Edge-Resident Intrusion Detection System (IDoS): Leveraging AI for Anomaly Detection in IoT Devices

[1] R. Saranya, [2] Dr. R. Saminathan, [3] Dr. N. Palanivel

[1] Research Scholar, Department of Computer Science
and Engineering
Annamalai University
Annamalainagar
Chidambaram - 608002
[2] Associate Professor, Department of Computer Science and Engineering
Annamalai University
Annamalainagar
Chidambaram - 608002
[3] Associate Professor, Department of Computer Science and Engineering
Manakula Vinayagar Institute of Technology
Puducherry

Email: [1] saranya.itmec@gmail.com, [2] samiaucse@yahoo.com, [3] npalani76@gmail.com

Abstract –This research presents a novel way to improve the security of Internet of Things (IoT) devices using an Artificial Intelligence (AI)-powered Edge-Resident Intrusion Detection System (IDoS). The suggested approach addresses latency, bandwidth, and scalability issues by bringing anomaly detection capabilities closer to the source of IoT data through the use of edge computing. By comparing normal and aberrant patterns in real-time device behaviour, the Edge-Resident IDoS uses sophisticated machine learning algorithms. This technology optimises bandwidth and minimises reliance on centralised cloud solutions by being deployed directly on edge devices or local servers. It also reduces latency. A proactive approach to security is ensured by AI-driven anomaly detection, which sees possible attacks before they escalate. The system's effectiveness in promoting safe and effective IoT installations is highlighted by discussing the practical ramifications, which include effects on compliance, energy efficiency, and reliability. The Edge-Resident IoT ecosystem, which contributes to robust and responsive IoT ecosystems, is a revolutionary step towards safeguarding connected settings through the convergence of edge computing and AI.

Key words – IoT security, Anomaly detection, Edge computing

I. Introduction

The interconnection of the Internet of Things (IoT) has revolutionized households, businesses, and industries, becoming an integral part of modern life. IoT gadgets are creating an abundance of data and making it harder to distinguish between the real and virtual worlds. Examples of these devices include autonomous vehicles, commercial sensors, smart thermostats, and connected appliances. Nevertheless, this revolution has also opened up new avenues for cybercrime, as attackers target weak points in order to steal confidential information, interfere with vital infrastructure, or even inflict bodily harm.

The proliferation of IoT devices has revolutionized our interactions with the outside world by enabling previously unheard-of levels of automation and connectedness in many areas of everyday life. IoT technology is now widely used in everything from wearables and smart homes to industrial sensors and smart cities. But the advantages of this networked environment are accompanied by growing worries about IoT device security. The quantity and variety of IoT devices have increased exponentially as a result of their broad use. These gadgets are used in many different fields and sectors, such as energy, transportation, healthcare, and agriculture. Cybercriminals have a vast attack surface due to the sheer number and diversity of devices, which makes it difficult to keep an eye on and safeguard each access point. The market for IoT security is expected to be valued at USD 20.9 billion in 2023 and grow at a compound annual growth rate (CAGR) of 23.1% from 2023 to USD 59.2 billion by 2028, on the basis of recent research reports [1].

Robust security elements are frequently overlooked in favour of utility, cost, and time-to-market in the design of many IoT devices. Manufacturers can put more emphasis on connectivity and user experience than on putting in place robust security measures. IoT devices may therefore have obsolete firmware, weak passwords, and no encryption, leaving them open to hacking [29-30]. The lack of industry-wide security guidelines for Internet of Things devices leads to disparate security procedures amongst manufacturers. In contrast to well-established sectors that have clear security guidelines, the Internet of Things ecosystem lacks a unified structure. Vulnerabilities go unchecked because of the difficulty in enforcing and maintaining uniform security procedures due to the absence of standardization.

Sensitive data, such as user behavior patterns and personal information, is frequently collected and transmitted by IoT devices. There could be serious privacy consequences if this data is compromised. IoT device vulnerabilities could be used by cybercriminals to obtain sensitive data without authorization, which could result in identity theft, financial fraud, or other nefarious acts. Numerous Internet of Things devices depend on embedded software and firmware, which could be vulnerable. It's possible that manufacturers won't give fixing these issues top priority or offer frequent updates and patches. Attackers may use outdated firmware and software to take over devices or obtain private information.

IoT devices with security flaws can create botnets, which can perform massive distributed denial-of-service (DDoS) attacks, disrupting infrastructure and services. Manufacturers, consumers, and legislators must collaborate to address the growing vulnerability of IoT devices. Strong authentication procedures, frequent security audits, secure communication frameworks, and rigorous security standards are essential. Prioritizing cybersecurity is crucial for maintaining the resilience, integrity, and privacy of connected ecosystems as the IoT develops.

Traditional intrusion detection systems (IDS) are effective in network security by monitoring traffic patterns for malicious activity [28]. However, they face limitations when implemented on IoT devices with limited resources, such as computational power, energy usage, bandwidth, and privacy issues. Sophisticated algorithms can negatively impact IoT devices' performance, especially those running on batteries or energy harvesting systems. Energy efficiency is crucial for these devices, as constant monitoring and analysis can be energy-intensive, making them unsuitable for continuous operation without frequent battery changes.

IoT devices often operate on constrained network bandwidth, which can be impacted by the amount of data generated and transmitted by traditional intrusion detection systems (IDS). This can lead to higher latency and bottlenecks, affecting the responsiveness of IoT apps. Traditional IDS may struggle to scale to monitor and analyze traffic from multiple devices, making it difficult to adapt to the various proprietary communication protocols in the Internet of Things ecosystem [33]. Additionally, IoT networks are dynamic, making it difficult to accurately detect and address security risks. Conventional intrusion detection systems often analyze network traffic, which may contain sensitive data, raising concerns about privacy. Balancing user privacy and effective intrusion detection is crucial in IoT deployments. Updating software on IoT devices can be challenging due to limited update capabilities. Conventional intrusion detection systems may require frequent upgrades to handle growing threats. Developing specialized solutions for resource-constrained IoT contexts is necessary to improve security, prioritize efficiency, low energy consumption, scalability, flexibility, and consideration of dynamic IoT installations.

A potential remedy for the security issues resource-constrained Internet of Things (IoT) devices confront is Edge-Resident Intrusion Detection Systems (IDoS) [2]-[3]. Edge-Resident Intrusion Detection Systems (IDoS) are installed directly on the edge devices inside an IoT network, in contrast to traditional Intrusion Detection Systems (IDS), which are usually centralised and run on servers or network gateways. There are a number of benefits to this deliberate positioning at the edge, nearer the points of data generation, processing, and transmission, including real-time anomaly detection, decreased latency, enhanced scalability, and resource efficiency [4] [27].

The security monitoring and detection features of Edge-Resident IDoS are dispersed throughout the edge devices in the network. By ensuring that security procedures are dispersed throughout the IoT ecosystem, this decentralized method helps mitigate the scalability issues associated with centralized IDS. IDoS reduces the need to send massive volumes of data to centralized servers for analysis by identifying and thwarting security risks at the edge. This lowers network latency, increasing IoT applications' responsiveness and raising system efficiency as a whole.

Because IoT devices have limited resources, including memory, computing power, and energy, Edge-Resident IDoS are made to be resource-efficient. These systems' lightweight design enables them to function well on devices with limited resources without noticeably sacrificing performance. When Edge-Resident IDoS detects an intrusion, it can act locally and swiftly. It is beneficial to use this localized threat response strategy to isolate compromised devices, stop threats from spreading, and preserve the integrity of the IoT network as a whole.

One can customize Edge-Resident IDoS to comprehend and examine the particular communication protocols frequently utilized in Internet of Things settings. This flexibility is essential for spotting and countering security risks in the dynamic and varied Internet of Things ecosystem. Because Edge-Resident IDoS eliminates the need to send sensitive data to centralized servers for analysis, privacy can be enhanced. Sensitive data can be monitored and threats identified locally, reducing the amount of time it is exposed to outside parties.

Because they can function offline, some Edge-Resident IDoS are appropriate for Internet of Things devices that might not always have connectivity. Even when a device is not actively connected to the central network, these technologies are still capable of identifying and addressing security concerns. Edge-Resident IDoS are in line with the recently developed edge computing concept, which involves processing and analysis closer to the source of the data. In Internet of Things contexts, this alignment improves the synergy between security methods and the distributed nature of edge computing.

A Comprehensive analysis of the unique features and needs of Internet of Things devices and networks is necessary for the implementation of Edge-Resident IDoS. Edge-Resident IDoS stands out as a possible option to solve the particular security concerns given by the proliferation of resource-constrained devices in various IoT deployments as the IoT environment continues to expand.

The main contributions of the research work are as follows

1. Edge-Resident Architecture: Focusing on local processing on IoT devices to reduce latency and maximise resource utilisation, this network edge-based architecture for intrusion detection is introduced.

2. Effective Pre-processing of Data: Offers innovative techniques to obtain and prepare data generated by the Internet of Things, tackling the difficulties posed by a variety of communication protocols and resource constraints.

3. Adaptive Response Mechanisms: Based on threat intensity and context, adaptive response mechanisms combine automatic responses, manual intervention, or a combination of the two to dynamically respond to recognised anomalies.

4. Contextual Anomaly Analysis: Increases the accuracy of the system in differentiating between normal and anomalous activity by including contextual information unique to IoT contexts.

The remaining parts of the paper are divided into the following sections: Section II explains the background details of edge computing, the challenges of implementing AI on edge devices and anomaly detection techniques; Section III describes the existing recent research works, Section IV describes the proposed Edge-Resident Intrusion Detection System (IDoS) and its functionalities; and Section V showcases the experimental setup and results. Finally, at the end, the paper is concluded.

II. Background Study

A. Edge computing

Edge computing is a distributed computing paradigm that relies less on centralized cloud servers and instead moves computation and data storage closer to the "edge" of the network, where data is generated. Data is transmitted to a centralized data centre for processing and analysis in traditional cloud computing [5]-[6]. But with edge computing, these operations are carried out nearer the data source, frequently on edge servers or local devices. The following are the elements of edge computing:

1. Edge Devices: These are the data-generating devices located at the network's edge. IoT devices such as cameras, sensors, and others are examples.

2. Edge Servers: These are computing units that are situated nearer to the edge devices and have local data processing and analysis capabilities.
3. Edge Gateway: It controls data flow and establishes a secure connection by acting as a bridge between edge devices and the cloud.

In addition to providing benefits in terms of latency reduction, bandwidth optimization, increased reliability, enhanced privacy and security, scalability, cost savings, and real-time analytics, edge computing brings processing and analysis capabilities closer to the point of data generation. This is especially advantageous for Internet of Things networks with a high number of interconnected devices. It cuts down on the amount of time data must travel from its source to the processing unit. This is critical for real-time processing applications like augmented reality, industrial automation, and driverless cars. Edge computing minimizes the need to send massive volumes of raw data to the cloud by processing it locally, which improves bandwidth efficiency and lessens network congestion.

By enabling devices to continue operating even in the event of a disruption in the connection to the central cloud, edge computing improves the dependability of Internet of Things systems. Independent completion of crucial activities is ensured via local processing. By keeping sensitive data closer to its source through local processing at the edge, the danger of data breaches during transmission to the cloud is decreased. It also makes it possible to apply security measures at the edge and respond to security risks more quickly. It permits scalable and distributed designs. The network can extend horizontally by adding more edge servers as the number of edge devices rises, meeting the expanding needs of Internet of Things applications. Through on-site data processing and less reliance on cloud connections, enterprises can save on bandwidth expenses and optimize resource use. Real-time data analysis at the place of origin is made possible by edge computing, which eliminates the waiting times involved in transferring data to a centralized location and allows for quicker decision-making.

B. Challenges of implementing AI on edge devices

Due to the constrained processing resources on edge devices, implementing AI presents a number of difficulties [7]. These are a few of the main obstacles:

1. Limited Processing capacity: In comparison to cloud servers or high-performance computing systems, edge devices—such as sensors, cameras, and Internet of Things (IoT) devices—frequently have less processing capacity. The intricacy and sophistication of AI models that can be used on these devices may be limited by this restriction.
2. Memory Restrictions: Since Edge devices usually have a small amount of memory, the size of AI models that can be installed on them is limited. The capacity to store and handle huge datasets may also be impacted by memory limitations.
3. Energy Consumption: Batteries or restricted power sources are used to power a large number of edge devices. Complex AI model execution can be computationally costly and may result in higher energy consumption, which could affect the device's overall sustainability and battery life.
4. Heat Dissipation: Because edge devices frequently lack advanced cooling mechanisms, heat might be produced when AI algorithms are performed intensively. The device's longevity and performance may be impacted by overheating, which may result in hardware problems.
5. Communication Bandwidth: Due to constrained communication bandwidth, it may not be feasible to transfer massive volumes of data between edge devices and centralized servers. For AI systems that need a lot of training data or regular updates, this can be a big problem.
6. Security Concerns: It's possible that edge devices can't implement strong security measures. This may increase their vulnerability to intrusions and jeopardise the privacy and confidentiality of the AI models and data.
7. Model Size and Complexity: Modern AI models with a lot of parameters, such as deep neural networks, can be computationally intensive. Model quantization, optimization strategies, or the usage of lightweight models that adhere to the limitations of the device may be necessary for the deployment of such models on edge devices.

8. Update and Maintenance Challenges: It might be difficult to update AI models on edge devices, particularly if they are placed in inaccessible or remote areas. It can be logistically difficult to keep models updated with security patches or new enhancements.
9. Edge device heterogeneity: Because edge devices are available in a variety of shapes and sizes, the environment is heterogeneous. It's challenging to create AI models that function well on various edge device types with differing capabilities.

Researchers and engineers have been working hard to provide specialized hardware, optimize algorithms, and create lightweight AI models to overcome these issues and enable effective AI processing on edge devices. Successful AI implementations on edge devices depend on striking a balance between the trade-off between model complexity and resource limitations [32].

C. Anomaly detection techniques anomaly detection techniques

An essential part of protecting Internet of Things (IoT) devices is anomaly detection, which aids in spotting unusual activity that could point to a malfunction or security risk [8]. For anomaly detection in the context of IoT devices, various methods are used. Selecting the right anomaly detection technique is crucial, and it depends on the IoT environment's features, the type of data being used, and the security specifications. A variety of techniques can frequently be applied to improve anomaly detection's overall efficacy in protecting IoT devices. Strong tools are introduced to the field of anomaly detection in IoT devices using machine learning techniques. By learning patterns from data and efficiently spotting anomalies, these algorithms can adapt to a variety of dynamic IoT contexts and improve the security and dependability of IoT systems. Table 1 offers a more thorough summary of anomaly detection methods, along with their benefits, drawbacks, and use cases.

Table.1 Summary of anomaly detection techniques

Anomaly Detection Techniques	**Description**	**Use Cases**	**Advantages**	**Challenges**
Statistical Methods	1. Mean and Standard Deviation: Basic statistical measurements used to identify anomalies in past data. 2. Z-Score: This tool uses standard deviations from the mean to identify anomalies.	Monitoring resources and time series analysis.	1. Straightforward and effectively computed; 2. Results that are simple to understand.	Limited ability to understand intricate patterns.
Machine Learning-Based Methods	1. Unsupervised Learning: Clustering (k-means), Dimensionality Reduction (PCA). 2. Isolation Forests: Randomly isolates anomalies using decision trees. 3. One-Class SVM: Trained on normal data to identify anomalies.	Industrial IoT, fraud detection, and network security.	1. The capacity to identify new and intricate patterns. 2. Capability to adjust to changing surroundings.	The need for enough labelled data in order to use supervised techniques.
Deep Learning Approache	1. Autoencoders: Compressed representation learning using neural	IoT image and speech recognition, predictive	Capability to identify complex patterns in data. appropriate for	Computationally demanding; strong hardware might be

s	network architecture. 2. Effective for sequential IoT data are Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM).	maintenance.	data that is consecutive.	needed.
Rule-Based Methods	Threshold-Based Rules: When values surpass predetermined thresholds, anomalies are identified. Expert Systems: Rule-based programs that identify anomalies by using subject expertise.	Quality assurance and environmental monitoring.	1. Openness in the defining of rules. 2. Including in-depth domain expertise.	Difficulties in establishing precise thresholds.
Network-Based Detection	Behavioural analysis: keeping an eye out for unanticipated links in communication patterns. Intrusion Detection Systems (IDS): Modified for the Internet of Things to identify unauthorised activity in network traffic.	Network surveillance and cybersecurity in smart cities.	1. Anomalies in the network are detected in real time. 2. Precise insight into network activity.	Potential for false positives.
Fuzzy Logic-Based Approaches	Fuzzy Clustering: More adaptable rule formulations for anomaly identification through the use of fuzzy logic.	Monitoring the environment and diagnosing illnesses.	1. Dealing with data uncertainty. 2. Latitude in defining rules.	A rise in implementation complexity.
Ensemble Methods	Combining Multiple Techniques: Increasing accuracy and resilience by integrating the results of various procedures.	Identification of anomalies in various IoT contexts.	1. Enhanced productivity all around. 2. Robustness against flaws in individual models.	A rise in the complexity of computation.
Time Series Analysis	Time-consuming and requires knowledge of the field.RNN with LSTM: Time-series data analysis for anomaly identification that captures temporal dependencies.	Forecasting energy use and doing predictive maintenance.	Useful for analysing data in a sequential manner.	computational complexity and overfitting risk.
Ensemble Methods (Continued)	Combining Multiple Models: Adding forecasts together to increase overall robustness and accuracy.	Identification of anomalies in various IoT contexts.	Enhanced productivity all around.	Arise in the complexity of computation.
Online Learning	Adaptive models are those that are updated continuously for changing environments depending on incoming input.	Adaptive systems with real-time anomaly detection.	Adaptability to shifting trends.	Model drift potential necessitates ongoing observation and

				updating.
Transfer Learning	Knowledge transfer: Applying lessons learned from trained models to enhance anomaly detection within particular domains.	Transferable knowledge and cross-domain anomaly detection.	effective application of prior information.	Difficulties in locating appropriate source domains and guaranteeing interoperability.
Feature Selection and Dimensionality Reduction	PCA: Cutting down on dimensions to pinpoint key characteristics for enhanced anomaly detection effectiveness.	Huge datasets and environments with limited resources.	Increased effectiveness of the model.	Diminished dimensionality results in information loss.
Hybrid Approaches	Combining Rule-Based and ML Techniques: Adding heuristics and domain knowledge to rule-based systems through the integration of machine learning.	Industrial Internet of Things, monitoring health.	Incorporates expertise unique to a given topic.	The difficulty of creating and sustaining hybrid systems.
Edge Computing	Models are deployed on edge devices, allowing for real-time anomaly detection without the need for centralised servers.	Real-time IoT applications and edge analytics.	Decreased latency.	Restricted computing power on peripheral devices.

III. Related Work

In a number of ways, edge computing was suggested as a way to improve the features and dependability of conventional Internet of Things applications [9]. The edge nodes can be used by the IoT application to handle management, storage, and computing duties. Reducing latency, managing the network in real-time, and improving data management are a few of the anticipated quality improvements. Security apps like an IDS might likewise be "migrated" to the edge in this situation. This change might help an intrusion detection system (IDS) by giving it access to greater processing power, which would allow it to run more sophisticated algorithms, as well as more storage capacity, which would allow it to store system logs for later analysis or to do memory-intensive operations. For real-time Internet of Things applications, an edge node may also be able to provide lower latency than the cloud. Furthermore, an edge-deployed intrusion detection system should be IoT-agnostic, which means it shouldn't rely on any particular IoT communication technology. When one of these IDSs is employed, several heterogeneous devices utilising various communication methods can be handled by it in a unified way, negating the need to implement an IoT-specific IDS for each subnetwork of devices.

The capacity to identify low-level threats created at the device level gives an IoT-specific IDS an advantage over an IoT-agnostic one. However, instead of deploying an IoT-specific IDS for each communication technology that is available, a single IoTagnostic IDS may handle a large number of IoT devices. Numerous IoT-specific IDSs have been developed for Bluetooth LoW Energy [10], Wi-Fi [11]-[12], LoRa [13], ZigBee [14], and LoRa [15]. Usually, expert systems record host-to-host communication and verify that each packet complies with network standards particular to the technology involved. Additionally, sophisticated systems are able to identify physical network layer (PHY) threats like jamming. An attacker executing a PHY assault typically transmits bits that do not adhere to the communication protocol, making it impossible for an external IDS to read the data and making the attack very challenging to identify.

A system based on the Artificial Immune System (AIS) was proposed by authors in [16]. The IDS is dispersed among cloud nodes, edge nodes, and Internet of Things devices. Lightweight detectors are installed on IoT devices. Alerts are examined and handled on the edge with the help of Smart Data ideas. Ultimately, the cloud groups the information and educates the sensors. In this approach, the cloud is used for the model training of the heavy-weight detectors, while IoT devices are solely used for their lightweight applications.

A method for identifying Sybil and jamming attacks by assessing the consistency of network characteristics across time was put forth by Verzegnassi et al. [17]. Network characteristics from IoT devices, such as average packet rate and signal quality, are passively gathered by the system and viewed by the gateway. The algorithm's output is the device network parameters' conformance value over time. A sudden shift in conformance values may be a sign that an assault is underway.

The authors of [18] suggest an anomaly detection system called Hypervisor Detector for the hypervisor layer. To increase the detection system's accuracy, they combine a hybrid approach—a fuzzy C-Means clustering method and an artificial neural network—with an FCM-ANN. The suggested method is put into practice and contrasted with the Classic ANN algorithm and the Naïve Bayes classifier. Experiments are conducted using the 1999 KDD Cup dataset from DARPA. The suggested approach outperforms the Naïve Bayes classifier and the Classic ANN in terms of anomaly detection accuracy and false alarm rate, even for infrequent attacks, according to a thorough theoretical and performance investigation.

An enhanced method that balances energy usage and detection accuracy has been proposed by Sedjelmaci et al. [19]. Their solution consists of an anomaly-based intrusion detection system (IDS), which uses more power to function but produces a more accurate analysis, and a signature-based IDS, which is more energy-efficient but may produce a high number of false positives. Only the signature-based intrusion detection system is operational. The anomaly-based IDS receives elevated alerts and has the authority to either confirm or reject them. Additionally, the system is designed as a security game, model in which the Nash Equilibrium is used by the anomaly-based IDS to carry out its predictions. One disadvantage is that the system's proper operation depends on the cloud's continuous operation. Techniques for detecting anomalies can be applied not just to identify network breaches but also to identify firmware problems in devices or departures from a system's typical state. Detection techniques for these anomalies have been presented in the context of Industrial IoT (IIoT).

Researchers in [20] have looked into how to identify dangerous edge devices. In fact, edge devices have the advantage of being able to store and handle data generated by thousands or even hundreds of IoT devices. An attacker may be able to access the data sent by connected IoT devices if they manage to take over such an edge node. The authors suggested an architecture that makes use of a Virtual Honeypot Device (VHD), an anomaly-based intrusion detection system, and a two-stage Markov Model. The two-stage Markov Model receives an alert that is raised by the IDS. The initial phase classifies the particular fog node, while the subsequent stage makes a prediction regarding the appropriate attachment of the VHD to the edge node that triggered the alert. Experts can later examine the logs of all connected edge nodes that are stored in the VHD.

On edge nodes, Schneible et al. [21] et al. presented a framework for carrying out distributed anomaly detection. The mechanism involves the deployment of Auto Encoder models across many edge nodes situated in disparate network locations. The traditional Auto Encoder method is used to detect anomalies. Additionally, the system demonstrates some adaptivity: as it is being deployed, the edge nodes update their models in response to fresh observations, spotting novel patterns in network traffic. Subsequently, a central authority receives the revised model from an edge node, compiles it, and distributes it to the remaining edge agents. Because only the models of the Auto Encoders are carried in the generated network traffic rather than all of the observed data, the scientists found that this strategy lowers the overhead bandwidth. In order to lessen traffic between edge nodes and the central authority, auto encoders were used in this context to detect anomalies and an automatic method to extract features compressing observed data. Even though edge nodes are more capable of processing than IoT devices, they lack the resources necessary to complete labor-intensive tasks like training big machine learning models.

It has been suggested to use a Multilayer Perceptron (MLP) model to create a vector space representation for a lightweight IDS [22]. The Australian Defence Force Academy Windows Dataset (ADFA-WD) and Australian Defence Force Academy Linux Dataset (ADFA-LD), two new generation system calls datasets containing exploits and attacks on a variety of applications, were used to test the described IDS. The simulation demonstrates that we can obtain 94% Accuracy, 95% Recall, and 92% F1-Measure in ADFA-LD and

74% Accuracy, 74% Recall, and 74% F1-Measure in ADFA-WD using a single hidden layer and a limited number of nodes. A Raspberry Pi is used to assess performance.

Passban IDS is a system designed by Eskandari et al. [23] that can provide an additional layer of security to directly connected Internet of Things devices. The system targets TCP/IP-based assaults, excluding those that rely on IoT technologies, like port scanning, HTTP and SSH brute force attacks, and SYN flooding. The system can be implemented on inexpensive edge devices and/or Internet of Things gateways, like the Raspberry Pi or its equivalent, and does not require complex computations. Despite the IDS's goal of shielding devices from comparatively few threats, the system has excellent accuracy and a very low false positive rate. One thing the system has going for it is that it is one of the few IDSs that is completely functional, from the web user interface-based warning system to the detection algorithm.

A single intrusion detection system (IDS) operating on the network perimeter is unable to watch, record, and analyse every event [26], according to Niedermaier et al. [24]. They suggested a distributed IDS built around a number of edge devices used by IIoT agents and a central unit that aggregates the logs they generate. Fundamentally, the IDS employs one-class classification methods for anomaly detection; the authors make the assumption that the agents are capable of learning the system's typical behaviour. Since the IDS doesn't require a lot of computation, low-power microcontrollers can operate it well. In addition, the authors have created a proof-of-concept implementation of the system, which is uncommon in publications of a similar nature.

A method for anomaly identification on power grid sensor readings was developed by Utomo et al. [25]. Anomaly warnings could be utilised to protect grid security by averting failures and blackouts, in addition to serving as a signal of unauthorised access. Because of the high degree of non-linearity in the readings, an artificial neural network (ANN) built using Long-Short Term Memory (LSTM) cells is employed to detect anomalies. Recurrent neural networks (RNNs), a form of artificial neural network architecture that is particularly good at processing data in sequence, such as a sequence of words in the field of natural language processing (NLP) or a succession of sensor inputs, are the family of neural networks that includes LSTM neural networks.

A system to carry out anomaly detection at the network edge gateways was proposed by Hafeez et al. [31]. The system solely relies on TCP/IP properties that the edge can observe, and it depicts the traffic with features that are independent of IoT communication technologies. This method has the benefit of allowing several systems with different IoT communication methods to be connected to a single IDS. Regarding the dataset, it is a collection of IoT data gathered from an actual test-bed. Additionally, they looked at the distribution of the several features that were taken into consideration, and they found that a heavy-tailed Gaussian suited well to most of the features. Fuzzy clustering is used to carry out the ultimate anomaly detection. They have achieved a minimal false positive rate and excellent accuracy on their bespoke dataset.

IV. Edge-Resident Intrusion Detection System (IDoS) Architecture

By monitoring and analysing network traffic at the network's edge, usually near the data source or destination, an Edge-Resident Intrusion Detection System (IDoS) can be deployed. Its placement offers the system a first line of defence against different types of cyberattacks by enabling it to identify and react to possible security threats instantly. The overall architecture is depicted in fig.1. Here's a summary of an Edge-Resident IDoS's general architecture:

Data Acquisition and Pre-processing Module

In order to gather and become ready for additional analysis, the Data Acquisition and Pre-processing module of an Edge-Resident Intrusion Detection System (IDoS) is essential. This module's main goals are to collect data in real-time, normalise and standardise it, reduce noise, aggregate it, and prepare it for feature extraction. This module collects raw data from several sources, including host-based and network sensors, then processes it to extract pertinent information and minimise noise. In order to collect raw network traffic data, network sensors are positioned strategically at the network's edge. Payload data, metadata, and packet headers are all included in this. To keep an eye on incoming and outgoing traffic, network sensors can be installed at switches, routers, or gateways. Application logs, file integrity, and system activity data are all gathered via host-based sensors that are installed on servers or individual devices. The behaviour of specific systems is shown by this data.

Network and host-based sensors provide the raw data. This data could contain timestamps, source and destination IP addresses, protocol usage, and communication trends. Host-based sensors could offer information on file access, system calls, and other pertinent activity. The formats and architectures of raw data gathered from various sensors can differ. To guarantee consistency, data normalisation entails standardising the format. Comparing and analysing data from many sources is made simpler by this step. Multiprotocol communication is common in network traffic. Identification and classification of the communication protocols are facilitated by protocol analysis. Deciphering the behaviour of network communication requires an understanding of the protocols. From the normalised data, pertinent features are located and extracted as part of the feature extraction process. Features are distinct qualities or properties that offer important data for intrusion detection. Unusual network patterns, unexpected system calls, and the quantity of unsuccessful login attempts are a few examples. Payload analysis for network data is looking through data packets to see what's within. Patterns or signatures connected to known assaults may surface from this investigation. Payload analysis is, nevertheless, done sparingly to strike a compromise between privacy concerns and the requirement for security.

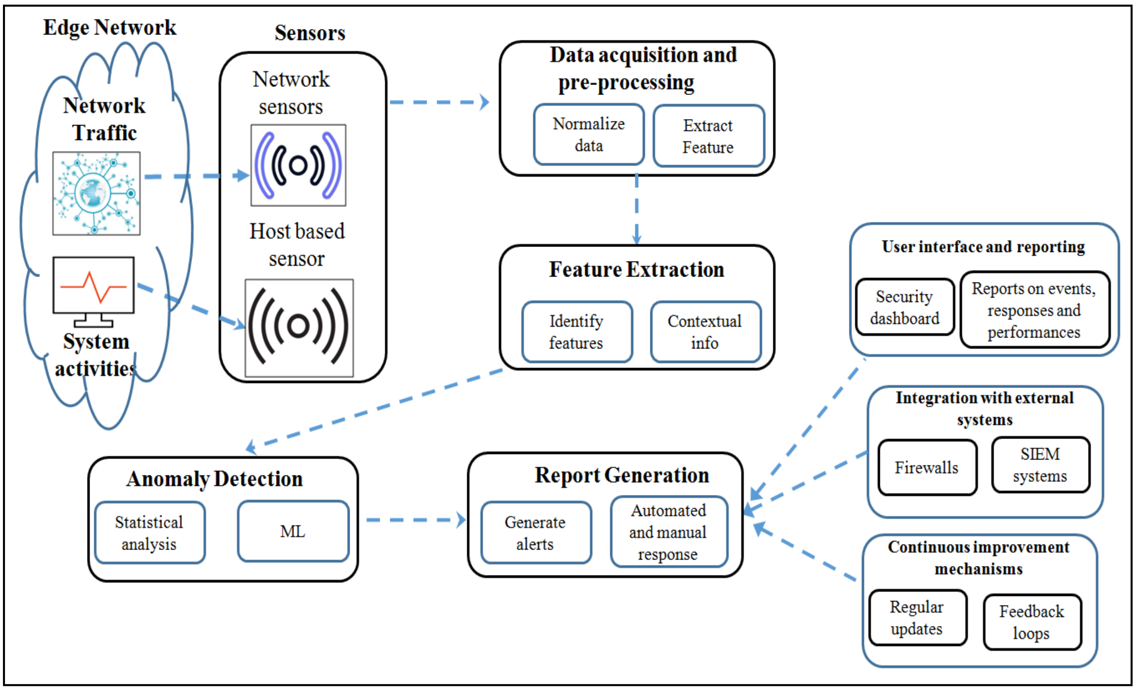

Fig.1. Overall architecture of the proposed system

To reduce noise and concentrate on important information, data reduction techniques are used. Less data means less resources are used and more efficiency is achieved in later stages of analysis. Accurate event correlation requires maintaining consistent timestamps across many data sources. A synchronised perspective of events is produced by timestamp normalisation, which aligns timestamps from several sensors.

Combining data from various sources into a single, cohesive dataset is known as data aggregation. This can be especially helpful for analysing multi-component coordinated attacks or for establishing connections between events that occur in various network segments. Pre-processed data is organised into a structure that may be examined further. To facilitate easy input of the data into the intrusion detection algorithms, this may entail arranging the data into tables, matrices, or other data structures.

Table 2. Data Acquisition and Pre-Processing Parameters

Parameter	**Description**
Data Sources	Network traffic, System logs and Endpoint activities.
Real-Time Collection	Yes.
Normalization and Standardization	Format standardization, value normalization.
Noise Reduction	Filtering irrelevant or redundant data.
Data Aggregation	Combine data from various sources.
Feature Extraction Preparation	Identify relevant attributes for analysis.
Packet Sniffing	Capture and analyze network packets.
Log Parsing	Extract relevant information from logs.
Endpoint Monitoring	Monitor activities on individual devices.
Data Compression	Reduce data volume through compression.
Timestamp Synchronization	Ensure accurate timestamps.

The main variables and responsibilities related to the Data Acquisition and Pre-Processing Module are listed on this table 2. It draws attention to the variety of data sources, the real-time nature of data collecting, and the range of methods used to prepare the data for efficient intrusion detection. The module's functions, like noise reduction and feature extraction preparation, are essential for preparing the data for the IDoS's later analysis. The feature extraction step of the Edge-Resident IDoS, which includes anomaly detection, signature-based detection, and decision-making procedures, uses the output of the Data Acquisition and Pre-processing module as input. This module is essential to guaranteeing that the information utilised for intrusion detection is precise, consistent, and pertinent to the system's security objectives.

Feature Extraction Module

An Edge-Resident Intrusion Detection System (IDoS) relies heavily on its Feature Extraction Module to distinguish between potential security threats and patterns of regular behaviour. The objective of this module is to locate and extract pertinent characteristics from pre-processed data, converting unprocessed data into a format that can be used with anomaly detection methods. Features could include things like the frequency of a certain protocol in network traffic, the length of a connection, or patterns of system behaviour like the quantity of unsuccessful login attempts. By removing redundant or less informative features, the module seeks to minimise dimensionality and improve the effectiveness of further analysis.

Additionally, to simplify the dataset, the Feature Extraction Module uses dimensionality reduction methods like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Compatibility with selected anomaly detection technologies, including machine learning models and statistical techniques, is guaranteed by feature transformation. To improve the system's ability to discern between regular and aberrant behaviour, the module may additionally use contextual feature engineering, which integrates extra data like time of day or past behaviour. This flexibility is essential because it enables the system to dynamically modify the feature set in real-time, guaranteeing resilience against shifting network circumstances and new threats.

The Feature Extraction Module is essentially a crucial bridge between unprocessed data and the subsequent anomaly detection procedure, which greatly enhances the precision and responsiveness of an Edge-Resident IDoS. With the help of efficient contextual feature engineering and its ongoing role updating and modifying the feature set, the system is better equipped to remain watchful in the face of constantly changing cyber security threats at the network edge.

Anomaly Detection Module

An essential part of an Edge-Resident Intrusion Detection System (IDoS) is the Anomaly Detection Module, which is meant to spot departures from established patterns of typical behaviour in system operations and network traffic. Through the use of statistical analysis, machine learning models, and signature-based identification, the module keeps an eye out for any odd patterns or outliers in real-time data streams. By creating models of typical behaviour from past data, statistical analysis allows the system to identify anomalies when observed patterns depart noticeably from predefined baselines. The module's capabilities are further enhanced by machine learning algorithms, which offer a dynamic and adaptive approach to anomaly identification by learning and reacting to changing patterns.

Additionally, signature-based detection is incorporated into the Anomaly Detection Module by comparing observed patterns to a database of known attack signatures. By using predetermined patterns, this method enables the system to quickly identify and react to recognised threats. The module also uses static and dynamic thresholding algorithms to detect when observed behaviour exceeds predefined limitations. By examining variations in respect to pre-established baselines and spotting connections between various features, behavioural analysis and feature correlation enhance the precision of anomaly identification. Monitoring in real-time guarantees that anomalies are discovered quickly, and contextual data—like the time of day and user roles—further improves the system's capacity to discriminate between benign and perhaps malevolent activity.

To summarise, the Anomaly Detection Module serves as the central component of the Edge-Resident IDoS, constantly analysing and monitoring system behaviours and network traffic in order to identify potential security concerns. With the help of its multifaceted approach, which combines statistical analysis, machine learning, signature-based detection, and contextual analysis, the system is able to precisely and quickly identify both known and unknown threats at the network edge.

Response Generation Module

An Edge-Resident Intrusion Detection System (IDoS)'s Response Generation Module is in charge of acting quickly to counter abnormalities or security concerns in order to lessen the impact of any incursions. An essential part of the overall IDoS architecture, this module helps the system respond to and protect against network edge security issues. The Response Generation Module's key goals are to notify security administrators, apply automated reactions, enable manual involvement when required, and keep thorough logs for study after an occurrence.

The Response Generation Module's ability to generate alerts is one of its primary features. The module instantly notifies security administrators by generating alerts upon detection of suspicious or unusual behaviour. The nature of the anomaly, its degree of severity, the entities that are impacted, and any pertinent contextual information are all covered in depth by these alerts. The alerting system makes sure that security staff can react quickly to possible threats so they can act appropriately and on time.

With the use of automatic reactions, the Response Generation Module is a technology that reduces the impact of threats without requiring human interaction. These can involve changing firewall rules, blocking harmful IP addresses, or temporarily deactivating user accounts. The module makes the Intrusion Detection System (IDS) more effective by facilitating prompt and planned reactions to particular security events. To guarantee proper replies, automation and human interaction must be balanced. Additionally, the module has manual intervention capabilities that let security administrators examine alerts and decide what actions to take before taking them. Additionally, the module keeps thorough logs of security events, responses, and results. These logs are crucial for compliance monitoring, post-incident analysis, and ongoing IDoS improvement.

Adaptive responses that can change based on the type and seriousness of threats detected are also included in the Response Generation Module. The IDoS can adapt its reaction actions dynamically to adapt to evolving threat scenarios and shifting network conditions. The module defines and implements response policies that outline the proper course of action for certain threats. These standards guarantee uniformity in the way security issues are handled by providing guidelines for automatic responses. Security administrators can evaluate warnings, consider the context of identified abnormalities, and decide on an appropriate course of action using the module's user-friendly interface. When an automated reaction could have serious repercussions or necessitates a sophisticated comprehension of the security context, this manual intervention option is useful.

Table 3. Components of the Security Reports

Report Element	Description
Security Events	Details of detected security events including timestamps, affected entities, and severity levels.
Response Actions	Summary of automated and manual responses taken to mitigate security threats.
System Performance	Metrics related to the overall performance of the IDoS, such as processing speed, resource utilization, and system uptime.
Anomaly Trends	Trends in detected anomalies over specific time periods helping identify emerging threats.
Compliance Status	Information on the IDoS's adherence to predefined security policies and compliance requirements.
Recommendations	Suggestions for improvements based on historical data and analysis, aiding in continuous enhancement.

The important components that are usually included in reports that the module generates are listed in table 3. Together, security events, response actions, anomaly trends, system performance indicators, compliance status, and recommendations offer a thorough picture of the security environment around the network. These reports can be used by security managers to make strategic decisions, pinpoint areas in need of development, and guarantee the IDoS's continued efficacy. The Report Generation Module's adaptability enables customisation in accordance with particular reporting requirements and organisational guidelines.

All things considered, the Response Generation Module is a vital component of the Edge-Resident IDoS architecture, offering a thorough and flexible method for efficiently addressing security threats. This module addresses abnormalities in real-time and reduces possible risks, thus improving the network's overall resilience and security posture—either by automatic reactions, manual intervention, or a combination of the two.

V. Experimental Setup

In the experimental setup, the aim was to check how well the suggested AI-powered Edge-Resident Intrusion Detection System (IDoS) enhances the security of Internet of Things (IoT) devices. For hardware, we used typical IoT devices with different computing abilities, along with local servers to run the Edge-Resident IDoS. On the software side, each IoT device got Suricata, a strong Network IDS, and we added the WAZUH Cyber Threat Intelligence platform to improve threat detection. The Edge-Resident IDoS used machine learning to spot anomalies in real-time on the edge devices. Data collection involved grabbing various IoT device behavior data to train the algorithms. We kept an eye on performance metrics like latency, bandwidth use, and scalability to see how well the system worked. To simulate cyber-attacks, we used Metasploit to test how the system responded to intrusion attempts and data theft. We also compared our system with traditional cloud-based solutions to show that the Edge-Resident IDoS used resources better. The assessment looked at practical effects, such as compliance, energy efficiency, and reliability in real-world IoT setups. This hands-on approach highlights the potential of the Edge-Resident IDoS in making connected environments more secure.

In the assessment, various aspects were scrutinized, including integrity checks, file types, and traffic logs, with a specific focus on operating systems (OS). The evaluation encompassed checking the reliability of the system in maintaining data integrity, examining the types of files processed, and analyzing traffic logs. Special attention was given to the specificity of traffic logs concerning different operating systems. This comprehensive examination aimed to ensure that the AI-powered Edge-Resident Intrusion Detection System (IDoS) could effectively handle diverse scenarios, providing a robust and nuanced approach to security in Internet of Things (IoT) environments.

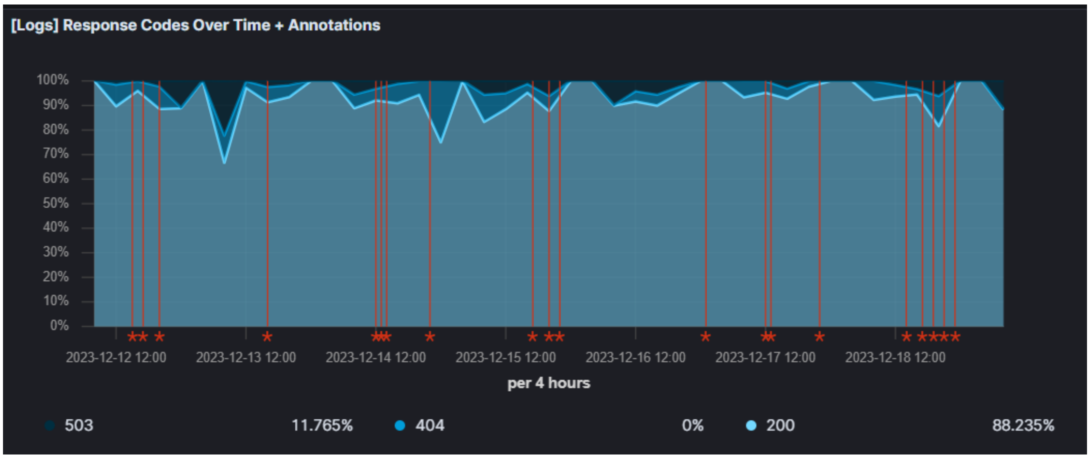

Figure 2. Sample Logs [Response Code Statistics]

The visualization in Figure 2 comprehends a system or application's temporal patterns and behaviors, based on its generated response codes and facilitates the identification of trends, patterns and anomalies within that behavior. Further context: annotations on this plot supply additional insights at specific points in time; they serve as markers expounding upon noteworthy occurrences throughout said period. Notable events, system changes, or incidents that may have influenced the observed response code patterns are what these annotations could highlight. They serve as markers for key occurrences; their role is to correlate spikes or dips in response codes with specific events.

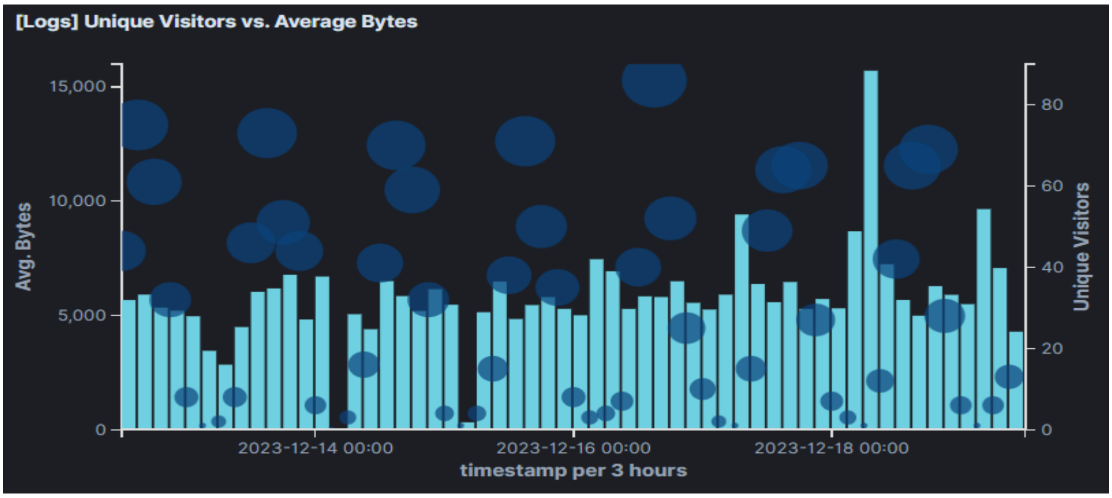

Figure 3. Sample Logs [Unique Visitors]

Figure 3 uniquely captures user engagement dynamics, Concurrently presenting two crucial metrics, "Unique Visitors" and the "Average Byte Consumption" associated with each visitor. The plot portrays a chronological narrative on the x-axis representing time intervals; meanwhile, counts for both unique visitors and their average byte consumption during corresponding time segments are encapsulated on the y-axis. Analyzing trends in unique visitors facilitates understanding of user interaction patterns, it measures platform popularity and identifies fluctuations in user activity. Concurrently, visualizing the average byte consumption per visitor illuminates both data transfer efficiency and individual interaction's digital footprints. The correlation between unique visitor count fluctuations and changes in average byte consumption reveals vital insights into resource use, considerations for user experience, and moreover content delivery’s overall efficacy. A holistic approach to these metrics ensures a thorough comprehension of user engagement; it also optimizes content distribution while tracking time-dependent performance dynamics within an IoT system.

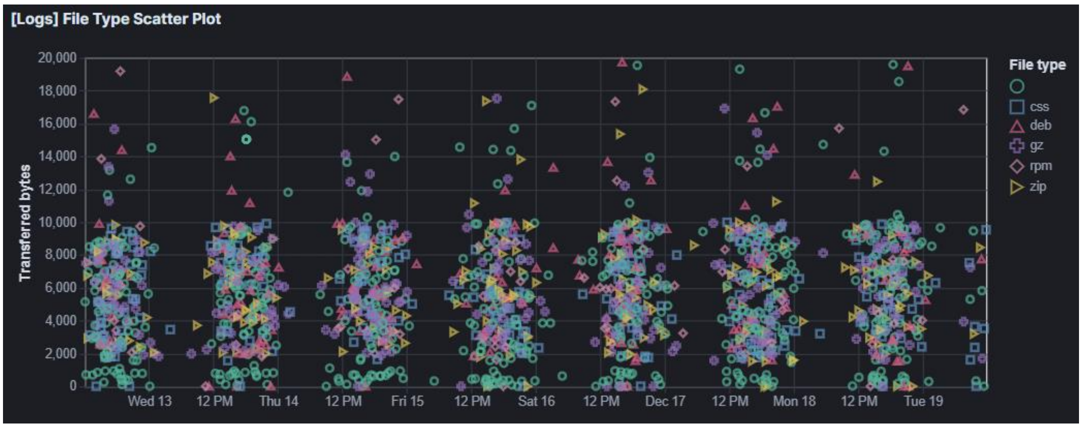

Figure 4. Sample Logs [Different File types]

A multifaceted representation in Figure 4 integrates the dynamics of "Different File Types" and their associated "Transferred Bytes" over time: The x-axis reveals temporal intervals; on the y-axis, we not only count various file types but also measure – by volume–the data each type transfers during corresponding time segments. This elaborate visualization—a scatter plot intrinsically—highlights specific file types such as CSS, DEB, GZ, RPM and ZIP. The scatter points represent instances of encountering these file types: their positions mirror not only their frequency but also the data they contribute. Through an examination of how often each file type appears and its associated transferred bytes, this plot provides comprehensive insights into content dynamics, resource utilization - even potential optimizations for efficient data transmission in IoT networks.

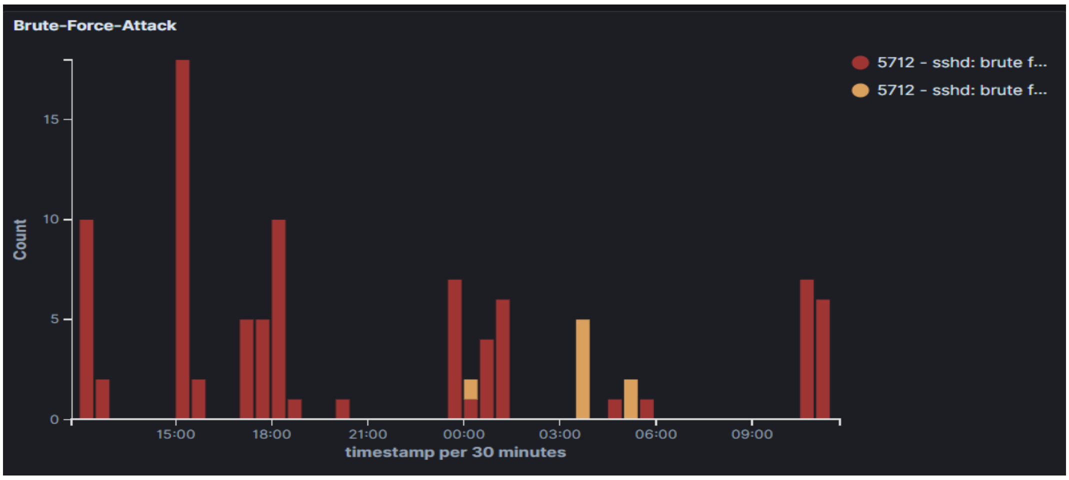

Figure 5. Screenshot of Brute force attack.

In Figure 5, we meticulously analyze Brute Force Attack Detection within the specific validation of an SSH (Secure Shell) daemon. The temporal intervals are delineated by timestamps on the x-axis and quantify each corresponding timeframe's count of detected brute force attacks along the y-axis. Discrete time points depict a bar for each attack frequency, providing visual insight into these security incidents' temporal distribution. The meticulous examination of temporal patterns in brute force attack occurrences over the specified duration becomes possible due to this representation's granularity. Figure 6 presents a granular analysis of "Suspicious Binary Detection," placing specific emphasis on technical validation. The x-axis precisely delineates temporal intervals via timestamps, while each corresponding temporal segment quantifies the count of identified suspicious binaries on the y-axis.

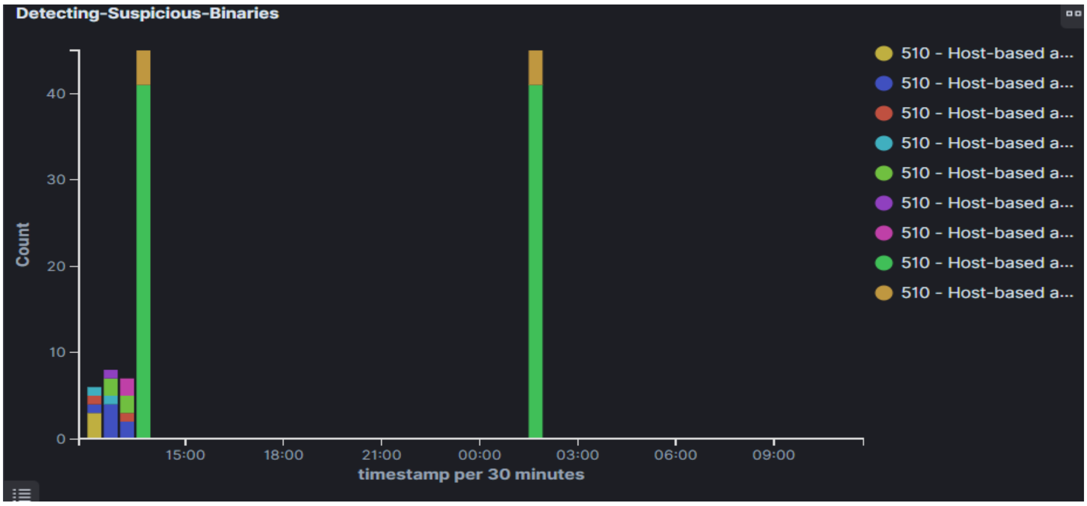

Figure 6. Screenshot of detecting suspicious binaries.

VI. Conclusion

This study concludes with an illustration of the Edge-Resident Intrusion Detection System (IDoS), a critical development in IoT device security achieved through integrating edge computing and artificial intelligence (AI). By strategically positioning the anomaly detection capabilities at the network's edge, this system mitigates issues related to latency, bandwidth, and scalability. Ultimately, for a broad spectrum of IoT applications, it renders the security solution more effective and responsive. AI-driven machine learning algorithms enable the real-time monitoring of device behavior, fostering a proactive stance towards threat identification. Further enhancing system resilience and optimizing bandwidth utilization involves the direct installation of Edge-Resident IDoS on edge devices or local servers; this reduces reliance on centralized cloud infrastructures. This proposed method addresses current security concerns within IoT ecosystems while also establishing robust, flexible and scalable security frameworks for our continually evolving world of interconnected devices.

References

[1] https://www.marketsandmarkets.com/Market-Reports/v2x-cybersecurity-market-194480977.html

[2] Gyamfi, Eric, and Anca Jurcut. "Intrusion detection in internet of things systems: a review on design approaches leveraging multi-access edge computing, machine learning, and datasets." Sensors 22.10 (2022): 3744.

[3] Spadaccino, Pietro, and Francesca Cuomo. "Intrusion Detection Systems for IoT: opportunities and challenges offered by Edge Computing and Machine Learning." arXiv preprint arXiv:2012.01174 (2020).

[4] Zhao, Xu, et al. "Research on lightweight anomaly detection of multimedia traffic in edge computing." Computers & Security 111 (2021): 102463.

[5] P. Ranaweera, A. D. Jurcut and M. Liyanage, "Survey on Multi-Access Edge Computing Security and Privacy," in IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1078-1124, Secondquarter 2021, doi: 10.1109/COMST.2021.3062546.

[6] M. Mukherjee, R. Matam, C. X. Mavromoustakis, H. Jiang, G. Mastorakis and M. Guo, "Intelligent Edge Computing: Security and Privacy Challenges," in IEEE Communications Magazine, vol. 58, no. 9, pp. 26-31, September 2020, doi: 10.1109/MCOM.001.2000297.

[7] Merenda, M.; Porcaro, C.; Iero, D. Edge Machine Learning for AI-Enabled IoT Devices: A Review. *Sensors* 2020, *20*, 2533. https://doi.org/10.3390/s20092533

[8] Fahim, Muhammad, and Alberto Sillitti. "Anomaly detection, analysis and prediction techniques in iot environment: A systematic literature review." IEEE Access 7 (2019): 81664-81681.

[9] [60] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab. "Edge computing enabling the Internet of Things". In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). 2015, pp. 603–608. doi: 10.1109/WF-IoT.2015.7389122.

[10] [49] Mateusz Krzysztoń and Michał Marks. “Simulation of watchdog placement for cooperative anomaly detection in Bluetooth Mesh Intrusion Detection System”. In: Simulation Modelling Practice and Theory 101 (May 2020), p. 102041. doi: 10.1016/j.simpat.2019.102041.

[11] [40] I. Butun, S. D. Morgera, and R. Sankar. “A Survey of Intrusion Detection Systems in WirelessSensor Networks”. In: IEEE Communications Surveys Tutorials 16.1 (2014), pp. 266–282.

[12] Kai Yang, Jie Ren, Yanqiao Zhu, and Weiyi Zhang. “Active Learning for Wireless IoT Intrusion Detection”. In: IEEE Wireless Communications 25 (Dec. 2018), pp. 19–25. doi: 10.1109/ MWC.2017.1800079.

[13] Mohamed Faisal Elrawy, Ali Ismail Awad, and Hesham F. A. Hamed. “Intrusion detection systems for IoT-based smart environments: a survey”. In: Journal of Cloud Computing 7.1 (Dec. 2018). doi: 10.1186/s13677-018 0123-6.

[14] Jegan Govindasamy and Samundiswary Punniakodi. “Energy Efficient Intrusion Detection System for ZigBee based Wireless Sensor Networks”. In: International Journal of Intelligent Engineering and Systems 10 (June 2017), pp. 155–165. doi: 10.22266/ijies2017.0630.17.

[15] Andrea Lacava, Emanuele Giacomini, Francesco D’Alterio, and Francesca Cuomo. “Intrusion Detection System for Bluetooth Mesh Networks: Data Gathering and Experimental Evaluations”. In: 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). 2021, pp. 661–666.

[16] Farhoud Hosseinpour, Payam Amoli, Juha Plosila, Timo Hämäläinen, and Hannu Tenhunen. “An Intrusion Detection System for Fog Computing and IoT based Logistic Systems using\ a Smart Data Approach”. In: International Journal of Digital Content Technology and its Applications 10 (Dec. 2016).

[17] E. G. Maria Verzegnassi, K. Tountas, D. A. Pados, and F. Cuomo. “Data Conformity Evaluation: A Novel Approach for IoT Security”. In: 2019 IEEE 5th World Forum on Internet of Things (WF-IoT). 2019, pp. 842–846.

[18] N. Pandeeswari and Ganesh Kumar. “Anomaly Detection System in Cloud Environment Using Fuzzy Clustering Based ANN”. In: Mobile Networks and Applications 21 (Aug. 2015). doi: 10. 1007/s11036-015-0644-x.

[19] H. Sedjelmaci, S. M. Senouci, and M. Al- Bahri. “A lightweight anomaly detection technique for low-resource IoT devices: A gametheoretic methodology”. In: 2016 IEEE International Conference on Communications (ICC). 2016, pp. 1–6.

[20] Rajinder Sandhu, Amandeep Sohal, and Sandeep Sood. “Identification of malicious edge devices in fog computing environments”. In: Information Security Journal: A Global Perspective 26 (July 2017), pp. 1–16. doi: 10.1080/19393555.2017. 1334843.

[21] Joseph Schneible and Alex Lu. “Anomaly detection on the edge”. In: MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM) (2017), pp. 678–682.

[22] Belal Sudqi Khater, Ainuddin Wahid Bin Abdul Wahab, Mohd Yamani Idna Bin Idris, Mohammed Abdulla Hussain, and Ashraf Ahmed Ibrahim. “A Lightweight Perceptron-Based Intrusion Detection System for Fog Computing”. In: Applied Sciences 9.1 (Jan. 2019), p. 178.

[23] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli. “Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices”. In: IEEE Internet of Things Journal 7.8 (2020), pp. 6882–6897.

[24] Matthias Niedermaier, Martin Striegel, Felix Sauer, Dominik Merli, and Georg Sigl. “Efficient Intrusion Detection on Low-Performance Industrial IoT Edge Node Devices”. In: ArXiv e-prints (2019). arXiv: 1908.03964 [cs.CR]

[25] D. Utomo and P. Hsiung. “Anomaly Detection at the IoT Edge using Deep Learning”. In: 2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW). 2019, pp. 1–2.

[26] Sibi Chakkaravarthy Sethuraman, Devi Priya, Saraju P Mohanty, “Flow based containerized honeypot approach for network traffic analysis: An empirical study", Computer Science Review, Elsevier, vol. 50, 100600, 2023.

[27] Devi Priya, Sibi Chakkaravarthy Sethuraman, Muhammad Khurram Khan, “Container Security: Precaution levels, Mitigation Strategies, and Research Perspectives", Computers & Security, Elsevier, vol. 135, 103490, 2023.

[28] Gopinath M, Sibi Chakkaravarthy Sethuraman, “A comprehensive survey on deep learning based malware detection techniques”, Computer Science Review, Vol. 47, February 2023, Elsevier.

[29] Devi Priya V S, Sibi Chakkaravarthy Sethuraman, “Containerized cloud-based honeypot deception for tracking attackers”, Scientific Reports, Nature, 2023.

[30] Dedipyaman Das, S.Sibi Chakkaravarthy, Suresh Chandra Satapathy, “A Decentralized Open Web Cryptographic Standard”, Computers and Electrical Engineering, Elsevier, Vol. 99, 107751, April, 2022.

[31] Ibbad Hafeez, Markku Antikainen, Aaron Yi Ding, and Sasu Tarkoma. “IoT-KEEPER: Detecting Malicious IoT Network Activity Using Online Traffic Analysis at the Edge”. In: IEEE Transactions on Network and Service Management 17.1 (Mar. 2020), pp. 45–59. doi: 10.1109/ tnsm.2020.2966951.

[32] S. Sibi Chakkaravarthy, Pranav Kompally, Saraju P Mohanty and Uma Chopalli,“MyWear: A Novel Smart Garment for Automatic Continuous Vital Monitoring”, IEEE Transactions on Consumer Electronics, IEEE, Vol. 67, No. 3, pp. 214-222, 2021.

[33] S. Sibi Chakkaravarthy, D. Sangeetha, Meenalosini Vimal Cruz, V. Vaidehi and Vaidehi V, “Design of Intrusion Detection Honeypot using Social Leopard Algorithm to detect IoT ransomware attacks", IEEE Access, IEEE, vol. 8, pp.169944-169956, 2020.