces classified by all five models and for all points in the original dataset. As expected, nonanomalous observations most frequently occur on ports 443 and 80, which are used for HTTPS and HTTP, respectively. The nonanomalous port distributions for all five algorithms are nearly identical, sharing the most frequent ports and deviating from each other by at most one percent. As ports become less frequent, the models' port frequencies become equal. The nonanomalous traces occur most frequently on common ports, with few exceptions like 95765 and 36836. However, because of the class imbalance, with 1% anomalous and 99% nonanomalous, the nonanomalous port distributions also reflect the port distribution for the entire dataset. The agreement between the nonanomalous port distribution and the overall port distribution, and the agreement among the anomaly detection methods confirms that all methods label points such that the nonanomalous traces are representative of the entire dataset.

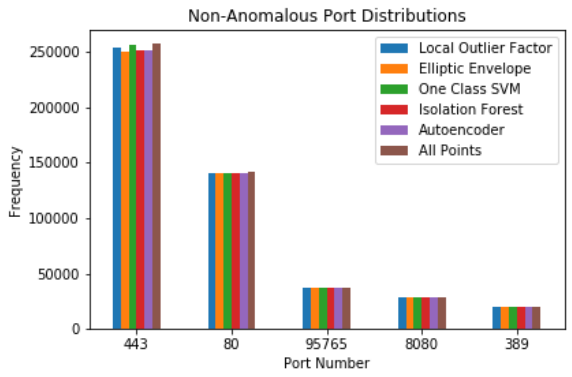

Fig. 1. Nonanomalous port distribution for all models and for the entire dataset.

In contrast, the port distributions for anomalous traces do not agree to the same extent. With the exception of One Class SVM, the anomalous traces labeled by all the algorithms occur on ports 443, 80, and 1433 within the top five most frequent, but differ in the rest of their distributions. Anomalous traces labeled by One Class SVM share no frequent ports with anomalous traces labeled by any of the other methods. While Isolation Forest, Elliptic Envelope, and the Autoencoder share many frequent ports, Isolation Forest and the Autoencoder share the most, having the same ten most frequent ports and similar frequencies for each. These three anomaly detection methods follow a similar port distribution, meaning their anomalous traces are involved in similar network activity. The agreement in port distribution among traces labeled by Isolation Forest and the Autoencoder mean that these algorithms label similar points as anomalies, while One Class SVM labels different types of traces as anomalies. Fig. 2 shows the port distributions for anomalies labeled by Isolation Forest and the Autoencoder.

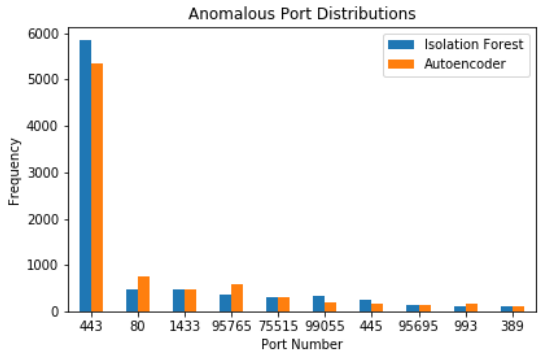

Fig. 2. Anomalous port distribution for Isolation Forest and the Autoencoder.

We divided the ports into three categories following IANA's port categorization, assigning ports as "System Ports" if they are in the range 0-1023, "User Ports" if they are in the range 1024-49151, and "Dynamic or Private Ports" if they are in the range 49152-65535 [9]. A chi squared test on the independence of anomalous label and port distribution under these categories shows that the anomalous and nonanomalous traces labeled by all four algorithms have statistically different port distributions. This result means that the anomalous and nonanomalous traces

occur on different ports and are thus involved in different network activity.

B. Protocol Analysis

Likewise, a difference in protocol distribution also shows a difference in network activity between anomalous and nonanomalous traces, as protocol also defines network activity type. All connections in this dataset are on Transfer Control Protocol (TCP) or User Datagram Protocol (UDP). TCP and UDP are both are transport protocols, meaning they establish a connection between the source and device to send packets to each other, but they differ in that TCP guarantees the delivery and correct order of packets and UDP does not. Without guaranteeing that packets are delivered correctly, UDP connections can be much faster, but because of its assurance of correct packet delivery, TCP is the most commonly used protocol.

Table 1 shows the percent of TCP connections for nonanomalous and anomalous traces. As expected, a vast majority of connections occur on TCP. Anomalous protocol distributions have many more connections on TCP than UDP, with over 99% TCP connections for anomalous traces classified by all five models except by Local Outlier Factor. Nonanomalous protocol distributions show roughly a 95%-5% split for connections on TCP and on UDP. A chi squared test on the independence of anomaly classification and protocol distribution showed that the protocol distributions for anomalous and nonanomalous traces are statistically different for all five algorithms, confirming that anomalous and nonanomalous traces are involved in different network activity.

TABLE I. PERCENT TCP TRACES

Model	Percent of TCP Traces	
	Nonanomalous	*Anomalous*
Isolation Forest	95.62%	99.32%
Autoencoder	95.62%	99.79%
Elliptoc Envelope	95.63%	98.06%
Local Outlier Factor	95.69%	92.07%
One Class SVM	95.61%	100.00%

C. Effect of Contamination

Cybersecurity experts estimate that at most 1% of enterprise network activity is malicious. Many unsupervised anomaly detection algorithms, like Isolation Forest, Local Outlier Factor, and Elliptic Envelope, require the contamination, the expected percent of anomalous observations, as one of the hyperparameters. We evaluated the models using 1% through 5% contamination to see the effect of contamination score on the difference between anomalous and nonanomalous traces. With higher contamination scores, the models achieved greater ratios of intracluster similarity of all traces to intracluster similarity of nonanomalous traces, meaning that as the number of traces classified as anomalous increases, the more similar the traces classified as nonanomalous become. This result makes sense, as when more observations that are different from the majority of observations are removed, the more similar the remaining, nonanomalous points should become.

Table 2 shows the intracluster similarity ratio for each model at each contamination. For Isolation Forest, the Autoencoder, and Elliptic Envelope, the ratios are high for 1% and increase through 5% contamination. One Class SVM and Local Outlier Factor do not find anomalies well for lower values of contamination, as their ratios of intracluster similarity for the nonanomalous data to the intracluster similarity of the entire dataset in less than one, meaning that these algorithms label observations as anomalies such that removing them from the dataset leaves a dataset less homogeneous than it was when it included the anomalous. Since Isolation Forest, the Autoencoder, and Elliptic Envelope are able to classify anomalies such that their intracluster similarity ratios are large means that it is possible to detect relatively different anomalies from the dataset, but One Class SVM and Local Outlier Factor are unable to do so. However, for higher contamination values, Local Outlier Factor is able to detect anomalies to improve the similarity of the nonanomalous subset, but not nearly as well as Isolation Forest, the Autoencoder, and Elliptic Envelope can.

TABLE II. RATIO OF INTRACLUSTER SIMILARITY AT EACH CONTAMINATION

Model	Contamination Percent				
	1%	*2%*	*3%*	*4%*	*5%*
Isolation Forest	2.527e6	1.106e7	2.506e7	4.706e7	6.798e7
Autoencoder	1.527e6	5.769e6	1.355e7	2.198e7	3.064e7
Elliptic Envelope	6.990e4	1.020e6	1.733e6	3.545e6	5.149e6
Local Outlier Factor	9.900e-1	9.792e-1	1.875	3.143	3.917
One Class SVM	9.901e-1	9.838e-1	9.839e-1	9.839e-1	9.839e-1

Evaluating the effects of different contamination values is important for evaluating our anomaly detection methods, as the purpose of anomaly detection on NetFlow data is to limit the number of observations to manually explore in packet capture to detect malicious activity. Though experts estimate at most 1% of network connections are malicious, labeling a higher percent of observations anomalous and therefore potentially malicious is still valuable as the cost of missing a potentially malicious actor on the network is greater than the cost of evaluating more connections in packet capture; filtering the NetFlow dataset to 5% of its original size is still a significant and worthwhile reduction in packet capture.

D. Labeling Agreement

We define agreement as the percent of observations classified the same, and we evaluate models' classification agreement at 1% contamination. Overall, all five models classify the traces the same 96.27% of the time, but this agreement is due entirely to agreement on nonanomalous traces, as all of the models have 0% complete agreement on any anomalous traces. This nonanomalous agreement is due in part to the dataset's class imbalance, with each model classifying 99% of observations as nonanomalous and 1% as anomalous. With such a class imbalance, it is much more likely for the algorithms to agree on nonanomalous observations since there is an overwhelming majority of nonanomalous points. Even by

random classification there would be high agreement. Conversely, the algorithms are less likely to agree on anomalous points as there are so few of them. As a result, we find that comparing percent agreement on anomalous observations is a better measure of labelling agreement, especially as we are predominantly interested in the anomalous observations.

Table 3 shows the pairwise percent agreement on anomalous traces. Isolation Forest and the Autoencoder have the largest agreement, with 68.11% of anomalous observations labeled the same. Elliptic Envelope and Isolation Forest have the second largest agreement, with 44.35% agreement on anomalous observations. Isolation Forest, the Autoencoder, and Elliptic Envelope individually have no agreement on any anomalous observations with either One Class SVM or Local Outlier Factor, but One Class SVM and Local Outlier Factor agree with each other on a few anomalies, achieving 0.4133% agreement on anomalous observations.

TABLE III. PERCENT AGREEMENT ON ANOMALOUS CLASSIFICATIONS

Model	**Model**				
	IF	***AE***	***EE***	***LOF***	***OCSVM***
IF	100%	-	-	-	-
AE	68.11%	100%	-	-	-
EE	44.35%	30.21%	100%	-	-
LOF	0%	0%	0%	100%	-
OCSVM	0%	0%	0%	0.4113%	100%

Interestingly, Isolation Forest and the Autoencoder performed best in terms of their intracluster similarity ratios and they agree the most. Elliptic Envelope follows Isolation Forest and the Autoencoder in terms of their intracluster similarity ratios, and its pairwise agreements with the two methods are the second and third greatest. Isolation Forest agrees more with Elliptic Envelope than the autoencoder does. Likewise, One Class SVM and Local Outlier Factor performed poorly in terms of their intracluster similarity ratios and individually they do not agree with any other algorithm, and only minimally with each other. Pairwise agreement and intracluster similarity are related; if two methods label the same points as anomalous, then they will have the same intracluster similarity ratio. The two evaluation methods thus rank the algorithms in the same order, with Isolation Forest performing best, followed by the Autoencoder and Elliptic Envelope.

Isolation Forest, the Autoencoder, and Elliptic Envelope all agree on 29.38% of anomalous traces. These traces account for almost all (99.7%) of the points in the pairwise agreement between the Autoencoder and Elliptic Envelope and are all included in the pairwise agreement between the Autoencoder and Isolation Forest, accounting for 57.1% of their intersection. Since all three algorithms agree on these observations, it is most likely that these observations are indeed anomalous. Additionally, since all three models classify these points as anomalous, all three models classify points with the same underlying structure as anomalous.

VI. CONCLUSION AND FUTURE WORK

In this paper, we find that all five models classify traces such that the anomalies and non-anomalies have statistically different port and protocol distributions, and thus all five unsupervised learning algorithms successfully separate the observations into two categories of network activity. However, we find that Isolation Forest, the Autoencoder, and Elliptic Envelope separate the anomalous and nonanomalous traces better than One Class SVM and Local Outlier Factor. Isolation Forest, the Autoencoder, and Elliptic Envelope detect anomalies at every contamination value since they have intracluster similarity ratios greater than 1 for every contamination. They also have much larger intracluster similarity ratios for each contamination than One Class SVM and Local Outlier Factor do, meaning they classify traces such that the difference between the anomalies and non-anomalies is much greater than it is when One Class SVM and Local Outlier Factor classify traces. Isolation Forest, the Autoencoder, and Elliptic Envelope also agree on anomalous classification over 29% of the time. This intersection shows that these three models find anomalies that have the same structure, and suggests the traces classified as anomalous by all three models are likely to be true anomalies.

In contrast, Local Outlier Factor and One Class SVM do not detect anomalies at every contamination value. They have intracluster similarity ratios less than 1 at low contamination values, indicating that they classify traces such that the anomalies are in fact not different from the non-anomalies. Additionally, these two models barely agree with each other on anomalous classification and do not agree at all with Isolation Forest, the Autoencoder, or Elliptic Envelope.

Though Isolation Forest, the Autoencoder, and Elliptic Envelope all effectively detect anomalies, we find that Isolation Forest performs best, classifying traces as anomalous such that the difference between the anomalous traces and the nonanomalous traces is the greatest. We also find that percent agreement with Isolation Forest also indicates performance, as the Autoencoder agrees most with Isolation Forest and has the second best intracluster similarity, and Elliptic Envelope agrees second best with Isolation Forest and has the third best intracluster similarity.

These findings are useful for cybersecurity researchers as unsupervised anomaly detection on NetFlow data limits the number of observations to evaluate in packet capture to ultimately detect malicious actors and is flexible to new types of malware. As Isolation Forest achieves a high intracluster similarity ratio at 1% contamination, our finding can be used to effectively reduce NetFlow to 1% of its original size, which significantly reduces the amount of data that systems administrators need to evaluate. Moreover, these models are useful for those who need to detect malicious activity on an enterprise network but lack labeled datasets to perform supervised learning.

However, the anomalous observations are not necessarily malicious and nonanomalous observations are not necessarily benign, as many malicious actors try to mimic normal network behavior so as to be undetected by network traffic monitoring systems, and unusual traffic does not necessarily mean malicious. Our results cannot be used to find true malicious

actors; they can only be used to detect suspicious traces in NetFlow data.

For future work we plan to perform the same models on labeled data to evaluate their true accuracy. We also plan to explore ensembling methods that combine our models' classifications.

ACKNOWLEDGMENT

"*The research reported in this document/presentation was performed in connection with contract number W911NF- 18-C-0019 with the U.S. Army Contracting Command - Aberdeen Proving Ground (ACC-APG) and the Defense Advanced Research Projects Agency (DARPA). The views and conclusions contained in this document/presentation are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of ACC-APG, DARPA, or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon*."

REFERENCES

[1] "The real and growing threat of cyber crime to corporations" 2016. cnbc.com/2016/09/24/the-real-and-growing-threat-of-cyber-crime-to-corporations-.html. Accessed: March 7, 2019.

[2] "The Evolution of Intrusion Detection/Prevention: Then, Now and the Future" 2017. secureworks.com/blog/the-evolution-of-intrusion-detection-prevention. Accessed: March 7, 2019.

[3] "Traffic Analysis for Network Security: Appraoches for Going Beyond Network Flow Data." 2016. insights.sei.cmu.edu/sei_blog/2016/09/traffic-analysis-for-network-security-two-approaches-for-going-beyond-network-flow-data.html. Accessed: February 27, 2019.

[4] "Network Flow Analysis." 2008. defcon.org/images/defcon-16/dc16-presentations/defcon-16-potter.pdf. Accessed: February 27, 2019.

[5] "Harnessing the Power of NetFlow and Packet Analysis" 2017. blogs.cisco.com/security/harnessing-the-power-of-netflow-and-packet-analysis. Accessed: March 7, 2019.

[6] Julina Zhang, K. Jones, Tianye Song, Hyojung Kang and D. E. Brown, "Comparing unsupervised learning approaches to detect network intrusion using NetFlow data," 2017 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, 2017, pp. 122-127.

[7] Gilberto Fernandes, Joel J.P.C. Rodrigues, and Mario Lemes Proença,"Autonomous profile-based anomaly detection system using principal component analysis and flow analysis," in Applied Soft Computing, vol. 34, pp. 513-525, 2015.

[8] V. L. Cao, M. Nicolau and J. McDermott, "Learning Neural Representations for Network Anomaly Detection," in IEEE Transactions on Cybernetics, vol. 49, no. 8, pp. 3074-3087, Aug. 2019.

[9] "Service Name and Transport Protocol Port Number Registry" 2019. iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml. Accessed: February 27, 2019.

[10] M. Turcotte, A. Kent and C. Hash, "Unified Host and Network Data Set", in ArXiv e-prints. Aug. 2017.

[11] "TCP (Transfer Control Protocol)." searchnetworking.techtarget.com/definition/TCP. Accessed: February 27, 2019.

[12] "What is connection-oriented?" searchnetworking.techtarget.com/definition/connection-oriented. Accessed: February 27, 2019.

[13] "Comparing anomaly detection algorithms for outlier detection on toy datasets" scikit-learn.org/stable/auto_examples/plot_anomaly_comparison.html#sphx-glr-auto-examples-plot-anomaly-comparison-py. Accessed: February 27, 2019.

[14] "Outlier Detection with Isolation Forest" 2018. towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45e. Accessed: February 27, 2019.