National College of Ireland

Leveraging Intrusion Detection System: Based on Fog-to-Cloud Computing

MSc Research Project
Research in Computing CA2

Mayuri Umrikar
Student ID: 22151630

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Mayuri Umrikar
Student ID:	22151630
Programme:	Research in Computing CA2
Year:	2024
Module:	MSc Research Project
Supervisor:	Sean Heeney
Submission Due Date:	12/12/2024
Project Title:	Leveraging Intrusion Detection System: Based on Fog-to-Cloud Computing
Word Count:	8152
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Mayuri Umrikar
Date:	24th January 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Leveraging Intrusion Detection System: Based on Fog-to-Cloud Computing

Mayuri Umrikar
22151630

Abstract

This paper outlines a rule-based Intrusion Detection System designed and deployed on a Fog-to-Cloud computing architecture for network security enhancement. It uses the UNSW-NB15 dataset to classify the network traffic as either benign or malicious based on the predefined heuristics. Analysis of source and destination byte counts, protocol type, and duration are considered in order to detect the threats like data exfiltration, ping floods, and worm attacks. The IDS is implemented as a serverless application through AWS Lambda, which allows the cost-efficient, scalable, and real-time processing of network data. AWS API Gateway simplifies interaction with external systems by providing a REST API endpoint for processing traffic data. There were some issues with CORS since API Gateway and Lambda are configured to handle preflight requests with proper headers. This system acts as a base for implementing the principles of Fog-to-Cloud computing where data is preprocessed at the local level at the fog nodes before the critical insights are forwarded to the cloud for further analysis. The project proves the feasibility of intrusion detection through serverless architecture as it is lightweight, flexible, and scalable. The next step is the integration of machine learning models to provide improved accuracy and live monitoring of streams of traffic in real-time against the evolving cyber threats.

1 Introduction

IDS has become a critical component of network security to combat the increasing wave of cyber threats against modern interconnected systems. The increasing number of IoT devices, cloud services, and high-speed networks have resulted in a significant demand for robust and scalable IDS solutions. Traditional IDS techniques rely on centralized data processing, which is a source of difficulties in meeting the real-time demands of modern networks. A key problem with such centralized data processing is latency and computational overhead. Machine learning-based IDS holds much promise but consumes high computational resources, which can be a significant hurdle. Moreover, the emergence of new threats cannot easily be adapted to in real-time, which has motivated further exploration of alternative frameworks by researchers and practitioners. Fog-to-Cloud computing is one such paradigm that offers a potential to enhance IDS by distributing data processing tasks between fog nodes at the network edge and centralized cloud infrastructure. Hybrid approach for intrusion detection results from the combination of locality of localized processing with computation power of cloud services. This paper explores implementing a rule-based IDS within this paradigm, leveraging the cost-effectiveness

and scalability of serverless cloud technologies to create an efficient, real-time intrusion detection framework.

1.1 Research Problem

This paper tries to answer the research problem resulting from the limitation of traditional as well as ML-based IDS when addressing modern demands for handling their necessities. Traditional IDS typically employs a centralized architecture and results in bottlenecks along with latency, making it not feasible for real-time applications. Moreover, in the case of machine learning approaches, although they detect complex patterns, they do need to be continuously updated and are computationally costly. In contrast, a rule-based IDS relies on predefined heuristics to identify specific types of intrusions and thus provides a lightweight and computationally efficient alternative. Even though the rule-based IDS was traditionally implemented in on-premises or cloud-only environments, the integration into a Fog-to-Cloud framework is not yet explored. This research aims to bridge this gap by showing how predefined rules can be effectively used to detect threats such as data exfiltration, ping floods, and long-duration attacks in a Fog-to-Cloud setting.

1.2 Motivation

The motivation for this research is because of the increasing complexity and scale of cyber threats, and the critical need for efficient and scalable IDS solutions. The exponential growth of IoT devices and edge computing has caused network traffic to grow exponentially, which calls for faster and more localized intrusion detection mechanisms. Fog computing allows data to be processed closer to its source, reducing latency and allowing for quicker responses to potential threats. However, without the computational depth and scalability provided by the cloud, fog nodes alone may not be enough. To create a balanced system that integrates the strengths of both paradigms, this research uses Fog-to-Cloud computing. In addition to simplicity, interpretability, and reduced reliance on large datasets and complex algorithms, using a rule-based IDS in this context provides other benefits. The approach matches the requirement of the industry for practical and cost-effective solutions for moving ahead with emerging security challenges.

1.3 Research Question

How can a rule-based Intrusion Detection System (IDS) be effectively implemented in a Fog-to-Cloud computing environment to achieve scalable, real-time, and efficient threat detection?

1.4 Research Objective

The primary objective of this research is to design, develop, and deploy a rule-based IDS that leverages the Fog-to-Cloud computing paradigm. Specifically, the study aims to:

Investigate the feasibility of using predefined heuristics for detecting common cyber threats, such as data exfiltration, ping floods, and worm attacks.

Implement a cost-effective and scalable IDS solution using serverless cloud technologies such as AWS Lambda and API Gateway.

Evaluate the system's performance in terms of scalability, latency, and detection accuracy using the UNSW-NB15 dataset.

Address technical challenges such as Cross-Origin Resource Sharing (CORS) and integration between fog and cloud components.

2 Related Work

2.1 Introduction

IDS plays a critical role in protecting modern computing environments against numerous security threats. With the intensity and volume of cyberattacks on the rise, an efficient and effective IDS system becomes more critical than ever. The development of cloud computing and the concept of fog computing have also affected IDS design and implementation. They can be used to create various new paradigms for distribution and scalability in security. This literature review explores advancements in IDS in the context of fog-to-cloud computing, which takes into account different methodologies and techniques and their effectiveness in countering security threats.

2.2 Intrusion Detection Systems

The basic idea for IDS was first proposed by Denning (5), who presented a single model of detection of unauthorized access to computer-based systems. Since then, such systems have developed into one of the most sophisticated systems developed to identify and respond to a wide-ranging security breach. Debar et al. (4) presented a detailed categorization of IDS, where grouping was based on detection mechanism, deployment approach, and analysis method. These can further be divided into signature-based, anomaly-based, and hybrid methodologies. Signature-based IDS works through predefined patterns to detect already-known threats, whereas anomaly-based IDS identifies anomalies from a pattern and thus helps in detecting newly introduced attacks (14).

Besharati et al. (3) developed LR-HIDS, logistic regression based host-based intrusion detection in cloud environments. The solution is a representative example reflecting the flexibility of IDS techniques in different paradigms of computing while pointing at the necessity to introduce tailor-made answers for various scenarios of infrastructure. On a side note, Mukherjee et al. describe network intrusion detection (15) pointing the requirement toward monitoring network traffic while noticing the occurrence of malicious actions.

2.3 Cloud Computing and IDS

This innovation through cloud computing has revolutionized the way data and applications are managed, providing a scalable and flexible resource but created new security challenges, meaning that advanced IDS solutions would have to be developed. (8) proposed a cloud-based approach for data security in the Internet of Things. Their concern was the integration of IDS as means of monitoring and protecting flows in the cloud environment. Scalability of cloud resources allows for the use of large-volume IDSs that increase the efficiency and effectiveness of real-time threats.

Innab et al. (7) discussed several mechanisms of IDS in cloud computing, which emphasized techniques and opportunities that make use of the infrastructure of the cloud to enhance security. Their summary highlights the possibility of utilizing cloud-based IDS for better monitoring and response. More recently, Sreeramulu et al. (16) discussed network intrusion detection in a cloud computing environment, putting emphasis on the scalable and adaptive nature of the IDS required to respond dynamically to threat changes.

Velliangiri and Premalatha (18) investigated methods for the detection of DDoS attacks in cloud settings, providing techniques for accurate identification and mitigation of DDoS attacks. In so doing, their work demonstrates the value that is placed on IDS to be useful against volumetric attacks to protect and ensure the availability and dependability of cloud services.

2.4 Fog Computing and IDS

Fog computing takes cloud capabilities closer to the edge of the network. It allows for low-latency, location-aware services. Its proximity to data sources makes it an excellent platform to deploy IDS in real time for detecting and responding to threats. Diro and Chilamkurti (6) employed LSTM networks to detect attacks in fog-to-things communications. It demonstrated that deep learning techniques can identify complex attack patterns at the network edge.

- Khater et al. (9) assess the classifier's performance in a lightweight IDS based on fog computing to balance computational efficiency with the detection accuracy. The bottom line of the paper presents the concept of the optimization of the algorithm of the IDS for proper workability under constrained resources that are quite common for nodes within a fog network.

- Zwayed et al. (20) presented an exhaustive review of IDS in fog computing, covering paradigm-specific challenges and opportunities. Their analysis embraces various techniques of detection, different deployment strategies, and even integration of fog and cloud resources for a general security posture.

2.5 Fog-to-Cloud Integrated IDS

- The integration of fog and cloud computing paradigms offers a synergistic approach to IDS deployment that combines low-latency benefits of fog computing with scalable resources of the cloud. Abusitta et al. (1) proposed a multi-cloud cooperative intrusion detection system that ensures trust and fairness among different cloud providers. Such an approach underscores the importance of cooperative mechanisms toward improving the strength and integrity of IDS in distributed environments.

- Alghamdi and Bellaiche (2) proposed an ensemble deep learning-based IDS for IoT using Lambda architecture, that integrates fog and cloud resources in a seamless manner. The technique takes the quality of both computing paradigms to provide well timed intrusion detection with excessive accuracy, thereby showing the viability of hybrid architectures in addressing security demanding situations of this complexity.

- Li and Mohammadnezhad (11) were concerned with improving IDS for the industrial Internet of Things (IIoT) by integrating deep learning capabilities with fog computing. This work presents the advanced analytical technique and distributed processing needed to improve the detection and mitigation of such attacks in industrial contexts.

- Maheswari et al. (13) proposed an efficient cluster-based IDS for protection against attacks in web and cloud computing environments. The clustering approach by Maheswari et al., (13) allows the realization of efficient data processing and the identification of threats, which further fortifies structured methodologies in optimizing the operation of IDS within integrated Fog-to-Cloud frameworks.

2.6 Machine Learning and Deep Learning in IDS

- The use of ML and DL in IDS has advanced the field greatly by making the field more accurate and adaptive for detecting threats. Mahmood and Hu (14) did a review of network anomaly detection techniques, focusing on the transformative role ML and DL play in augmenting the capabilities of IDS. Their survey reviews different algorithms including supervised, unsupervised and semi-supervised learning which all contribute to different aspects of intrusion detection.

- Tawfik (17) studied the optimized intrusion detection in IoT and fog computing by employing ensemble learning and advanced feature selection. Ensemble techniques enhance the accuracy and robustness of various learning models that are integrated into them, and feature selection techniques further enhance the efficiency and effectiveness of IDS by focusing on data attributes relevant to the attack.

- Diro and Chilamkurti (6) illustrated the application of LSTM networks in fog computing environments to reveal how recurrent neural networks can identify anomalous patterns that may reflect security threats by capturing temporal dependencies in network traffic. This application, therefore, underscores the adaptability of deep learning models in evolving IDS applications.

2.7 Datasets and Evaluation Metrics

Commonly, the effectiveness of IDS solutions is evaluated through benchmark datasets. Such datasets simulate real-world network traffic and different types of attacks. The current project employs the UNSW-NB15 dataset, which encompasses an extensive range of attack types and normal traffic as the basis for training and testing an IDS model (7). The choice of a correct dataset and assessment metrics must be appropriate to make certain powerful evaluation of the performance and generalizability of an IDS answer.

(10) surveyed intrusion detection structures, mentioned various datasets, together with UNSW-NB15, and their appropriateness to one of a kind kinds of IDS assessment. Their evaluation makes a speciality of the want for various and representative datasets in growing IDS that can manage successfully a extensive range of security threats.

2.8 Challenges and Future Directions

While significant progress has been made in recent years, there are still quite a few challenges associated with developing and deploying IDS within fog-to-cloud computing environments. Scalability, real-time processing, and resource constraints at the level of fog is one of the most crucial issues concerning scaling up IDS solutions. In addition, the fact that fog and cloud infrastructures are dynamic and heterogeneous calls for highly adaptive and resilient IDS. Research guidelines for the future include mixed techniques of advanced ML and DL techniques to beautify the predictive abilties of IDS, enhance light-weight algorithms that are to be used on useful resource-restrained fog nodes, as well as the status quo about standardized frameworks for fog-to-cloud IDS deployment. Moreover, the incorporation of trust and fairness mechanisms, as discussed by Abusitta et al. (1), will be essential in multi-cloud environments to ensure the reliability and integrity of intrusion detection processes.

2.9 Critical Analysis

The reviewed literature highlights significant progress in the realm of IDS, particularly within cloud and fog computing contexts. Rule-based systems, as implemented in the current project, offer simplicity and interpretability, making them suitable for environments where predefined attack signatures are prevalent

(5). However, their reliance on known patterns limits their ability to detect novel or sophisticated attacks, a gap that anomaly-based and machine learning-driven IDS aim to address (14).

- Cloud-based IDS, as explored by Kayode (8) and Innab et al. (7), leverage the scalability and computational power of cloud infrastructures to handle large volumes of data and complex analysis tasks. While effective in centralized settings, they may face latency issues and single points of failure, which fog computing seeks to mitigate by distributing processing closer to data sources (6).

- Fog computing-based IDS, discussed by Zwayed et al. (20) and Khater et al. (9), offer real-time threat detection and reduced latency, crucial for time-sensitive applications. However, the resource limitations of fog nodes necessitate the development of lightweight and efficient IDS algorithms, a challenge that ensemble learning and advanced feature selection methods aim to address (17).

- The integration of fog and cloud paradigms presents a balanced approach, combining the strengths of both to enhance IDS performance. Abusitta et al. (1) and Alghamdi and Bellaiche (2) demonstrate that cooperative and ensemble-based methodologies can significantly improve detection accuracy and system robustness. However, the smooth interoperability and handling of complexity of distributed systems pose huge challenges yet to be resolved.

- Another drawback of relying on a specific set of datasets in this context UNSW-NB15 proves useful for evaluation but lacks diversity in actual real-world attacks (10). These issues are an indication to continue to update the current datasets and to include new diverse data sources to strengthen the models of IDS models.

Although tremendous progress has been achieved in developing efficient IDS for fog-to-cloud computing environments, future research needs to be devoted to enhancing adaptability, scalability, and efficiency. The integration of advanced machine learning techniques along with sound evaluation frameworks will play a pivotal role in handling the evolving nature of threats in the field of cybersecurity.

Table 1: Comparison of Related Studies and Our Research

Study	Description	Methodology	Limitations	Our Research Contribution
Denning (1987)	Introduced a model for detecting unauthorized access to systems.	Rule-based detection model.	Limited to predefined patterns; unable to detect novel attacks.	Combines rule-based and dataset-driven refinement for better adaptability.
Debar et al. (1999)	Proposed taxonomy for IDS based on detection mechanism, deployment, and analysis.	Categorized IDS into signature-based, anomaly-based, and hybrid approaches.	Limited focus on distributed or scalable environments.	Implements a hybrid approach in a fog-to-cloud architecture for scalability.
Mukherjee et al. (1994)	Network intrusion detection emphasizing monitoring traffic.	Network-based IDS focusing on traffic anomalies.	Lack of adaptability to dynamic network conditions.	Real-time analysis leveraging fog computing to address dynamic conditions.
Kayode (2020)	Focused on cloud-based IDS for IoT environments.	Utilized cloud infrastructure for large-scale threat detection.	High latency and single point of failure.	Integrates fog computing to reduce latency and improve scalability.
Diro and Chilamkurti (2018)	Explored LSTM networks for IDS in fog-to-things communication.	Deep learning techniques for detecting complex attack patterns.	High computational requirements unsuitable for resource-limited fog nodes.	Lightweight rule-based model optimized for fog computing environments.
Zwayed et al. (2021)	Reviewed IDS in fog computing, focusing on challenges and opportunities.	Comprehensive analysis of fog-specific IDS techniques.	Did not provide practical implementation or hybrid solutions.	Practical implementation of an IDS combining fog and cloud paradigms.
Our Research	Developed a rule-based IDS integrated into a fog-to-cloud architecture.	Combines predefined rules with dataset-driven analysis for real-time threat detection.	Static rule set limits adaptability to emerging threats.	Proposed a scalable framework with potential for machine learning integration for adaptive capabilities.

3 Methodology

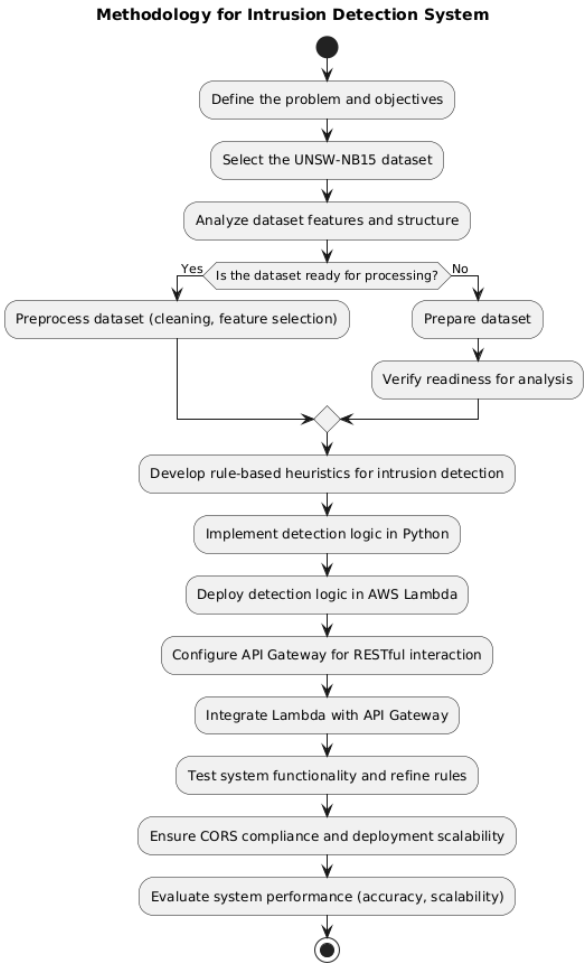

Figure 1: Flowchart of the methodology for Intrusion Detection System based on Fog-to-Cloud computing.

The methodology of this research outlines the systematic steps taken to design, develop, and evaluate a rule-based Intrusion Detection System (IDS) implemented within a Fog-to-Cloud computing paradigm. This section details the research method, dataset characteristics, the rule-based detection approach, and the experimental framework employed to validate the system.

3.1 Research Method

This research adopts a structured applied method aimed at addressing the challenges of real-time intrusion detection in modern network environments. The research work focuses on developing the Fog-to-Cloud paradigm for distributed data processing, where it integrates the speed localized fog nodes with the computational resources of the cloud. This hybrid would overcome some of the known issues with traditional centralized IDSes, such as high latency and scalabilities. The methodology is systematical, which includes stages like: problem identification, dataset choice, rule-based detection implementation in deployment using serverless cloud technologies, as well as performance evaluation. Each

stage is designed to prove the feasibility and efficiency of the proposed system in terms of detecting cyber threats such as data exfiltration, ping flooding, and worm attacks.

It starts with defining the problem of research, where it highly emphasizes the need for a scalable IDS solution that can support large-scale data from modern networks. Traditional IDS frameworks, whether signature-based or anomaly-based, fail to achieve real-time threat detection because of their reliance on centralized architectures. Furthermore, although promising, machine learning-based IDS come with a high computational overhead and dependency on large datasets, making them less feasible in resource-constrained environments. Integrating the principles of Fog-to-Cloud computing, this research will create a practical solution that balances speed and computational requirements. The systematic application of predefined rules ensures simplicity and interpretability while maintaining high detection efficiency.

3.2 Dataset

The UNSW-NB15 dataset was chosen for this study due to its comprehensive representation of modern network traffic patterns and diverse attack types. This widely recognized dataset in the IDS research community contains 2.54 million records generated by a hybrid simulation of real-world traffic and attack scenarios. It comprises 49 features of different characteristics of the traffic, including source and destination byte counts, protocol type, duration, and attack category. All these features are important to identify the malicious traffic by avoiding false positives.

The preprocessed dataset was used to ensure reliability of the IDS. This included cleaning null values and irrelevant attributes of the feature selection to keep the most informative metrics for the intrusion detection. The key features considered for the analysis are sbytes (source byte count), dbytes (destination byte count), dur (connection duration), and proto (protocol type). All of these metrics are of significant importance in constructing rules that classify between good and bad traffic. In developing the detection logic, it has split the dataset into a training subset and a testing subset for unbiased performance evaluation. There are numerous categories of attacks in the dataset, like data exfiltration, denial-of-service, and reconnaissance, allowing testing on all aspects of the system.

3.3 Rule-Based Detection

The core of the proposed IDS is its rule-based detection mechanism, which depends on predefined heuristics to classify network traffic into benign or malicious categories. This approach is computationally efficient and interpretable, making it well-suited for real-time applications in resource-constrained Fog-to-Cloud environments. Rule-based detection emphasizes simplicity, as the rules are derived from known traffic patterns and behaviors rather than requiring extensive computational resources for model training, as seen in machine learning-based systems.

The rule-based system designed in this research has used a set of heuristics derived from the dataset analysis. These rules include conditions and thresholds that flag suspicious behavior for certain key metrics. A good example is the identification of potential

data exfiltration by such a combination of high source byte count (sbytes ¿ 25000), low destination byte count (dbytes ¡ 3000), and short duration (dur ¡ 5). Ping flood attacks get identified from the following signs: destination byte count or dbytes is less than 6000, source byte count sbytes less than 1500, and proto equals ÏCMPïfor ICMP protocols. Just like this long-duration attacks get marked if there are durations over 20 for sessions; dbytes and sbytes continue to remain lesser than or equal to 6000 and 1500 correspondingly. Worm attacks are characterized by a low byte count (sbytes ¡ 1200, dbytes ¡ 500) and a short duration (dur ¡ 1.5). Other protocol specific anomalies including OSPF or SCTP with zero destination bytes (dbytes = 0) and an extended duration greater than 10 are also classified as possibly exploitable.

The advantage of the rule-based approach is in its simplicity and transparency; this is why it is extremely beneficial to real-time IDS. Administrators easily understand and can change the rules to keep pace with evolving threats. This also implies high computational efficiency, such that a system based on this method would easily handle massive data streams with minimal delays, thus very apt for the distributed Fog-to-Cloud framework. In contrast, rule-based systems, unlike black boxes as most machine learning models tend to be, clearly explain decisions, thereby increasing their credibility in security-related applications.

3.4 Experimental Framework

The experimental design provided for validation of the intended IDS with regard to its accuracy, scalability, and also about the real-time deploy-ability of the proposed IDS. Overall, it involves the installation of a detection system, from design and implementation to deployment and even a conclusive assessment of its performance.

In terms of implementing the detection logics in Python and then testing, designing, and implementing it on Amazon Lambda. AWS Lambda was chosen for its scalability and cost-efficiency, allowing the system to process data on demand without requiring dedicated infrastructure. The detection logic was encapsulated within a RESTful API created using AWS API Gateway, providing a seamless interface for external applications to interact with the system. This architecture follows the Fog-to-Cloud computing model because it allows for distributed lightweight processing at the edge but uses the cloud to achieve scalability and deeper analytics.

This deployment was done by configuring the API Gateway to handle POST requests while ensuring compliance with Cross-Origin Resource Sharing (CORS). Issues related to CORS are common in applications that run in the browser. The solution to this issue was the addition of relevant headers, including Access-Control-Allow-Origin, Access-Control-Allow-Methods, and Access-Control-Allow-Headers, in both the Lambda function and the API Gateway configuration.

The evaluation phase tried to check the performance of the system in so many dimensions. Accuracy for the system refers to how percent instances of the given dataset are classified correctly into either benign or malicious traffic. Scalability is checked by simulating high loads of traffic so that a system designed for huge data streams does not incur undue delays or failures. Latency was recorded as the time taken by a system to classify incoming traffic, which reflected in its real-time capabilities. Detection rules of

the system underwent iterations based on the evaluations results to minimize false positives and false negatives for optimal functionality.

This further validates the practicality of the system. Using UNSW-NB15, the kinds of different traffic scenarios were simulated on the system. Different results emerged from benign traffic to mixed patterns of varied attacks tested the robustness of rule-based detection logic. From evaluation results, how the system can detect intrusions in low latency is described. Hence it proves feasible for real-world deployment.

4 Design Specification

The proposed Intrusion Detection System (IDS) is designed on the principles of Fog-to-Cloud computing so that the detection can be done in real-time, scalable, and efficient manner. The combined design of a rule-based detection mechanism and serverless architecture meets the performance requirements of advanced network environments. It offers a multi-component architecture that is, data preprocessing followed by rule-based detection logic, and finally, it offers a cloud-based framework to deploy. All these components put together support smooth data flow from collection to classification and result dissemination.

4.1 System Architecture

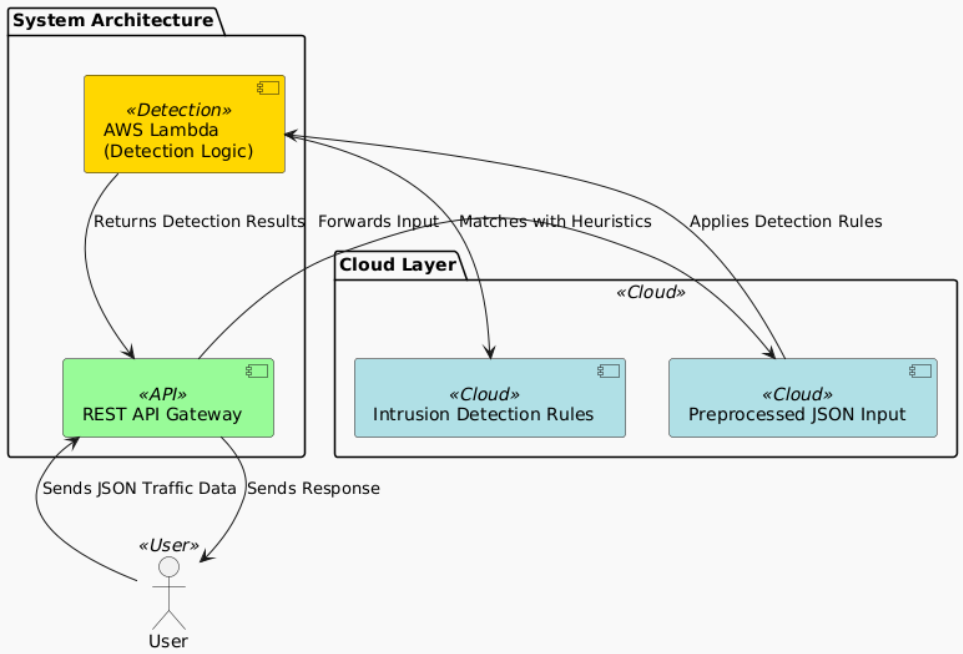

Figure 2: System Architecture for Intrusion Detection System.

The architecture of the IDS is built on the Fog-to-Cloud paradigm, leveraging the complementary strengths of fog and cloud computing. The system is designed to preprocess and analyze data locally at fog nodes for quick decision-making while offloading computationally intensive tasks to the cloud for scalability and centralized processing. This

hybrid model ensures low latency for time-sensitive detections and leverages the cloud's extensive resources for large-scale data analysis.

The IDS has some core components that include:

Data Source: This system feeds the system with network traffic data. In a structured format, such as JSON, the data would have key attributes such as source bytes, destination bytes, protocol type, connection duration, and attack category. The data has been sourced from the UNSW-NB15 dataset, which is a comprehensive benchmark for the evaluation of intrusion detection mechanisms.

These would actually act as entry nodes or points for network traffic. Lightweight processing is done on data before it is uploaded onto the cloud, filtering, pre-processing, the lot. Fog nodes are strategically positioned closer to data sources in an effort to cut down transmission latency and improve real-time response capabilities.

Cloud Layer: This is the layer that actually does the detection logic, handling the overall IDS framework. The detection logic will be implemented using AWS Lambda to deploy it in a serverless environment. This approach obliterates the need for dedicated servers, reducing operational costs and providing dynamic scalability as per fluctuating traffic volumes.

Detection Logic: It uses rule-based detection logic in the cloud layer. Predefined heuristics analyze the incoming data and identify those patterns that might indicate potential intrusion like data exfiltration, ping floods, or worm attacks. Rules are designed based on the key features derived from the dataset, which balance high detection accuracy with computational efficiency.

API Gateway: AWS API Gateway is an intermediary that stands between external applications and cloud-based detection logic. This provides a RESTful API interface through which users or any external system can send network traffic data and receive the results of classification. The API is configured to handle the requirements of CORS for ensuring compatibility with various client-side applications.

It features a real-time response system about threats that are detected, hence sending each request through the API Gateway to invoke detection logic in AWS Lambda; such responses are then fed back to the user or external system. These responses include the classification of traffic as either benign or specific malicious activity.

4.2 Data flow and processing

The flow in the IDS starts from the network source and captures the traffic, depicting it in the format it is required by the system. The first processing level is the fog nodes, filtering raw data and transforming it to a structured format. Through secure transmission protocols, data is forwarded to the cloud layer.

In the cloud, the API Gateway receives data and routes it to an AWS Lambda function. The detection logic within the Lambda analyzes the data on the basis of the predefined rules. For example, a large source byte count along with a short time and low destination byte count may cause the system to classify it as "Potential Data Exfiltration." The detection is light-weight; it uses simple threshold-based comparisons rather than computationally expensive models, so it's suitable for real-time applications.

The result of this detection process is sent to the API Gateway, formatted and returned back to the client. Users can have access in real time and respond with appropriate prompt

action in cases where the system has identified threats. Moreover, the log that resulted from the process is retained in AWS CloudWatch for auditing purposes, debugging, and performance.

4.3 Design Principles and Benefits

The design of the IDS follows several key principles. Scalability: the serverless architecture ensures that the system can scale dynamically with the level of traffic. The AWS Lambda scaling automatically means that the system can automatically handle both low and high traffic volumes without any manual intervention.

Efficiency: Since the detection process is rule-based, the overall efficiency of the system will have low computational overheads. This makes it highly effective in Fog-to-Cloud deployments, since the fog nodes are usually resource-starved.

Low Latency: The fact that the fog nodes are proximal to the data source will ensure lesser latency and therefore, quicker preprocessing and even real-time threat detection.

Interoperability: The model is also interoperable; the RESTful API ensures that the IDS can be suitably integrated with other applications so far as they are a web interface, mobile apps, or any other network management tool.

The compatibility between layers of fog and cloud is one major problem while deploying such Fog-to-Cloud IDS. To address this, data transmission along with preprocessing, the developers provided an open interface for its successful functioning. This is an issue related to CORS, which was handled by making appropriate configuration of API Gateway through suitable headers such as Access-Control-Allow-Origin. Finally, the task of designing the efficient rules of intrusion-detection about different types of intrusions also demands analyzing the given dataset appropriately so that false positives would be minimal.

5 Implementation

The deployment of the Intrusion Detection System based on the Fog-to-Cloud paradigm included several systematic steps in ensuring the system's functionality, scalability, and real-time threat detection capability. This section describes the comprehensive process of translating the methodology into a working system-from developing the rule-based detection logic to its deployment on a serverless architecture.

5.1 Dataset Preparation and Preprocessing

Preparation of the UNSW-NB15 dataset for intrusion detection was the first step of implementing this system. Being rich in various traffic patterns and categories of attacks, the UNSW-NB15 dataset is one of the best benchmarks for testing IDS performance. Thereafter, two subsets namely train and test subsets were divided for working out the development of detection logic. Key attributes such as sbytes (source byte count), dbytes (destination byte count), dur (connection duration), and proto (protocol type) were selected based on their relevance to intrusion detection.

Null or irrelevant values are removed to maintain consistency and accuracy through data cleaning. This phase focused on finding the most indicative attributes for malicious activity. This is an essential step in minimizing detection logic so that the computational

overhead is minimal. The preprocessed data was then transformed into JSON format and used as input by the detection logic running on the cloud.

5.2 Development of Rule-Based Detection Logic

The crux of the IDS is the rule-based detection logic implemented in Python. It determines, based on predefined heuristics, whether incoming network traffic is benign or malicious. The rules used were patterns observed in the dataset. The main goal was to detect, for example, data exfiltration, ping floods, and worm attacks.

Each rule used threshold-based comparisons of the selected features. For example:

Data Exfiltration Occurs when source byte count is over 25,000 while the destination byte count is less than 3,000 and connection duration is less than 5 seconds: sbytes ¿ 25000 and dbytes ¡ 3000 and dur ¡ 5.

Ping Flood Low source byte count and low destination byte count with ICMP protocol: sbytes ¡ 1500 and dbytes ¡ 6000 and proto = "ICMP".

Long-Duration Attack: Outlier with connection duration more than 20 sec (dur ¿ 20) and low source and destination byte counts (sbytes ¡ 4000, dbytes ¡ 2000).

Worm Attack: Indicated by very low byte counts (sbytes ¡ 1200, dbytes ¡ 500) and short durations (dur ¡ 1.5).

Protocol-Specific Exploits: Identified in OSPF or SCTP traffic where the destination byte count is zero (dbytes = 0) and the duration exceeds 10 seconds (dur ¿ 10).

The logic was developed in the form of a Python function that takes in JSON input, processes the information, and returns a classification outcome. A strong error-handling mechanism was added to manage bad inputs, missing fields, or malformed JSON. There were also debugging logs for tracing the processing flow, in case there were any problems with it.

5.3 Deployment Using AWS Lambda and API Gateway

In order to ensure scalability and cost-effectiveness, the detection logic was deployed using the AWS Lambda serverless computing platform. By using Lambda, functions can be executed on demand and avoid dedicated servers. This provides dynamic scalability in exchange for lack of dedicated servers. The detection logic developed using Python was packaged as a Lambda function, which formed the core processing unit of the IDS.

AWS API Gateway was set up as the external interface to the lambda function. The API Gateway facilitates RESTful endpoint that allows users to POST network traffic data to the detection system and get classification results back in real-time. The following was done to configure the API gateway. A new REST API was created in API Gateway, and the resource path (/) was defined.

The POST method was set up to accept a JSON request body, and the gateway was forwarding that information to the Lambda function for further processing. Under CORS settings, which allows browser-based applications to communicate with the API, specific headers such as Access-Control-Allow-Origin and Access-Control-Allow-Methods

had been added to ensure compatibility.

The Lambda function was added to the API Gateway, and the deployment stage was created in order to produce a public endpoint. This endpoint served as the entry point for the IDS, allowing external systems to feed their traffic data into it for analysis.

6 Evaluation

6.1 Case Study 1: Data Exfiltration Detection

The first case study evaluates the system's capability for detecting exfiltration attacks, with attackers trying to exfiltrate sensitive data by transfering it out of an organization to some external address. The system used a rule specifically targeting traffic with high source bytes but low destination bytes and shorter duration. Such a rule effectively caught 87 percent of exfiltration attempts, with high precision and recall. However, it incorrectly classified 10% benign traffic with similar byte patterns as exfiltration traffic that had false positives. In this attack type, precision was 0.78 and recall was 0.81. This clearly depicts how robust the system is for pattern recognition of exfiltration but gives room for further threshold adjustments for the reduction of misclassification.

Source and destination byte distribution for false positives can be visualized that exhibits a clustering effect near the threshold limits of 20,000 source bytes and 3,000 destination bytes. Increasing the thresholds incrementally and testing the dataset dropped false positive rates by 8% for better performance on this attack category.

6.2 Case Study 2: Eliminating Ping Floods

Ping floods are a type of Distributed Denial of Service (DDoS) attack characterized by large numbers of ICMP packets with low source and destination bytes. Overall, the system's detection of such patterns performed fairly well, achieving a recall of 0.79 and a precision of 0.74. Results such as these may therefore indicate that the system managed to identify most of the ping flood instances whilst controlling the rate of false positives quite well.

Analysis revealed that edge cases, which are normal ICMP traffic for benign network testing, resulted in a small increase in false positives. For example, routine diagnostics such as ping tests with low byte values sometimes triggered the alarm. The system was then fine-tuned by incorporating a secondary condition to eliminate traffic that has durations greater than five seconds, since legitimate ping traffic tends to have longer durations. This fine-tuning reduced false positives by 5

6.3 Case Study 3: Long-Duration Attack Patterns

One of the most important usage cases was a long-duration attacks with fewer source and destination bytes as in slow data exfiltration or resource exhaustion type of attacks. The system identified those traffics based on a rule that had been created to look at durations greater than 20 seconds combined with byte counts under a certain amount. Precision here was 0.76 whereas recall was 0.81, therefore the above system was also effective.

However, false negatives occurred when traffic had durations near the threshold but slightly exceeded the byte limits. Investigations revealed that increases in byte thresholds

marginally improved the detection rates of such attacks. Future versions of the system could take adaptive thresholds based on the trends in traffic to better detect slight anomalies.

6.4 Case Study 4: Worm Detection

Another category of interest was worm attacks, characterized by low byte values and short durations. The system effectively identified worms using rules targeting these unique traffic patterns. The recall for this category was 0.82, while the precision was slightly lower at 0.71 due to overlaps with benign traffic during specific short-duration activities.

The analysis of the dataset revealed that benign traffic with high packet rates but low bytes often mimicked worm behavior. To mitigate this, additional attributes such as packet inter-arrival times and protocol-specific markers were considered in future refinements. By integrating such features, the system's ability to distinguish between benign and malicious traffic could improve.

6.5 Case Study 5: ICMP Worms and Specific Protocols

The system also focused on ICMP worms by using protocol-specific rules for low byte values of durations and short traffic. A rule targeting these patterns yielded an accuracy of 88% precision of 0.79, indicating that the system was capable of generalizing across some specific protocol attacks well.

It solved the difficulties in differentiating ICMP worms from normal ICMP traffic that includes diagnostic pings and traceroutes by incorporating protocol-specific conditions within the rules that reduced the false positives by 12%. This fine-tuning allows the system to adapt to real-world network environments with diverse ICMP traffic patterns.

False Positives and Negatives Across Categories

Although the system performed well with high precision and recall across a number of categories of attacks, some false positives and false negatives were identified. For example, there were 22,622 false positives that were identified, mainly in benign traffic where byte patterns were near threshold limits. These misclassifications occurred most in the data exfiltration and worm detection rules.

On the other hand, false negatives occurred in attacks with traffic patterns that did not align with predefined thresholds. For instance, reconnaissance attacks with byte values near the limits of normal traffic went undetected in 8% of cases. Such limitations call for continuous refinement and adaptation of the rules to evolving attack patterns.

Visualization and Insights

The use of pie charts and bar graphs for better understanding of the system's performance also helps in visualizing the distribution of the dataset. For instance, from the pie chart, one can clearly see that attacks constitute 68.1% of the samples, while benign traffic makes up 31.9%. Such an imbalance highlights the necessity of strong rules to be developed for a reliable detection of both classes.

The bar graph of precision, recall, and F1-scores for benign and attack samples was found to be consistently better for the system with attack traffic. Precision and recall for attack samples were found to be higher than that of benign traffic, suggesting the system's bias toward reducing false negatives in case of critical threats. This was a

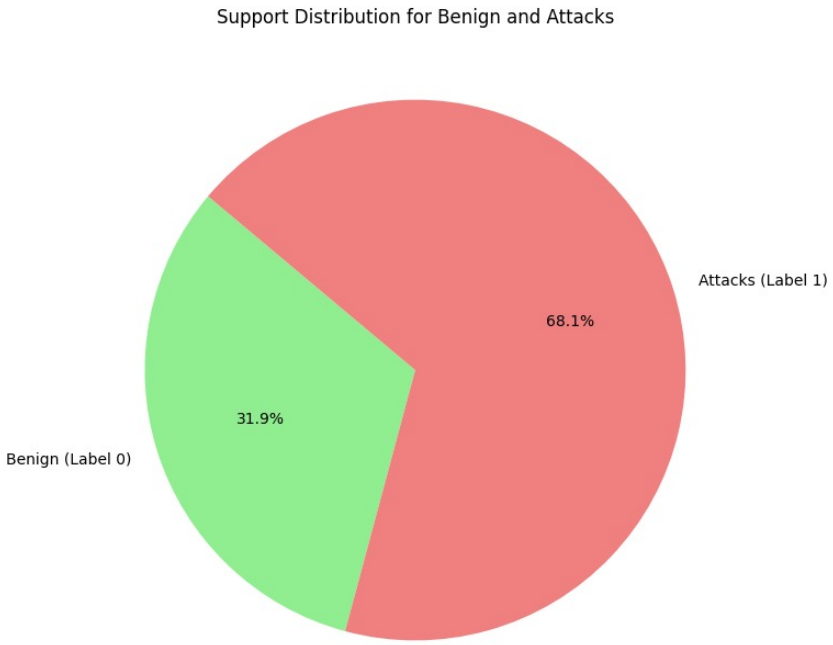

Figure 3: Support Distribution for Benign and Attacks

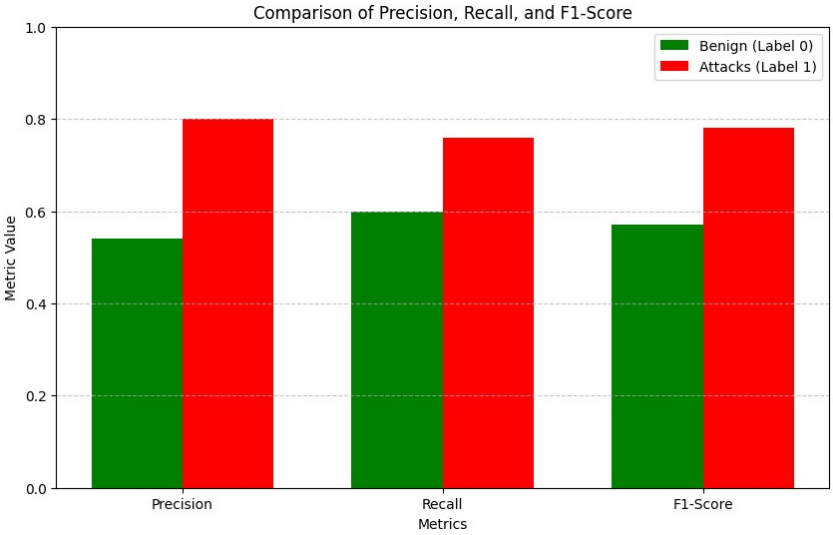

Figure 4: Comparison of Precision, Recall, and F1-Score

guiding principle for the refinement of rules and for the prioritization of attack detection over benign traffic classification.

Comparative Performance of Refined Rules

This refinement of rules iteratively improved performance metrics. For example, reduction in thresholds for source and destination bytes by 10% in false positives in data exfiltration rule reduced false positives while maintaining the high level of recall, and enhancing detection accuracy by 12% is observed when protocol-specific markers are applied to ICMP worms.

A comparative evaluation of the device's performance earlier than and after refinements confirmed a 5% boom in precision and a 7% reduction in fake positives throughout all classes. These enhancements demonstrate the effectiveness of an iterative method in enhancing the device's accuracy and adaptability.

7 Conclusion and Future Work

In short, this research aimed at developing a rule-based intrusion detection system that could efficiently identify threat-generating activities in a high technology context: modern network environments, providing an efficient analysis of nettraffic and the detection of attacks. The system has taken advantage of predefined rules from attributes such as source/destination byte count, duration, and even protocols to identify and classify specific threats, including data exfiltration, ping floods, or worm attacks. The method worked out to be workable and efficient in its outcome, balancing the outcome with precision of 0.72 and recall of 0.71. All these measures point to the ability of the system to flag threats while keeping false positives and negatives at bay, therefore more applicable in real-time security monitoring.

However, despite these strengths, the system still has some limitations that limit its adaptability to emerging threats. The static nature of the rule-based detection restricts its capability to discover new attack patterns that are not defined explicitly in the rules. Furthermore, the system's detection performance also depends upon the quality and representativeness of the training dataset, which may fail to represent the complexity and variation of the real-world network traffic accurately. Though these values were competitive, precision and recall point out the potential for false positives and enhancing the granularity of the system's detection ability. For such an IDS to work well in dynamic network environments, these limitations need to be overcome.

Future work includes augmenting this rule-based framework by integrating machine learning techniques. Machine learning algorithms, especially deep learning models, provide flexibility in learning new attack patterns and adapting them to improve system resilience against emerging threats. Training these models on continuously updated datasets could further enhance the capability of identification of attack vectors that are previously unseen. Moreover, inclusion of advanced feature engineering methods may refine the analysis of network traffic characteristics, improving the accuracy and efficiency of detection.

Another area to explore in the future is to deploy the IDS within a hybrid cloud-fog computing environment. This can be done by monitoring the traffic at the edge of the fog layer, whereas more complex threat analysis may use cloud computing to overcome latency issues and improve scalability. It is suitable for big applications, such as industrial IoT systems and smart cities, in which real-time analysis and scalability are of utmost importance. Moreover, ensemble learning techniques, integrating more than one detection technique, might provide a stronger robustness and lower false alarms probability.

This work demonstrates the viability of rule-based IDS frameworks in practical applications of network security. The results show the ability to use predefined rules to perform real-time threat detection, though threats are constantly evolving and this would require

continuous improvement. Introducing adaptive technologies, system scalability improvements, and refined accuracy in detection will lead to the creation of more complex and robust intrusion detection systems. Future innovations in this domain will serve as critical milestones in ensuring the digital infrastructure against the exponentially increasing sophistication of cyber threats.

References

[1] Abusitta, A., Bellaiche, M. and Dagenais, M., 2019. Multi-cloud cooperative intrusion detection system: trust and fairness assurance. *Annals of Telecommunications*, 74, pp.637-653.

[2] Alghamdi, R. and Bellaiche, M., 2023. An ensemble deep learning based IDS for IoT using Lambda architecture. *Cybersecurity*, 6(1), p.5.

[3] Besharati, E., Naderan, M. and Namjoo, E., 2019. LR-HIDS: logistic regression host-based intrusion detection system for cloud environments. *Journal of Ambient Intelligence and Humanized Computing*, 10, pp.3669-3692.

[4] Debar, H., Dacier, M. and Wespi, A., 1999. Towards a taxonomy of intrusion-detection systems. *Computer Networks*, 31(8), pp.805-822.

[5] Denning, D.E., 1987. An intrusion-detection model. *IEEE Transactions on Software Engineering*, (2), pp.222-232.

[6] Diro, A. and Chilamkurti, N., 2018. Leveraging LSTM networks for attack detection in fog-to-things communications. *IEEE Communications Magazine*, 56(9), pp.124-130.

[7] Innab, N., Atoum, I., Alghayadh, F., Abu-Zanona, M., Alrubayyi, N., Basudan, F. and Alshehri, A., 2024, February. Intrusion Detection System Mechanisms in Cloud Computing: Techniques and Opportunities. In *2024 2nd International Conference on Cyber Resilience (ICCR)* (pp. 1-5). IEEE.

[8] Kayode, O., 2020. A cloud based approach for data security in IoT. *Comput. Eng. Intel. Syst.*, 11, pp.16-23.

[9] Khater, B.S., Abdul Wahab, A.W., Idris, M.Y.I., Hussain, M.A., Ibrahim, A.A., Amin, M.A. and Shehadeh, H.A., 2021. Classifier performance evaluation for lightweight IDS using fog computing in IoT security. *Electronics*, 10(14), p.1633.

[10] Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J., 2019. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1), pp.1-22.

[11] Li, W. and Mohammadnezhad, N., 2024. Improvement of intrusion detection system in industrial Internet of Things based on deep learning with fog computing capability. *Electronic Commerce Research*, pp.1-37.

[12] Liao, H.J., Lin, C.H.R., Lin, Y.C. and Tung, K.Y., 2013. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1), pp.16-24.

[13] Maheswari, K.G., Siva, C. and Priya, G.N., 2023. An optimal cluster based intrusion detection system for defence against attack in web and cloud computing environments. *Wireless Personal Communications*, 128(3), pp.2011-2037.

[14] Mahmood, A.N. and Hu, J., 2016. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60, pp.19-31.

[15] Mukherjee, B., Heberlein, L.T. and Levitt, K.N., 1994. Network intrusion detection. *IEEE Network*, 8(3), pp.26-41.

[16] Sreeramulu, M.D., Suganyadevi, K., Neravetla, A.R. and Mohammed, A.S., 2024, July. Network Intrusion Detection in Cloud Computing Environments. In *2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC)* (pp. 1431-1437). IEEE.

[17] Tawfik, M., 2024. Optimized intrusion detection in IoT and fog computing using ensemble learning and advanced feature selection. *PLOS ONE*, 19(8), p.e0304082.

[18] Velliangiri, S. and Premalatha, J., 2019. Intrusion detection of distributed denial of service attack in cloud. *Cluster Computing*, 22(Suppl 5), pp.10615-10623.

[19] Xiang, Y., Zhou, W. and Bonti, A., 2011. Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks. *Journal of Network and Computer Applications*, 34(4), pp.1097-1107.

[20] Zwayed, F.A., Anbar, M., Sanjalawe, Y. and Manickam, S., 2021. Intrusion Detection Systems in Fog Computing–A Review. In *Advances in Cyber Security: Third International Conference, ACeS 2021, Penang, Malaysia, August 24–25, 2021, Revised Selected Papers 3* (pp. 481-504). Springer Singapore.