**National University Of Computer And Emerging Sciences**

**Masters of Science In Cyber Security**



**Real-Time Detection and Prediction of Zero-Day Attacks in Network Traffic Using Ensemble Unsupervised Learning**

**Kanwar Azlan (23K – 7709)**

**Supervisor: Asst. Prof. Dr. Fahad Samad**

**Master’s Thesis Report I**

**FALL [2025]**

# **ABSTRACT**

In the recent times, Internet networks are the one of the most in demand and critical to protect and safeguard from the zero day attacks. Multiple types of malicious traffic can be used to inject the payloads into the encrypted network traffic so that the system can be compromised. Existing methodologies proposed are effective in detecting zero day attacks in network traffic but have limitations like they used Traditional methods of Supervised Machine learning and Deep learning models which are not much effective in detecting unknown attacks. Unsupervised learning surpassed the existing models and can have best results in detecting zero day or unknown attacks as it does not require pre-defined labels on dataset. In this paper, we have proposed a solution of predicting unknown attacks by using Ensemble (i.e. combining multiple models for getting better results) Unsupervised learning models (FAST ABOD, Isolation Forest and K-means). We will use not only one but multiple ML models to detect and predict zero day attacks in encrypted network traffic so that we can have a better approach.

# **INTRODUCTION**

Widespread use of traffic encryption has been made to safeguard data transmitted over the Internet. In 2019, more than 80% of websites switched to HTTPS to guard against data breaches. The cypher suite for this kind of security has two drawbacks, though. Specifically, encrypted communication enables adversaries to hide their malevolent actions, such as ransomware campaigns, vulnerability-exploiting, and data exfiltration. The proportion of encrypted malicious traffic on the Internet is rising quickly, and it now accounts for more than 70% of all malicious traffic ([2]). As the paper ([1]) concludes The zero-day attack detection methods based on supervised and hybrid learning utilize supervised learning or a combination of supervised and unsupervised learning. When provided with appropriate training data sets, these methods can accurately detect zero-day attacks. However, the data sets typically lack representation of zero-day attacks, which means the detection schemes need to assume that zero-day attacks behave similarly to known attacks. This assumption has not been validated yet. Another paper ([3]) suggested that Supervised classifiers and deep learners are generally effective at identifying familiar attacks, but they struggle to detect unfamiliar attacks, such as new types of attacks, modified versions of existing exploits, or threats that the intrusion detector is not equipped to handle. So to Overcome these issue we have proposed an ensemble unsupervised learning that has these benefits. Like, An area of machine learning techniques called unsupervised learning uses unlabeled data to identify patterns. In order to identify zero-day assaults, the zero-day attack detection model aims to learn a compressed representation of typical data. Zero-day attack data is not needed for the outlier detection algorithms' training. Since the zero-day attack data is usually unavailable during the model training phase, this is a huge advantage ([1]). The best way to identify unknown attacks, according to the results, is to use unsupervised meta-learners, which can identify known assaults more accurately than a few supervised classifiers ([3]). After these motivations, we are proposing ensemble learning approach to overcome the efficiency. Firstly we will gather the dataset of the network traffic, then we will preprocess the dataset. It is an efficient way because it can remove outliers, normalize the data and extract important features out of it. Then we will pass the dataset without labeling to the Unsupervised ML model to detect and predict the unknown attacks. Then finally we compare the results of multiple Unsupervised ML models used so that we can detect and predict the zero day.

# **LITERATURE REVIEW**

In existing machine learning-based zero-day attack detection research, the data used for both training and testing is limited. Evaluation results often fail to demonstrate that the proposed zero-day attack scheme provides satisfactory detection accuracy and is uniformly effective against different types of zero-day attacks. Although many studies have been conducted on intrusion and malware detection using various approaches, none have focused on ML-based zero-day attack detection schemes. It is highly desirable to conduct a comprehensive review of existing ML-based models to compare and contrast their advantages and disadvantages and to identify major design and evaluation gaps ([1]).

Another study states that, Existing methods cannot detect malicious traffic encrypted with unknown patterns by learning the traffic characteristics of a single flow. However, it is still possible to detect such attack traffic, because these attacks involve multiple attack stages with different flow interactions between attackers and victims, which are distinct from benign flow interaction patterns ([2]). Multiple survey conducted on network traffic analysis as it is the hot topic for cyber security researches and also the attackers, another survey concludes, the goal of this survey is to identify the means to achieve encrypted network traffic analysis and inspection effectively.

This survey will help researchers in the field to (i) understand the challenges of traffic inspection when the network traffic is encrypted or tunneled, (ii) discover the use cases and applications of encrypted traffic analysis, (iii) acquire knowledge of the methods that are used to achieve encrypted traffic analysis, (iv) deduce which techniques are appropriate with respect to the objectives of a system, (v) recognize the constraints each method presents, and finally, (vi) identify the publicly available datasets that are appropriate for use ([4]).

Although, as most of the prior work have been done on using Machine Learning models to mitigate and cater the unknown behavior or attacks in the network traffic. Many different types of models have been used previously and have different types of approaches in network anomaly detection. Like the paper ([5]), propose a localized ADS scheme called Scalable and Energy-Efficient Anomaly Detection Scheme (SEEADS). The scheme consists of a detection activation module, a lightweight pre-detection module, a severe anomaly, and a dynamic strategy selection module. The complexity is reduced by developing a localized and adaptive heavy vehicle detection module called an anomaly detection scheme based on semi-supervised evolutionary local learning (LESLA).

In another study, He proposes Adriot, a structure of the detection of the IoT network that uses an edge computer to identify potential threats. This area is approved by a traffic car, a preliminary processor for traffic, and an abnormal detection module composed of an abnormal collection dedicated to each type. Each detector is generated by an LSTM auto encoder in an unsupervised manner ([6]). In recent developments in NIDS, DL algorithms are often used to accurately detect malicious activities. However, the inherent vulnerability of DL algorithms to adversarial examples opens up new attack surfaces for attackers ([7]). The paper examines which classifiers show the best detection performance across datasets and training options, and illustrates the differences between supervised (i.e., SUP, DEEP, META-SUP) and unsupervised (i.e., UNS, META-UNS) approaches. Discusses the ability to detect unknowns and their impact on detection performance. We then compare the performance of the best supervised classifier (XGBoost) with the best unsupervised classifier (FastABOD Boost). Shows that META-UNS improves all metric scores over UNS classifiers. In particular, the Boosting FastABOD ensemble, generates ACC scores that outperform DEEP classifiers; this halves the accuracy gap between META-SUP and UNS classifiers. This is important because unicard classifiers are traditionally considered much worse than SUP, deeply and meta-up-classifier, and are much less used in practice ([3]).

The Table I below shows the comparison of the existing solutions on predicting unknown attacks on encrypted network traffic:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IDEA | RESEARCH GAP | METHODOLOGY | TECHNOLOGY | LIMITATIONS | MITIGATIONS | YEAR |
| A Review of ML based zero day detection. | Signature based detections | Comparison of different ML models to predict | SL, UN SL, DL models and other ML models. | Zero day attack Dataset is limited. | Honeypots can be used to gain better dataset for prediction. | 2023 |
| Comparison of SL, UN SL, META Learning Models. | Traditional models face difficulty in detecting zero day attacks. | Tried Different classifiers of ML models on different type of dataset. | ML models  47 classifiers on 11 public dataset. | SL, META SL, are not effective in detecting zero day attacks. | META UN SL can be used as the future direction. | 2023 |
| Survey of encrypted network traffic analysis | Spreading of malware, taking advantage of encryption protocol | Comparison of domain specific network traffic analysis using Ml models. | Categorization of existing solutions of ML models based on network traffic. | High False positive, Privacy Concern | Better classification accuracy, reduce false positive and good performance. | 2021 |
| Scalable and Energy efficient anomaly detection | Security and Scalability concern in SDN networks | Semi-unsupervised learning, use previous prediction of each packet. | Detection activation model, Heavyweight and lightweight model. | It works under an assumption, Can cause delay. | Maliciousness estimation update scheme can be used for more efficiency. | 2021 |
| Edge Assisted anomaly detection framework | Low power devices were used, resource constraint for single device | First traffic capturer, then preprocessor and collection of anomaly detection. | Edge computing devices, LSTM autoencoder | Making assumptions about the data flow that is benign, certain attacks can still happen. | IoT based attacks can be detected with edge assisted architecture | 2022 |
| Federated graph neural network for anomaly detection in CAN | Existing CAN bus lack real time performance. | Generated directed graph for CAN message, Node attributes denote data contents, then make predictions | Federated learning approach, Graph neural network (GNN) | Assumption of vehicle’s from same model, lack of public specifications | Can detect multi-stages attack types in CAN messages | 2023 |
| Flow interaction graph analysis | DL based model used, graph dependent on network patterns | Real time graph generation, categorize them based on Ml models for detecting unknown attacks | GNN, Deep learning models. | Dependencies on network patterns, Effective Model but not optimal solution used. | GNN based approach to detect unknown attacks from analyzing network patterns | 2024 |
| Fine grained unknown class detection against open set attacks | Previous models cannot detect anomaly type, Network traffic could be ever changing | Splitting the data, then leveraging the concept of isolation, advancing the incremental model to cope up with network traffic. | Novel Tree based Model, Supervised Learning | Suffers with large number of data, The scheme should be customized for every user. | FOSS identified known and unknown attacks to legitimate traffic in incremental model. | 2024 |
| Network based anomaly detection using GRU | DL models suffers real time detection, availability of data, false positive rate. | Improve the data in training, CIA should be used for security. Used GRU NN for better accuracy. | SMOTE Algorithm, GRU, NN, DL models. | Greater false alarm rate. High processing power | Provide technique for detecting VANET network traffic anomalies. | 2024 |
| Mitigating poisoning attacks in Semi-supervised learning | False data patterns, sample dependencies, noisy data. | Feature extraction, poisoning attack impact estimation, detect anomalies with clustering algorithms | gradient descent algorithm, DL based models, VRNN | The complexity of the PRAD based approach is still questionable. | PRAD framework for detecting encrypted traffic anomalies. | 2024 |
| Unsupervised anomaly detection for network security | False positive, no prior info, anomaly patterns | Flow generation, feature extraction, anomaly detection then experiment to validate. | MSC, K-means++ Apriori, random projections, Unsupervised Modes | Depending on IP address of network in packet info, More scalability and reliability | Used unsupervised learning MSCA to increase robustness of detection by multiple independent results | 2023 |
| Anomaly detection in SDN | Roughly categorizations of existing models, no sufficient coverage of anomalies. | Flow selection for forwarding graph, finding optimal no. of rules, rule installation, then detecting anomalies | Depth first traversal (DFT), flow selection algorithm, ILP model | Rule based approach is used instead of dynamic. | Proposed FADE framework to detect anomalies and categorize them in SDN networks | 2021 |

*Table 1 Literature Review*

Apart from the existing literature we have studied, We have chosen to work with an existing study included as are baseline in the paper. They have used “Flow interaction graph based analysis for detecting unknown malicious network traffic”. According to them, existing methods cannot detect malicious traffic encrypted with unknown patterns by learning the traffic characteristics of a single flow. However, it is still possible to detect such attack traffic, because these attacks involve multiple attack stages with different flow interactions between attackers and victims, which are distinct from benign flow interaction patterns ([2]).

They have kept the technique of graph based analysis by following multiple stages or phases in the framework. The graph construction consist of the following phases. The raw network packets were captured from the MAWI Dataset ([21]). The dataset is provided by the MAWI working Group of traffic Archive. Now from the network flow, the flows will be classified as Short and Long flows depending upon the hyper parameters defined in ([22]). They classified Short and Long Flows based on the ‘Pkt\_Timeout’, ‘Judge\_Interval’, ‘FLOW\_LINE’.

**EXPERIMENTAL SETUP:**

As per the paper ([2]), the flow interaction graph analysis framework provided each steps to detect anomalies in encrypted network traffic, we have replicated the phases.

**Dataset:**

The raw network packets were captured from the MAWI Dataset ([21]). The dataset is provided by the MAWI working Group of traffic Archive. The dataset contains the relevant features including *Source\_IP, Destination\_IP, Length, Time, Protocol, Info*. Below is the visual representation of the dataset is something like this:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DATASET | INSTANCES | FEATURES | SAMPLE | DESCRIPTION |
| MAWI working Group of traffic Archive | 69 Million | Time  Source  Destination  Protocol  Length  Info | 3.5+ Million | The Dataset is collected on a random day of January, 2020.  File type: .pcap |

*Table 2 MAWI Group: Encrypted Network Traffic Dataset.*

We have used Flow classification Algorithm and short Flow Aggregation Algorithm for the implementation provided in ([22]).

**RESULTS:**

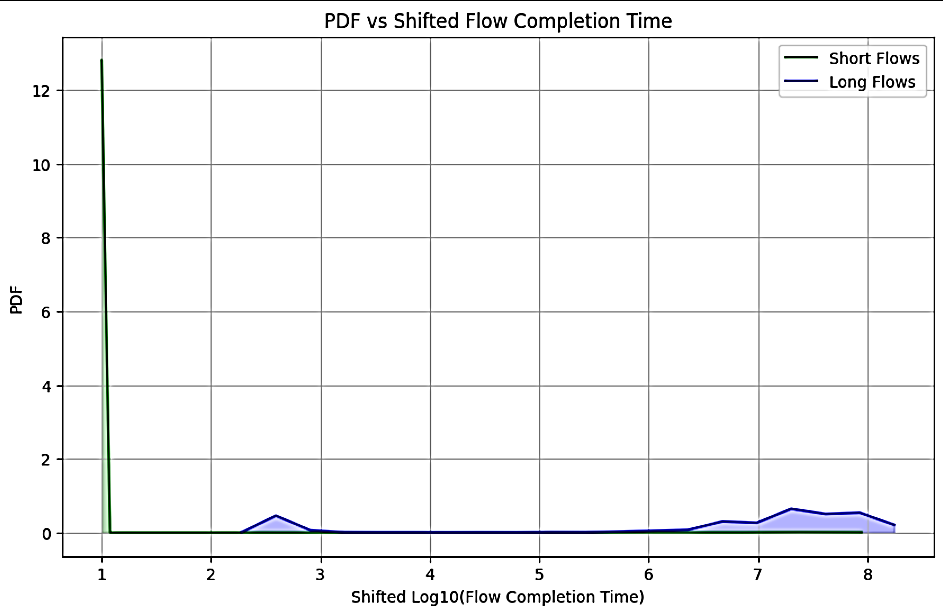
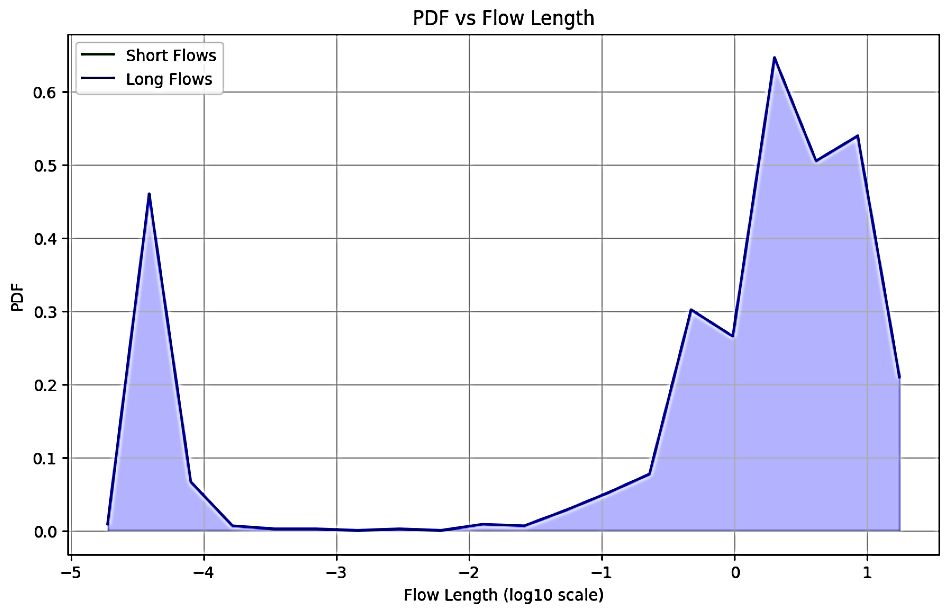
For the Implementation of Interaction Flow graph analysis framework, there are few phases we need to cover. The main two Phases are Graph Construction and Graph Preprocessing.

**Graph Construction:**

* **Flow Classification:**

The network flow, the flows will be classified as Short and Long flows depending upon the hyper parameters defined in ([22]). They classified Short and Long Flows based on the ‘Pkt\_Timeout’, ‘Judge\_Interval’, ‘FLOW\_LINE’. For the classification they have used Random Sample of encrypted network traffic of Jan 2020 (MAWI Dataset). Query packet information from the data layer's high-speed packet detection engine to get raw information. Packet characteristics such as (destination address, port number, protocol, length, inter-arrival time). These features can be extracted from encrypted text traffic for generic detection. **Figure 1(a)** Shows the Flow completion time and PDF computed on the data points so that we can visualize the short and long flows. We observe the short flows exceeds the percentage of long flows, having higher distribution.

**Figure 1(b)** Shows the Flow length distributions of the short and long flows. In Flow length, the percent of long flows are higher than short ones. We have scaled both the distribution on Log10 scale for better visualization.

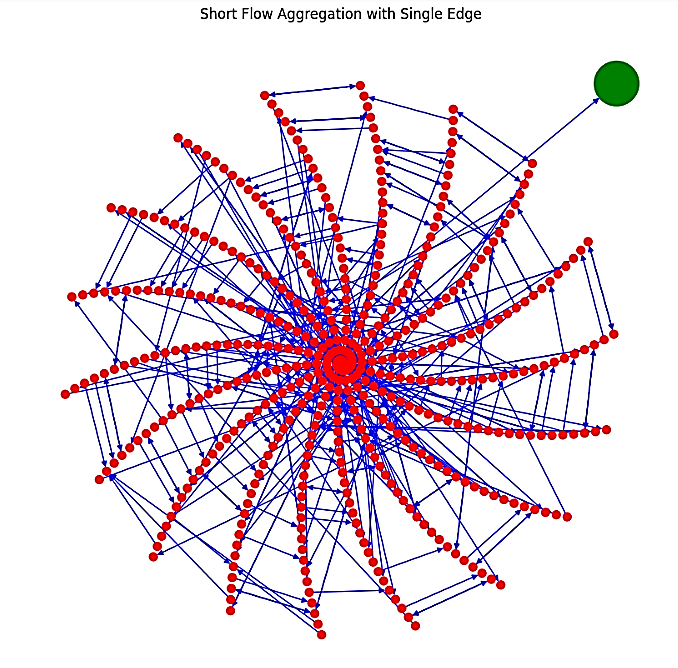
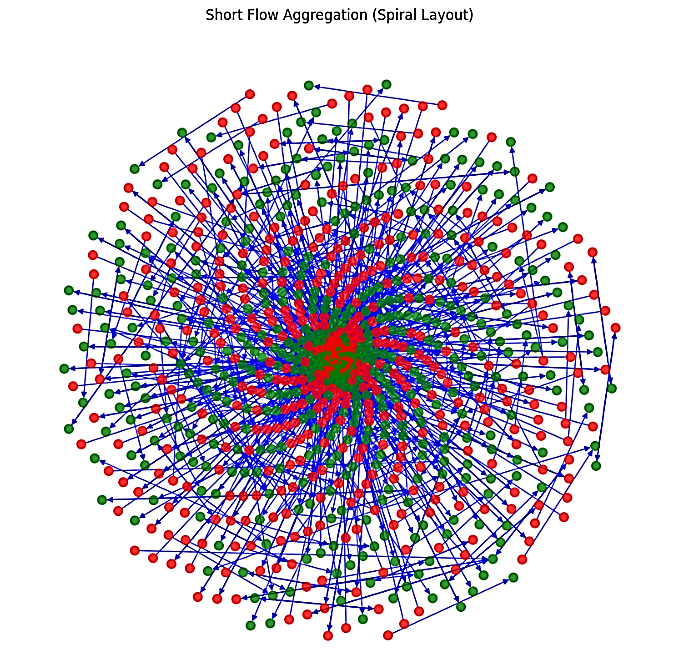
 

*Figure 1(b). Flow Length Distribution*

*Figure 1(a). FCT Distribution*

* **Short Flow Aggregation:**

After the flow classification, Short flow aggregation was performed so that we can reduce the density of the graph ([2]). By comparing the traditional flow graph as edges with our aggregated graph form the Real world backbone traffic dataset ([21]). The hyper parameter used in short flow aggregation will be AGG\_LINE (see details in ([22])) which ensure that the flows are repetitive enough. **Figure 2(a)** Describes the traditional flow as edges with the features like (Source, Destination). We saw that the density of the graph is much high as the edges are very tightly coupled and there is no clear separation of short and long flows. So In **Figure 2(b)** we aggregated short flows i.e. Green edges summed up for short flow aggregation. Along with PKT\_TIMEOUT and FLOWLINE we used AGG\_LINE metric so that we can ensure the aggregation of the short flows. By this way we effectively reduce the storage overhead and density of the graph. We observe that the graph reduces 99.80% of vertices and Edges as well.



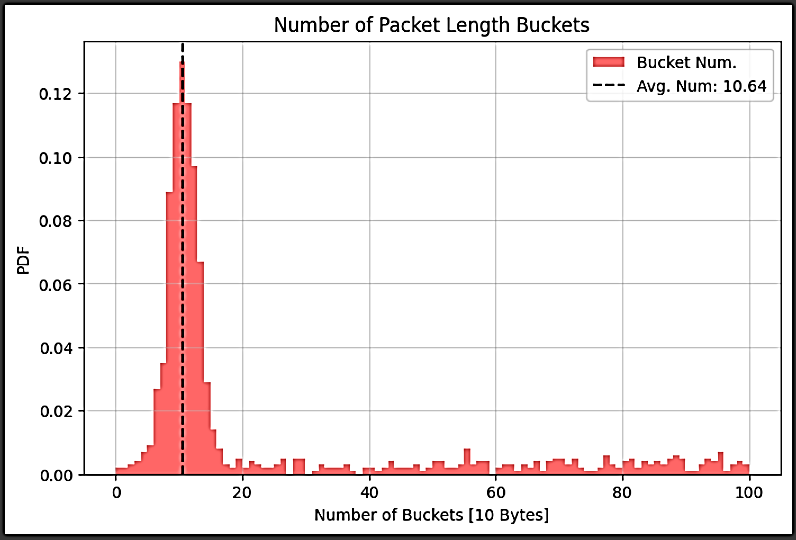
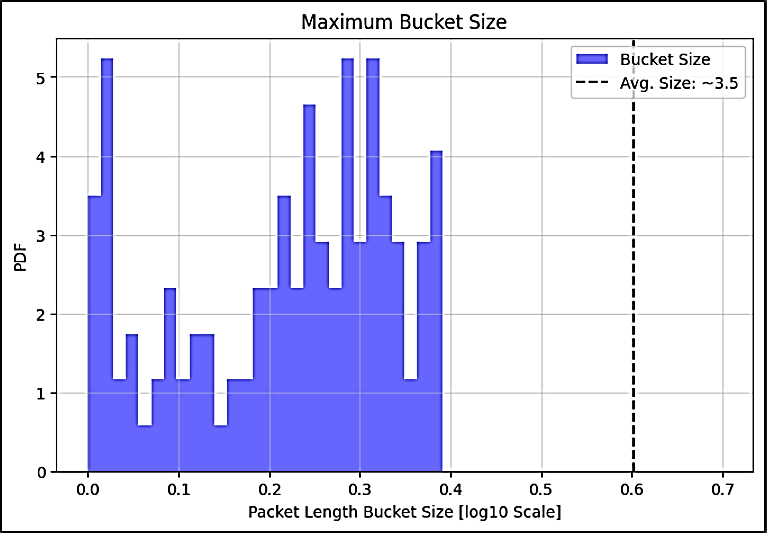
*Figure 2(b). Short Flow Aggregation*

*Figure 2(a). Traditional Flow Edges*

* **Feature Distribution Analysis for Long Flows:**

With the help of histogram we represent the per-packet distribution of long flow so that we preserve long per packet feature sequence since they are centrally distributed. Specifically we construct a hash table for each per packet feature sequence. We segregate the histogram with Number of packet Length buckets and Maximum bucket Size.

**Figure 3(a)** Describes the Number of buckets Calculate the hash code by dividing the packet characteristics by the bucket width and incrementing the indexed counter. Hash code. Finally, note down the hash code and its associated counters in the form of a histogram. For the problem of coarse grained flow it is insufficient to make malicious encrypted network traffic detection with small no. of packets, Also it arises the concern of plotting histogram which is not so good in this case. **Figure 3(b)** shows the maximum bucket size for long flows in the same data set. Most packets are in long flows have similar packet lengths and arrival interval. We have scaled the values of packet length and bucket size to Log10 for better visualization.

*Figure 3(a). Number of Packet Length Buckets*

*Figure 3(b). Maximum Bucket Size*

**Graph Preprocessing:**

* **Connectivity Analysis:**

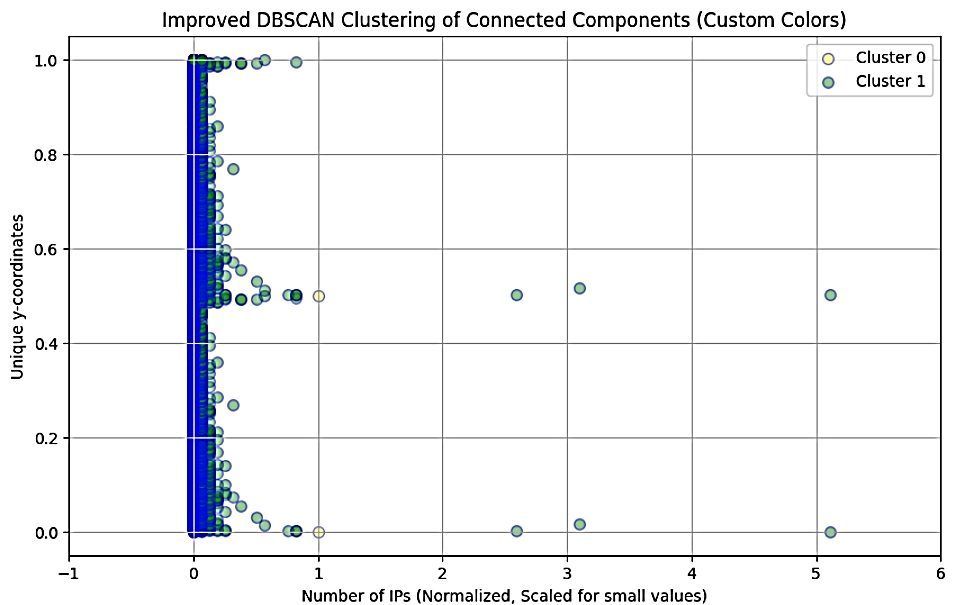
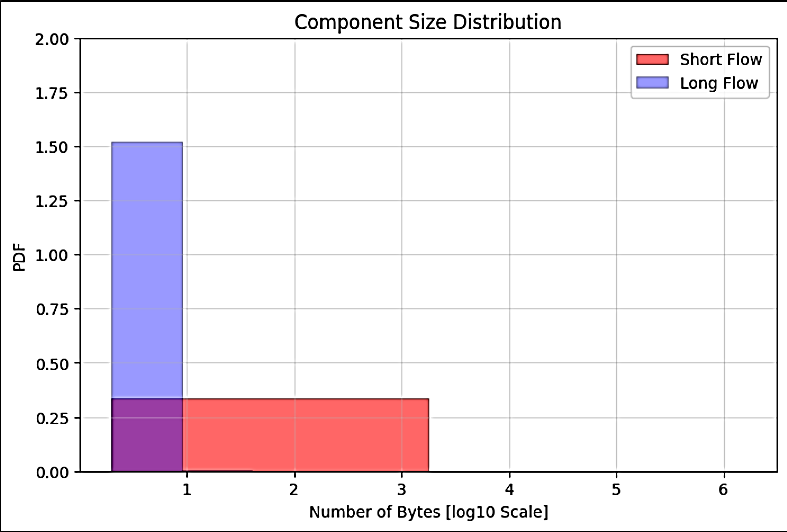
The Connectivity Analysis is being done for the real time graph based learning detection. We collected the connected components for Short and Long Edges and Plotted them in the Graph. It is been observed that few edges were carrying similar interaction patterns.

Therefore, we perform clustering based on high-level statistics of related components. **Figure 4(a)** defines the component size distribution for connected components of short and long flows. We can see that there are few amount of components that have similar interaction patterns and fee edges associated with them.

**Figure 4(b)** illustrates that the high level pre clustering performed on the similar edges of the components. The graph contains the most dense region between -1 to +1 range of scaling vector. We extracted the following features out to perform the pre clustering using Min Max normalization and acquire centers using Density based Clustering i.e. DBSCAN:

* Number of Short Flows
* Number of Long Flows
* Number of Edges denoting Short Flows
* Number of Edges denoting Long Flows

The Features we contain are (Source Address, Destination Address, Length and total packet count).



*Figure 4(b). PCA Decomposed Features*

*Figure 4(a). Component Size Distribution*

# **RESEARCH GAP**

Many supervised, unsupervised and deep learning models used to detect zero day by have limitations like labeled dataset, traditional approaches, relies on network patterns. As the paper cited, Traffic encryption is always used by attackers to encrypt their harmful actions. Since malicious traffic is encrypted, it can easily bypass traditional threads just like harmless threads detection. In particular, the existing encrypted traffic detection methods are supervised which rely on the prior knowledge of known attacks (e.g., labeled datasets). Detecting unknown encrypted malicious traffic, which does not require prior knowledge, is still an open problem ([2]).

# **PROBLEM STATEMENT**

The proposed interaction graph analysis heavily relies on interaction patterns between flows for predicting zero day attacks which is not a good approach as many patterns have abnormal behavior and are not certainly sophisticated. Also this is an open door to the type of attacks which mimic the normal traffic flow behavior.

# **PROPOSED MEHTODOLOGY**

To Cater the limitations of the flow interaction graph analysis, Our proposed approach is to use unsupervised Learning algorithms and train them with the Dataset and make the prediction results better from the graph analysis.

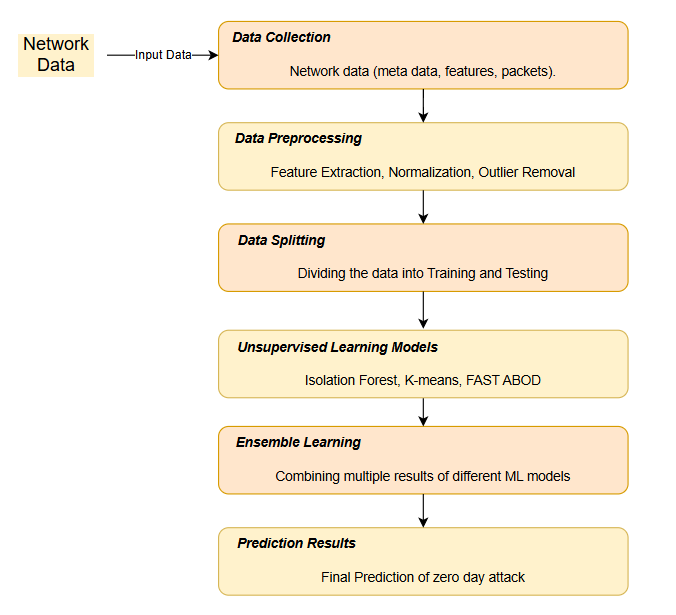
The framework can be described as, Firstly we will gather the dataset of the network traffic, then we will preprocess the dataset. It is an efficient way because it can remove outliers, normalize the data and extract important features out of it. Then we will pass the dataset without labeling to the Unsupervised ML model to detect and predict the unknown attacks. Then finally we compare the results of multiple Unsupervised ML models used so that we can detect and predict the zero day.

To concise the idea of the proposed framework we summed up the details in the annexure, i.e. **Table 3**, defines the objectives and tasks of each of the following phases of our proposed methodology. It defines the importance and usage of each phase along with its functionality:

|  |  |  |  |
| --- | --- | --- | --- |
| PHASES | OBJECTIVE | TASKS | OUTPUT |
| Data Collection | Gathering of the network packets from the dataset | Collecting the raw encrypted network traffic packets that serve as an Input for the process. | Network data including features and packet information |
| Data Preprocessing | Clean and transform the raw data | Extraction of the relevant features then remove the outlier components and scale all the features at the same level. | Relevant extracted features with outlier removal |
| Data Splitting | Divide the processed data into training and testing phase | Making of training set that used to train the model and testing set that used to test the model performance. | Training and Testing dataset |
| Unsupervised learning models | Use unsupervised models to identify anomalies | Detect the anomalies by isolating outliers, clustering the data points. | Prediction Results of multiple models |
| Ensemble learning | Combine the multiple results | Aggregation of the outputs based on techniques like voting or weighted average etc. | Values of the prediction results |
| Prediction Results | Final prediction results of zero day attacks | Made the single output from the multiple outputs | Single detection result. |

*Table 3 Details of Proposed Methodology.*

Below is a flow of our architectural diagram of our proposed methodology in **Figure 5.**



*Figure 5. Flow Architectural diagram*

# **PROJECT TIMELINE**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **TIMELINE** | | | | | | | | | | | | | | | | | |
| **Processes** | **Sep** | | **Oct** | | **Nov** | | **Dec** | | **Jan** | | **Feb** | | **Mar** | | **April** | | **May** | |
| Proposal submission and Thesis Defense |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Data Collection |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Data Preprocessing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Training of the Data on ML models |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Predicting The Unknown Attacks |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Comparison of different results |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

*Table 4 Project Timeline*

# **CONCLUSION**

Another paper ([3]) suggested that Supervised classifiers and deep learners are generally effective at identifying familiar attacks, but they struggle to detect unfamiliar attacks, such as new types of attacks, modified versions of existing exploits, or threats that the intrusion detector is not equipped to handle. So to Overcome these issue we have proposed an ensemble unsupervised learning that has these benefits. Like, An area of machine learning techniques called unsupervised learning uses unlabeled data to identify patterns. In order to identify zero-day assaults, the zero-day attack detection model aims to learn a compressed representation of typical data. Zero-day attack data is not needed for the outlier detection algorithms' training. Since the zero-day attack data is usually unavailable during the model training phase, this is a huge advantage ([1]). Apart from the existing literature we have studied, We have chosen to work with an existing study included as are baseline in the paper. They have used “Flow interaction graph based analysis for detecting unknown malicious network traffic”. According to them, existing methods cannot detect malicious traffic encrypted with unknown patterns by learning the traffic characteristics of a single flow. However, it is still possible to detect such attack traffic, because these attacks involve multiple attack stages with different flow interactions between attackers and victims, which are distinct from benign flow interaction patterns ([2]). The dataset is provided by the MAWI working Group of traffic Archive. Now from the network flow, the flows will be classified as Short and Long flows depending upon the hyper parameters defined. Query packet information from the data layer's high-speed packet detection engine to get raw information. Packet characteristics such as (destination address, port number, protocol, length, inter-arrival time). These features can be extracted from encrypted text traffic for generic detection. After the flow classification, Short flow aggregation was performed so that we can reduce the density of the graph ([2]). By comparing the traditional flow graph as edges with our aggregated graph form the Real world backbone traffic dataset ([21]). The hyper parameter used in short flow aggregation will be AGG\_LINE. With the help of histogram we represent the per-packet distribution of long flow so that we preserve long per packet feature sequence since they are centrally distributed. Specifically we construct a hash table for each per packet feature sequence. We segregate the histogram with Number of packet Length buckets and Maximum bucket Size. The Connectivity Analysis is being done for the real time graph based learning detection. We collected the connected components for Short and Long Edges and Plotted them in the Graph. It is been observed that few edges were carrying similar interaction patterns. The graph contains the most dense region between -1 to +1 range of scaling vector. We extracted the following features out to perform the pre clustering using Min Max normalization and acquire centers using Density based Clustering i.e. DBSCAN.

# **REFERENCES**

[1] Yang Guo, computer communications, A Review of Machine Learning-Based Zero-Day Attack Detection: Challenges and Future Directions , 2022.

[2] Chuanpu Fu , Qi Li , Senior Member, IEEE, and Ke Xu , Fellow, IEEE, Member, ACM, IEEE/ACM TRANSACTIONS ON NETWORKING, Flow Interaction Graph Analysis: Unknown Encrypted Malicious Traffic Detection 2024.

[3] Tommaso Zoppi∗ , Andrea Ceccarelli, Tommaso Puccetti, Andrea Bondavalli, Computers & Security, Which algorithm can detect unknown attacks? Comparison of supervised, unsupervised and meta-learning algorithms for intrusion detection, 2023.

[4] EVA PAPADOGIANNAKI, SOTIRIS IOANNIDIS, ACM Computing Surveys, A Survey on Encrypted Network Traffic Analysis Applications, Techniques, and Countermeasures, 2021.

[5] Bizhu Wang , Yan Sun , and Xiaodong Xu , Senior Member, IEEE, IEEE INTERNET OF THINGS JOURNAL, A Scalable and Energy-Efficient Anomaly Detection Scheme in Wireless SDN-Based mMTC Networks for IoT, 2021.

[6] Ruoyu Li , Qing Li , Member, IEEE, Jianer Zhou, and Yong Jiang , Member, IEEE, IEEE INTERNET OF THINGS JOURNAL, ADRIoT: An Edge-Assisted Anomaly Detection Framework Against IoT-Based Network Attacks, 2022.

[7] Ke He , Dan Dongseong Kim , and Muhammad Rizwan Asghar, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey, 2023.

[8] Sangjun Kim and Kyung-Joon Park , Applied Sciences, A Survey on Machine-Learning Based Security Design for Cyber-Physical Systems, 2021.

[9] Willian Tessaro Lunardi , Member, IEEE, Martin Andreoni Lopez , Member, IEEE, and Jean-Pierre Giacalone, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, ARCADE: Adversarially Regularized Convolutional Autoencoder for Network Anomaly Detection, 2023.

[10] Diego Madariaga , Graduate Student Member, IEEE, Javier Madariaga, Martín Panza, Javier Bustos-Jiménez , and Benjamin Bustos, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, Detecting Anomalies at a TLD Name Server Based on DNS Traffic Predictions, 2021.

[11] Ghada Elbez , Member, IEEE, Klara Nahrstedt , Fellow, IEEE, and Veit Hagenmeyer , Member, IEEE, IEEE TRANSACTIONS ON SMART GRID, Early Attack Detection for Securing GOOSE Network Traffic, 2024

[12] Qi Li , Senior Member, IEEE, Yunpeng Liu, Zhuotao Liu , Peng Zhang , and Chunhui Pang, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, Efficient Forwarding Anomaly Detection in Software-Defined Networks, 2021.

[13] Ramani Sagar, Rutvij Jhaveri and Carlos Borrego, Electronics, Applications in Security and Evasions in Machine Learning: A Survey , 2020.

[14] Nadia Niknami , Graduate Student Member, IEEE and Jie Wu , Fellow, IEEE, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, Entropy-KL-ML:Enhancing the Entropy-KL-Based Anomaly Detection on Software-Defined Networks, 2022.

[15] Hengrun Zhang , Member, IEEE, Kai Zeng , Member, IEEE, and Shuai Lin, Member, IEEE, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, Federated Graph Neural Network for Fast Anomaly Detection in Controller Area Networks, 2023.

[16] Ziming Zhao , Student Member, IEEE, IEEE/ACM TRANSACTIONS ON NETWORKING, FOSS: Towards Fine-Grained Unknown Class Detection Against the Open-Set Attack Spectrum With Variable Legitimate Traffic, 2024.

[17] Yating Liu, Student Member, IEEE, Yuantao Gu , Senior Member, IEEE, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, MSCA: An Unsupervised Anomaly Detection System for Network Security in Backbone Network, 2023.

[18] Zhangfa Wu , Huifang Li , Yekui Qian , Yi Hua , and Hongping Gan , Member, IEEE, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, Poison-Resilient Anomaly Detection: Mitigating Poisoning Attacks in Semi-Supervised Encrypted Traffic Anomaly Detection, 2024.

[19] Ying Zhong , Zhiliang Wang , Xingang Shi, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, RFG-HELAD: A Robust Fine-Grained Network Traffic Anomaly Detection Model Based on Heterogeneous Ensemble Learning, 2024.

[20] Rasheed Ahmad1  · Izzat Alsmadi2  · Wasim Alhamdani1  · Lo’ai Tawalbeh3, Zero‑day attack detection: a systematic literature review, 2023.

[21] WIDE. MAWI Working Group Traffic Archive. Accessed: May 2022. [Online]. Available: <http://mawi.wide.ad.jp/mawi/>

[22] Supplementary Material. Accessed [Online]. Available: <https://ieeexplore.ieee.org/ielx7/90/10640200/10474390/supp13370851.pdf?arnumber=10474390>