Original Article

http://napas.org.ng

Computer Network Traffic Analysis Framework for Anomaly Detection using Unsupervised Machine Learning

Ge, T.[1], Agaji, I.[2], Blamah, N.V,[3] and Ogala, E.[2]

[1.]Department of Mathematics and Computer Science, Benue State University, Makurdi
[2.]Department of Computer Science, Joseph Sarwuan Tarka University, Makurdi
[3.]Department of Computer Science, University of Jos, Jos

Abstract

Numerous novel traffic classification methods have been conceived and put into practice. The ability to recognize and categorize network traffic from diverse domains holds substantial importance in network management but there is no existing framework that addresses the demand using unsupervised learning techniques. This study developed a K-means based framework that possessed the capacity to both identify and classify network traffic data originating from various domains using unsupervised machine learning. A mathematical model for the k-means-based network analysis framework was designed and implemented in Python to evaluate key performance metrics of V-measure, Mutual information, Normalized mutual information, Rand index and Adjusted rand index using diverse datasets representative of different domains. In order to detect patterns suggestive of typical network behavior and deviations indicating possible security breaches, the framework was trained using UNWSW –NB15 dataset and tested using Diverse Network Traffic Dataset. Additionally, the framework adopted dynamic thresholding and window sliding techniques that allowed it to adjust to changing threat environments and maintained anomaly detection efficacy over time. For each window, the sum of malicious flows was calculated, and if this sum exceeded the defined threshold, the window was classified as malicious, otherwise it was classified as normal. The threshold and window size were the key factors that impact the accuracy of this classification (66.22% anomalous, 33.78 benign for UNWSW-NB15 dataset and 54.44 anomalous, 45.56 benign for Diverse Data set). It also places a strong emphasis on scalability and interpretability, allowing for easy integration into current network infrastructures in a variety of organizational contexts. The results from this study indicated that (for both the training and test datasets) V-measure equaled Normalized Mutual Information (0.7356 training, 0.065 test datasets respectively) and correctly aligned each other as they both reached their lowest and highest values at the same number of clusters. The rand index, adjusted rand index and mutual information reached their maximum values at different number

OPEN ACCESS
Correspondence:
Ge, T
Emails:
tivlumunge@yahoo.com

Specialty Section;*This article was submitted to Sciences a section of NAPAS.*

Submitted: ***10th October, 2024***
Accepted: 15th November. 2024

Citation: Ge,T., Agaji, I., Blamah, N.V and Ogala, E. (2025). Computer Network Traffic Analysis Framework for Anomaly Detection using Unsupervised Machine Learning

Effective Date ***7(2)*** *100 -* ***113***

Publisher:*cPrint,Nig.Ltd*
Email:*cprintpublisher@gmail.com*

of clusters but decreased as the clusters increased beyond the optimal point.

Keywords: Cybersecurity, detection, machine learning, network anomaly, network security and threat.

Introduction

Network traffic refers to the volume of data packets in motion within a network at a specific moment. These packets originate from applications or services running on a source computer system and are destined for applications or services on a target computer system (Kim et al., 2023). Network traffic analysis (NTA) is a vital component to understand the requirements and capabilities of a network. It is an essential way to monitor network availability and activity to identify anomalies, maximize performance and keep an eye out for attacks. Analysts are consistently faced with incomplete and ever-increasing user demands and uncertainty about evolution of network systems (Barut et al., 2020).

Networks are of different types and can be categorized based on several factors. These factors involve parameters like latency, packet loss and throughput. In order to design high performance networks or guarantee performance of any type of networks, detailed analysis of the factors is a crucial step. The primary step is the study of the traffic on the network so as to have a clear understanding of the traffic in the network. As a consequence, the type of traffic model used to understand the flow of traffic in the network, and how closely the model depicts the real-time characteristics of the network, become vital parameters (Zeinali and Barenji, 2023).

While other network security tools such as firewalls and intrusion detection system (IDS)/ Intrusion Prevention Systems (IPS) focus on monitoring virtual traffic that crosses the perimeter of a network environment, network traffic analysis solutions are focused on all communications whether those are traditional TCP/IP style packets, virtual network traffic crossing a virtual switch (vSwitch) traffic from within cloud workloads, and API calls to SaaS applications or server-less computing instance. These solutions also focus on operational technology and Internet of Things (IoT) networks and are otherwise completely invisible to the security team (Kizza, 2024). Given the rapid expansion of network users and the introduction of networking services, the significance of network traffic analysis has surged. However, the existing methods employed for this purpose exhibit limitations. As noted by Chandrasekaran (2009) and Shikhaliyev (2017), numerous traffic models have been proposed recently, yet they lack a singular model that can comprehensively capture the diverse traffic characteristics across all network types and scenarios. Even with the introduction of various computational techniques for effective network traffic analysis, a notable challenge remains in detecting network intrusions. This has been emphasized by Ji et al. (2020), who assert that the dynamic alterations in network traffic patterns pose a persistent research challenge in the realm of network security. This study develops a framework that differentiates between normal and abnormal data, while also enabling the analysis of network traffic data from dispersed domains through the utilization of unsupervised learning techniques.

The rest of the paper is organized as follows: The next sections describe related methods used in network traffic analysis, followed by the methodology deployed in which the framework architecture is described in details with a flowchart. The resulting experiments are then presented with their results and discussion of the results together. Finally, conclusion is drawn with recommendation.

Related Works

Different studies have attempted the use of learning methods to analyse network traffic data. Foremski (2013), presented the Payload Based approach for analysing traffic data (deep pocket inspection (DPI) technique). In terms of network traffic classification, this technique produced accurate findings. The contents of packets were inspected to look for network application signatures in the data. This is the first alternative to using ports. This method was designed specifically for peer-to-peer (P2P) applications. Conti et al. (2018), carried out an in-depth investigation into network traffic analysis. The work was divided into three categories: (1) the analysis' goal, (2) the network point where traffic is observed, and (3) the mobile platforms chosen. The work looked at a variety of algorithms, including Naive Bayes, C4.5 decision trees, Random forests, and k-means, etc. A comparative analysis of methods, validation approaches, and final outcomes for mobile devices

were in the work. D'Alconzo et al. (2019), proposed a Network Traffic Monitoring and Analysis (NTMA) system to collect, store, and process massive sets of historical data for post-mortem analysis in the face of challenges faced by big data approaches such as Volume, Velocity, Variety, and Veracity. According to the study, NTMA is a critical component for network management, particularly for ensuring the proper operation of large-scale networks like the Internet. To determine to what extent the potential of big data is being explored in NTMA, D'Alconzo et al. used past research on NTMA that leverage big data methodologies. Singhal, et al. (2013) developed a methodology to classify network traffic based on unsupervised machine learning principles. A 'learner' and a 'classifier' were the two components of their method. The goal of the 'learner' was to deduce a relationship between flows and traffic class from a set of data. The learnt mapping is then utilized to create a classifier. The work developed and evaluated a technique that enabled the building of a traffic classifier using flow statistics from both labelled and unlabelled flows. This methodology is arguably a semi-supervised method since in as much as the labelled set was small as compared to the unlabelled set.

Iorliam (2016) applied the Benford's law and Zipf's law on network traffic data for intrusion detection purposes and observed that naturally generated network traffic data should follow these Power laws whereas malicious network traffic data should deviate from these Power laws. The study utilized the Benford's law and Zipf's law and analysed the flow size difference of network traffic data and detected malicious traffic over the Internet with AUC values within the range of 0.6858 and 1.0 Wang et al. (2017) proposed Seed Expanding Algorithm for detecting Network Traffic attacks before they damage the system. The algorithm employed a Two-Seed-Expanding network traffic clustering scheme that clustered traffic attack into different attack phases. The algorithm preprocessed network traffic including constructing the network flow, changing continuous valued attributes into nominal attributes by adopting the discretization method, and further turning into binary features. Seed-Expanding then computes a weight for each flow and interactively selected seeds to expand based on the features until all flows are divided into clusters. The results of their experiment showed that preprocessing greatly improved the clustering performance and the Two-Seed-Expanding method was better than K-means and other kinds of Seed-Expanding in attack flow clustering. Zola et al. (2022), presented a three phased novel methodology which converted network traffic classification into node behavior classification. The first phase focused on creation of temporal dissection and Traffic Dissection Graph (TDG) as well as implementing graphs enrichment process. The second phase utilized two techniques: a combination of Random Under-Sampling (RUS) and Random Over-Sampling (ROS) called R-hybrid and a combination of Random Under-Sampling, Synthetic Minority Over-Sampling Technique (SMOTE) and Random Over-Sampling called SM-hybrid. After applying RUS, R-hybrid replicated not only the most relevant subgraphs, but also their node behaviours using a ROS strategy. After applying RUS, SM-hybrid replicated the most relevant subgraphs. The third phase, node behaviour classification was performed by training Deep Learning models. This methodology may not work well in analyzing data in critical infrastructure where timing is fundamental in order to apply counter measures and reduce the impact of cycler attack very quickly.

Sanz et al; (2020), proposed a light weight malware detection system by means of network behavior analysis that relied on light weight machine learning techniques to monitor network behavior of suspicious applications. A realistic and up-to-date network traffic dataset made up of 359 goodware and malware applications was constructed to evaluate the system.

Ji et al. (2020) conducted a study on network security, focusing on innovative methods – data-filtration and transformation, pixel-based visual representation, graph representation and coordinated muti-view (CMV) and user interactions - to detect intrusive network activities. The study highlighted the challenges of identifying and understanding unique characteristics of suspicious activities, and the role of visualization systems in assisting data understanding. The work identified four key approaches for effective network traffic visualization systems: data filtration and transformation, pixel-based visualization, graph representation, and coordinated multi-views. The study also evaluated prototype visualizations, assessing implementation complexity, data preprocessing requirements, and network pattern discernibility. The study's focus on analyzing

abnormal network activities through raw traffic data, highlights the complexities of identifying anomalous or malicious activities. Results from the work showed that scatter plot and bar and line graphs are the most commonly used data representation techniques. For identifiability of abnormal events and activities, heatmap and pixel-based visualization are commonly utilized because all data elements are represented as colorful visual representation so that patterns can easily be discovered.

Wu *et al.* (2020) applied AE and DNN based hybrid deep learning method for malicious code detection. Specifically, AE was adopted to reduce dimensions of original data and focused on the main and important features. Afterward, the DBN-based learning model was used to do the detection of malicious code, which consisted of multilayer RBM and a layer of BPNN. Defining each layer of RBM as unsupervised trained and BP as supervised trained, the optimal hybrid model was finally obtained by fine-tuning the whole network. Experiments showed that detection accuracy of the hybrid network was higher than other previous DBN-based networks. Firstly, it was challenging to modify deep learning methods as real-time classifiers for attack detection. In most of the previous works, they only reduced feature dimension for less computation cost during phase of feature extraction. Secondly, most of the deep learning techniques were appropriate for analysis of image and pattern recognition.

Aidahoul *et al.* (2021) proposed a model fusion that combined binary normal/attack Deep Neural Networks (DNN) to detect attacks and multi-attacks DNN to categorize attacks. The binary model included feature pre-processing and DNN, while the multi-class model included feature pre-processing and DNN. The proposed model fusion outperformed the baseline model in terms of average macro precision, recall, F1 score, and Fβ score. It significantly reduced the false alarm rate by 5.3%. However, the authors failed to employ other deep learning models like 1D convolutional neural network (CNN) for spatial features and long short-term memory (LSTM) for temporal features. They suggested unsupervi sed learning of LSTM autoencoder as a better solution for large-scale datasets.

Mutmbak, *et al.* (2022) proposed a model for network intrusion (anomaly) detection based on machine learning algorithms. The proposed model consisted of six phases. The phases were dataset analysis, pre-processing, feature selection, parameter tuning, training and testing. The work used five machine learning algorithms for the classification of network anomaly detection, which were K-neighbors, logistic regression, SVM, decision tree and random forest. Performances of the Machine Learning algorithms were observed on the basis of their accuracy, recall, precision and F1 score. CICIDS2017 dataset was used for training and testing, which consisted of seven different types of attacks. Results indicated that the Random Forest algorithm performed better as it achieved the highest accuracy and precision rates of 98.63% and 99.80, respectively for the classification of anomaly detection, followed by Decision Tree algorithm, while, the K-Neighbors algorithm had the least accuracy and precision rates. In order to identify possible problems such as low throughput, in large scientific organizations' network where petabytes of data are transferred regularly, *Syal et al.* (2019) developed a supervised data analytics system for mining network traffic data. The proposed model utilized the approach of assigning binary classification labels to network transfers using an adaptive threshold based on the throughput mean and building a classification model to predict new data labels in real-time so as to identify traffic with low throughput. Using a linear Support Vector Machine classification algorithm that could handle up to one year's worth of network traffic data, the authors tested the system with datasets from eight Data Transfer Nodes (DTNs) at major computer centres. The tests results indicated the system's ability to accurately identify large windows of low throughput. However, problems were recorded in the cases of isolated or alternating intervals during testing of the model as reported by the authors. Also, the results from the proposed method were not compared with existing systems so as to establish any improvement over the existing systems. The use of other machine learning techniques such as CNN and Random Forest would have further validated the system. Song *et al.* (2020) developed a software-hardware complex (SHC) system to detect network anomalies by formalizing normal and anomalous behavior based on the Hurst (H) parameter of network traffic. The system includes hardware, algorithms for analyzing captured traffic, statistical load parameters, and flow regulation. The system used a table of reference values for each

new subscriber and had two stages: training and detection. The system compared its performance with SolarWinds Deep Packet Inspection and found that the software components improved efficiency by identifying non-standard factors and dependencies. However, the authors suggested a shift towards a semi-controlled detection technique, where the "norm" class values would be known in advance. Cherie (2020) created a new cyber threat detection framework using a custom search engine library, a machine learning-based engine, and various algorithms. The Apache Lucene.Net search engine library was customized to function as a cyber threat detector, and Microsoft ML.NET was used to train the engine. The framework improved cyber threat detection capabilities, including self-learning and predicting attack details. However, the framework lacked feature extraction, a host-based anomaly detection engine, a heuristic-based analysis, and a robust reporting service. To advance the framework to a complete cyber threat detection application, a feature extractor component and a separate analysis framework were needed. The framework's functionality and usability could have been improved with a robust reporting service. Chen *et al.* (2020) proposed an innovative approach was proposed focusing on the modeling and identification methods of network attacks, called the FEW-NNN method. The FEW-NNN method can improve the accuracy and efficiency of flow-based network traffic attack detection and discover attacks in network traffic that contained possible network attack activities. By using the KDD99 and CIC-IDS-2017 datasets as the samples for the study, it was observed that FEW-NNN method improved accuracy and efficiency of flow-based network traffic attack detection. Zhai *et al.* (2023) used the K-means method to convert unlabeled data into labeled data, and then used the SVM algorithm iteratively to train the final maximum margin hyperplane. Thus, the K-means method saved the cost of manual labeling, and iteratively solving the maximum interval hyperplane. During the process of identifying anomalous data in substations, a technique known as Principal Component Analysis (PCA) was employed to reduce the dimensionality of the input data. This approach amounts to preserving the most influential feature dimensions while disregarding those dimensions that exhibit minimal variance, thereby achieving effective dimensionality reduction for the data. The K-means method was used to cluster the unlabeled substation data after dimensionality reduction.

Rao *et al.* (2024) proposed a novel approach to network anomaly detection utilizing a Hybrid Convolutional Neural Network (CNN) and Generative Adversarial Network (GAN) architecture. The hybrid model leveraged the strengths of both CNN and GAN to enhance the detection of network anomalies. The CNN component was designed to extract high-level features from network traffic data, allowing it to capture complex patterns and relationships within the data. Simultaneously, the GAN component acted as a generator and discriminator, learning to generate normal network traffic patterns and distinguishing anomalies from them. To train the hybrid model, employing a large dataset of labelled network traffic, encompassing both normal and anomalous behavior. During training, the GAN generated synthetic normal traffic, creating a diverse set of normal data to train the CNN and help it generalize better to variations in network traffic. In experiments, the hybrid CNN-GAN model demonstrated superior performance in detecting network anomalies compared to traditional methods. It exhibited a high detection rate while minimizing false positives, making it a promising tool for enhancing network security using MATLAB software. The proposed approach contributed to the ongoing efforts to safeguard critical network infrastructures against evolving cyber threats by harnessing the power of AI-driven anomaly detection. **Paradhi *et al.* (2024)** proposed an unsupervised machine learning approach for anomaly detection in network traffic. The work employed several unsupervised methods, including **K-Means clustering, Self-Organizing Maps (SOM), one-class Support Vector Machine (SVM)**, and **DBSCAN.** These algorithms were selected due to their ability to detect unusual patterns in network traffic without relying on labeled datasets. The study primarily focused on using clustering techniques to identify outliers and anomalies in network data, such as deviations in traffic volume, protocol behavior, and packet headers. The experiments demonstrated that these unsupervised methods could detect network anomalies efficiently, with a particular emphasis on the superior performance of clustering algorithms like DBSCAN. This was shown through their ability to identify abnormal traffic patterns, indicating potential security breaches. The

proposed approach was noted for its effectiveness in managing evolving cyber threats without needing predefined labels, making it suitable for dynamic and large-scale network environments. However, the framework was limited in its capacity to handle false positives and did not address the detection of specific attack types across all classes, indicating a potential gap in the comprehensive application of the algorithms. The challenge of real-time anomaly detection and tuning to reduce false positives also remained unaddressed.

Li *et al.* (2024) proposed an intelligent deep learning-based solution for network traffic prediction and anomaly detection within campus networks. The method integrated **Convolutional Neural Networks (CNN)** and **Long Short-Term Memory (LSTM)** networks, which were capable of extracting spatial features and modeling temporal behavior simultaneously. This combination allowed for improved accuracy in both traffic prediction and anomaly detection, addressing the evolving security challenges faced by campus networks. The authors also designed an adaptive threshold anomaly detection algorithm that adjusted its sensitivity to traffic fluctuations, improving the balance between detection accuracy and recall rates and introduced an anomaly visualization scheme using heatmaps to display spatiotemporal distributions of anomalies, which provides valuable decision support for network administrators. The large-scale experiments conducted demonstrated that this method effectively identified a wide range of security threats, including **DDoS attacks, scanning probes,** and **botnets**, achieving a detection rate exceeding 90% while maintaining a low false positive rate. Compared to traditional statistical and machine learning methods, the proposed solution exhibited better adaptability and generalization, making it suitable for dynamic and unknown traffic patterns in campus networks. However, the proposed framework did not address adversarial attacks directly, and its implementation in real-time environments may face computational overhead challenges. The study also did not delve into the integration of reinforcement learning mechanisms, which could have enhanced the model's dynamic adaptation and real-time decision-making capabilities. Future work aimed at improving the real-time performance and robustness by optimizing the lightweight implementation of deep learning models and exploring **active immunity mechanisms** to increase resilience against adversarial attacks.

Aboho and Agaji (2024) used an ensemble model for the detection of phishing URLs. The model combined predictive capabilities of Naïve Bayes and Random Forest to make a final prediction using a voting classifier. A dataset comprising 45, 0214 URLs from Phish Tank and Kaggle was collected and the Count Vectorizer function from Scikit-Learn library was applied to transform the URLS into numerical features. Results from the experiment indicated that the voting classifier achieved an accuracy of 5.36%. The performance of the classifier using various metrics like accuracy, precision, recall and F1-score showed the ability to differentiate between legitimate and phishing web addresses was evaluated. David and Thomas (2021) proposed a dynamic thresholding algorithm that provided a high detection rate with less false positives and less processing time. The approach did not depend on network topology, packet arrival pattern etc. hence avoided signature metric. Different traffic features were used to identify and discriminate DDoS attacks from the flash crowd. Using Wireshark tool, different packet header features were extracted. Based on traffic features, two attributes A1 and A2 were calculated. The mean (μ) was calculated by using a sliding window concept. DDoS attack was detected when there was a violation of threshold.

Research Methodology

The proposed system represents a significant leap forward in network traffic analysis, meticulously rectifying the deficiencies and gaps present in the current system by constructing an intricate framework, which harness the potential of unsupervised learning method to effectively encapsulate the intricate and evolving attributes of network traffic within a spectrum of diverse domains. Its core objective centers around bolstering the precision of traffic classification through an automated approach, effectively identifying both legitimate and potentially malicious data instances. Also, the proposed system strives to synthesize a comprehensive overview of recent advancements in network traffic classification, effectively bridging knowledge gaps in the domain and fostering the assimilation of cutting-edge strategies. The integration of unsupervised learning techniques unveils concealed patterns and

anomalies within network data, subsequently elevating the system's ability to detect anomalies. Ultimately, the framework furnishes network administrators and security personnel with a panoramic vista of traffic analysis. This holistic perspective holds the potential to optimize the allocation of network resources, fortify security measures, and enhance the overall efficiency of network management. The objectives underlying the proposed framework are multifaceted. The architecture of the framework is shown in Figure 1.

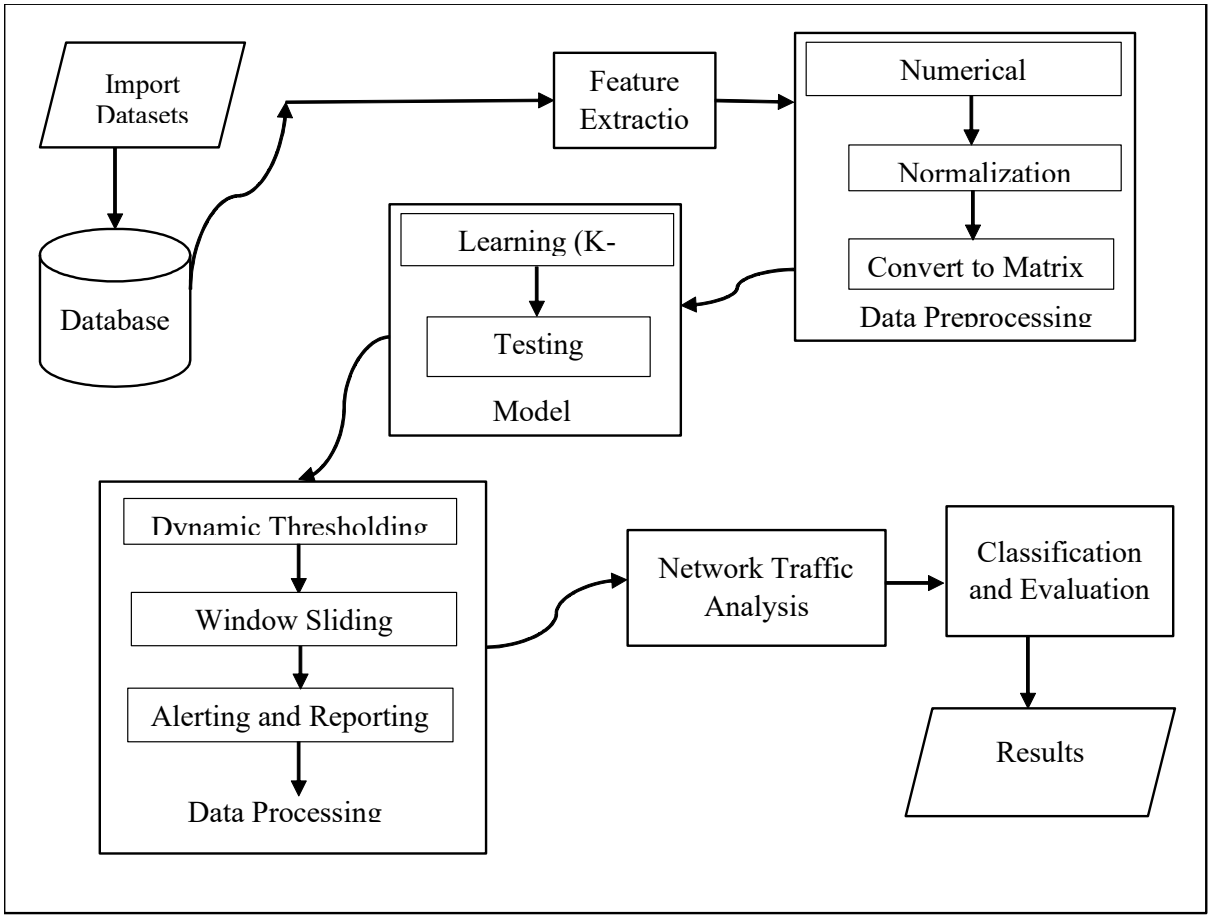

Figure 1: The Model Architecture of the Framework

.In Figure 1, network traffic datasets were obtained and stored into a database and preprocessed. The collected network traffic data was transformed into a CSV format that was suitable for the machine learning algorithm. In feature extraction component, features from the dataset were extracted for use in the machine learning model. The extracted features were processed and formatted for machine learning algorithm. Data was filtered and normalised. In learning model, the k-means machine learning algorithm was used to test the preprocessed data. The machine learning model was used to analyze, classify and evaluate network traffic. This involved detecting anomalies and classifying traffic based on various criteria including thresholding and window sliding techniques. The evaluation of the performance metrices was carried out using V-measure, Rand Index, Adjusted Rand Index, Mutual Information and Normalized Mutual Information and the results were displayed for feedback.

Method of Data Collection

The datasets collected for the purpose of this study was the secondary datasets which were obtained from Kaggle website. The datasets were UNWSW-NB15 dataset and Diverse Network Traffic Datasets collected respectively from (https://www.kaggle.com/datasets/mrwellsdavid/unwsw-nb15) and (https://www.kaggle.com/code/adepvenugopal/network-datasets) .

Detailed Design of the Framework

The flowchart was used for detailed design of the framework. The detailed design is shown in Figure 2

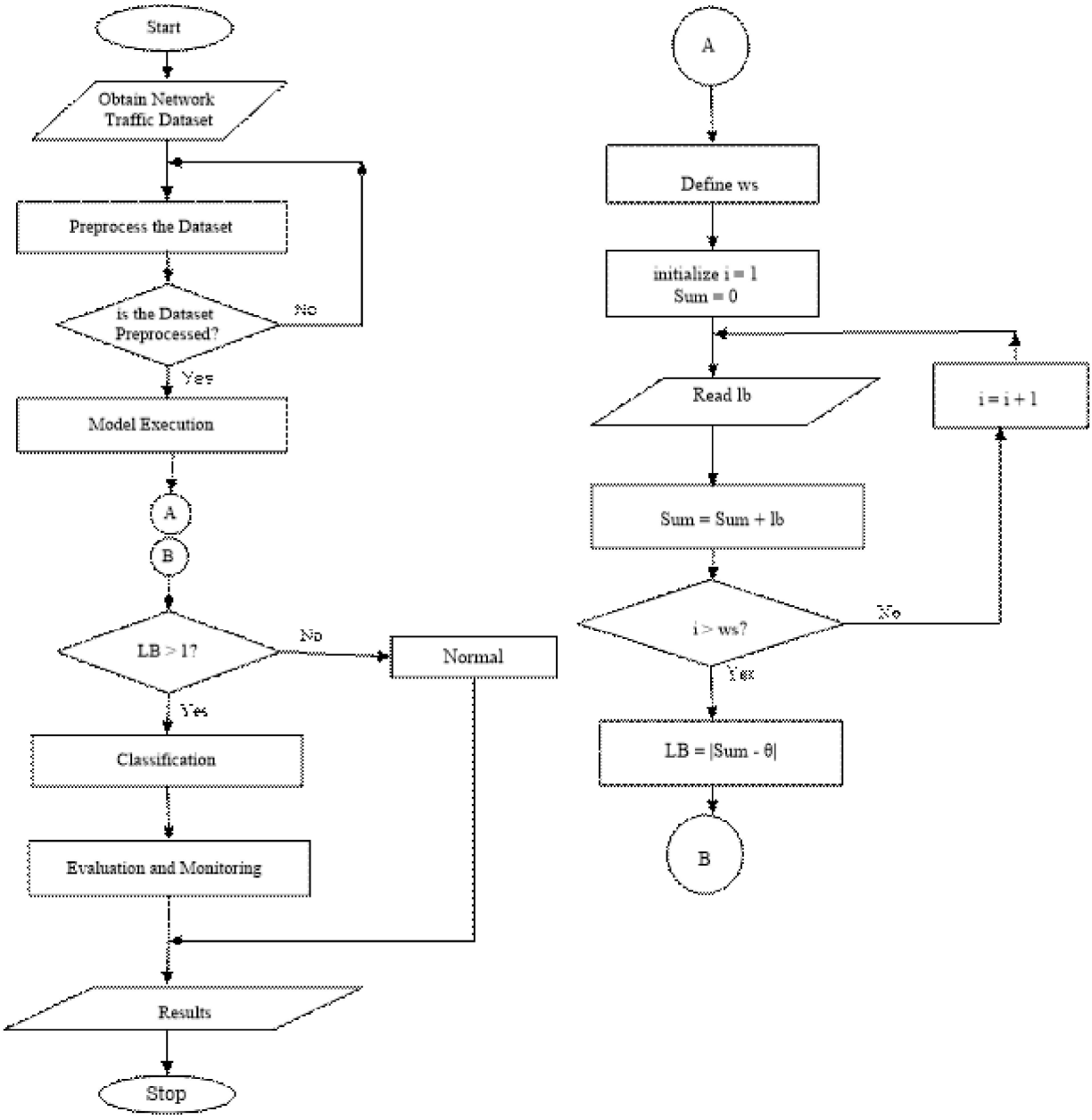

Fig 2: The Flowchart of Framework

Figure 2 shows the System Flowchart of the Framework. In Figure 2 datasets are first obtained from secondary sources and stored in a database, preprocessed to reduce dimensionality, outliers and then formatted to suit the K-means learning model. The K-means model takes the formatted dataset and executes ensuring the activation of the thresholding condition through the window sliding technique is performed in which **LB** is the Label window, **ws**, the window size, **lb** the flow window, **sum** is the summation of flows within a window and **i** is the count of the flows within a window size. This process determines the status of the dataset, whether benign or malicious and thus classify appropriately. Further analysis is the evaluation of the performance metrics for V-measure, Mutual Information, Normalized Mutual Information, Rand Index and Adjusted Rand Index. The results are displayed graphically and numerically for decision making.

Any other similar dataset can be obtained and subjected to pass through the framework.

Experiments

A total of two experiments were carried out in this study using python programming language on Google Colab. The first experiment was to train, classify and evaluate the framework using UNWSW-NB15 dataset which has different network domains. The second experiment was

conducted to test the implemented framework using the Diverse Network Traffic Dataset, in order to classify the dataset and evaluate the framework.

Experiment 1: Training, Classification and Evaluation of the Framework

The experiment was conducted using the unsupervised K-means model and UNWSW-NB15 dataset of different domains. The experiment enabled the framework to learn how to detect benign and anomalous data and evaluate the performance of a given data set in terms of V-measure, Rand Index, Adjusted Rand Index, Mutual Information and Normalized Mutual Information. It served as base experiment.

Inputs

The primary input was the dataset containing network traffic data from various domains. This dataset includes features such as packet size, protocol type, source and destination IP addresses, and timestamps. Additionally, parameters for configuring the K-means algorithm, like the number of clusters and convergence criteria, were considered in the input. **X_data** Contained input features such as packet size, protocol type, source and destination IP addresses, and timestamps (independent variables) for preprocessing. These features were crucial for the model to analyze and cluster network traffic data effectively. **Y_data** Contained the target variable (dependent variable), representing the classification of the network traffic domain to be predicted. Scaling is applied to ensure consistency for modeling. The sample data is shown in Figure 3

	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sload	dload	sloss	dloss	sinpkt	dinpkt	sjit	djit	swin
31	1.924287	tcp	-	FIN	10	8	530	354	8.834441	1983.072266	1288.789062	2	1	213.738785	261.252014	13125.111328	399.405151	255
32	2.773779	tcp	-	FIN	14	8	7954	354	7.570899	21302.345703	894.087097	4	1	213.367615	381.293427	18158.228516	788.701355	255
34	1.465041	tcp	-	FIN	10	8	2516	354	11.603770	12368.253906	1692.785400	2	1	157.703445	195.494720	8764.723633	346.639313	255
35	0.983874	tcp	http	FIN	10	8	816	1172	17.278635	5976.375000	8342.531250	2	2	109.319336	124.932861	5929.211914	192.590408	255
40	1.535254	tcp	http	FIN	10	10	826	1266	12.375802	3876.882812	5940.385254	2	2	170.481888	159.070557	11933.065430	244.118011	255
43	0.759306	tcp	-	FIN	10	8	534	354	22.388865	5067.785645	3266.140381	2	1	84.242668	98.501709	4716.886719	140.050613	255
44	1.294036	tcp	-	FIN	10	8	2516	354	13.137192	14002.701172	1916.484497	2	1	143.781784	173.331146	9390.189453	257.271362	255
45	1.059359	tcp	http	FIN	10	8	830	1134	16.047441	5641.147461	7498.874512	2	2	117.706558	132.899277	6102.668457	255.756226	255
46	0.624048	tcp	-	FIN	10	6	534	268	24.036613	6166.192383	2871.573975	2	1	69.229668	106.834801	3412.856201	145.655869	255
47	1.190292	tcp	-	FIN	10	8	2516	354	14.282210	15223.155273	2083.522461	2	1	132.107117	154.911713	8972.551758	220.423187	255

Figure 3: The UNWSW-NB15 Dataset Sample

Results

The results of Experiment 1 were as shown in Figure 4 – 5 and Tables 1 – 2.

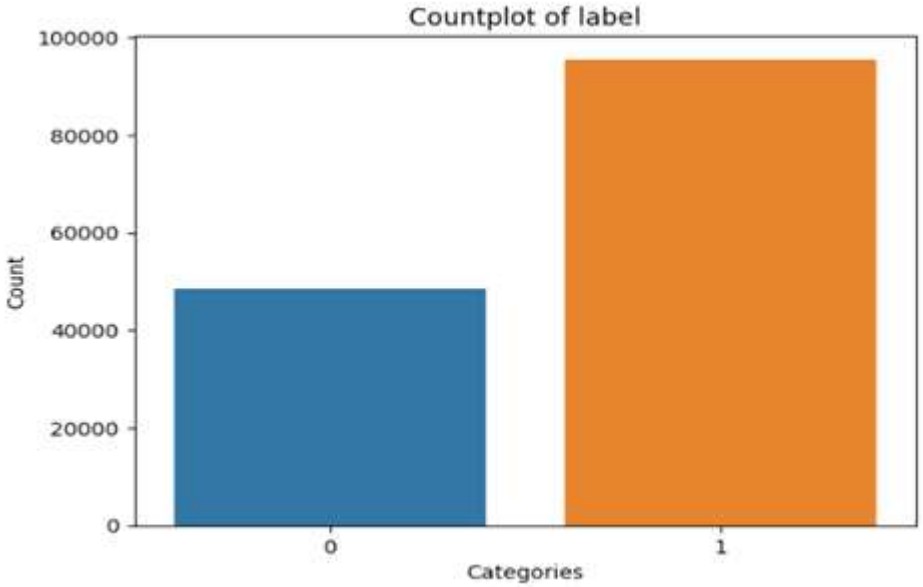

Figure 4: Count Plot of Packet Labels for UNWSW-NB15 Dataset

Figure 4 presents the count plot bars representing the count of occurrences for each packet label category in UNWSW-NB15. The figure shows 0 for normal packets and 1 for abnormal packets.

Results and Discussions of Experiment 1

Table 1: Thresholding and Classification

Dataset	Threshold	K-means Classification	
		Anomalous (%) Abnormal	Benign (%) Normal
Exp 1 UNWSW- NB15	0.9404	66.22	33.78

Table 1 is the numeric result as obtained in Experiment 1, showing 66.22% anomalous and 33.78% benign and a threshold value of 0.9404. This is similar to the graphical result obtained in Figure 4.

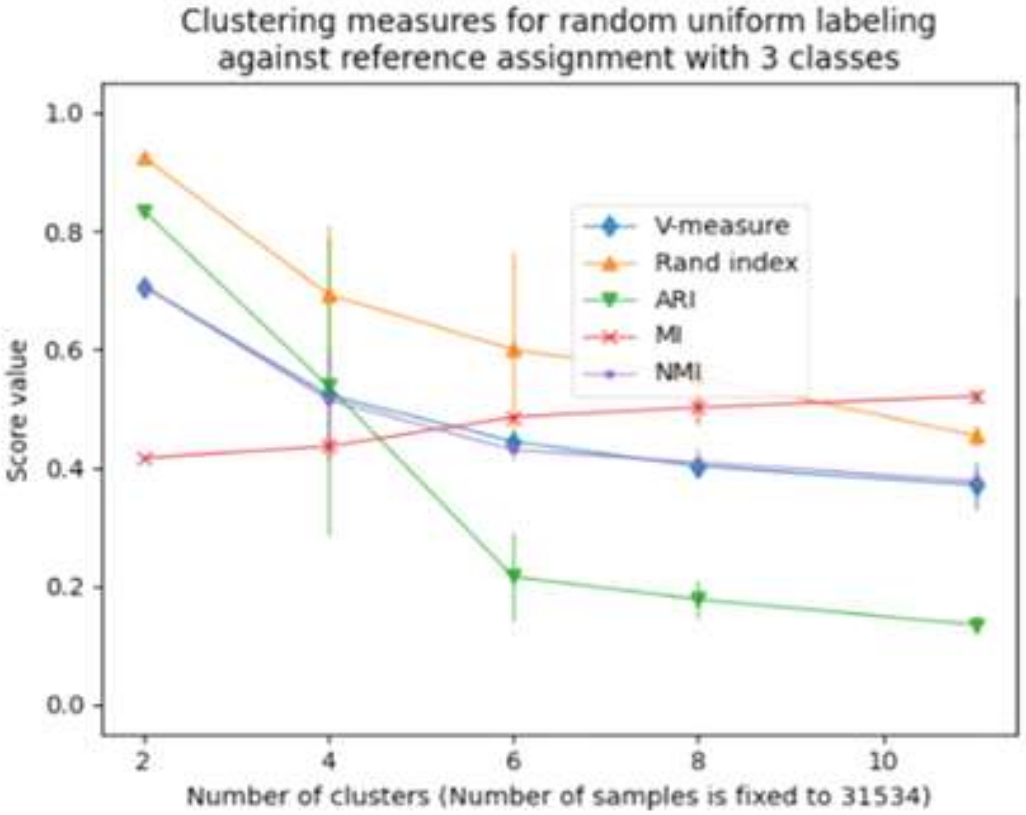

Figure 5: A Plot of Evaluation Metric Scores for UNWSW-NB15 Dataset

In Figure 5, several metrics (V-measure, Rand Index, Adjusted Rand Index (ARI), Mutual Information (MI), and Normalized Mutual Information (NMI)) were used to assess the quality of clustering. And the results obtained showed that: V-measure yielded its highest score of approximately 0.7356 when k=2 and the lowest value of 0.4444 at k =7. It can be observed that as the number of clusters increased beyond 2, the V-measure generally decreased, indicating potential diminishing returns in clustering performance. Rand Index achieved its peak value of approximately 0.9436 at k=2 and a 0.5190 value at k = 9 as its minimum. This high degree of the peak value is in agreement with the effectiveness of the framework in capturing the underlying structure of the network traffic data. Adjusted Rand Index (ARI), contrary to V-measure and Rand Index, the ARI reached its maximum score of approximately 0.8753 when k=4. This discrepancy suggests that while fewer clusters may optimize homogeneity and completeness, a slightly higher number of clusters may better capture the true clustering structure, as indicated by ARI. This result also meets expectation that it is better used for large number of clusters. Mutual Information attained its highest score of

approximately 0.5670 when k=10. This suggests that the clustering algorithm captures a significant amount of information about the underlying data structure, particularly with a larger number of clusters. Again, this meets expectation that it increases as the number of clusters increase.

Normalized Mutual Information (NMI) achieved its maximum score of 0.7356 when k=2, aligning correctly with the V-measure results as it is expected. This indicates a strong agreement between the true labels and the predicted clusters, further supporting the effectiveness of the framework.

The results show that V-measure, Rand Index, Adjusted Rand Index and Normalized Mutual Information show higher scores compared to Mutual Information. Basically, this indicates that the clustering method performed well in terms of grouping data points together that belong to the same class. However, Mutual Information showed comparatively lower performance. Hence, this implies that while the clustering is good generally, it might not perfectly capture all the relationships between the true labels and the clusters. The results of each performance metrics of the UNWSW-NB15 dataset used for the framework and their maximum score is as shown in Table 2.

Table 2: Generic Result of the Performance Metrices for UNWSW-NB15 Dataset

K	V-measure	Rand index	ARI	MI	NMI
2	0.735622	0.943630	0.8743568	0.425853	0.735622
3	0.711410	0.943367	0.874427	0.433478	0.711410
4	0.700833	0.943296	0.875348	0.453235	0.700833
5	0.511233	0.666293	0.387693	0.462418	0.511233
6	0.503032	0.665997	0.387530	0.462450	0.503032
7	0.444359	0.5951491	0.287667	0.465243	0.444359
8	0.513319	0.606956	0.313542	0.566263	0.513319
9	0.455335	0.519014	0.198788	0.566608	0.455335
10	0.455533	0.520751	0.200982	0.566962	0.455533
	V-measure: Max Score = 0.7356 at k = 2	**Rand index: Max Score = 0.9436 at k = 2**	**ARI: Max Score = 0.8753 at k = 4**	**MI: Max Score = 0.5670 at k = 10**	**NMI: Max Score = 0.7356 at k = 2**

The numeric results in Table 2 are visualization of each of the performance metrics for varying number of clusters. The performance of our framework for anomaly detection was based on the evaluation metrics such as V-measure, Rand Index, Adjusted Rand Index (ARI), Mutual Information (MI), and Normalized Mutual Information (NMI). The framework was tested under varying numbers of clusters (k) to determine the optimal configuration for anomaly.

Hence, the network traffic analysis framework demonstrates strong performance in anomaly detection, as evidenced by high scores across multiple evaluation metrics. While the optimal number of clusters may vary slightly depending on the evaluation metric used, the framework consistently performs well as it is expected. These findings underscore the potential of our framework for effective anomaly detection in network traffic data.

Experiment 2: Test Dataset for Classifying and Evaluating the Framework

The experiment was conducted to show how the framework can classify and evaluate data from different domains. Diverse Network Traffic Dataset was used as

Inputs:

The input is a dataset from a Diverse Network Traffic. This dataset has similar features as the first dataset such as packet size, protocol type,

source and destination IP addresses, and timestamps used in experiment 1. **X_data:** Contains input features such as packet size, protocol type, source and destination IP addresses, and timestamps (independent variables) for preprocessing. **Y_data:** Contains the target variable (dependent variable), representing the classification of the network traffic domain to be predicted. Scaling is applied to ensure consistency for modeling.

Figure 6 is the Sample data used for this Experiment

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	root_shell	su_attempted
0	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0
1	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0
2	2	tcp	ftp_data	SF	12983	0	0	0	0	0	0	0	0	0	0
3	0	icmp	eco_i	SF	20	0	0	0	0	0	0	0	0	0	0
4	1	tcp	telnet	RSTO	0	15	0	0	0	0	0	0	0	0	0
5	0	tcp	http	SF	267	14515	0	0	0	0	0	1	0	0	0
6	0	tcp	smtp	SF	1022	387	0	0	0	0	0	1	0	0	0
7	0	tcp	telnet	SF	129	174	0	0	0	0	1	0	0	0	0
8	0	tcp	http	SF	327	467	0	0	0	0	0	1	0	0	0
9	0	tcp	ftp	SF	26	157	0	0	0	0	1	0	0	0	0

Figure 6: Diverse Network Traffic Dataset Sample

Results and Discussion of Experiment 2

The network traffic domain classification is shown in Table 3.

Table 3: Thresholding and Classification

Dataset	**Threshold**	**K-means Classification**	
		Anomalous (%)	**Benign (%)**
Exp 2 Diverse Network Dataset	0.9404	54.44	45.56

Table 3 shows that with a threshold of 0.9404 the Diverse Network Traffic Dataset was with 54.44% anomalous and 45.56% benign values.

Performance Metrics of the Framework

The results of each performance metrics of the Diverse Network Traffic dataset used on the framework and their maximum score is as shown in Table 4.

Table 4: Generic Result of the performance metrices for Diverse Network Traffic Dataset

K	**V-measure**	**Rand Index**	**ARI**	**MI**	**NMI**
2	0.000166	0.943630	0.000401	0.000031	0.000166
3	0.005588	0.943367	0.037434	0.002669	0.005588
4	0.005884	0.943296	0.037535	0.002827	0.005884
5	0.062123	0.666293	0.020480	0.042485	0.062123
6	0.065485	0.665997	0.023283	0.050482	0.065485
7	0.049100	0.595149	0.018727	0.040237	0.049100
8	0.048913	0.606956	0.018578	0.040105	0.048913
9	0.048375	0.519014	0.000664	0.046151	0.048375
10	0.056672	0.520751	0.005541	0.056152	0.056672
	V-Measure: Max Score = 0.0655 at k = 6	**Rand Index: Max Score = 0.9436 at k = 2**	**ARI Max Score = 0.0375 at k = 4**	**MI: Max Score = 0.0562 at k = 10**	**NMI: Max Score = 0.0655 at k = 6**

The numeric results in Table 4 are the visualization of each of the performance metrics using Diverse Network Traffic Dataset. The Table shows the performance evaluation of our network traffic analysis framework for anomaly detection using various evaluation metrics. V-measure achieved its highest score of approximately 0.0655 when k=6. As the value of k increased beyond 6,

the V-measure generally fluctuated, with no significant improvement observed. The lowest value is observed as 0.0001655 at k = 2. Rand Index attained its peak score of approximately 0.9436 when k=2. Subsequent increases in k led to a decrease in the Rand Index. Adjusted Rand Index (ARI) reached its highest value of approximately 0.0375 when k=4. Similar to the Rand Index, the ARI generally decreased as k increased beyond the optimal point. Mutual Information (MI) achieved its maximum score of approximately 0.0562 when k=10. This suggests that the clustering algorithm captured a significant amount of information about the underlying data structure, particularly with a larger number of clusters and Normalized Mutual Information (NMI) aligned correctly with V-measure, reaching its highest score of approximately 0.0655 when k=6. As with V-measure, subsequent increases in k did not result in improvements of NMI.

The framework was evaluated with different numbers of clusters (k) to determine its effectiveness in identifying anomalies.

Conclusion

In conclusion, our network traffic analysis framework for anomaly detection has shown robust performance across various evaluation metrics, despite slight variations in the optimal number of clusters suggested by different metrics. While the choice of evaluation metric may influence the determination of the optimal number of clusters, our framework consistently demonstrated strong performance across different settings. The variation in the optimal number of clusters highlights the importance of comprehensive evaluation and consideration of trade-offs between different aspects of clustering performance. Despite this variation, the framework's ability to adapt to different clustering configurations underscores its robustness and applicability to diverse datasets and anomaly detection scenarios. Moving forward, future research could focus on further refining the framework's algorithms, exploring additional evaluation metrics, or incorporating domain-specific knowledge to enhance anomaly detection accuracy and efficiency. By continuously improving and adapting the framework, we can address emerging challenges in network traffic analysis and contribute to the development of more effective anomaly detection solutions.

References

Aidahoul, N., Karim, H.A., and Wazir, A.S.B. (2021). Model Fusion of Deep Neural Networks for Anomaly Detection. *Journal of Big Data*. https://doi.org/10.1186/s40537-021-00496-w

Aboho M D and Agai I (2024) Ensemble Model for the Detection of Phishing URLs. Ilournal of Computer Science and Information Technology, 7(1) : 1-25

Chandrasekaran, B. (2009). *Survey of network traffic models*. Washington University, St. Louis. http://www.cse.wustl.edu/~jain/cse567-06/ftp/traffic_model3/index.html

Chen, L., Gao, S., Liu, B., Lu, Z., and Jiang, Z. (2020). FEW-NNN: A fuzzy entropy weighted natural nearest neighbor method for flow-based network traffic attack detection. *China Communications*, 17(5), 151–167. doi:10.23919/jcc.2020.05.013

Cherie, Meshesha K., (2020). "Network Traffic Analysis Framework For Cyber Threat Detection" *Masters Theses and Doctoral Dissertations*. 343. https://scholar.dsu.edu/theses/343

Conti, M., Li, Q. Q., Maragno, A., and Spolaor, R. (2018). The dark side (-channel) of mobile devices: A survey on network traffic analysis. *IEEE communications surveys and tutorials*, *20*(4), 2658-2713.

David, J and Thomas C. (2021). Discriminating flash Crowds from DDoS attacks Using Efficient Thresholding Algorithm. Journal of Parallel and Distributed Computing 152 pp 79-87. Elsevier Inc.

D'Alconzo, A., Drago, I., Morichetta, A., Mellia, M., and Casas, P. (2019). A survey on big data for network traffic monitoring and analysis. *IEEE Transactions on Network and Service Management*, *16*(3), 800-813.

Foremski, P. (2013). On different ways to classify Internet traffic: a short review of selected publications. *Theoretical and Applied Informatics*, *25*.

Iorliam, A. (2016). Application of power laws to biometrics, forensics and network traffic analysis. University of Surrey (United Kingdom).

Ji, S. Y., Jeong, B. K., & Jeong, D. H. (2020). Evaluating visualization approaches to detect abnormal activities in network traffic

data. *International Journal of Information Security*, 1-15.

Kim, J., Kim, Y., Yegneswaran, V., Porras, P., Shin, S., & Park, T. (2023). Extended data plane architecture for in-network security services in software-defined networks. *Computers & Security*, *124*, 102976.

Kizza, J. M. (2024). System intrusion detection and prevention. In *Guide to computer network security* (pp. 295-323). Cham: Springer International Publishing.

Li, J., Linsangan, N.B., Dong, H., (2024). Campus Network Traffic Prediction and Anomaly Detection Based on Deep Learning. *International Journal of Emerging Technologies and Advanced Applications* 2024, 1(7), 1-12.

Mutmbak, K., Alotaibi, S., Alharbi, K., Albalawi, U., and Younes, O. (2022). Anomaly Detection using Network Metadata. *International Journal of Advanced Computer Science and Applications* volume 13, Number 5.

Paradhi, D., Ansari, M. N., & More, S. (2024). Anomaly Detection in Network Traffic Using Unsupervised Machine Learning. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*. DOI: 10.48175/ IJARSCT-19264.

Rao, V. S., Balakrishna, R., El-Ebiary, Y. A. B., Thapar, P., Saravanan, A. K. and Godla, S. R. (2024), AI Driven Anomaly Detection in Network Traffic Using Hybrid CNN-GAN. Journal of Advances in Information Technology, Vol. 15, No. 7

Sanz, I. J., Lopez, M. A., Viegus, E.K. and Sanches, V.R. (2020), A Light Weight Network based Android Malware Detection System. IFIP Networking Conference IEEE. PP.695-703. Shikhaliyev, R. (2017). The conceptual model for the intellectual monitoring system of computer networks. *Problems of information technology*, *8*(2), 26-30.

Singhal, P., Mathur, R., and Vyas, H. (2013). Network Traffic Classification based on Unsupervised Approach. *International Journal of Computer Applications (0975–8887)*.

Song, W., Beshley, M., Przystupa, K., Beshley, H., Kochan, O., Pryslupskyi, A., and Su, J. (2020). *A Software Deep Packet Inspection System for Network Traffic Analysis and Anomaly Detection. Sensors, 20(6), 1637.* doi:10.3390/s20061637

Wang J, Yang L, Wu, J. and Abawajy, J. H. (2017). Clustering Analysis for Malicious Network Traffic. IEEE International Conference on Communication.DO1:10.1109/ icc.2017.7997375

Wu, Y., Wei, D., & Feng, J. (2020). *Network Attacks Detection Methods Based on Deep Learning Techniques: A Survey. Security and Communication Networks, 2020, 1–17.* doi:10.1155/2020/8872923

Zeinali, M., & Barenji, R. V. (2023). Communication Networks Characteristics Impact on Cyber-Physical Systems. In *Control Engineering in Mechatronics* (pp. 189-202). Singapore: Springer Nature Singapore.

Zhai, W., Wu, Y., Qi, B., Xue, T. and Wu. Q. (2023) "Abnormal data detection method of smart substation based on K-means-SVM," *Proc. SPIE 12922, Third International Conference on Electronics, Electrical and Information Engineering (ICEEIE 2023)*, 129220Q (27 October 2023); doi: 10.1117/ 12.3008728

Zola, F., Seguola-Gil L., Bruse, J. L., Galar, M. O., Orduna-Urrutia, R. (2022). Network Traffic analysis through node behavior classification: Graph-based approach with temporal dissection and data-level preprocessing. Computers and Security Vol 115, page 1-18.