Title

Unsupervised Anomaly Detection in Daily WAN Traffic Patterns

Permalink

https://escholarship.org/uc/item/3n2021jm

ISBN

978-3-030-63392-9

Authors

Campbell, Scott

Kiran, Mariam

Wala, Fatema Bannat

Publication Date

2020

DOI

10.1007/978-3-030-63393-6_16

Unsupervised Anomaly Detection in Daily WAN Traffic Patterns

Scott Campbell, Mariam Kiran, and Fatema Bannat Wala

Energy Sciences Network (ESnet),
Lawrence Berkeley National Laboratory,
Berkeley, CA, USA
(scottc,mkiran,fatemabw)@es.net

Abstract. Growth in large-scale experiments using high capacity reliable networking as part of their design is creating a need for better monitoring and analysis of observed traffic. Network providers need intelligent solutions that can help quickly identify and understand anomalous behaviors at the network edge, allowing reactions to unexpected traffic or attacks on facilities and their peerings. However, due to lack of labeled data in network traffic analysis and user diversity, we introduce novel methods that process very large network datasets quickly for outlier identification.
In this paper, we leverage artificial intelligence (AI), network research, and edge computing to collect and train unsupervised classification algorithms using streaming data pipelines from multiple months of network flow records. Once trained, individual classifiers quickly observe and flag alerts in hourly behaviors. Our work describes building the data pipeline as well as addressing issues of false positives and workflow integration.

Keywords: network anomaly detection, NetFlow Data, unsupervised clustering methods, K-means, Gaussian Mixture Models

1 Introduction

Large experimental facilities, with their high-speed networks and traffic production rates, face enormous data movement challenges in supporting distributed science workflows. In these wide area networks (WANs), service providers need reliable solutions that can help quickly identify and understand anomalous behaviors at the network edge in near real-time, raising alarms and identifying unexpected attacks [3]. Many traditional approaches used in the security community for quickly identifying anomalous behaviors in a large wide area network designed for big data flows, relies on either performance metrics collected from tools like perfSONAR [9] or characterizing data volume observed in a particular period [19]. There is a need to develop efficient ways in which anomalous behaviors can be recognized quickly in large data volumes in near real-time based on the WAN network traffic patterns and high packet flow rates.

Network traffic classification has been extensively studied over the years [23], but classifying flows based on their behaviors, applications, and quality of service is a formidable task. Machine learning solutions can automate some of these efforts and find patterns to classify 'normal' versus 'abnormal' behaviors, providing some insight to security professionals in identifying potential network threats. Most network intrusion detection systems (NIDS) use flow statistics and features to build outlier detection algorithms, such as using random forest trees [24], fingerprinting [17], and behavioral comparisons. These rely on an offline analysis of large network traces (or network flow data) by using clustering to group similar flows together. For example, in studying network traffic entropy [20] found varying patterns of inbound and outbound traffic on weekdays versus weekends on real internet service providers (ISPs). Similar characteristics could be identified in common host connections, flow sizes, and topology used.

Identifying behavior patterns among hosts and how they connect to various endpoints is a common preliminary approach for anomaly detection in network communications [6]. Various dedicated solutions focus on identifying important features such as packet payload, port numbers, protocols [8] [21], and classification techniques to help identify potential threats [25]. However, the lack of labeled data sets makes it difficult for one to cluster results without knowing what each of the classes represents and measure the accuracy of classifiers where minimal information is available [12].

In this paper, we propose to explore unsupervised network traffic classification information, based on K-means and Gaussian methods, to address the issues of unsupervised machine learning for WAN-security. We develop novel methods to recognize anomalies in each method, by estimating how far the data point is from each cluster and density information. Specifically, our major contributions are as follows:

- We propose a novel anomaly finding approach that works with unsupervised clusters to identify potential outliers. With K-means we calculate the furthest data point from all clusters and in Gaussian models, we calculate the least density of data points in each cluster.
- We provide a detailed analysis of two classification techniques - K-means and Gaussian Mixture Models (GMM), used for the benefit of network traffic classification. We observed that feature selection affects the anomalies found.
- Our analysis is done on 3 real WAN data centers from January to May 2020, where we study weekend, weekday traffic patterns.
- We built an extremely efficient data pipeline by pre-processing data for the machine learning algorithms to use, offline training of the clusters, and online-anomaly detection.

The rest of the paper is organized as follows: Section 2 describes the background and literature review, Section 3 describes the key points and the motivation for this work. Section 4 provides the details on the overall methodology conducted, the data sets used, the feature extraction and the machine learning approaches explored. Section 5 gives the details of the primary analysis conducted for data set visualization. Section 6 illustrates the findings and results. Finally, Section 7 presents the discussion and conclusion of the research.

2 Related Work

Understanding network behavior patterns are crucial to network management and security tasks. Network traffic classification research has developed many approaches using statistical, supervised, and unsupervised machine learning techniques to categorize traffic patterns to understand activity across site endpoints, hours, days, and months.

Understanding security incidents is a classical challenge in network research. However, processing large amounts of network flow capture in meaningful time is itself a formidable challenge. Researchers have provided some solutions such as summary tools for identifying distributions of packet features (IP addresses and ports) [14], detecting volume surges, or changes in origin-destination [13] to help isolate anomalies or flow arrival time and packet types [17]. Techniques from statistical or machine learning solutions have been extensively provided to help summarize 'normal' and 'abnormal' traffic behaviors, but often are designed for specific data sets and network environments [1]. With the growing complexity of networks and devices themselves, network service providers need intelligent solutions that can quickly identify and understand anomalous behaviors at the network edge, raising alarms to prevent unexpected network attacks on their sites or peerings.

Networks sample packets using monitoring tools, extracting features that describe the behavior [16] [18]. Feature selection can play a significant role in the anomalies identified [22] [10]. Most current work maps traffic profiles to applications or protocols used [21]. Others have used machine learning to find day and night patterns to identify potential DDOS attacks, but in all cases, lack of labeled data makes it difficult to assess the accuracy of the results [7], [8], [11].

Recent methods used Gaussian Mixture Models to characterize NetFlow data into two categories elephant and mice flows [11], but showed that flow characteristics differ across the sites involved. Deep learning approaches have achieved accuracy of up to 96% for clustering [15], but require labeled data.

Compared to current solutions, this paper provides an end-to-end solution for identifying anomalous traffic patterns from multiple sites and leverages unsupervised machine learning algorithms to help raise alarms. Our work builds a data pipeline from individual NetFlow recorders, processes these as quick Splunk data summaries and runs machine learning code to identify potential anomalies. We also perform offline training and online detection using techniques - K-means, Gaussian Mixture Models - to show how the classifiers show different performances.

3 Key Points and Motivation

This section discusses our assumptions and study motivation.

3.1 Assumptions

Our goal is to build lightweight classifiers that will identify potential anomalies in network traffic. We take one hour blocks of sampled NetFlow records and apply statistical and counting measurements as summaries to feed to the classifiers. Our work relies on the following assumptions:

- **Building 'normal' behavior classifiers.** Deviations from normal traffic behaviors or stable measurements can be identified and are interesting to both the network engineering and security groups because of their unusual characteristics. These deviations are identified via testing against models trained with traffic observed from normal situations. Examples of deviations might be bursts of new addresses or ports (both in or out of a site) as well as more subtle changes like the shape of data measurements. How we identify these is to a large degree the motivation for this work. Due to the lack of labeled data and a diverse set of users, we base our approaches on [14] where using summarizing techniques we will create hourly patterns to train our classifier as normal behaviors
- **Hourly summaries can help identify morning and afternoon patterns.** This assumption relies on the hypothesis that network usage differs during regular working hours when the users are expected to be more vigilant versus hours in the evening. Since our data sets primarily consist of the research-based WAN network traffic, the chances of observing distinct patterns in the hourly summaries in our training data sets are low.
- **Offline training for classifiers.** We expect to be able to train the classifier using unsupervised clustering methods (mainly K-means and GMM) using a data set known to exhibit normal network traffic patterns. Hence, once the classifier learns what the normal traffic behavior looks like, it can then decide if a given test data set exhibits normal patterns or if it contains anomalous behavior.
- **Online access to the trained clusters to find patterns on the fly.** We expect that once the classifier is trained offline using the datasets known to have normal traffic patterns, it can then be used in near real-time to detect a given pattern exhibiting any abnormal behavior on-the-fly. The classifier then assesses how far the given test pattern falls from the normal clusters within a given threshold.

3.2 Intuition behind our methods

Research WANs versus Commodity WANs. For the initial experiment, we chose a dataset based on network traffic from the DOE Open Science HPC Facilities. We illustrate our approach with real NetFlow traffic traces from a DOE research WAN (ESnet, www.es.net), across 3 data centers between the months of January and May 2020. The expectation is that the traffic profile for these facilities will have less interactive human activity (such as web browsing) which exhibits a strong diurnal weekday pattern [2], and a far greater proportion of long duration, high volume data transfers than would be expected in Commodity traffic [4].

Network traffic monitoring tools. Traffic traces are collected via tools such as Simple Network Management Protocol (SNMP), sflow, and Netflow. Some (like SNMP) can be used to collect time-stamped information on CPU, memory utilization, and interface counters at end-points. Sflow and NetFlow records provide a local router view and provide details of fine-grained traffic view including key features such as protocols (e.g. TCP, UDP, etc), interfaces, source, and destination IP addresses and even flow speeds [5].

Unsupervised clustering methods for anomaly finding. ESnet has a unique perspective with regard to the behavior of network traffic in and around large multiuser facilities. Anomaly detection in data sets can be used for both security as well as traffic engineering. The selection of data sources is dedicated high-performance computing within three large scale office of science computing sites named Site-1, Site-2, and Site-3 for paper anonymity.

3.3 Unsupervised clustering algorithms

In this section, we review the clustering algorithms we use to build our classifiers for unlabelled traffic data. In particular, reviewing K-means and Gaussian mixture models.

K-means Clustering technique. Given a set of data blobs, the K-means algorithm can quickly label these into clusters such a way to closely match relevant data points together. This is calculated based on iterations of distances between the clusters to form circular shapes.

Why is this good? K-means is a good approach to explain how data sets with seemingly unrelated features can be grouped, just based on their empirical distances.

Gaussian Mixture Models (GMMs). Gaussian mixture models work to find multi-dimensional Gaussian probability distributions that best fit training data. Based on calculating density estimation and probability that a data point belongs to a cluster, this method works well to generalize non-uniform data.

Why is this good? In K-means, there is no intrinsic probability measure or uncertainty in the clusters. GMMs are better to characterize different shapes of the data which do not exist as clear circles.

4 Methodology

We propose to develop an end-to-end traffic classification mechanism that will work in three phases (Fig. 1): First, the Trace collection phase uses data pipelines that create hourly summaries of NetFlow records for each site from the routers.

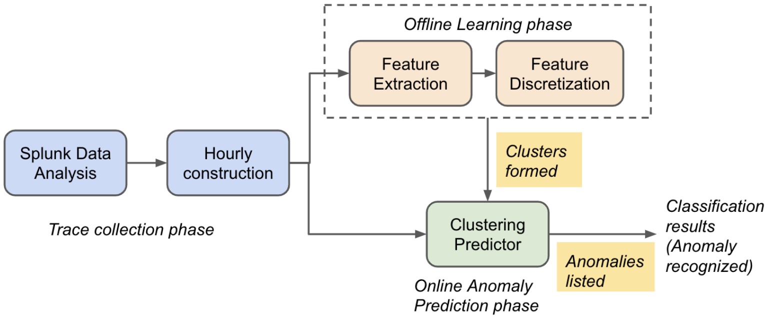


Fig. 1: Overall methodology of recognizing anomalies in Netflow characteristics.

Second, an offline learning phase will use clustering methods to group similar traffic flows together. To verify this behavior we will divide our data into training and test data and compare the results found by the classifier. And lastly, we will deploy these classifiers with the pipelines to perform online clustering as data is collected. We use 3 data centers as we anticipate different traffic patterns.

4.1 Trace collection: Building Streaming Data Pipelines

Phase one consists of trace collection and data reduction. ESnet sees, in aggregate, between 20-50 million net flow records per hour on average which follow a classic weekday, diurnal pattern. As shown in Fig. 2, this data is gathered through a set of flow collectors and sent to a splunk instance which indexes and stores the records in a performant searchable format. Flow data at ESnet is sampled at 1:1000 before being sent to a collection which plays an important role in the type of analysis that is possible. Values that can be approximated by large sample sets work well, but exact enumerations are not possible. For example, looking at the exact number of packets to port 80/tcp, or if a specific IP has been seen are not possible with sparsely sampled data, but estimating the ratio of 80/tcp vs. 443/tcp is possible.

To analyze the classification techniques we used data sets from three DOE data center sites, we will be referencing them as Site 1, Site 2, and Site 3 in this paper. For this, the raw data is filtered for site Autonomous System Number and or network subnet to define a site or region of interest. In an effort to reduce the effects of random scanning and background noise/radiation an additional filter was imposed which removed records containing less than 64 bytes.

This filtering reduces the data volume down to around 1.3 million records per hour. The data summary process walks through this data, breaks it into one-hour blocks, and generates a set of summary statistics based on counting and heuristics for each block. The reduction in data volume for the summary data set (millions of flow records down to one set of statistical/count measurements) lets us process large windows of data for the model building and comparison in very little time.

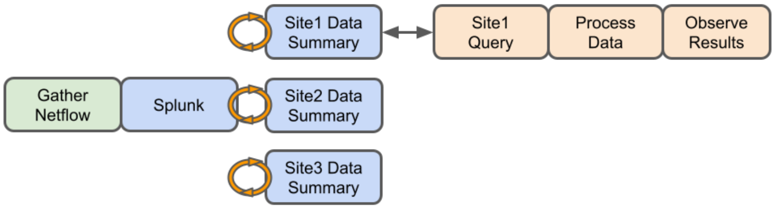


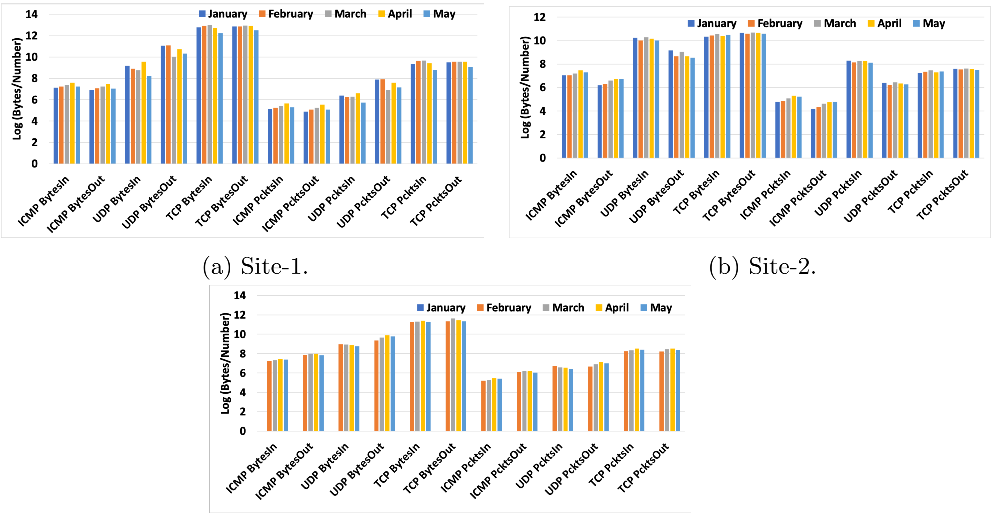
Fig. 2: Workflow for data analysis.

Data summary consists of the one-hour measurement blocks, features concerning byte counts, packet counts, unique server IP, and unique server port are broken out into direction (inbound vs. outbound) as well as protocol (TCP, UDP, and ICMP). The standard deviation for byte and packet counts are broken out similarly. Flow records are combined based on the heuristic that the lowest port represents the service which is based on the classic fixed service port and ephemeral client-side port. This is not always true in terms of dynamically generated services and data transfers (for example Globus GridFTP), but since we are looking at aggregate behavior across a large number of sessions these ephemeral services should average out.

A feature is a property of a data sample, where average, mean, median, and standard deviation can also be features. Unsupervised feature extraction helps identify patterns from features in trace data.

Type of feature	Feature description
Byte Count (TCP, UDP, ICMP)	Integer
Packet Count Inbound (TCP, UDP, ICMP)	Integer
Packet Count Outbound (TCP, UDP, ICMP)	Integer
Std Dev Bytes Inbound (TCP, UDP, ICMP)	Float
Std Dev Bytes Outbound (TCP, UDP, ICMP)	Float
Std Dev Packets Inbound (TCP, UDP, ICMP)	Float
Std Dev Packets Outbound (TCP, UDP, ICMP)	Float
Unique Server IP Inbound recorded this hour	Integer
Unique Server IP Outbound recorded this hour	Integer
Unique Server Port Inbound recorded this hour	Integer
Unique Server Port Outbound recorded this hour	Integer
Hour	date-hour
Weekday	date-wday

Table 1: Features unsupervised clustering from hourly NetFlow summaries.



(a) Site-1.

(b) Site-2.

(c) Site-3.

Fig. 3: Traffic Distribution across all sites in months Jan-May 2020.

Training and Test data. We use January and February 2020 data as training data, and March-May 2020 as test data. The TCP, UDP, and ICMP patterns are shown in Fig. 3 over the 5 months.

4.2 Offline Learning in Classifiers

For phase two, we use K-means and GMM methods to train our classifiers into unsupervised clustering methods. Fig. 4 shows how the clusters are formed on training data. The test data is then grouped into one of the clusters.

4.3 Online Anomaly Finding

In phase three, we use K-means and GMM models to perform anomaly findings. Because of the lack of labeled data, we cannot specifically identify an anomaly unless all anomalies are grouped in particular clusters. To counter this, we define an anomaly that falls far from the 'normal' behavior in the training data sets. This is calculated in each clustering technique separately as shown in Equations 1, 2. We calculate an anomaly based on how far the data point is from the centroid and the density of the cluster. K-means assumes circular clusters, where we calculate centroid and radius of the original clusters. In GMM, we use Gaussian distribution to calculate the probability of each data point and list the least probability as a possible anomaly.

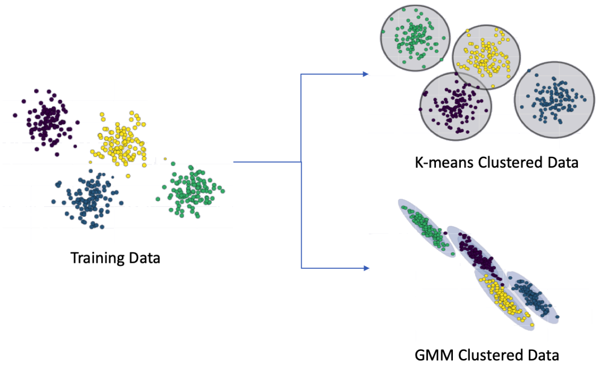


Fig. 4: Calculating anomalies based on how far the point is from the cluster.

In K-means, we can calculate the distance to each cluster by,

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{1}$$

where clusters of k groups can assign data points based on the euclidean distance function. The higher the distance from all clusters, the higher the probability of the data point to be anomalous.

For calculating anomalies with GMMs, we use the expectation and maximization method to calculate the probabilities of a data point belonging to a cluster. This probability can be defined as,

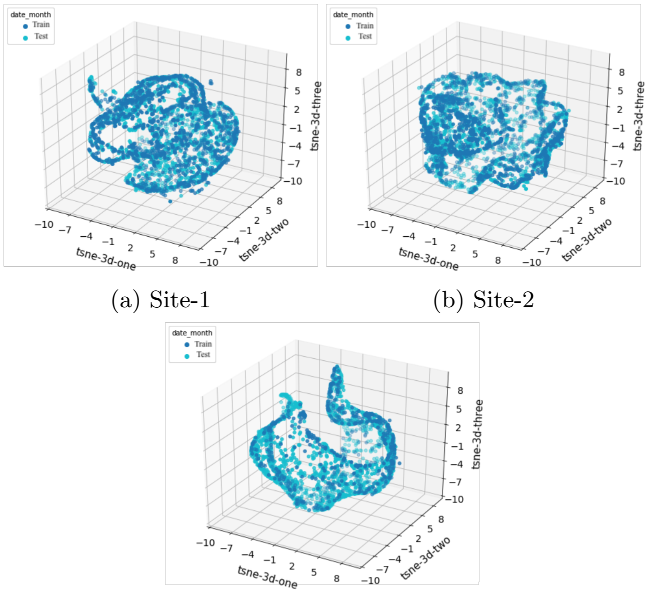
$$w_j^{(i)} = \frac{g_j(x)\phi_j}{\sum_{l-1}^{k} g_l(x)\phi_l} \tag{2}$$

where $g_j(x)$ represents the multivariate Gaussian of each cluster and ϕ_j represents the prior probabilities. These can be printed out to denote an average probability that they belong to a cluster. We use a threshold of -0.5 to denote that this is a very low probability that the data point belongs to a cluster and label these as anomalies.

5 Preliminary Analysis

We visualize the data using PCA (Principal Component Analysis) and t-SNE (t-Distributed Stochastic Neighbor Embedding) to represent a high-dimensional dataset (38 features, shown in Table 1) in a low-dimensional space of 2,3 dimensions. Fig. 5 shows the Site-1, Site-2 and Site-3 divided into training and test visuals. The sub-figures show different behaviors in the months, particularly in test data, impacted with COVID-19 work changes.

In contrast to PCA which simply maximizes the variance, t-SNE creates a reduced feature space where similar samples are modeled by nearby points and dissimilar samples are modeled by distant points with high probability. We got optimum results and the KL divergence was minimum for the 3 dimensions reduction (n=3) of the original data set with t-SNE algorithm with perplexity = 40 and 300 iterations. For Site-3 we see a tight clustering for weekdays, but from March-May'20, it shows a diverse traffic profile in Fig. 5, showing that the profile does change and would be picked up as anomalous.



(a) Site-1

(b) Site-2

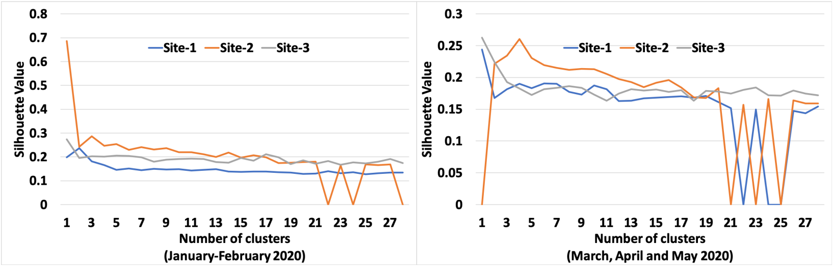
(c) Site-3

Fig. 5: TSNE visualization of Training vs Test Data of all Sites.

6 Experimental Results and Discussions

6.1 Silhouette analysis for Optimal Clustering

We perform a silhouette analysis to study the optimal number of clusters in the training and test data sets. This informs the unsupervised clustering results. Fig. 6 shows these measures of how close each point is in one cluster to points in neighboring clusters with the maximum value gives the optimum number of clusters. We find that optimal clusters in Site-1 are 3, Site-2 and Site-3 is 2 for training. We also performed a similar analysis for test data, showing that there is considerable variability in characteristics.

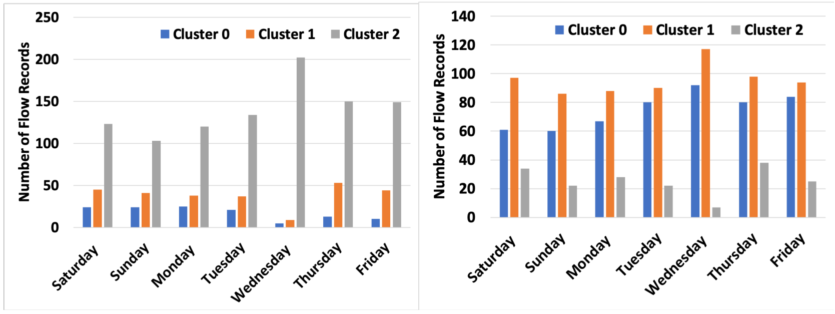


(a) Training Data.

(b) Testing Data.

Fig. 6: Silhouette analysis to gain optimal clusters in the data.

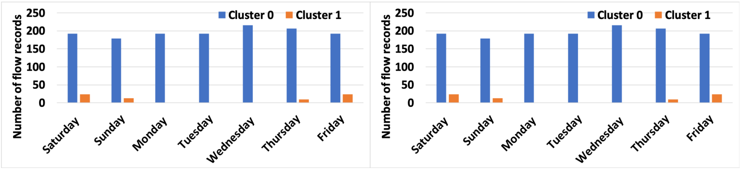
6.2 Clustering Weekdays and Weekends in Training Data



(a) K-means on Training Data.

(b) GMM on Training Data.

Fig. 7: Listing weekdays recognized in each cluster in Site-1.

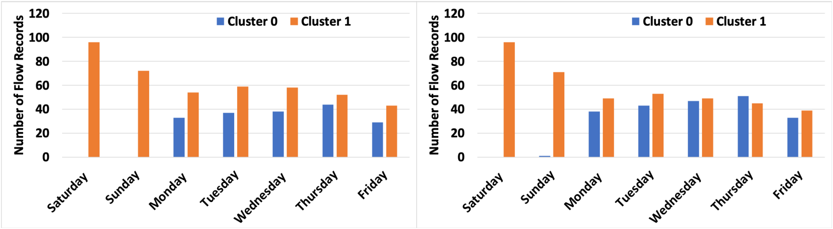


(a) K-means on Training Data.

(b) GMM on Training Data.

Fig. 8: Listing weekdays recognized in each cluster in Site-2.

We listed how the days were being recognized in each of the clusters. Fig. 7 shows that K-means and GMM both cluster data differently and there are no



(a) K-means on Training Data.

(b) GMM on Training Data.

Fig. 9: Listing weekdays recognized in each cluster in Site-3.

distinct patterns between the weekdays. Comparatively in Fig.s 8 and 9, we do find that there are individual clusters that can identify specific days of the week such as Tuesdays in Cluster 0 in Site-2, and Saturdays in Cluster 1 in Site-3. However, since these also appear in other clusters, it is difficult to run test data and measure this assumption. This shows that the clusters selected in training data are insufficient to recognize individual days in the test data across all sites.

6.3 Identifying Outliers in Test Data

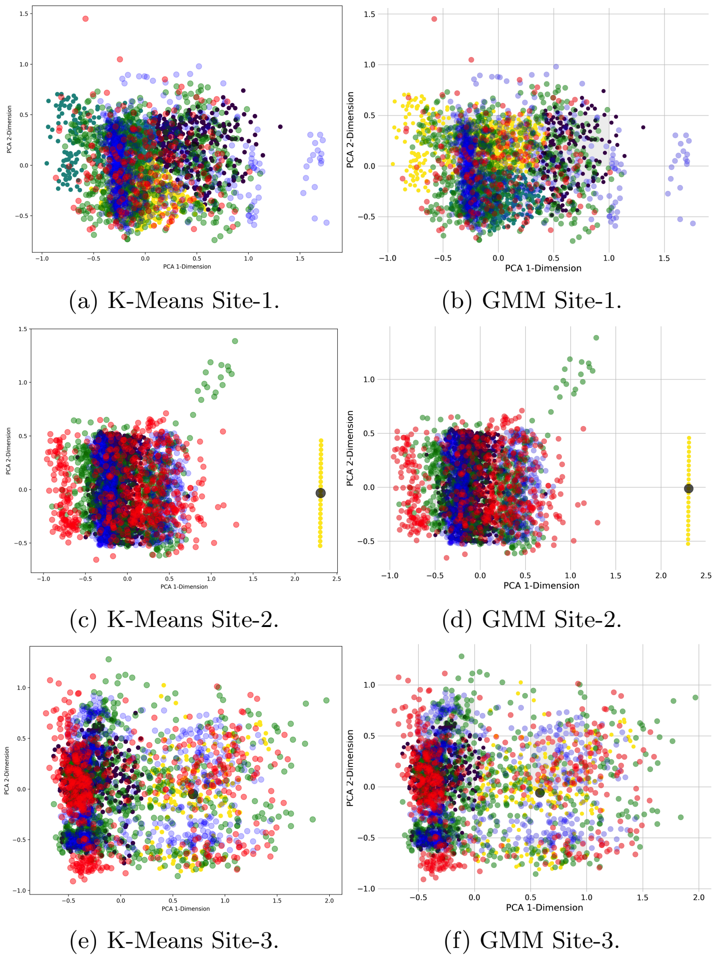
Fig. 10 shows the representation of the test data sets with the training data clusters based on K-means and GMM results. In Fig. 10b, we witness that some behaviors in April and May are recognized as anomalies. As GMM calculates anomalies based on ellipsoid density, it recognizes lesser anomalies that K-means which uses only centroids and cluster density to calculate anomaly boundaries. In Fig. 10d, most of the March and May data sets are recognized as anomalous behaviors, in Fig. 10f, nearly all March and May are recognized as anomalies.

The results are summarized in Fig. 11 which shows total anomalies in each site's behavior. GMM is able to recognize fewer anomalies and we know from background information that there were no anomalies recorded in the real dataset. This is an unsupervised technique that lists how many records fall outside the common clusters formed in the training data sets, and because the behavior patterns changed in the months of March onwards these fall outside the clusters formed.

6.4 Impact of Selected Feature Discretization using Domain Knowledge

Feature discretization takes a subset of features (knowledge-informed) in the data summary object for training and testing rather than the entire object. This not only gives a much better focus on the type of anomaly to look for, but also allows the analyst to better understand *what* specifically has changed in testing.

Specific feature selection is typically driven by the combination of fields that contain data related to the characteristic to measure and are informed by feedback from a domain expert. Individual fields are defined in Table 1 and can



(a) K-Means Site-1.

(b) GMM Site-1.

(c) K-Means Site-2.

(d) GMM Site-2.

(e) K-Means Site-3.

(f) GMM Site-3.

Fig. 10: Plotting all sites training and test data. Colors present: March (green), April (blue), May (red). Others colors (yellow, purple) are training data clusters.

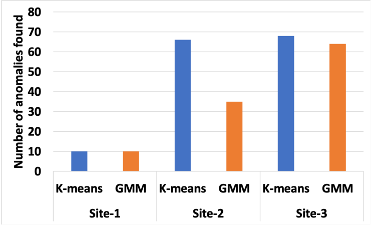


Fig. 11: Total anomalies at each Site during March - May '20.

be categorized into data volume (bytes and packets), connections (host, port, direction), and descriptive statistics of the data volume. An example would be the count of unique outbound network addresses. This can be captured by the features: 'ServIPOut' (for TCP, UDP, ICMP). A more complex example might have many more fields. Training and test groups are generated using the same ratios as for K-means and GMM.

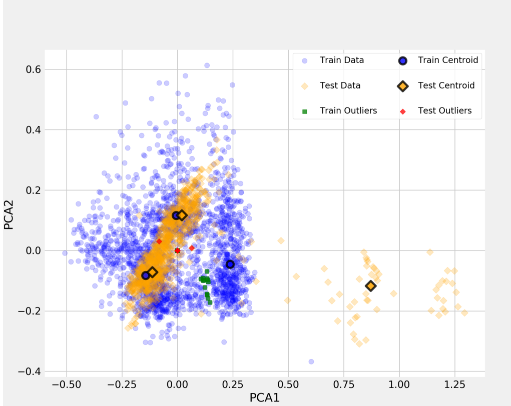


Fig. 12: Feature discritization sample showing training (blue), test (orange) data.

Training data is normalized via MinMax to prevent biases in clustering from large values, then data dimensionality is further reduced via PCA. After running through GMM we end up with a set of matrices that hold (amongst other things) labels for cluster assignment as well as predict the posterior probability of each data point. Since cluster assignment is driven by the probability that a data point belongs to a cluster, a simple threshold test can be used to identify low likelihood events.

Outlier detection. Represented graphically, a 2D view of train and test clusters can be seen in Fig. 12. Here training data is in blue diamonds and test data in orange circles. The usual color per cluster is not used here since we are looking at the probability of assignment to any cluster rather than the actual cluster membership. Outliers for test data are in red triangles and for completeness outliers in the training data, are in green squares. As mentioned above, outliers identify when the assignment probability returned by GMM clustering is below a threshold. More detailed information about Fig. 12 will be found in the next section.

Addressing Field decomposition. Knowing that there are outlier elements in the test data can be informative - in this case, we can identify the outliers as outgoing IP services since both the x-axis (PCA-1) and y-axis (PCA-2) are composed of these features. To get greater details it is necessary to examine the PCA eigenvectors in more detail.

```
index: [‘ServIPOut: 1’, ‘ServIPOut: 6, ‘ServIPOut: 17’]
X: eigenvalue:  0.035  percent:  0.5484  coeff:  [-0.979 -0.188 -0.074]
Y: eigenvalue:  0.022  percent:  0.3407  coeff:  [-0.085  0.047  0.995]
              residual percent:  0.11081
```

Here ‘ServIPOut: 1’, ‘ServIPOut: 6’, ‘ServIPOut: 17’ represent the count of unique external destination addresses during the 1 hour sample window. In terms of how they relate to the coordinate PCA axis seen in Fig. 12 we look at the set of weights or coefficients assigned to each component eigenvector in the figure. The text above defines the various weights assigned for each of the values, so in this case we can see that the singular majority of the x-axis (first component) is ICMP (-0.979) and the y-axis is UDP (0.995).

The outlier test data centroid around x=0.85 is an interesting artifact worth understanding. The outlier test data represents two individual UDP scans that happened in the same week in late April directed at Site-1. Examining the original flow data we see two reasons why this ended up in the data. First, the byte sizes for the per packet scanning was above the threshold which defined background radiation described in Section 4.1 . Second, the number of addresses and destination ports covered in the scan was 2-3x larger than what is typically seen in scanning during the training period.

In order to automate the analysis of traffic data, we look at the set of assignment probabilities returned from the training model. Looking at average and variance for the set provides a naive measurement of how good the model fits the test data in a general sense, while skewness and kurtosis provide a measure of asymmetry and the presence of outliers from a normal distribution.

7 Conclusions

In this paper, we analyzed traffic profiles and used these to predict anomalous traffic patterns. In security research, we assume that daily patterns are enough to recognize anomalous behaviors as we classify based on the hour of the day. However, with the changes in COVID working patterns, this assumption did not hold as most behaviors in the test data were labeled as anomalies, even when they were not.

Further our unsupervised clustering technique proved useful to find outliers in unlabelled data sets. GMM was able to provide better results than K-means which assumed a more uniform circular pattern of characteristic profiles, finding more false anomalies.

In the future, we will be deploying additional online classifiers to collect anomalies at each site's edge. Further, these techniques will be adapted to work with lower-level granular data. For example to find the reasons why certain data points are considered outliers such as a new site appearing or a unique transfer size which has never been done before. Our results show the potential to be deployed across many other ESnet network peerings and points of presence in DOE.

Acknowledgements

This work was supported by the U.S. Department of Energy, Office of Science Early Career Research Program, under LAB 16-1625 for Large-scale Deep Learning for Intelligent Networks FP00006145.

References

1. Ahmed, M., Mahmood, A.N., Hu, J.: A survey of network anomaly detection techniques. Journal of Network and Computer Applications 60, 19 – 31 (2016), http://www.sciencedirect.com/science/article/pii/S1084804515002891
2. Benson, T., Akella, A., Maltz, D.A.: Network traffic characteristics of data centers in the wild. SIGCOMM conference on Internet measurement (2010)
3. Campbell, S.: Esnet wan security project updates. In: Internet2 Technology Exchange (2018)
4. Campbell, S., Lee, J.: Prototyping a 100g monitoring system. Euromicro International Conference on Parallel, Distributed and Network-based Processing (2012)
5. Claise, B.: Cisco systems netflow services export version 9. Internet Engineering Task Force [IETF] (2004)
6. Dewaele, G., Himura, Y., Borgnat, P., Fukuda, K., Abry, P., Michel, O., Fontugne, R., Cho, K., Esaki, H.: Unsupervised host behavior classification from connection patterns. International Journal of Network Management 20(5), 317–337 (2010), https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.750
7. Erman, J., Arlitt, M., Mahanti, A.: Traffic classification using clustering algorithms. In: SIGCOMM Workshop on Mining Network Data. p. 281–286. ACM (2006), https://doi.org/10.1145/1162678.1162679
8. Erman, J., Mahanti, A., Arlitt, M., Cohen, I., Williamson, C.: Semi-supervised network traffic classification. In: International Conference on Measurement and Modeling of Computer Systems. p. 369–370. ACM (2007), https://doi.org/10.1145/1254882.1254934
9. James Zhang, Ilija Vukotic, R.G.: Anomaly detection in wide area network meshes using two machine learning algorithms. Elevier Future Generation Computer Systems (2019)
10. Kim, H., claffy, k., Fomenkov, M., Barman, D., Faloutsos, M., Lee, K.: Internet traffic classification demystified: Myths, caveats, and the best practices. In: Conference on emerging Networking EXperiments and Technologies (Dec 2008)
11. Kiran, M., Chhabra, A.: Understanding flows in high-speed scientific networks: A netflow data study. Future Generation Computer Systems 94, 72 – 79 (2019), http://www.sciencedirect.com/science/article/pii/S0167739X18302322
12. Kohout, J., Pevný, T.: Unsupervised detection of malware in persistent web traffic. In: IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 1757–1761 (2015)
13. Lakhina, A., Crovella, M., Diot, C.: Diagnosing network-wide traffic anomalies. In: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. p. 219–230. ACM (2004), https://doi.org/10.1145/1015467.1015492
14. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. SIGCOMM Comput. Commun. Rev. 35(4), 217–228 (2005), https://doi.org/10.1145/1090191.1080118

15. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., Lloret, J.: Network traffic classifier with convolutional and recurrent neural networks for internet of things. IEEE Access 5, 18042–18050 (2017)
16. Nguyen, T.T.T., Armitage, G.: Training on multiple sub-flows to optimise the use of machine learning classifiers in real-world ip networks. In: IEEE Conference on Local Computer Networks. pp. 369–376 (2006)
17. Nguyen, T.T.T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Communications Surveys Tutorials 10(4), 56–76 (2008)
18. Nguyen, T.T.T., Armitage, G., Branch, P., Zander, S.: Timely and continuous machine-learning-based classification for interactive ip traffic. IEEE/ACM Transactions on Networking 20(6), 1880–1894 (2012)
19. Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. IEEE Symposium on Security and Privacy (2010)
20. Tari, R.M.: On the move to meaningful internet systems. Confederated International Conferences On the Move to Meaningful Internet Systems 2 (2007)
21. Zander, S., Nguyen, T., Armitage, G.: Automated traffic classification and application identification using machine learning. In: IEEE Conference on Local Computer Networks. pp. 250–257 (2005)
22. Zhang, H., Lu, G., Qassrawi, M.T., Zhang, Y., Yu, X.: Feature selection for optimizing traffic classification. Computer Communications 35(12), 1457 – 1471 (2012), http://www.sciencedirect.com/science/article/pii/S0140366412001259
23. Zhang, J., Xiang, Y., Wang, Y., Zhou, W., Xiang, Y., Guan, Y.: Network traffic classification using correlation information. IEEE Transactions on Parallel and Distributed Systems 24(1), 104–117 (2013)
24. Zhang, J., Zulkernine, M.: Anomaly based network intrusion detection with unsupervised outlier detection. In: IEEE International Conference on Communications. vol. 5, pp. 2388–2393 (2006)
25. Zhao, J., Huang, X., Sun, Q., Ma, Y.: Real-time feature selection in traffic classification. Journal of China Universities of Posts and Telecommunications 15, 68 – 72 (2008), http://www.sciencedirect.com/science/article/pii/S1005888508601582