Cluster Analysis and Statistical Modeling: A Unified Approach for Packet Inspection

Sheikh Muhammad Farjad* and Asad Arfeen†
National Centre for Cyber Security
Department of Computer & Information Systems Engineering, NED University of Engineering and Technology
Karachi, Pakistan
Email: *smfarjad@outlook.com, †arfeen@neduet.edu.pk

***Abstract*—A secure network layer capable of distinguishing between malicious and genuine traffic flows is the need of every transit service provider, edge network, corporate customer, and a common Internet user. With the emergence of advanced technologies, the demand for security has been drastically increased over the past decade. The analysis of network traffic is essential for various tasks like security, capacity planning, and visibility at various levels. In this paper, a novel architecture is proposed which exploits two powerful techniques for network traffic inspection, that is, cluster analysis and statistical modeling, and unifies them in a single framework. The proposed architecture leverages the clustering technique and statistical modeling for analyzing and inspecting the network traffic. Instead of selecting NetFlow records as the primary format, this research paper presents an approach that employs Packet Capture (PCAP) data format for network analysis. The clustering technique can be used for classifying benign and malicious traffic but there may arise many uncertainties caused by various dynamic factors due to emerging application mixture. Our proposed model uses statistical modeling for supplementing the results obtained from clustering. This unified approach for traffic analysis reduces the chances of the false alert generation that substantially deteriorates the security ecosystem. The proposed architecture inspects different parameters of network traffic to uncover any strong correlation for identifying malicious network traffic flows.**

***Index Terms*—Network traffic analysis, machine learning, cluster analysis, statistical modeling, exploratory data analysis, packet inspection, cybersecurity systems.**

I. Introduction

The world is getting revolutionized by the wave of the fourth industrial revolution. The rapidly emerging fields of the Internet of Things (IoT) and Artificial Intelligence (AI) have significantly affected the world by implementing the technology of automation while making it more feasible as compared to the prior techniques. The industrial sector already relies on robotics for manufacturing and production processes. The domestic sector has also been started being influenced by the wave of automation. The smart home is one of the most beneficial applications of IoT [1]. With the increase in the emergence of innovations, the threat of cyber-attacks has also raised rapidly [2]. The vulnerabilities in the design of newly developed innovations open an entry point which can be utilized to penetrate the infrastructure with malicious incentives. The networking links of devices is the crucial component that is frequently subjected to the threat of cyber-attack.

All devices and systems are connected through a network. A compromised network containing vulnerabilities can result in major cyber-attacks rendering huge losses for the corresponding organization. This is why the practice of securing the network is an essential and most prominent part of cybersecurity endeavors. The network traffic can be analyzed for drawing out insightful results about the behavior and other speculations of the network. Network monitoring is not performed only for detecting anomalies but it is used for other purposes also, including the censoring of contents and evaluation of the network performance metric. Several security tools provide the functionality of network monitoring and management but these tools are often subjected to problems like false alert generation and there also requires qualified personnel for analyzing the illustrations and other parameters acquired from these tools. The analysis of huge piles of network logs requires expertise for drawing out insights.

The domain of data analysis is widely being used by cybersecurity researchers for developing the robust intrusion detection system (IDS) to detect and prevent cyber-attacks [3]. This paper proposes a unified model that leverages notable AI techniques, that is, cluster analysis and statistical modeling, for determining the different parameters and anomalous behavior of network traffic. The proposed model can also be used for inspecting the packets. The emerging domain of AI replaces the conventional Deep Packet Inspection (DPI) approach with an efficient and cost-effective alternative which requires minimal manual interference, and it results in fewer deviations and false alerts.

II. The Unified Model

The cluster analysis cannot exclusively perform efficiently due to consistent deviations and dependent focal points. The centroids obtained from cluster analysis cannot be independently classified whether these belong to the cluster of malicious traffic or benign traffic. This is why statistical modeling is employed in the proposed model to supplement the cluster analysis for the classification of data points. The statistical modeling also explores different network characteristics of the packets.

978-1-7281-6840-1/20/$31.00 © 2020 IEEE

Fig. 1 illustrates the process of packet inspection based on the unified approach presented in this paper. The initial three stages involve data preprocessing which includes data collection, data cleaning, and data annotation. The practice of data preprocessing is the mainstay of any machine learning model and exploratory analysis. After passing through these preprocessing stages, the data is fed into the core model which performs the respective operations including feature selection, clustering and classification, information visualization, and statistical modeling. It is worth mentioning that the flow diagram illustrated in Fig. 1 presents a generalized view of applying the proposed approach over the specific dataset and the additional steps can be eliminated according to the scenario. For example, if the data already exists in the annotated form then the stage of data annotation can be ignored.

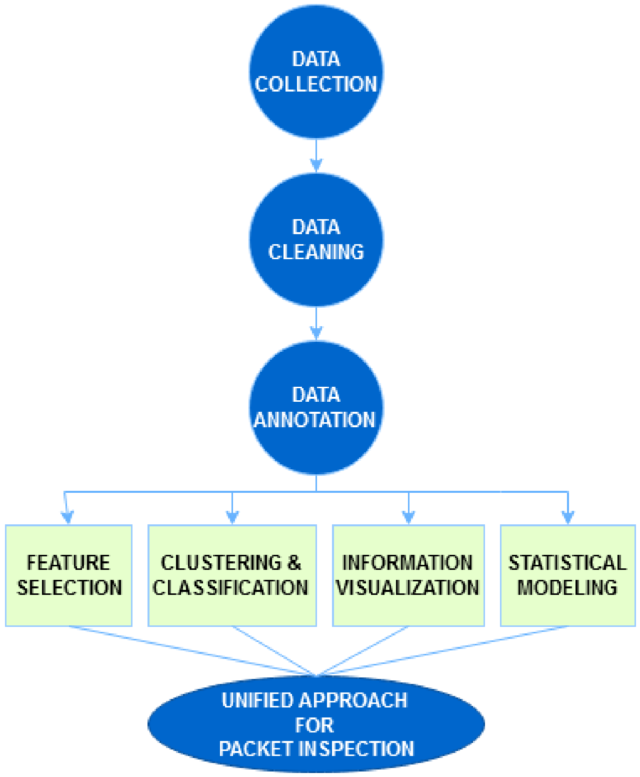

DATA COLLECTION
DATA CLEANING
DATA ANNOTATION
FEATURE SELECTION
CLUSTERING & CLASSIFICATION
INFORMATION VISUALIZATION
STATISTICAL MODELING
UNIFIED APPROACH FOR PACKET INSPECTION

Fig. 1. Process of packet inspection based on unified approach.

A. Data Collection

The first stage of any AI model is data collection. The approach presented in this paper aims to target real-time network traffic. Generally, network traffic is captured, stored, and processed in the following two ways:

- NetFlow Records: The majority of cybersecurity systems are developed for processing NetFlow records. NetFlow is still being used by many researchers for carrying out their research in the domain of network security. NetFlow is a protocol that is used for collecting and monitoring network traffic. NetFlow records are generated by NetFlow-enabled routers. NetFlow collectors collect and process the records for further operations.
- PCAP Format: Packet Capture (PCAP) format is a de facto standard for capturing network packets. PCAP is a binary format that has support for nanosecond-precision timestamps and this is why it contains broad spectra of information. It consists of a global header that is followed by individual packet headers and data.

In our model, we did not opt for the conventional approach and selected PCAP format as our fundamental data format because PCAP format is independent of vendors and it has community support which can help in developing applications depending upon it. The network packets are collected by using Wireshark[1] utility and the captured packets can be stored in PCAP files. For generating the malicious network traffic, we simulated an attack on our machine using Scapy[2]. All network activities were recorded by Wireshark and then the recorded activities were exported in PCAP format. Fig. 2 illustrates how malicious packets were merged with benign packets for creating a PCAP file to test our model.

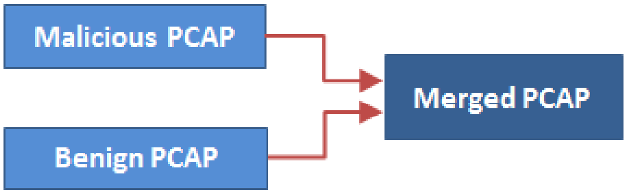
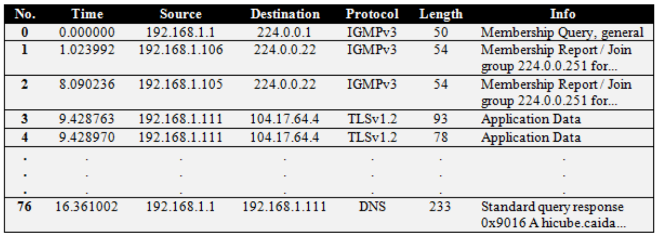


Fig. 2. Merging benign and malicious packets for creating testing dataset.

B. PCAP Analysis

The utility of Scapy can be used in the development of scripts for manipulating and sniffing network packets. The piles of packets collected by Wireshark can be exported as Comma-Separated Values (CSV) format also, but it causes loss and ambiguity in different fields, a CSV version of the PCAP file is illustrated in Fig. 3.



No.	Time	Source	Destination	Protocol	Length	Info
0	0.000000	192.168.1.1	224.0.0.1	IGMPv3	50	Membership Query, general
1	1.023992	192.168.1.106	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for...
2	8.090236	192.168.1.105	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for...
3	9.428763	192.168.1.111	104.17.64.4	TLSv1.2	93	Application Data
4	9.428970	192.168.1.111	104.17.64.4	TLSv1.2	78	Application Data
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
76	16.361002	192.168.1.1	192.168.1.111	DNS	233	Standard query response 0x9016 A hicube.caida...

Fig. 3. The exported packets in CSV format.

It can be observed that there exists only a limited number of fields in CSV format which are not sufficient to determine the malicious activity in the network. This is why the original PCAP format is always preferred over CSV format. The tool of Scapy can be used for reading and writing PCAP files along with minor details. The merged PCAP file, consisting of malicious and benign packets, is processed by using Scapy

[1] https://www.wireshark.org/
[2] https://scapy.net/

and other libraries, and the data extracted from the PCAP file are stored in a data frame.

C. Data Cleaning

Every model is designed to receive input data with several constraints. These constraints improve the quality of data by enforcing standardization, which also increases the readability of data for the model. Once the data extracted from the PCAP file are stored in a data frame, various techniques can be applied to the data frame for cleansing the data. In data cleansing, following major steps are taken:

- 'None' in the cells of the data frame is replaced with '0'.
- The time is converted from exponential format to floating-point format.
- 'NaN' in the cells of the data frame is also replaced with '0'.

While implementing the approach presented by this paper, the merging of two files for creating a single file resulted in repeated indexes. This is why the old indexes were dropped and new indexes were assigned to the packets. The specific features also undergo z-score normalization before subjecting to the process of K-means clustering. The z-score normalization can be defined as shown in (1), where x represents data point, μ represents mean, and σ refers to standard deviation.

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

III. Cluster Analysis

Cluster analysis is one of the prominent techniques of unsupervised learning. It is a significant component of the data mining paradigm that is used for providing valuable insight into the data under consideration. Cluster analysis leverages different clustering algorithms for grouping data instances into distinct clusters based on the information obtained from the data.

A. K-Means Clustering

K-means clustering is a prototype-based clustering algorithm. The primitive goal of K-means clustering is to determine K non-overlapping clusters in the given dataset. Each cluster is represented by its centroid which purports to be the mean of all points lying in the cluster. The optimal centroid of a cluster is determined by considering the measure that there should be minimum squared Euclidean distance between the centroid and its neighboring data points existing in the same cluster. Therefore, the objective function, sum of the squared errors (SSE), for K-means clustering algorithm can be defined as follows:

$$SSE = \min_{\{\mu_k\}, 1 \leq k \leq K} \sum_{k=1}^{K} \sum_{x \in C_k} ||x - \mu_k||^2 \tag{2}$$

where x represents the data instance of the dataset $X = \{x_1, x_2, x_3, \ldots, x_N\}$ which belongs to the cluster C_k, K is the number of clusters assigned by the user, and μ_k is the centroid of cluster C_k which can be formulated as follows:

$$\mu_k = \sum_{x \in C_k} \frac{x}{n_k} \tag{3}$$

where x and n_k represent the data point and the total number of data points of the cluster C_k, respectively.

The clustering process of the K-means algorithm is initiated by selecting K centroids. Each data instance is assigned to the closest centroid C_k while predicating on the measure shown in (2). The collections of data instances assigned to the centroids form clusters and the centroids are updated according to the data instances of the respective clusters. This process keeps repeating until the convergence is achieved and centroids become stable.

B. Implementation of K-means Clustering

There are several reasons for preferring the K-means algorithm over other existing clustering algorithms. K-means is a robust and highly efficient algorithm that is relatively simple to implement when compared with other clustering alternatives. In the K-means algorithm, the convergence is guaranteed, and it can often result in clusters tighter than those which are generated by hierarchical clustering. K-means clustering has a wide range of applications. It is mostly used in document clustering, image segmentation, and image compression. K-means algorithm can be used either to get meaningful intuition from the data or it can exploit the clustering-and-predicting approach. The clustering-and-predicting approach develops clusters and then predicts the behavior of data points lying in the respective clusters [4], [5].

The clustering-and-prediction approach is the primitive factor that makes K-means clustering an appropriate choice for clustering network data. According to this approach, the packets are first clustered and then the prediction is made whether the particular packets are malicious or not. Fig. 4 describes the implementation of K-means clustering in our proposed model.

C. Clustering Evaluation

There are several measures for assessing the results obtained from cluster analysis. In this paper, silhouette analysis was used for determining the optimal number of clusters for the data under consideration. The silhouette score S_i of an instance i can be calculated as follows:

$$S_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \tag{4}$$

where a_i is the average dissimilarity of the ith-instance to all other instances in the same cluster, and b_i is the minimum of average dissimilarity of the ith-instance to all other instances in other clusters. The silhouette score has a range of $[-1, 1]$.

The measure of silhouette score was calculated for different numbers of clusters and the best value of silhouette score was observed when the number of clusters was equivalent to 2 as

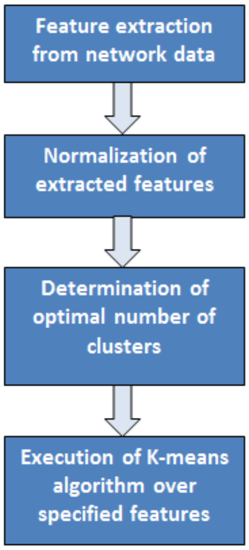

Feature extraction from network data
Normalization of extracted features
Determination of optimal number of clusters
Execution of K-means algorithm over specified features

Fig. 4. An illustration of K-means clustering process for network data.

shown in Fig. 5. This implies that the implementation of the K-means clustering algorithm over the data under consideration evaluates more accurate results when the number of clusters k is kept equivalent to 2.

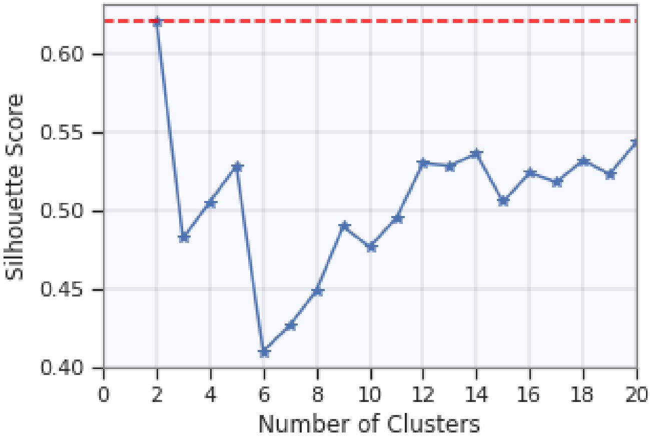


Fig. 5. Silhouette analysis over network packets' features.

IV. Statistical Modeling

Statistical modeling is already playing a very significant role in communication networks. It is used for estimating the cost and behavioral characteristics of the network. Statistical modeling can be employed for estimating and determining network parameters without confronting any problem which may often arise due to the lack of knowledge about network topology. It has always been the mainstay of data analytics paradigm. The endeavor of statistical modeling can be leveraged for determining the correlation amongst different features of the data, and interactive visualization can be created for elaborating significant characteristics of the parameters under consideration. The statistical analysis purports to be an efficient approach for detecting anomalies and potential cyber threats [6], [7]. The domain of big data analytics further enhances the statistical learning for anomaly detection [8].

Moreover, the results obtained from the cluster analysis can be further augmented by using statistical modeling. In order to draw insights from the clusters, information-theoretic measures can be utilized and enhanced results can be used for elucidating network statistics. In our proposed approach, statistical modeling is utilized for exploring multi-faceted data through interactive and descriptive visualization.

A. *Exploratory Data Analysis*

The model, we proposed in this paper, uses Exploratory Data Analysis (EDA). EDA is used for summarizing different characteristics of data in the form of graphs and figures. The technique of EDA further elucidates data through various perspectives, this helps in visualizing data for extracting different information from it [9].

Our proposed unified model exploits EDA for detecting malicious actors in the network by exploring and analyzing all respective features of network traffic obtained from the PCAP file. It is used for illustrating different features in the form of graphs that help in determining the malicious host or IP address. The information gained from EDA is used for determining the cluster that represents malicious packets and this process is executed by finding out the cluster assigned to malicious IP address procured from EDA, the cluster that is assigned to malicious IP is 'blacklisted' and every packet belonging to the cluster is considered to have a higher probability of being malicious.

B. *Implementation of EDA*

The idea of supplementing cluster analysis with EDA purports to be an efficient approach because it provides the exact results with high accuracy. EDA is conventionally used for drawing insights into the data before applying any machine learning model. The endeavor of EDA can also be utilized for exploring different aspects of the data under consideration. In the approach presented by this paper, EDA is employed for exploring network statistics which can significantly supplement the results obtained from cluster analysis.

Tables I and II are describing complete network statistics regarding IP addresses and ports, respectively. This information can be used for assessing the threat vectors in network data. It must be brought under consideration that most frequent addresses are not necessarily prone to suspicion; the other relevant factors, including the size of payloads, are also used in supplementation for determining the suspicious ports and addresses.

Fig. 6 and Fig. 7 illustrate the source IP addresses and destination IP addresses, respectively, which are involved in the transmission of massive payloads. The large size of payloads can be an alarm for Distributed Denial of Service (DDoS) attacks and this is why these IP addresses are of paramount importance.

TABLE I
IP ADDRESSES INVOLVED IN COMMUNICATION

Source IP Addresses	Destination IP Addresses
172.28.0.3	172.28.0.2
172.28.0.2	172.28.0.3
172.28.0.1	172.28.0.1
74.125.204.95	74.125.204.95
169.254.169.254	169.254.169.254
64.233.189.95	64.233.189.95
64.233.187.95	64.233.187.95
74.125.203.95	74.125.203.95
108.177.97.95	108.177.97.95
10.1.10.53	84.54.22.33
84.54.22.33	10.1.10.53
75.75.75.75	75.75.75.75

TABLE II
MOST FREQUENTLY ADDRESSED PORTS

Most Frequent Source Ports			Most Frequent Destination Ports		
8080	53762	53938	53	57016	57190
9000	53770	53946	443	57030	57196
38922	53780	53956	6000	57038	57212
39044	53788	53962	34002	57044	
39108	53802	53968	34092	57064	
45808	53808	53980	56884	57072	
46686	53816	53992	56900	57086	
46696	53834	55278	56912	57098	
46752	53844	55282	56920	57114	
53406	53856	57594	56930	57120	
53680	53870	60251	56948	57134	
53690	53884	60668	56956	57140	
53702	53892	60672	56962	57152	
53720	53904	60778	56974	57160	
53726	53912	60782	56992	57168	
53734	53922		56998	57174	
53746	53932		57010	57184	

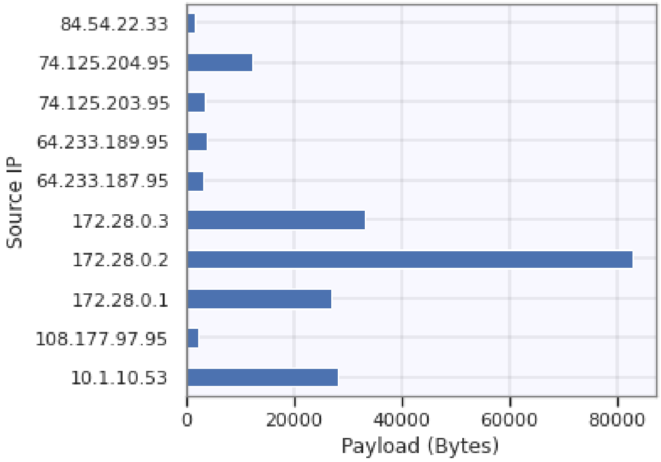


Fig. 6. Source IP addresses and their respective payload size.

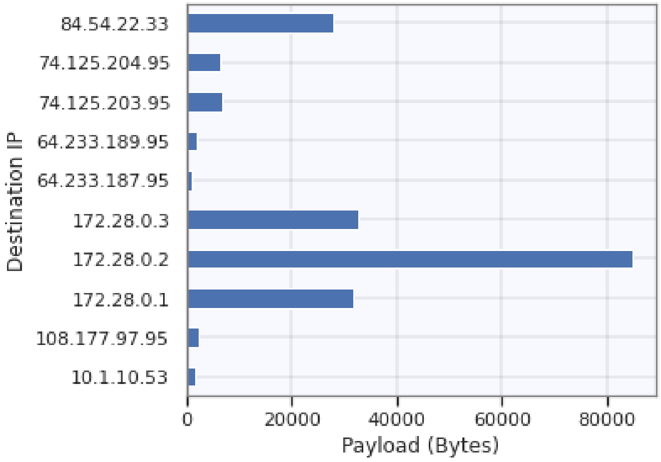


Fig. 7. Destination IP addresses and their respective payload size.

The ports refer to logical constructs that are assigned to dedicated services or processes. The communication protocols use ports for identifying and binding application layer services. The applications also utilize specifically reserved ports for communicating with the end-points. Since the port numbers represent the reserved services and different protocols depend upon ports, the analysis of ports can be used for identifying any malicious activity in the network. Fig. 8 and Fig. 9 highlight the source and destination ports, respectively, which are involved in the communication of massive payloads. Similar to IP addresses, the ports can be used as entry points for injecting malicious code and these illustrations can be used for blocking out the suspicious ports.

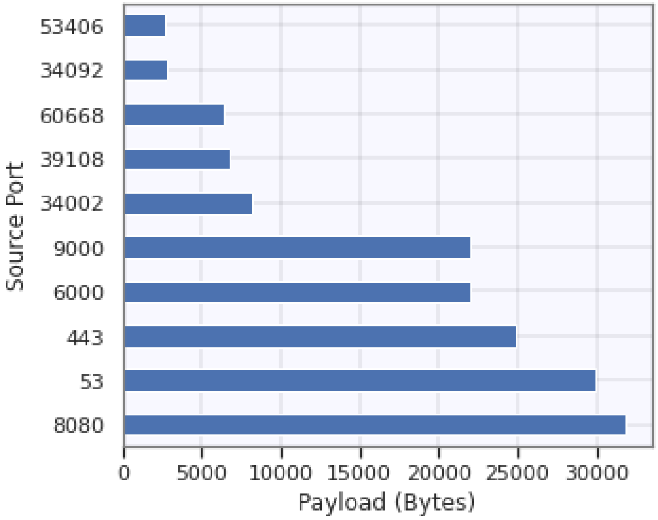


Fig. 8. Source ports and their respective payload size.

V. RESULTS

The unified approach presented in this paper was subjected to specially crafted packets in the form of the PCAP file.

TABLE III
DATA FRAME OF PACKETS AFTER PASSING THROUGH CLUSTER ANALYSIS

id	src	dst	sport	dport	window	payload	chksum	len	ttl	time	cluster
14029	172.28.0.2	172.28.0.3	9000	34002	501	439	66589	491	64	1.587672e+09	0
37391	172.28.0.2	172.28.0.3	9000	34092	501	788	43227	840	64	1.587671e+09	0
1	10.1.10.53	84.54.22.33	53	53	0	933	84212	961	64	1.532199e+09	1
52801	172.28.0.3	172.28.0.2	34092	9000	501	0	27817	52	64	1.587671e+09	0
10094	84.54.22.33	10.1.10.53	53	53	0	56	127751	84	122	1.532199e+09	1

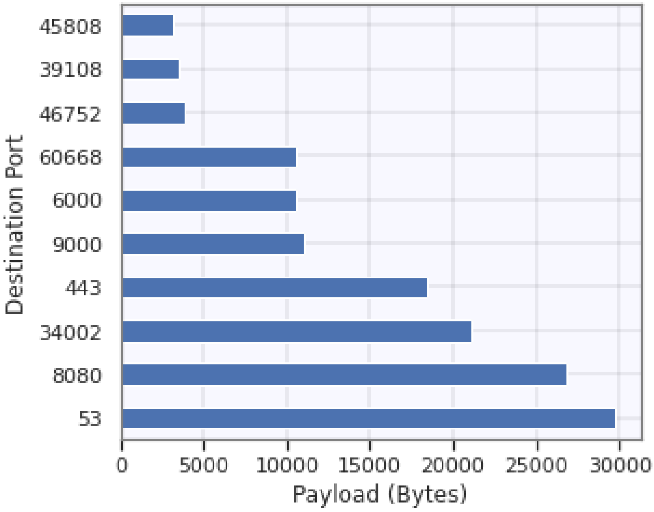


Fig. 9. Destination ports and their respective payload size.

The packet analysis was based on clustering and statistical modeling. The outcome of cluster analysis was found highly consistent with the expected result, for instance, the cluster analysis of packets resulted in two optimal clusters, and the packets were also expected to be categorized into two clusters, that is, benign and malicious. A tag specifying the cluster was attached to each packet. The packets after going through cluster analysis appear as shown in Table III.

Besides cluster analysis, the technique of EDA also resulted in validated network statistics that were used for supplementing the results of cluster analysis. An analysis of Fig. 8 and Fig. 9 infers that the following IP addresses have higher probabilities of being malicious: 172.28.0.1, 172.28.0.2, and 172.28.0.3. The IP address 172.28.0.2 has the highest probability of exhibiting malicious behavior.

VI. RELATED WORK

In the domain of cybersecurity, the researchers have been working on anomaly detection for decades. There are primarily two standards for analyzing network data: packet inspection and NetFlow records' analysis. The packets provide a more detailed spectrum of information when compared to that of NetFlow records. Many researchers have worked on analyzing NetFlow records for determining the anomalies in network data. Monitoring NetFlow records is relatively easier but it does not provide a detailed spectrum of network parameters; while, packets provide relatively more network parameters that increase the accuracy of anomaly detection, but packets are difficult to analyze.

TABLE IV
SUMMARY OF MOST FREQUENT ADDRESSES

Top Destination Addresses	Most Frequent Source Address	Most Frequent Destination Address
172.28.0.3 172.28.0.1 74.125.204.95 169.254.169.254 64.233.189.95 64.233.187.95 74.125.203.95 108.177.97.95	Count: 1281 IP Address: 172.28.0.2 Frequency: 571	Count: 1281 IP Address: 172.28.0.2 Frequency: 648

In the literature, various machine learning based traffic analysis techniques have been presented. Adi et al. [10] used four machine learning models (Naıve Bayes, Decision Tree, JRip, and Support Vector Machines) for analyzing network traffic. The research conducted by Farnaaz et al. [11] used Random Forest for network analysis and reported the result with low false alarms. Kemp et al. [12] used NetFlow records for the detection of slow read attacks and compared the results with prior researches. Li et al. [13] proposed a passive method that uses traffic association and machine learning for online Ethereum node detection in NetFlow records. Liu et al. [14] performed the technique of CNN on NetFlow records for predicting the possible network attacks. Hou et al. [15] also used Random Forest technique on NetFlow data for DDoS detection and they reported to highly reduce the false alarms.

Beazley et al. [9] used the technique of Exploratory Data Analysis (EDA) on 'Unified Host and Network Dataset'[3] for discriminating normal and abnormal network behavior. Yang et al. [16] used machine learning and deep packet inspection for identifying different application traffic. The absence of supplementary technique for enhancing the accuracy and dependency of proposed models upon NetFlow records are the main issues that are common in all the aforementioned research works, despite the fact that PCAP format provides more descriptive

[3] https://csr.lanl.gov/data/2017.html

data. This gap was attempted to be filled by our proposed approach.

VII. Conclusion and Future Scope

In this paper, a unified approach for packet inspection has been proposed which leverages the domains of cluster analysis and statistical modeling for detecting and determining malicious activities and network characteristics, respectively. The proposed model initiates processing by taking the PCAP file as input data. The PCAP file undergoes a thorough analysis and the respective data contained by PCAP file are extracted and stored in a well-organized and readable format. After passing through the initial steps of data preprocessing, the data obtained from the PCAP file are subjected to cluster analysis and statistical modeling for generating insights into the data. The proposed model predicates on silhouette analysis for validating the results obtained from cluster analysis. Statistical modeling further augments the validity of overall results generated by the unified approach proposed in the paper.

In the future, we aim to overcome the big data problems which are often rendered by K-means clustering. For instance, if the number of clusters increases, the K-means clustering starts suffering from the empty clustering problem, and the number of iterations also increases for attaining the convergence point, which drastically degrade the performance of K-means algorithm. This malfunctioning behavior of K-means clustering for handling a large amount of data makes it infeasible for solving big data problems. These shortcomings are to be circumvented in the future for enhancing the performance and reliability of the proposed model.

References

[1] S. Dey, A. Roy, and S. Das, ”Home Automation Using Internet of Thing,” in 7th IEEE Annual Ubiquitous Computing Electronics & Mobile Communication Conference, 2016.

[2] J. Ho, ”Efficient and Robust Detection of Code-Reuse Attacks Through Probabilistic Packet Inspection in Industrial IoT Devices,” IEEE Access, vol. 6, pp. 54343-54354, 2018.

[3] ”Traffic Analysis for Network Security: Approaches for Going Beyond Network Flow Data,” 2016. Available: https://insights.sei.cmu.edu/sei_blog/2016/09/traffic-analysis-for-network-security-two-approaches-for-going-beyond-network-flow-data.html

[4] X. Wang, R. Chen, F. Yan, Z. Zeng, and C. Hong, ”Fast Adaptive K-Means Subspace Clustering for High-Dimensional Data,” IEEE Access, vol. 7, pp. 42639-42651, 2019.

[5] S. Wang et al., ”K-Means Clustering With Incomplete Data,” IEEE Access, vol. 7, pp. 69162-69171, 2019.

[6] N. Moustafa, K. R. Choo, I. Radwan, and S. Camtepe, ”Outlier Dirichlet Mixture Mechanism: Adversarial Statistical Learning for Anomaly Detection in the Fog,” IEEE Transactions on Information Forensics and Security, vol. 14, no. 8, pp. 1975-1987, Aug. 2019.

[7] F. Li, A. Shinde, Y. Shi, J. Ye, X. Li, and W. Song, ”System Statistics Learning-Based IoT Security: Feasibility and Suitability,” IEEE Internet of Things Journal, vol. 6, no. 4, pp. 6396–6403, Aug. 2019.

[8] M. Tang, M. Alazab, and Y. Luo, ”Big Data for Cybersecurity: Vulnerability Disclosure Trends and Dependencies,” IEEE Transactions on Big Data, vol. 5, no. 3, pp. 317–329, 1 Sept. 2019.

[9] C. Beazley et al., ”Exploratory Data Analysis of a Unified Host and Network Dataset,” in 2019 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2019, pp. 1–5.

[10] E. Adi, Z. Baig, and P. Hingston, ”Stealthy denial of service (dos) attack modelling and detection for http/2 services,” Journal of Network andComputer Applications, vol. 91, pp. 1–13, 2017.

[11] N. Farnaaz and M. Jabbar, ”Random forest modeling for network intrusion detection system,” Procedia Computer Science, vol. 89, pp. 213–217, 2016.

[12] C. Kemp, C. Calvert, and T. Khoshgoftaar, ”Utilizing Netflow Data to Detect Slow Read Attacks,” in 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, 2018, pp. 108–116.

[13] Z. Li, J. Hou, H. Wang, C. Wang, C. Kang, and P. Fu, ”Ethereum Behavior Analysis with NetFlow Data,” in 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, 2019, pp. 1–6.

[14] X. Liu, Z. Tang, and B. Yang, ”Predicting Network Attacks with CNN by Constructing Images from NetFlow Data,” in 2019 IEEE 5th Internatl Conference on Big Data Security on Cloud (Big Data Security), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), Washington, DC, USA, 2019, pp. 61–66.

[15] J. Hou, P. Fu, Z. Cao, and A. Xu, ”Machine Learning Based DDos Detection Through NetFlow Analysis,” in MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, 2018, pp. 1–6.

[16] B. Yang and D. Liu, ”Research on Network Traffic Identification based on Machine Learning and Deep Packet Inspection,” in 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2019, pp. 1887–1891.