A Novel Approach for Zero-Day Attack Detection and Prevention

Dr. Jyoshna Bejjam[1] Rachit Rahul Das[2], Ankur Banerjee[3], Srujan Landeri[4], Mohammad Arshad Ali[5], Nikhil Guru Venkatesh[6]

Associate Professor, Keshav Memorial Institute Of Technology, Hyderabad, India
Student, Keshav Memorial Institute Of Technology, Hyderabad, India
Student, Keshav Memorial Institute Of Technology, Hyderabad, India
Student, Keshav Memorial Institute Of Technology, Hyderabad, India
Student, Keshav Memorial Institute Of Technology, Hyderabad, India
Student, Keshav Memorial Institute Of Technology, Hyderabad, India

Abstract

Zero-day network interruption assaults comprise a regular online protection danger, as they look to take advantage of the weaknesses of an organization framework. Zero-day attacks have always been a major contributor in data leaks which has led to loss of money, time, and resources. Our solution to detect and prevent such attacks is a desktop application that monitors your network traffic in real time and looks for any anomalies or malicious activity that may be happening and works to minimize the damage caused by prevention. Our Zero-day attack detection and prevention system is a software designed to protect the user's machine from malicious connections and stop it if any are attempted. Our software utilizes a network flow collection tool called CIC Flowmeter to collect network flows from the user in real time. These flows are analyzed using a two layer approach. Protection is employed by blocking network port access for specific IP Addresses that have been flagged as malicious.

Keywords: Zero day attacks, Attack detection, Machine Learning.

1.Introduction

In today's ever-changing world, networks and digital systems have proliferated to every path of life. It has revolutionized how we live and conduct business, research and polity. However, growing with the advantages of seamless connectivity is the threat of mounting cyber attacks. With a large portion of our public and private data being hosted on digital technologies, protecting computer systems and networks from malicious attacks has become a critical concern in the current digital era. Cybercriminals, leveraging the power of the internet, are constantly searching for vulnerabilities in hardware and software that can be exploited to steal or corrupt data. Within the rapidly expanding cyber security landscape, the complexity and scale of cyber attacks has grown exponentially. Amongst these attacks the most challenging to counter is the zero-day attack. A zero-day attack occurs when criminals exploit unknown vulnerabilities in hardware or software, unbeknownst to manufacturers. Akin to being caught unawares, this means that there are no defenses or safeguards to prevent these assaults. Zero-day attacks are particularly tricky to deal with, as they catch organizations off guard. This leaves developers with little time to identify and mitigate the damage. If undetected, this attack may inflict irreparable damage to essential system architecture. Cloud bleed, Stuxnet, Pegasus and WannaCry are all devastating examples of the effects of an unregulated zero-day attack on organizations and common day-to-day users. The inherent efficacy of zero-day exploits has driven multiple organizations to pool resources, knowledge and experience to study, record and mitigate zero-day attacks. One such initiative: Google's Project Zero, dedicated to analyzing zero-day attacks, regularly publishing their tracking records of publicly known zero-day vulnerabilities. As per 'Ponemon Institute', a prestigious research institution, the average cost incurred per zero-day attack is estimated to be around $1.2 Million. The conventional techniques, which are generally divided into two types: Signature-based Detection and Anomaly-based Detection, are used to combat and identify zero-day attacks. Signature-based detection is used in cybersecurity to identify and mitigate known threats, such as malware and attacks. It involves creating predefined signatures or patterns that represent known malicious activities. When incoming network traffic matches these signatures, it indicates the presence of an already known threat. While signature-based detection has its merits, it also has notable disadvantages. One of the major

drawbacks is its inability to detect zero-day attacks, which exploit previously unknown vulnerabilities. Since the signatures for zero-day attacks are not yet known or available, these attacks can go undetected. Additionally, maintaining an up-to-date signature library can be challenging and time-consuming. Furthermore, signature-based detection is susceptible to false negatives when encountering variations of known threats with altered signatures or polymorphic malware that changes its code structure. We can use anomaly-based and machine learning-based detection to overcome the limitations of signature-based detection methods to detect zero-day vulnerabilities. These methods can identify previously unknown threats by analyzing patterns and deviations from normal behavior, making it reliable to detect zero-day attacks with distinctive patterns.

Contrarily, machine learning-based detection can gain knowledge from enormous datasets, enabling models to analyze trends and raise detection accuracy with time unlike Signature-based Detection techniques. Machine learning-based detection provides advantages, but it also has drawbacks. As the samples of zero-day vulnerability are not available in the datasets. This method makes a crucial assumption: that zero-attacks are identical to or similar to the attacks that currently exist and are used to train the models. Further evaluating the models is difficult due to the unavailability of true zero-day attack samples which makes it difficult in testing and evaluating the performance of the models.

2. Literature Survey

In the realm of network security and zero-day attack detection, several notable approaches and techniques have emerged. The CICFlowMeter, a robust Python library, plays a crucial role in meticulous network traffic analysis, providing features for extracting flow-based attributes and real-time data processing[1]. Hindy et al. (2020) introduce a groundbreaking method utilizing deep learning techniques, demonstrating exceptional proficiency in identifying previously unknown attacks and significantly advancing network security[2]. Guo (2022) offers a comprehensive review of machine learning-based methods, not only addressing current challenges but also charting potential directions for future research endeavors[3]. Hairab et al. (2023) pioneer an anomaly detection approach for zero-day attacks, combining Convolutional Neural Networks (CNNs) with regularization techniques to showcase the potential of CNN-based models in bolstering security measures[4]. Rushdan et al. (2019) explore the dynamic landscape of Software-Defined Networks (SDNs) in the context of zero-day attack detection and prevention[5]. Vaisla's (2014) work provides valuable insights into the analysis and identification of zero-day attacks, contributing significantly to the understanding of these threats[6]. Comar et al. (2013) present a hybrid approach that melds supervised and unsupervised learning for zero-day malware detection, introducing a promising methodology that leverages the strengths of both learning paradigms[7]. Zoppi et al. (2021) showcase unsupervised algorithms designed to detect zero-day attacks, offering critical insights into the application of unsupervised learning in cybersecurity[8]. Holm (2014) investigates the efficacy of signature-based intrusion detection systems in zero-day attack identification, highlighting their robustness in the modern security landscape[9]. Sun et al. (2016) develop a probabilistic approach to identifying zero-day attack paths, contributing to a deeper understanding of attack vectors and pathways[10]. Serinelli et al. (2021) conduct a thorough analysis of open-source datasets, validating Intrusion Detection System (IDS) implementations for both well-known and zero-day attack detection[11]. Musca et al. (2013) propose a method centered around honeypots for detecting and analyzing zero-day attacks[12]. Wang et al. (2010) introduce the concept of k-zero day safety, providing a metric for quantifying network security risk against unknown attacks, offering a more quantitative approach to this challenge[13]. Kaur and Singh (2015) present a hybrid real-time system for zero-day attack detection and analysis, addressing the pressing need for timely identification of previously unknown threats[14]. Patidar and Khandelwal (2019) delve into the application of machine learning techniques for zero-day attack detection, demonstrating their effectiveness in enhancing cybersecurity measures[15].

3.Proposed Methodology

Our proposed work focuses on developing a robust threat engine capable of identifying and blocking zero-day attacks using a two-layer model approach. The core objective of our research is to analyze real-time network traffic and detect zero-day attacks using a network flow collection tool. Leveraging machine learning techniques, the threat engine will unveil potential threats and subsequently integrate with a firewall to enact prompt blocking measures against

the identified attacks. Our work aims to enhance cybersecurity measures by proactively detecting and neutralizing threats.

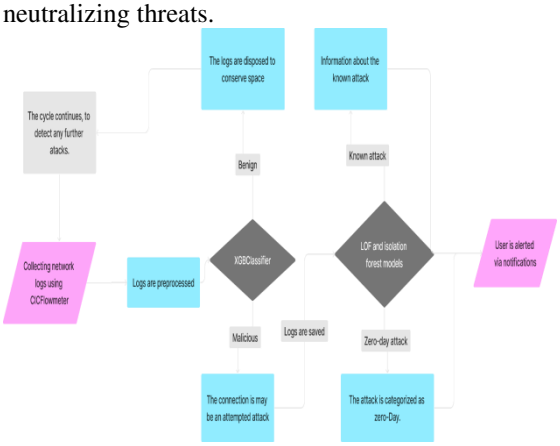


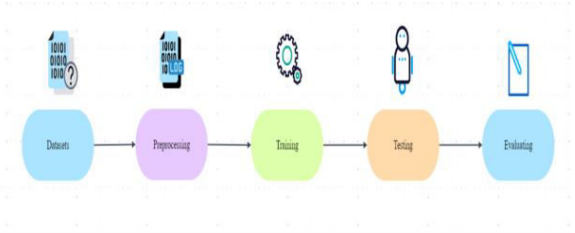
Fig1: Workflow of the project

In our two-layer model, we adopt a comprehensive approach to detect and classify network flows as benign or malicious. The model consists of two distinct layers, each serving a specific purpose in the detection process. The first layer: we employ a single classifier model that takes the network flows collected by CICFlowMeter as input. The classifier's primary task is to classify each flow as either benign or potentially malicious. This initial classification helps us segregate flows that require further scrutiny for probable security threats. Once the classifier identifies flows as potentially malicious, they are passed to the second layer, which comprises a series of sophisticated anomaly detection models.

These models are specialized in recognizing specific types of attacks and patterns associated with known malicious activities. During this phase, each flow is subjected to in-depth analysis. If a flow exhibits patterns that match those of a known attack, it is classified as an "inlier." The presence of inliers suggests that the model successfully identifies the pattern of a known attack, allowing for prompt subjugation. On the other hand, if a flow deviates significantly from recognized attack patterns, it is labeled as an "outlier." Outliers indicate flows with unidentified patterns, indicative of potential zero-day attacks.

Identifying zero-day attacks is crucial, as they exploit previously unknown vulnerabilities, posing significant risks to network security. Upon the detection of any malicious activity, a call to action is sent to the integrated firewall. It block incoming connections from the same source, preventing further compromise and safeguarding the user system from potential threats.

By adopting this two-layer model, we enhance the accuracy and effectiveness of our network flow analysis. The initial classifier efficiently narrows down the scope of flows that require detailed examination, while the anomaly detection models in



the second layer provide a robust mechanism to identify both known and previously unknown

Fig2: Phases of the project

malicious activities. The integration with the firewall ensures swift response and proactive protection, bolstering the overall cybersecurity measures for a safer and more resilient network environment.

3.1 Datasets

The training dataset used in this paper is CSE-CIC-IDS 2017 and 2018 Data Set. The two datasets contains six different intrusion types, Brute-force, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside, with a total of 14 different intrusions, namely, Botnet attack, FTP-BruteForce, SSH-BruteForce, BruteForce-Web, BruteForce-XSS, SQL Injection, DDoS-HOIC attack, DDoS-LOIC-UDP attack, DDoS-LOIC-HTTP attacks, Infiltration, DoS-Hulk attack, DoS-SlowHTTPTest attack, DoS-GoldenEye attack, and DoS-Slowloris attack. These two datasets were collected from servers at the Canadian institute of cybersecurity and posted on its website to be open source. The benign data is collected for a week from a typical research network. The traffic includes routine daily activities such as emailing, searching, news, video streaming, etc. All attacks and benign traffic are labeled and are used for training the detection models. The data set has 80 bi-directional flow features.

3.2 Preprocessing

During the preprocessing phase of our model, we tackle two crucial aspects: computational efficiency and maintaining accuracy while preserving correlations. We meticulously evaluated the impact of various scalers on the model's correctness by leveraging TSNE and UMAP visualizations at each preprocessing step. Our approach involved selecting different scalers for the two model layers.

For the first layer, we opted for the Quantile Transformer scaler. This choice was deliberate, as it efficiently transforms data into uniform or Gaussian distributions, proving beneficial for the initial classifier. On the other hand, we employed two distinct scalers for the second layer: the Power Transformer and the Min-Max Scaler. The Min-Max Scaler was chosen to ensure feature uniformity and data normalization within a specific range, which aids in anomaly detection in the second layer.

It's important to note that during this process, extensive data cleaning is performed. Records with null values were removed and distributions were analyzed for any outstanding outliers. The decision to remove such records was taken after careful evaluation of the proportion of these bad samples in the overall dataset. Our study concluded that the presence of these records was only minimal in number, and could be removed without effecting the efficiency of both layers.
Addressing skewed data distributions, the Power Transformer scaler not only normalized data but also made it more suitable for certain anomaly detection algorithms.

Post data transformation, we observed that some columns were populated with identical values in each record, primarily zero. Indicating that they lacked predictive importance. To streamline computational load during model training, we removed these columns, as well as other non-contributing attributes which are: protocol numbers, source and destination IP addresses, and port numbers. Following feature engineering, the number of features utilized reduced to 40 for layer 1 and 53 for layer 2.
The graphs provided illustrate the transformed data for both layer 1 and layer 2.

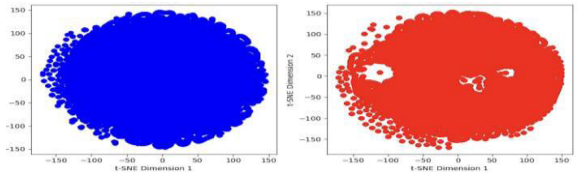


Fig 3:Dataset-1 distribution of benign connections
Dataset-2 - distribution of malicious connections

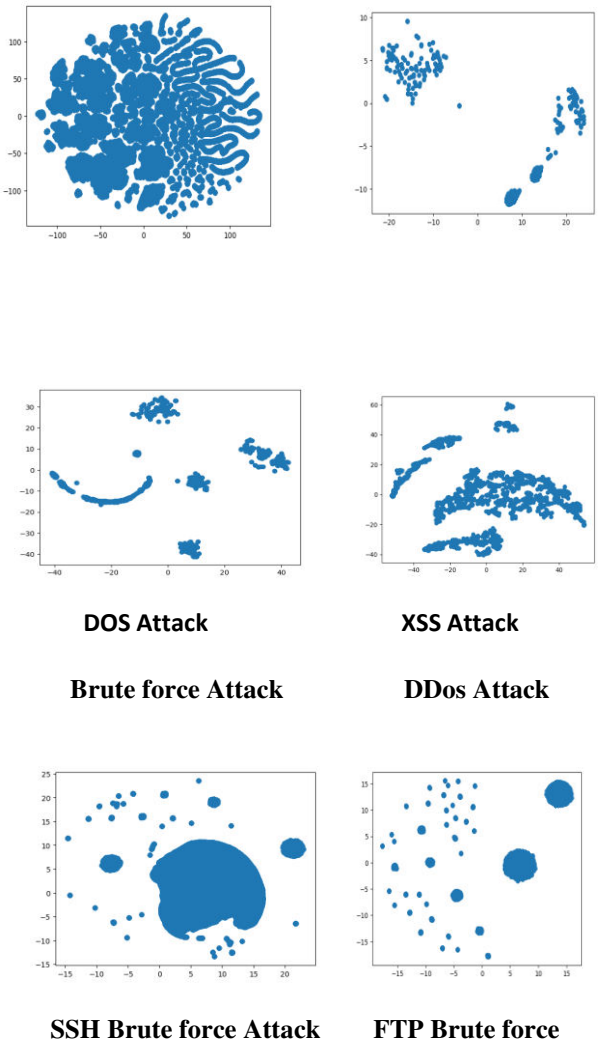


Fig 4: Various Types of Attacks

3.3 Training

In our research to find the best model for zero-day attack detection we experimented with various models such as Random forest, LightGBM and XGBClassifier whose subsequent explanations are given below. For the first layer it was required to select the best model among them. Our criteria for choosing the best model was accuracy. Among them XGBClassifier was chosen because of its superior accuracy.

First layer

3.3.1 Random Forest

Random Forest is a versatile and powerful ensemble machine learning algorithm used for both classification and regression tasks. It operates by constructing multiple decision trees during training and combining their predictions to make more accurate and robust forecasts. Each decision tree is built on a random subset of the training data and a random subset of features, reducing overfitting and improving generalization. Random Forest's ensemble approach ensures that it can handle complex relationships in data, handle missing values, and provide feature importance scores for variable selection. This algorithm is widely employed in various fields, from finance and healthcare to image recognition and natural language processing, due to its reliability and ability to deliver high-quality predictive models. The overall accuracy of this layer is evaluated to stand at around 80%..

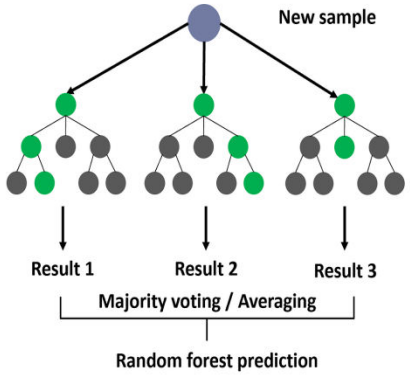


Fig : 5 Random Forest Prediction

3.3.2 Light Gradient Boosting

Light Gradient Boosting Algorithm (LightGBM) is a cutting-edge gradient boosting framework developed by Microsoft, known for its exceptional speed and efficiency in machine learning tasks. It employs innovative techniques such as leaf-wise tree growth and histogram-based data binning, resulting in faster training times and reduced memory usage. LightGBM can handle large datasets, offers built-in support for categorical features, and provides regularization options to prevent overfitting. Its ability to parallelize and distribute computations, along with GPU acceleration, makes it scalable for both small and large-scale machine learning projects. With its ease of integration through various programming language APIs, LightGBM has become a preferred choice in competitions and real-world applications, delivering high-quality predictive models swiftly and effectively. The resultant accuracy was around 70%.

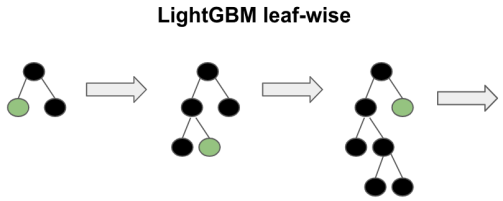


Fig: 6 Light Gradient Boosting

3.3.3 XGBoost

XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leafs that contains a continuous score. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

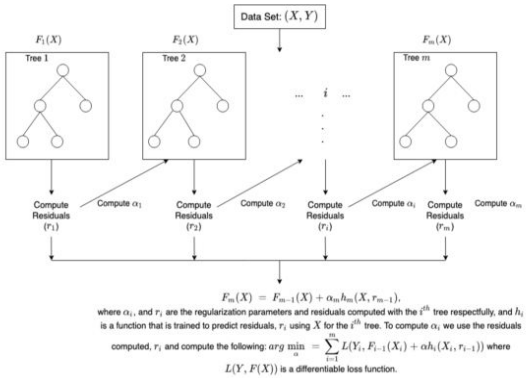


Fig 8

The initial layer harnesses the power of the XGBClassifier algorithm, renowned for its robust performance in classification tasks. XGBoost (Extreme Gradient Boosting) is an ensemble learning algorithm grounded in decision tree frameworks. It employs boosting techniques to combine the predictions of multiple weak learners (in our case, decision trees) into a strong predictive model. The algorithm iteratively refines its predictions by adjusting the weights of misclassified instances at each step. This iterative approach enables the model to focus progressively on challenging-to-predict instances.

We have thoroughly assessed a range of models, including Decision Trees, Random Forest, Stacker Model, Logistic Regression, LGBM, CatBoost, and the XGBClassifier, to ascertain the optimal choice. After a comprehensive evaluation, the XGBClassifier stood out, excelling in accuracy and minimizing false positives, thus emerging as our prime selection.

This classifier is trained on a dataset comprising 4 million records, with around 2.7 million representing attacks and nearly 1 million representing benign connections. Achieving a balance between these two classes was the result of careful experimentation, where various configurations were tested on the XGBClassifier model. The configuration yielding the highest accuracy was chosen.

Noteworthy hyperparameters, such as the learning rate (set at 0.1), maximum depth of the trees (max_depth=8), and the number of boosting rounds (n_estimators=450), were meticulously selected. These parameters wield considerable influence over the model's performance and convergence. The learning rate controls the step size during each iteration, max_depth determines the tree's maximum depth, and n_estimators defines the total number of boosting rounds.

The accuracy on real attack data was 82%.

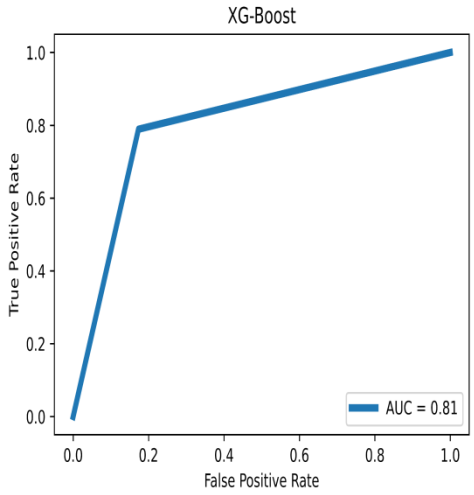


Fig:9 The roc curve .

3.4 Second layer:

The subsequent layer, dedicated to anomaly detection models, underwent meticulous testing using two widely recognized outlier detection techniques: the Local Outlier Factor (LOF) and the Isolation Forest. Both models were trained on a subset of 2.7 million attack records.

The Local Outlier Factor (LOF) assesses the local density deviation of a data point with respect to its neighbors. It flags data points with significantly lower density as outliers. On the other hand, the Isolation Forest operates by creating a series of isolation trees. It isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature. The number of splits required to isolate a data point is indicative of its anomaly status.

To train these models, the dataset was initially segregated into various attack types. Each model was then individually trained on a specific attack type. For these models, we employed the Min-Max Scaler and the Power Transformer. The Min-Max Scaler ensures that the data is uniformly distributed within a specific range, thereby aiding in effective isolation of anomalies. The Power Transformer, on the other hand, addresses skewed data distributions and enhances the suitability of data for anomaly detection algorithms.

In the Isolation Forest, we set the hyperparameters as follows: n_estimators=100, max_samples='auto',

contamination=0.1, and max_features=1.0. For the LOF model, the parameters were chosen as n_neighbors=20, contamination=0.1, and novelty=True. These values were carefully selected after experimentation to ensure optimal performance and sensitivity.

The transformation techniques employed here played a pivotal role in achieving remarkably accurate outlier predictions. This comprehensive approach enabled our model to effectively identify anomalies, contributing to its robust performance in real-world scenarios.

3.5 Testing

The first layer was tested with previously unseen data derived from the CIC IDS-2018 dataset. The second layer was tested on data that it was trained on to derive the accuracy.

3.5.1 Layer 1

Model	Accuracy
XGBClassifier	97.22%

3.5.2 Layer 2

Model	Attack	Accuracy	Overall Accuracy
Local Outlier Factor	DoS Hulk Bot XSS DDoS GoldenEye DoS slowloris SSH Bruteforce Brute force	83.77% 95.90% 99.34% 98.16% 99.91% 89.35% 96.46%	90.58%

3.6 Evaluating

Evaluation was done by collecting data using CICFlowmeter. 20000 rows of benign data and 13000 rows of attack data was collected in real time to test these models.

Model	Benign(accuracy)	Malicious (accuracy)
XGBClassifier	80%	80%
Local outlier factor	99%	99%
Isolation forest	99%	99%

3.7 Prevention

When we identify attacks, it's crucial to halt them without delay. Our method involves swiftly denying access to IP addresses on the particular ports where the attacks are pinpointed by the creation of custom rules. This process utilizes the built-in pfctl firewall on MacOs. Whenever an attack is identified, a new rule is created and we put a halt to incoming and outgoing traffic from that port and IP address indefinitely. The block remains in place until the user decides to lift it. The process of blocking is automated using AppleScript. The automation works by dynamically creating new rules every time an attack is encountered and these rules are updated automatically to the firewall.

3.7.1 Windows

The Microsoft Defender Firewall, often referred to as the Windows Firewall, is a formidable and user-friendly cybersecurity tool developed to efficiently filter incoming and outgoing network connections. This firewall empowers users with the ability to selectively block access from specific IP addresses or completely halt data transmission through designated network ports. The firewall's functionality is realized through the creation of "Rules," which can be established either via the intuitive built-in graphical user interface (GUI) application or through manual configuration utilizing shell scripts or system commands.

Of particular significance is the application's intrinsic capability to identify and mitigate potentially malicious network connections on specific ports. This is seamlessly accomplished through the integration of an embedded shell script, which dynamically formulates new firewall rules in response to the detection of hazardous connections.

4.Results and Discussion

Real time data was considered for testing our model and the accuracy for the best model for the first layer was XGBClassifier with 80% accuracy and the best models for the second layer were Local Outlier Factor and Isolation Forest with accuracies 99% for both respectively.

4.1 Application Overview

This section gives an overview of The developed threat detection and mitigation system, a sophisticated desktop application created to strengthen the network security.. The application smoothly integrates front-end and back-end components using the renowned Electron framework to offer a strong defensive mechanism against harmful network activities.

4.2 Framework and Implementation:

The front-end part of the Application is expertly designed and makes use of three web technologies: HTML, CSS, and JavaScript. End users are given the ability to dynamically track and closely examine real-time logs that are derived from ongoing packet transmissions within the network interface. These logs are cleverly preprocessed by the program, preparing them for analysis in the back-end portion. The strength of sophisticated, a flow is a detailed summary of all packets collected by a network interface during a specified duration of time. Flows contain data pertaining to source and destination IP addresses, port numbers, protocol types, packet sizes, and communication durations.

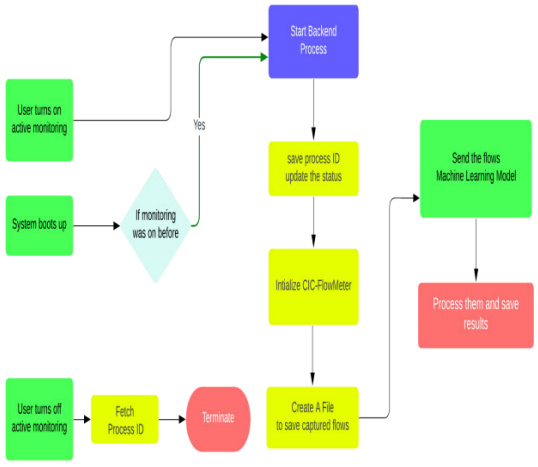


Fig10: Use case of the application

CICFlowmeter processes this captured data to extract essential features that can be further used for analysis. It relies on two critical dependencies: Scapy and Npcap. Scapy is a powerful Python library that facilitates packer manipulation, generation, and Machine Learning Models, applied inside the boundaries of two different layers, is the foundation of the latter component. The log collection is done by CICFlowmeter,which is an open-source network flow collection tool developed to assist in the monitoring of networks by the Canadian Institute for Cybersecurity. It provides an essential kit of tools that can be used to build powerful and robust Cybersecurity applications. The tool's primary function is its ability to analyze large-scale network flow data that is received through a network interface. Network flow data is the information that is collected during interactions between different devices and applications on the same network. To be more precise decoding at a low level. Npcap (libcap for macOS) is a packet capture library that enables network interfaces to capture live traffic from network interfaces in real time.

4.3 Working of the Application:

After receiving the logs, the system's first layer carefully examines their contents and uses Machine Learning algorithms to predict whether they are benign or malicious. This result then serves as the foundation for the following layer, in which it applies a more in-depth examination to the prediction.

The application demonstrates its strong nature when a malicious action is discovered. The system effectively thwarts the danger by blocking both the source IP address and the port through which the incursion happened by immediately activating the built-in firewall mechanism. The end-user is proactively informed about the discovered harmful activity at the same time. This communication includes details on the threat in addition to a measurable confidence score that provides more context for the prediction's accuracy.

Additionally, the application stores thorough recordings of every discovered malicious behavior on the user's local storage to aid with painstaking post-incident analysis. These data provide in-depth information about the attack, enabling a complex assessment of the dangers encountered.

The platform also accepts user-generated logs, enhancing its functionality and allowing its Machine Learning models to predict suspicious activity. The user's proactive network security management is aided by the predictive analysis performed on these logs, which were gathered using CIC-Flowmeter.

Socket implementation ensures real-time synchronization and interactivity by orchestrating the symbiotic communication between the front-end and

back-end components. This extensive, multi-layered design effectively illustrates how cutting-edge technologies work together to provide a strong, effective protection mechanism against changing cyberthreats.

Limitations:

- The first layer model will only work with data derived from the CICFlowmeter tool.

References

1 Hindy H, Atkinson R, Tachtatzis C, Colin JN, Bayne E, Bellekens X. Utilising deep learning techniques for effective zero-day attack detection. Electronics. 2020;9(10):1684.
2 Guo Y. A review of Machine Learning-based zero-day attack detection: challenges and future directions. Comput Commun. 2022.
3 Ibrahim Hairab B, Aslan HK, Elsayed MS, Jurcut AD, Azer MA. Anomaly detection of zero-day attacks based on CNN and regularization techniques. Electronics. 2023;12(3):573.
4 Huthifh Al-Rushdan, Mohammad Shurman, Sharhabeel H. Alnabelsi, Qutaibah Althebyan. Zero-Day Attack Detection and Prevention in Software-Defined Networks
5 Vaisla Dr. K. Analyzing of zero day attack and its identification techniques. Proceedings of the first international conference on advances in computing & communication engineering (ICACCE-2014). 2014;1:11-3.
6 Comar PM, Liu L, Saha S, Tan P-N, Nucci A. Combining supervised and unsupervised learning for zero-day malware detection. Proceedings of the IEEE Infocom, Turin, Italy. *2013*;2013:2022-30.
7 Zoppi T, Ceccarelli A, Bondavalli A. Unsupervised algorithms to detect zero-day attacks: strategy and application. IEEE Access. 2021;9:90603-15.
8 Holm H. Signature based intrusion detection for zero-day attacks: (not) A closed chapter? 47th Hawaii International Conference *on* System Sciences, Waikoloa, HI, USA, 2014; *2014*. p. 4895-904.
9 Sun X, Dai J, Liu P, Singhal A, Yen J. Towards probabilistic identification of zero-day attack paths IEEE Conference on Communications and Network Security (CNS), Philadelphia, USA, 2016; *2016*. p. 64-72.
10 Blaise A, Bouet M, Conan V, Secci S. Detection of zero-day attacks: an unsupervised port-based approach,Computer Networks.
11 Blaise A, Bouet M, Conan V, Secci S. Detection of zero-day attacks: an unsuperised port-based approach. Comput Netw. 2020;180:107391.
12 Serinelli BM, Collen A, Nijdam NA. On the analysis of open source datasets: validating IDS implementation for well-known and zero day attack detection. Procedia Comput Sci. 2021;191:192-9.
13 Musca C, Mirica E, Deaconescu R. Detecting and analyzing zero-day attacks using honeypots 19th International Conference on Control Systems and Computer Science, Bucharest, Romania, 2013; *2013*. p. 543-8
14 Wang L, Jajodia S, Singhal A, Noel S. k-zero day safety: measuring the security risk of networks against unknown attacks. In: Computer security–ESORICS 2010. Proceedings of the 15: 15th European Symposium on Research in Computer Security, Athens, Greece, Sep 20-22, 2010. Berlin, Heidelberg: Springer; 2010. p. 573-87.
15 Kaur R, Singh M. A hybrid real-time zero-day attack detection and analysis system. IJCNIS. 2015;7(9):19-31.
16 Patidar P, Khandelwal H. Zero-day attack detection using machine learning techniques. Int J Res Anal Rev. 2019;6(1):1364-7.

Rachit Rahul Das
I am currently a Computer Science student at Keshav Memorial institute of technology. I aspire to continue my academic pursuits and contribute significantly to the field of computer science, Machine learning and cyber security. I am eager to collaborate with experts in the field, broaden my horizons, and further my educational journey through advanced studies.
Email: rachit1031@gmail.com

Landeri Srujan
A dedicated undergraduate studying Computer Science Engineering with Artificial Intelligence and Machine learning major at Keshav memorial institute of technology, Hyderabad. I am passionate about learning and research, excited to collaborate with industry experts to expand my knowledge and contribute to my field.
Email: srujanlanderi@gmail.com

Mohammad Arshad Ali
I am an enthusiastic student currently pursuing a Bachelor in Computer Science at Keshav Memorial institute of Technology, With a keen interest in Artificial intelligence, Cybersecurity, and emerging technologies. I embarked on this research project to expand my knowledge and contribute to the field.
Email: arshadali02177@gmail.com

Nikhil Guru Venkatesh
I was born in August of 2003 and have always held a high regard for science and computers. I'm currently pursuing a dual degree in Computer Sciences from the Keshav Memorial Institute of Technology with a minor in Artifical Intelligence and Machine Learning and Data Science from the Indian Institute of Technology Madras. The idea of machines being capable of doing autonomous tasks have held me hostage for some time, and it gives me great joy to be able to collaborate with other like minds in this field. I have a great interest in Machine Learning and it's possibilities and never shy away from the prospect of new challenges and research opportunities. It would be a privilege to collaborate with experts in the field and learn more from them on my journey.
Email: nikhilalloverthe world@gmail.com

Ankur Banerjee
I am a computer science student at Keshav Memorial Institute of Technology, and I am enthusiastic about AI and ML. I have also developed a keen interest in cybersecurity. This project has served as a means to gain more knowledge about this domain while simultaneously creating something that might be useful in our day-to-day lives.
Email: ankur0402official@gmail.com

Dr. B. Jyoshna Bejjam, Associate Professor in Department of Computer Science and Engineering at Keshav Memorial Institute of Technology Hyderabad. Having 20 Years of Teaching Experience. Published various International and National journals, Interested to do research area in Data Security and Cyber Security, implemented various projects. I am currently involved in AI projects and working with AI tools such as Llama2 and BERT Email: drjyoshnabejjam@gmail.com