Spectral Clustering Technique for Classifying Network Attacks

Anna Little
Jacksonville University
Jacksonville, Florida 32211
Email: alittle2@ju.edu

Xenia Mountrouidou
Wofford College
Spartanburg, South Carolina 29303
Email: mountrouidoup@wofford.edu

Daniel Moseley
Jacksonville University
Jacksonville, Florida 32211
Email: dmosele@ju.edu

***Abstract*—The problem of differentiating regular from anomalous traffic has been studied extensively. However, the classification of anomalous traffic to different types of attacks remains a difficult and widely unexplored area. In the age of big data analytics it is becoming paramount to automate the security process, such as reaction to attacks and mitigation based on their type. This paper investigates the use of spectral clustering techniques for classifying computer network attacks. Spectral clustering is a highly robust classifier for big data, and is found to accurately and efficiently classify the attack data using a minimal number of select features. Extensive investigation into feature selection and weighting is discussed. Our classification results can easily be adapted by an Intrusion Detection System (IDS) for real-time attack detection, and the classification information used to mitigate future attacks.**

***Index Terms*—Clustering algorithms, classification algorithms, computer security, intrusion detection.**

1. Introduction

While much research has been done in distinguishing normal network traffic and flash crowd events from traffic anomalies and attack traffic, not much work has been done in classifying different types of attacks. Though it is important to be able to distinguish attack from benign traffic, it is also vital to know what kind of attack is underway to deploy the most appropriate countermeasures.

There are several reasons why distinguishing a specific network attack type is important. First, it is helpful to employ sophisticated techniques in an automated manner, without the need for human intervention, such as network administrators reviewing network logs. Second, information sharing in computer security is vital to alert organizations about new attacks. The importance of information sharing is demonstrated by the considerable work that has been done on protocols to share information about attacks and perpetrators, i.e. [1] and [2]. Thus, the quality of information and details about specific attacks is essential. Finally, Advanced Persistent Threats (APT) start with simple network scanning or Denial of Service (DoS) attacks. Being able to distinguish these may prevent more sophisticated attacks.

In this paper we present a classification algorithm that is completely automated and can be used to classify representative computer network attacks. Our algorithm specifies three different groups of attacks: Probe, DoS, and User to Root (U2R) as well as normal traffic with good accuracy and efficiency. These types of attacks are frequently encountered in computer networks or may be the precursor of a larger, more sophisticated attack.

One of the pressing issues in present-day networks is the amount of information that is produced due to high speed, high volume transactions. Classification of large volumes of traffic is considered a difficult big data problem. The k-means algorithm is one of the most common classification algorithms used; however it lacks accuracy and sophistication. Neural networks is another common method for classifying network attacks [3].

What distinguishes our algorithm from other papers on classification is the low number of features used to classify attacks to different groups. Furthermore, we are using spectral clustering for classification, and our algorithm's efficiency is apparent from the minimal amount of features that are needed to extract an accurate classification. Finally, we rely only on network traffic data (tcpdump [4]); thus this algorithm can be used with Intrusion Detection Systems (IDSs).

2. Related Work

There is a large body of work revolving around detection of network attacks aiming to improve the accuracy and efficiency of IDSs. Our method of classifying attacks focuses on spectral clustering, which can be used to group a variety of network attacks and to distinguish traffic anomalies from regular traffic. Therefore, we have divided the related literature in three categories:

- methodologies applied to distinguishing regular user traffic or flash crowd traffic from network attack traffic
- algorithms used to classify attacks to specific groups, and
- clustering algorithms for classification and machine learning.

Our solution deals with clustering, attack detection, and classification. Thus, we present a brief review of each category below.

The problem of detecting a computer network attack, i.e., distinguishing regular user traffic from attack traffic, is fundamental when building online Intrusion Detection Systems (IDS). Real time systems such as Snort [5] and Bro [6] have been developed. Snort is based on a set of rules that recognize attack signatures and patterns in network traffic and raise alerts when an anomaly in traffic is detected. Bro is state based, relying on a system of events and event handlers that establish policies on how to react to an attack. The limitation of IDSs is that they usually only detect attacks and traffic anomalies. They do not offer information or a grouping of the types of attacks. A survey on Anomaly-based Network IDSs (A-NIDS) is presented in [7]. IDSs are divided in three categories: statistics based, knowledge based, and machine learning based. Still, there are open issues with the accuracy of all these IDSs especially when it comes to recognizing specific attacks.

Another difficult problem is discriminating attack traffic from flash events, i.e. unusually high regular user traffic. In [8] the authors use Pearson Correlation and devise a statistical measure to distinguish traffic anomalies, specifically Denial of Service (DoS), versus flash events. In [9] the authors use source IP entropy and traffic cluster entropy to distinguish Distributed Denial of Service (DDoS) attacks from flash events. Both of these detection mechanisms are based on statistical measures and attackers can "train" their traffic generator to mislead them. A novel approach that is based on creating a robust regular user traffic model is presented in [10], where the authors use an anomaly detection score and the Bayes rule to differentiate user traffic versus attack traffic.

Most of the classification research that has been done places great importance on the features used, having differing feature sets even when using the same data. In [11], the author developed a decision tree based classifier using the 41 features provided with the KDD Cup 99 data set [12] to distinguish DoS attacks. In [13] the authors used 7 features to detect DDoS attacks again using the same data sets. These features included number of UDP echo packets to a specified port, number of connections to the same host, number of ICMP echo reply packets from the same source, number of connections that have SYN errors using the same service, number of connections having the same window size, sequence number, and packet length, number of packets in the URG flag set in TCP header, and type of service. In [14] the authors provide reduced sets of features (each a subset of the 41 features from the KDD Cup data) for each type of attack valuable for distinguishing the four types of attacks. Similarly, in [15] the authors discuss what features distinguish Probe, DoS, R2L, and U2R attacks and additionally which features are not useful. Our classification algorithm uses spectral clustering with only six significant features; this sets it apart from the above classification systems in terms of efficiency and accuracy.

Several classification systems use neural networks to improve detection accuracy and flexibility for attack classification. Using a keywords list the authors in [16] calculate a score, then apply detection and classification with neural networks. This method improves accuracy and lowers false alerts, but it is heavily dependent on a preexisting keyword list. In [3] the authors detect DDoS attacks using neuro-fuzzy networks. Again, in this work only one type of attack is detected.

Once appropriate features have been selected, various clustering algorithms in machine learning can be used to classify the attacks. Clustering algorithms fall into two broad categories: hierarchical and partitioning [17]. Hierarchical algorithms create a multiscale clustering structure by iteratively merging or dividing clusters. Partitioning algorithms produce a unique partition of the data into generally disjoint clusters. Partitioning algorithms include relocation, density-based, and spectral methods [17]. Relocation algorithms such as k-means [18] and MCLUST [19] build convex clusters and fail when there are nonlinear boundaries between clusters. Density-based methods such as DBSCAN [20] and spectral methods such as those in [21] and [22] are more reliable for non-convex clustering structures.

3. Processing the Data

We used the 1999 DARPA data set [23] to verify the accuracy and efficiency of our spectral clustering algorithm. An older version of this dataset was used for the famous Knowledge Discovery in Databases challenge (KDD cup [12]). Although the KDD cup included a variety of audit as well as traffic data and was processed to distinguish traffic flows, our algorithm utilized the raw network traffic logs. Therefore, with some adjustments our classification algorithm can be used efficiently by an IDS. Furthermore, what sets our methodology apart from the related work is the low amount of features required as well as the spectral clustering accuracy of classification.

Although there has been criticism about the 1999 DARPA dataset regarding its appropriateness to evaluate an IDS as well as its timeliness [24], it still represents one of the most complete datasets and has set a standard for IDS accuracy evaluation. The DARPA dataset was created in a realistic network testbed that emulates the structure of an Air Force base network. The background traffic of this dataset emulates users on thousands of hosts modeled by realistic government sites. More than 200 instances of 58 distinct attack types were launched against Unix and Windows NT machines. For the evaluation of our classifier it supplies a variety of attacks mixed with normal traffic and this sets it apart from datasets that include only one type of attack.

Moreover, we have produced realistic Distributed DoS (DDoS) TCP SYN flood network traffic in a real world cloud infrastructure, Global Environment for Network Innovations (GENI) [25]. We have implemented a botnet of ten zombie nodes, a victim node that ran Apache server, and a node that produced background data based on a realistic long tail traffic distribution. To this end, we have emulated a real world, high speed attack using the GENI cloud infrastructure. We used tcpdump to collect these data.

3.1. Types of Attacks

A summary of the attacks that are grouped by our novel spectral clustering algorithm appears in Table 1. Our classification algorithm distinguishes three groups of attacks: Denial of Service (DoS) and Distributed DoS (DDoS), Network Scans (Probes), and User to Root (U2R). DoS and DDoS attacks are the most common and easy to deploy. The goal of these attacks is to flood a network with traffic in order to prevent legitimate users from accessing network services. Although today's computer networks may overcome DoS attacks by reserving more bandwidth, they are still a considerable threat. A recent 400 Gigabyte DDoS attack detected by Arbor Networks [26] demonstrates the gravity of these attacks.

TABLE 1. ATTACK CATEGORIES

Attack Category	Attack Names
DoS	apache, TCP SYN flood, udpstorm
Probe	ipsweep, ls, portsweep, queso, ntis, satan, resetscan, ntinfoscan, insidesniffer
U2R	anypw, casesen, eject, fdformat, loadmodule, ntfsdos, perl, ps, sechole, sqlattack, xterm, yaga, ffbconfig

There is a wide range of DDoS attacks [23], from floods, amplification, and high bandwidth attacks (e.g. SYN flood, "ping of death", and smurf) to low bandwidth attacks that target large files and other website or database vulnerabilities (e.g. crashiis, land, and teardrop). Due to their wide range of characteristics, from low to high bandwidth and a variety of sources and destinations, it is inherently difficult to create a single group of DoS attacks. Therefore, we have excluded the low bandwidth attacks and focused on high bandwidth and amplification attacks (SYN flood, apache).

The second set of attacks that is classified by our algorithm are network probing attacks. The goal of a probe attack is to scan a network for open backdoors and vulnerabilities. A probe attack is not detrimental by its nature. However, it can easily be combined with a sophisticated attack vector and perform considerable damage, such as data exfiltration. These attacks have common characteristics, such as high variance of destination ports, deterministically contiguous destination addresses or ports, and high volume of error packet traffic in response to the probing requests.

The third set of attacks that the spectral classification algorithm distinguishes is User to Root (U2R) attacks. U2R attacks assume that the attacker already has user access to a host. Consequently, the goal of the attacker is to obtain root access. The prevalent characteristic of these attacks is transfer of executable files that can cause buffer overflow. Buffer overflow is caused when a static buffer is not checked and additional data is written to it so that system stack data is overwritten. Careful manipulation of the stack can cause command injection and consequently exfiltrate root access. Another characteristic of these attacks is a set of packets with specific string patterns, which aim to inject specific commands in the system.

A fourth group of attacks named Remote to Local (R2L) attacks, is part of the DARPA dataset. Remote to Local attacks aim at gaining local access to a machine. Although the attacker sends packets over a network to gain access and these packets are captured from network traffic data (tcpdump) they are not meaningful unless combined with audit logs. Just like a regular IDS, our classification algorithm uses only network traffic logs. For this reason our method has the advantage that with modifications it can be used in real time by an IDS or to extract post forensic information using only network data. However, R2L attacks are out of the scope of our spectral clustering algorithm since they need audit data to be used in their classification.

3.2. Data Filtering

As mentioned above, the data used came from the 1999 DARPA datasets weeks 4 and 5 [23] as well as the DDoS data created using the GENI cloud infrastructure [25]. Instead of the audit data as used in the KDD Cup, we used the inside sniffing data exported from Wireshark to a comma separated (csv) file. The data was filtered using the "Detection Scoring Truth" file as provided on the DARPA website. This file includes the time stamps for each attack, the source IP address of the attack, the destination, and data on what type of attack.

Many of the attacks had multiple parts, sometimes with time gaps between parts. To prepare the data for computations, we extracted the traffic corresponding to the parts of the attacks isolating destinations as indicated by the truth file. We put each part in a separate file, to be combined later in the computation process. Computations such as the inter-arrival rate would have been incorrect had we combined the parts at this point.

4. Features

Initially, we considered a set of sixty data features from [27] that ranged from protocol and flag signatures, to packet size, inter-arrival rate, source and destination ports, and packet error rates. Furthermore, we merged individual attack signatures such as command injection, SYN packets, and executable file transfer to create several new features. Because the presence of irrelevant features can drastically degrade the performance of spectral clustering [28], we reduced the number of features to the six features shown in Table 2. We used features which met both of the following criteria:

1) The mean feature value was significantly different across the four categories considered, as quantified by the results of a one-way ANOVA. (For flags, we required that the percentages were significantly different as quantified by a chi-square test.)
2) The distributions of the feature value did not have extensive overlap for at least two of the categories.

TABLE 2. DATA FEATURES AND ONE-WAY ANOVA RESULTS

Data Features	p-Value
Coefficient of variation of inter-arrival rate	0.0097
Coefficient of variation of packet size	0.0000
C or Javascript code flag	0.0000
FTP in protocol and "*.c" in content flag	0.0021
DoS smart feature	0.0000
Third moment of packet size flag	0.0000

Several of the features that were considered, such as packets per second or inter-arrival rate, are sensitive to the magnitude or scale of the attack. Similarly, the standard deviation of inter-arrival times are going to be sensitive to scale. To mitigate this sensitivity, we also looked at the coefficient of variation of these values which is defined to be the standard deviation divided by the mean. These values measure variability while adjusting for the scale of the attack. We also considered the third central moment of several metrics which measures skewness.

One class of features that we found useful were flags with thresholds determined by statistical analyses. For example the code and ftp flags were characteristic of U2R attacks and had a significant p value. Surprisingly, the skewness of the packet size being higher than a specified threshold also proved to be a useful feature for grouping normal traffic. This reflects the fact that attack traffic may have predictable packet size whereas normal traffic packet size typically exhibits a right-skewed distribution.

Many of the features that were evaluated are not endemic to a whole subcategory of network attack. However, these still may provide useful data in identifying a specific attack (e.g. udpstorm, apache, etc) and can be combined. For example, the flags that were used were based on SYN packets per second, ECHO packets per second, HTTP and malformed percent, and source bytes from the same destination were combined using a logical "or" to become what we call the DoS smart feature in Table 2.

The first four features in Table 2 were helpful in differentiating DoS and Probe attacks from U2R attacks and normal traffic. The DoS smart feature was deployed to distinguish DoS attacks and the last feature was useful for distinguishing normal traffic.

There is always a tradeoff between performance and accuracy for classification algorithms. Our goals were twofold; first we were concerned with learning relevant features. Because of the limited sample size the attacks were not split into training and test data so that the feature selection process could be fully optimized. Secondly, our goal was to discover attack signature patterns which was achieved by grouping features. Consequently, we have devised a fairly accurate algorithm with only six significant features. Thus, our algorithm can be easily adopted and used by an IDS. In addition, the spectral clustering algorithm can be used to raise an alert for a potential attack, and to extract attack data in a database for future information sharing that includes information on the specific group of the attack.

5. Clustering

Before discussing the details of our implementation, we first give a brief review of spectral clustering.

5.1. Spectral Clustering Algorithm

Given n data points $x_1, \ldots, x_n$ and a similarity function $f(||x_i - x_j||, \sigma)$, a weight matrix W is defined by:

$$W_{ij} = f(||x_i - x_j||, \sigma)$$

The similarity between two data points depends both on the distance between the points and a scaling parameter σ; for example, the Gaussian similarity function

$$f(||x_i - x_j||, \sigma) = e^{-||x_i - x_j||^2/2\sigma^2}$$

is frequently used. The scaling parameter determines the local structure of the connections between data points. A degree matrix D is then defined by:

$$D_{ii} = \sum_{j=1}^{n} W_{ij}$$

and the Laplacian matrix L by:

$$L = D - W.$$

The eigenvalues and eigenvectors of L are then used to cluster the data; normalizing the Laplacian before computing the spectral decomposition results in more balanced clusters [29]. Note that the number of clusters k is a required input parameter.

We apply the spectral clustering algorithm proposed by [22]. First the Laplacian is normalized as follows:

$$L_{\text{sym}} = D^{-1/2} L D^{-1/2}$$

Let V be the n by k matrix whose columns are the eigenvectors corresponding to the k smallest eigenvalues of L_{sym}. The rows of V are then normalized to obtain a new matrix Y:

$$Y_{ij} = \frac{V_{ij}}{\sum_{j=1}^{n} V_{ij}}$$

Now viewing the rows of Y as a collection of n data points in $\mathbb{R}^k$, the k-means algorithm is applied to cluster the data.

5.2. Implementation

Although spectral clustering is a powerful tool for finding non-convex clusters in big data, results can be sensitive both to the choice of similarity function and the scaling parameter σ. We defined the similarity function as follows:

$$W_{ij} = e^{-\frac{(\alpha_1||x_{i1} - x_{j1}||^2 + \ldots + \alpha_6||x_{i6} - x_{j6}||^2)}{2\sigma^2}}$$

where x_{im} denotes feature m of data point x_i, and the features have been normalized so that $|x_{im}| \leq 1$.

The weights $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_6)$ and the scale σ were optimized by maximizing a performance score. To evaluate

the performance of spectral clustering for a specific combination of parameters, a matrix C was computed were C_{ij} was the number of points in cluster i that belong to the attack category j. The maximal number of points belonging to a single category ($\max_j C_{ij}$) was computed for each cluster, and the performance score was then defined by adding these maximal counts together and dividing by the sample size n:

$$\frac{\sum_{i=1}^{4}(\max_j C_{ij}(\sigma, \boldsymbol{\alpha}))}{n}$$

Note that when the clustering algorithm is performing well, C is close to a diagonal matrix and the score is close to 1.

Because spectral clustering employs k-means after computing the spectral embedding, and k-means is sensitive to the initial placement of the centroids, for each fixed weight combination spectral clustering was in fact applied 50 times and the mean and standard deviation of the performance scores were computed. Letting $C_{ij}(\sigma, \boldsymbol{\alpha}, \omega)$ denote the matrix C obtained in run ω when parameters $\sigma, \boldsymbol{\alpha}$ are used, define:

$$\text{score}(\sigma, \boldsymbol{\alpha}) = \frac{1}{50}\sum_{\omega=1}^{50}\left(\frac{\sum_{i=1}^{4}(\max_j C_{ij}(\sigma, \boldsymbol{\alpha}, \omega))}{n}\right)$$

$$\text{stdev}(\sigma, \boldsymbol{\alpha}) = \frac{1}{49}\sum_{\omega=1}^{50}\left(\frac{\sum_{i=1}^{4}(\max_j C_{ij}(\sigma, \boldsymbol{\alpha}, \omega))}{n} - \text{score}(\sigma, \boldsymbol{\alpha})\right)^2$$

The optimal weights were found by maximizing score$(\sigma, \boldsymbol{\alpha})$ as the weights α_i ranged from 1 to 11 and σ ranged from the 10^{th} to 50^{th} percentile of pairwise distances. Approximate maxima were detected from coarse parameter increments and were then refined using finer parameter increments. If two combinations of parameters had similar mean performance scores, the combination with the smallest standard deviation was preferred. It was found that multiple parameter combinations gave equivalent results indicating the robustness of our features and stability of our algorithm; the parameters $\sigma = 15$ and $\boldsymbol{\alpha} = (2, 7, 6, 4, 7, 11)$ were used for all reported results and figures. For this parameter choice stdev$(\sigma, \boldsymbol{\alpha}) = 0$, indicating complete stability of results. Both the weight optimization and spectral clustering algorithm were implemented in MATLAB R2015a. For comparison purposes k-means was also applied directly to the data, but spectral clustering, which has superior performance on irregularly shaped clusters, was found to be both more accurate and more stable.

6. Results

Spectral clustering results are shown in Table 3. The first column shows the classification accuracy for each category (assuming attacks are assigned to the most prevalent category within each cluster), while the second column indicates what percentage of each cluster consisted of the most prevalent category. The classification accuracy was very high for DoS and Normal but a little lower for Probe and U2R (Probe attacks were often misclassified as U2R attacks and U2R attacks were often misclassified as Probe attacks or Normal traffic), with an overall classification accuracy of 78%. Compared to other machine learning algorithms such as Expert-2 (74% accuracy) and Forensics (87% accuracy) our the classification algorithm accuracy is , however the above algorithms only detect 50% of attacks that are in-spec [12]. Compared to these, our classification algorithm detects all attacks that are in-spec.

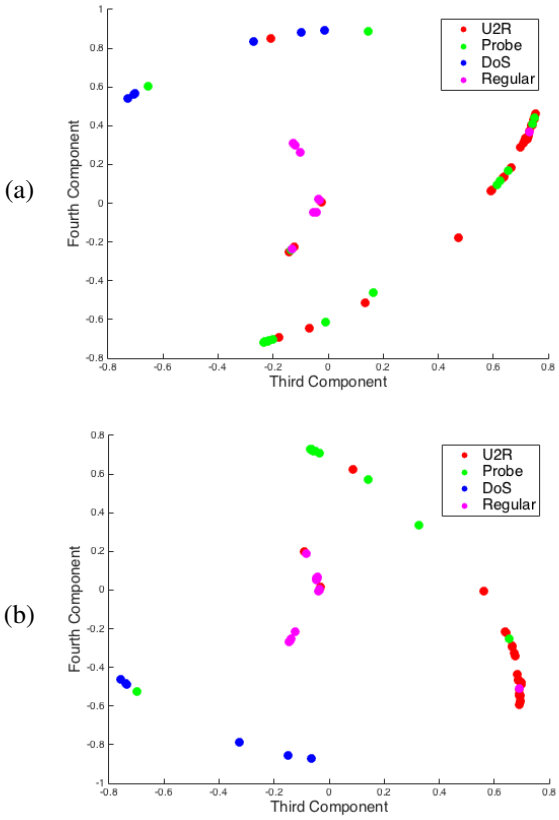

Figure 1. Third and Fourth Components of Spectral Embedding for (a) All Categories and (b) Restricted Categories

TABLE 3. SPECTRAL CLUSTERING RESULTS

Category	Number in Category	Percent of Category in Cluster	Percent of Cluster in Category
DoS	14	100%	78%
Probe	30	67%	80%
U2R	33	64%	78%
Normal	26	96%	76%
Overall	103	78%	78%

Figure 1(a) shows the third and fourth components of the spectral embedding, which clearly separate the four categories; DoS attacks are shown in blue, Probe in green, U2R in red, and Normal in pink. The lower classification accuracy for Probe and U2R is due to the diverse nature of these types of attacks, some of which were not characterized by the selected features or need additional audit data to be

characterized. It is worth noting that if certain subcategories of attacks are removed, the classification accuracy increases dramatically. Removing the ntfsdos, perl, and xterm from the Probe attacks considered and the insidesniffer, ntinfoscan, ntis, mscan, and resetscan from the U2R attacks considered, we obtain the results in Table 4 with an overall classification accuracy of 90%. The third and fourth components of the spectral embedding are shown in Figure 1(b).

Other algorithms such as Expert-2 (90% accuracy) and DMINE (100% accuracy) from [12] only detect 50% and 40% of the attacks that are in-spec respectively. Our clustering algorithm compares favorably, using a wider array of the documented intrusions.

TABLE 4. CLUSTERING RESULTS (RESTRICTED CATEGORIES)

Category	Number in Category	Percent of Category in Cluster	Percent of Cluster in Category
DoS	14	100%	88%
Probe	23	87%	91%
U2R	24	79%	90%
Normal	26	96%	89%
Overall	87	90%	90%

7. Conclusions and Future Work

We have presented an algorithm based on spectral clustering that is robust and efficient in the classification of computer network attacks utilizing a minimal number of features. What differentiates our work is that we have automated the process of grouping attacks to improve countermeasure deployment and information sharing. Furthermore, to our knowledge this is the first time spectral clustering has been used in attack classification. In the future we would like to modify this algorithm for real-time intrusion detection, refine and further automate our feature selection, and consider a wider spectrum of attacks.

References

[1] S. Barnum, "Standardizing cyber threat intelligence information with the structured threat information expression (STIX™)," 2014. [Online]. Available: http://stixproject.github.io/getting-started/whitepaper/

[2] "The trusted automated exchange of indicator information (TAXII™)." [Online]. Available: http://taxiiproject.github.io/getting-started/whitepaper/

[3] S. S. P. Arun Raj Kumar, "Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems," *Computer Communications Journal*, vol. 36, pp. 303–319, 2013.

[4] "Tcpdump and libcap." [Online]. Available: http://www.tcpdump.org/

[5] M. Roesch, "SNORT?lightweight intrusion detection for networks," in *Proceedings of LISA*, 1999, pp. 229–238.

[6] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer Networks Journal*, vol. 31, pp. 2435–2463, 1999.

[7] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1, pp. 18–28, 2009.

[8] W. Z. Theerasak Thapngam, Shui Yu and G. Beliakov, "Discriminating ddos attack traffic from flash crowd through packet arrival patterns," in *Proceedings of The First International Workshop on Security in Computers, Networking and Communications*, pp. 229–238.

[9] M. Sachdeva and K. Kumar, "A traffic cluster entropy based approach to distinguish ddos attacks from flash event using DETER testbeds," *ISRN Communications and Networking*, 2014.

[10] M. V. Mahoney and P. K. Chan, "Learning nonstationary models of normal network traffic for detecting novel attacks," Technical Report, Tech. Rep., 2002.

[11] H. Waguih, "A data mining approach for the detection of denial of service attack," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 2, no. 2, pp. 99–106, 2013.

[12] "Kdd cup 1999 data," 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[13] P. A. R. Kumar and S. Selvakumar, "Distributed denial of service attack detection using an ensemble of neural classifier," *Computer Communications*, vol. 34, no. 11, pp. 1328–1341, 2011.

[14] P. G. Jeya, M. Ravichandran, and C. Ravichandran, "Efficient classifier for r 2 l and u 2 r attacks," *International Journal of Computer Applications*, vol. 45, no. 21, 2012.

[15] S. S. Kaushik and P. Deshmukh, "Detection of attacks in an intrusion detection system," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 2, no. 3, pp. 982–986, 2011.

[16] R. K. C. Richard P. Lippmann, "Improving intrusion detection performance using keyword selection and neural networks," *Computer Networks Journal*, vol. 34, pp. 597–603, 2000.

[17] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping multidimensional data*. Springer, 2006, pp. 25–71.

[18] S. Lloyd, "Least squares quantization in pcm," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.

[19] C. Fraley and A. Raftery, "Mclust: Software for model-based cluster and discriminant analysis," *Department of Statistics, University of Washington: Technical Report*, no. 342, 1998.

[20] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[21] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.

[22] A. Y. Ng, M. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[23] M. L. Lab, "Intrusion detection attacks database," 1999. [Online]. Available: http://www.ll.mit.edu/ideval/docs/attackDB.html

[24] J. McHugh, "Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," *Transactions on Information and System Security*, vol. 3, no. 4, pp. 262–294, 2000.

[25] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "GENI: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, pp. 5–23, 2014.

[26] A. Networks, "Arbor data set," 2015. [Online]. Available: https://www.arbornetworks.com/images/documents/Data20Sheets/DS_TMS_EN.pdf

[27] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 darpa off-line intrusion detection evaluation," Tech. Rep. 4, 2000.

[28] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *The Journal of Machine Learning Research*, vol. 7, pp. 1963–2001, 2006.

[29] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.