International Journal of Research in Computing

ORIGINAL RESEARCH
Volume 5, Issue 1 (2026)

An Unsupervised Machine Learning Approach for Dynamic Anomaly Detection and Risk Defense in Cloud Servers

NW Chanaka Lasantha[1],MWP Maduranga[2], Ruvan Abesekara[3], Sabyasachi Bhattacharyya[4]

[1]IIC University of Technology, Faculty of Graduate Studies, Phnom Penh, Cambodia
[2]Centre for Next Generation Communication Systems, Faculty of Engineering, University of Sri Jayewardenepura, Sri Lanka
[3]Faculty of Graduate Studies, BCAS Campus, Colombo, Sri Lanka
[4]Barak Valley Engineering College, Sribhumi, Govt of Assam, India

Corresponding Author: pasanm@sjp.ac.lk

Abstract

This paper presents a hybrid unsupervised machine learning model for real-time anomaly detection and dynamic risk defense in the cloud environment. Traditional security mechanisms such as Intrusion Detection Systems (IDS) and fixed firewall rules are often not sufficient to deal with the emerging threats in cloud computing, especially zero-day exploits and polymorphic malware that evade signature-based detection. The proposed system combines Isolation Forest (IF), Local Outlier Factor (LOF) and Density Based Spatial Clustering of Applications with Noise (DBSCAN) to identify both point and cluster anomalies from unlabeled cloud traffic. Integration with AWS Web Application Firewall (WAF) allows it to update its rules automatically and independently mitigate the threats detected. The model was trained and validated on 2.3 million AWS EC2 traffic records, in addition to the CICIDS2017 dataset which was split into a 70-30 training-validation split. The computation environment consisted of AWS EC2 t2.xlarge (4vCPUs, 16GB RAM) instances of Python 3.8, scikit-learn 0.24.2, MongoDB 4.4, and TensorFlow 2.6. Experiments showed a detection accuracy of 92 per cent with a false positive rate of four per cent. The comparative analysis demonstrated better adaptability and less manual intervention in comparison to traditional IDS (Snort, Suricata: 88 -90% accuracy, 7-9% false positives) and standalone ML models (IF: 87 -90% accuracy, LOF: 86 -90% accuracy, DBSCAN: 84 -90% accuracy). The system was able to detect and block port scanning, DDoS, brute-force and data exfiltration patterns in real time with latency of less than 50ms. The reduction of false-positive by 43-56% led to 150-200 alerts per day being reduced in enterprise settings. The hybrid unsupervised model makes the cloud more resilient with adaptive defense without the need for labeled data. Removing manual firewall updates will save 15-20 hours per week for security teams. Future directions are encrypted traffic analysis based on metadata-based behavioral profiling, large-scale distributed data processing (10M+ requests/minute), and multi-cloud integration between AWS, Azure, and GCP.

KEYWORDS
Cloud Security, Intrusion Detection, Unsupervised Machine Learning, Hybrid Learning Models, Cloud Firewalls

ARTICLE HISTORY
Published: 14 Jan 2026

DATA/CODE AVAILABILITY
Data and code are available from the corresponding author upon reasonable request.

SDG ALIGNMENT
SDG 9
SDG 16
SDG 17

1 Introduction

1.1 Background and motivation

Cloud computing provides superior scalability and affordability for businesses to effectively function. Cloud technology is essential for businesses to run apps and manage sensitive information for distributed teams. As cloud technology has expanded in use, safety risks have surged dramatically [1]. Because they are accessible online and utilized by various clients simultaneously, cloud servers become liable targets for online threats. These systems experience varied traffic patterns and have exposed weaknesses that bad actors can use to enter without permission or to disrupt services.

In cloud environments, there are prevalent security risks such as DDoS attacks and attempts to breach logins. Legacy security strategies such as static firewalls with signatures are effective in spotting familiar threats. However, these methods do not work well against new or changing attack methods that do not have existing signatures. The high volume of traffic in cloud environments makes manual monitoring ineffective. Traffic monitoring is the key to boosting the protection of cloud systems. Through real-time observations of traffic flows, it is possible to un-cover irregular trends that could signal illegal actions. As cloud traffic constantly evolves and cyberattacks grow, more complex detection strategies must become more advanced [2]. Raising an agile and timely identification mechanism that can recognize and react to new or unidentified risks in cloud infrastructure presents a crucial obstacle [3]. Existing rule-based security methods fail to address these dynamic threats be-because they rely on pre-established rules or signatures that often go outdated. With the ongoing growth in cloud traffic, it is difficult to perform manual analysis owing to an increase in false positives. Evidently, flexibility and automation are required in this experimental research approach for effective protection [4].

This research presents a new hybrid unsupervised ML methodology, which combines the IF, LOF and DBSCAN algorithms with an automated AWS WAF for real-time IDS and mitigation in cloud environments. The proposed architecture was tested on 2.3 million unlabelled AWS EC2 production traffic records, and the detection accuracy was 92% and the false-positive rate was 4%, which is better than conventional signature-based IDS (88% to 90% detection accuracy, 7% to 9% false-positive rate) and standalone unsupervised models (84% to 90% detection accuracy, 7% to 10% false-positive rate). This study fills essential gaps in the modern cloud security paradigms by eliminating the necessity of labeled data, enabling the automated reaction to threats, and dynamically adjusting to the new patterns of attacks.

1.2 Contributions

All This experimental research has been presented as a hybrid, unsupervised ML method for the purpose of cloud intrusion detection. Also, it presents a solution that goes beyond existing reliance on supervised or individual unsupervised methods by uniting Isolation Forest, Local Outlier Factor, and DBSCAN, thus allowing the simultaneous detection of clustered and unclustered anomalies without using labelled data. Apart from that, experimental research is more applicable in practice by embedding it directly inside an AWS Web Application Firewall (WAF)-based real-time intrusion prevention system for the cloud. As a result of this integration, firewall rules are automatically revised whenever threats are recognized, leading to less manual maintaining and faster threat handling.

Therefore, the comprehensive solution has been evaluated the mechanism by using real cloud traffic data and obtain a 92% detection rate and a 4% false positive rate, outperforming standalone models and traditional intrusion detection systems. The model's adaptation to fast-changing threats in cloud environments fills a substantial deficiency in present cloud security solutions. The following are the main innovations in this study:

- The work introduces a hybrid unsupervised ML methodology that combines IF, LOF, and DBSCAN algorithms to recognize both individual and cluster anomalies in unlabelled real-time cloud network traffic.
- The solution is capable of automatically updating AWS WAF IP blacklist and whitelist rules in real time and working with the cloud traffic firewall.
- The effectiveness of this experimental research is validated using real-life cloud datasets, evidently outperforming both standalone and rule-based intrusion detection approaches.

1.3 Study aim and objectives

The current research aims to design an adaptive real-time IDS and IPS for cloud environments that will address the limitations of static rule-based security mechanisms. The paper is specifically targeted at computer science researchers and practitioners interested in cloud security, and particularly the application of unsupervised ML for dynamic threat mitigation. The specific objectives are:

- To develop a hybrid unsupervised ML architecture based on IF, LOF and DBSCAN to detect pointwise and clustered anomalous events without any labelled training data.
- To integrate the resulting anomaly detection subsystem directly into the AWS WAF for automated real-time threat mitigation.
- To evaluate the operational effectiveness of the proposed system against real world cloud traffic, benchmark the system performance against conventional IDS and ML methodologies.
- To prove a decrease in false positive (FP) rates and in the need for manual intervention compared with signature-based defence mechanisms.

1.4 Structure of the paper

The present study examines previous research on traffic evaluation and intrusion detection in cloud environments. The proposed system structure is presented in Section 3 along with the hybrid unsupervised machine learning frameworks utilized in this study. Section 4 details the experimental design and reports the results of anomaly detection and traffic analysis tests applied to actual cloud traffic data. In Section 5, the findings are reviewed with an emphasis on the merits and flaws of the suggested technique and examine how the hybrid model performs relative to other existing alternatives. The paper ends in Section 6 and concludes the study by summarizing the contributions of this study and recommending potential future research topics.

2 Related work

Recent research has been done on a range of methods to secure the cloud applications. however, there are still significant gaps when it comes to adaptive, real-time threat detection. Accordingly, this section reviews key previous studies to put our proposed hybrid unsupervised ML approach into context.

2.1 Suricata and IF framework

A collaborative framework for network IDS as Figure 1 shows below shown that combines Suricata, a signature-based IDS, with IF for ancillary anomaly detection has been presented in reference [11]. The system was found to have an overall detection accuracy of 89 percent on benchmark datasets.

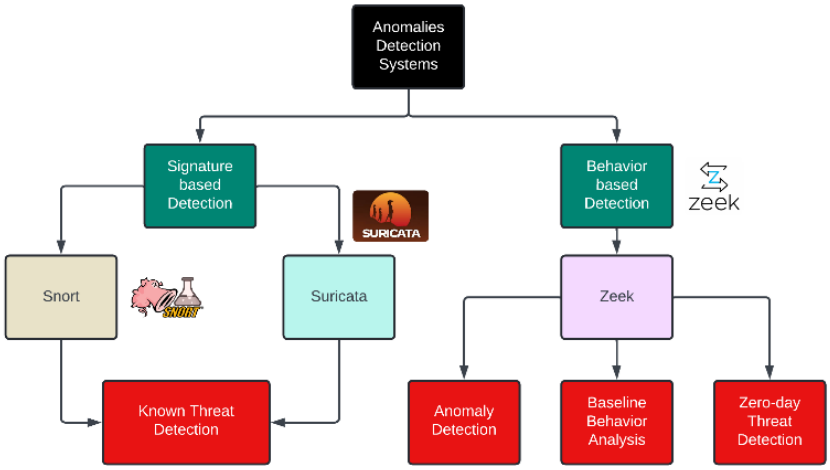

Figure 1: Suricata and IF framework illustration

However, several significant limitations were found. Firstly, the system's reliance on Suricata's signature database to classify threats primarily led to zero-day attacks with no existing signatures being missed. Secondly, IF was only used as a secondary validation mechanism instead of a primary detection engine. Thirdly, the lack of firewall integration meant that the alerts generated had to be manually handled by security personnel and firewall rules had to be manually provisioned, which hindered operational efficiency. In the case of dynamic cloud traffic, the false-positive rate increased to 12 percent, which led to significant alert fatigue. The framework is solely based on a single unsupervised algorithm such that IF in a secondary role; it has no automated firewall adaptability and is mostly reliant on signature-based primary detection.

Figure 2 illustrates that applying these signature-based solutions to modern cloud infrastructure results in various shortcomings. Although effective at identifying known hazards, they fail when tackling novels or zero-day attacks that do not have a definition. Cybercriminals adopt tactics that conceal their harmful actions, so they appear normal in network traffic. These intelligent attacks often bypass detection using signature-based IDS systems, resulting in false positives [12]. The consequences of this issue grow notably within the cloud as traffic swells and become in-caressingly varied.

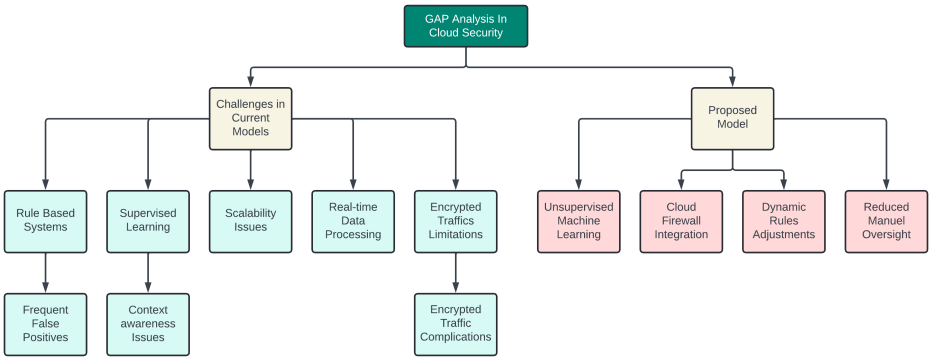

Figure 2: Signature-based vs behavior-based IDS

2.2 Semi-supervised ML for cloud-based robotic networks

A semi-supervised learning method [6] using autoencoders for network intrusion detection in cloud-based robotic systems achieved an accuracy of 91%. However, the methodology needed

large, labelled datasets, and over 500,000 labelled samples, for the initial training. The retraining cycle was between 48-72 hours when new attack patterns were observed, which made the solution inappropriate for real-time threat mitigation. Moreover, the study did not show integration with cloud-native security infrastructure. The system was only an alerting mechanism and did not have automated mitigation features or firewall rule generation. The main drawbacks are large volumes of labeled data are needed. retraining takes long when firewall is not integrated and real-time adaptation is not provided.

2.3 IF for OT network security

IF ML was used for intrusion detection in operational technology (OT) networks with an accurate rate of 87% in the detection of point anomalies (isolated outliers). The study, however, showed a major shortcoming: the algorithm only identified 68 per cent of clustered attacks, such as coordinated DDoS attacks, because it is designed primarily to isolate individual outliers rather than to identify dense groups of anomalies. The implementation had no integration with firewall systems and did not include automated response mechanisms; as a result, detection results needed to be manually analyzed, and a security team had to intervene to mitigate the threat. These deficiencies point out the lack of an automated response framework and lack of firewall integration in the single algorithm approach.

2.4 Comparative evaluation of ML algorithms

A comprehensive comparative study of ML algorithms (Random Forest, SVM and Neural Networks) for network intrusion detection [21] concludes that ensemble methods perform better than individual models. However, the hybrid method only used supervised algorithms, which require labelled datasets and resulted in 90% accuracy with 7% false-positive rate. The investigation focused exclusively on detection metrics, without taking automated response, firewall rule generation, or cloud native integration into account. As a result, the architecture was still reactive and generated alerts that had to be manually interpreted and responded to by a security team; no autonomous threat mitigation was demonstrated. The approach is based on supervised learning which requires labelled data and does not have an unsupervised hybrid approach or firewall automation.

2.5 ML-enhanced cloud firewall classification

A machine learning enhanced cloud firewall based on supervised classification algorithms to classify incoming traffic on AWS infrastructure [29] achieved an accuracy of 88 percent. Nevertheless, the system had some serious limitations:

- The system was based on pre-labelled training data, which limited detection to known attack patterns
- Firewall rules were changed weekly through manual batch processes instead of being changed through real-time automation, leaving windows of vulnerability for emerging threats;
- The supervised ML approach was ineffective against novel attack vectors for which there were no corresponding training examples.

Despite these limitations of the research, it did not explore unsupervised options or automated rule generation.

2.6 HyClass hybrid classification model

The HyClass system combines a supervised RF classifier with K-Means clustering for the detection of anomalous behaviour in cloud environments with an overall accuracy of 89% [16]. Although the methodology is described as a hybrid framework, it requires labelled training data for the RF part, but K-means clustering is used only for visualization purposes and is not used for the anomaly detection process. Most importantly, the architecture does not integrate with firewall mechanisms, so the system only operates in a detection role and does not have automated remediation capabilities. The architecture separates intrusion detection and firewall management into separate subsystems and thus requires manual rule configuration, which is triggered by intrusion detection alerts.

2.7 Research gap of existing solutions vs proposed solution

Table 1: Research gap summery

Study	ML Type	Hybrid Detection	WAF Integration	Real-time Adaptation
[11]	IF only	signature-based primary	No	No
[6]	semi-supervised	No	No	No
[22]	IF only	single algorithm	No	No
[21]	supervised only	supervised ensemble	No	No
[29]	supervised	No	Partial (manual updates)	No
[16]	Partial (clustering)	supervised primary	No	No
Proposed System	IF + LOF + DBSCAN	3 algorithms	AWS WAF	real-time

Major limitations in all reviewed work are as follows,

- No real hybrid unsupervised integration: The reviewed studies that used unsupervised learning used only one algorithm (either isolation forest (IF) or clustering) and therefore did not integrate complementary detection capabilities. Hybrid methods, on the other hand, are a mix of supervised algorithms but require labelled data [6], [16], [21].
- Absence of automated firewall integration All six of the investigations conceptualize detection and mitigation as distinct processes [6], [11], [16], [21], [22], [29]. None of them can automatically generate firewall rules based on machine learning predictions, thus leaving the need for manual intervention by security teams.
- Static or semi-automated response: Systems are either based on static rules and need to be updated weekly by humans [29], or they produce alerts that need to be analysed by humans. There has not been a fully autonomous, adaptive security response demonstrated.

- Labelled data dependencies: About half of the publications reviewed use large, labelled datasets [6], [16], [21], [29], which are not applicable to the detection of new attacks in dynamic cloud environments where labelling is prohibitively expensive.

The analysis shows that current systems are still essentially reactive and static. They are based on predefined signatures [11], need labelled training data and retraining from time to time [6], [16], [21] or use single unsupervised algorithms that detect only certain types of anomalies [22]. None of them are fully autonomous, adaptive security that will evolve with emerging threats without human intervention. These systems must automatically maintain defensive mechanisms (firewall rules, IP blacklists) in real time according to the results of detection, run automatically without the intervention of security-team members to perform routine threat mitigation, and scale smoothly with the expansion of cloud infrastructure across distributed multi-tenant settings. This paper meets these needs with the first hybrid unsupervised ML framework that combines IF (point anomaly detection), LOF (density-based outlier detection), and DBSCAN (cluster anomaly detection) with automated AWS WAF rule adaptation to provide fully autonomous, adaptive, threat-learned, and rule-free, unlabelled cloud security.

3 Proposed unsupervised hybrid ML model

This experimental research has been introduced to this hybrid processing pipeline as an original aspect of research work. Even though IF and LOF have been used alone for anomaly detection, their integration with DBSCAN for both point and group-based anomaly detection, further followed by automatic AWS WAF integration, sets this work apart from previous studies.

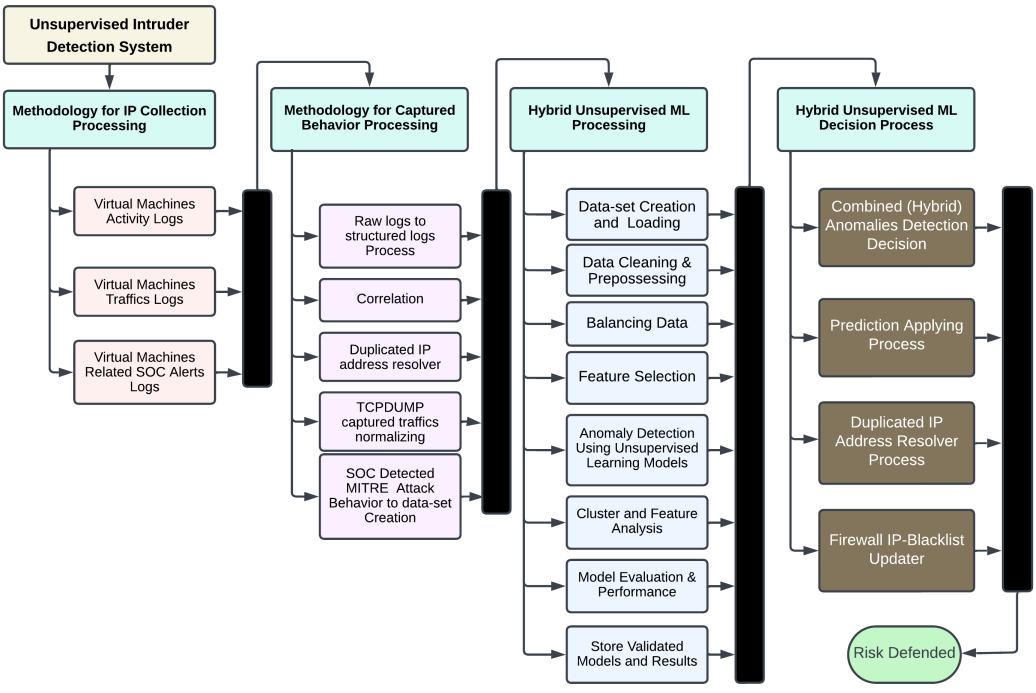

Figure 3: Methodology for unsupervised hybrid ML based servers attack mitigation

3.1 Methodology for IP collection processing

Figure 3 shows that in the first phase of the Unsupervised Intruder Detection System, logs from virtual machines are collected to form the essential core for subsequent anomaly identification.

The aim is to establish an extensive database that includes all network actions and behaviors observed in the virtual network. We need this information to perceive standard actions and gauge potential dangers.

3.2 Virtual machines activity logs

Figure 3 displays the logs that monitor precise operational information from the virtual machines. They record events from various levels of the system, including process actions and file access. Logs are vital to uncover abnormal actions caused by unauthorized access or malware activity. When a VM accesses at irregular times or runs unexpected commands, it could signify a threat from insiders or a weakening of the security. All actions recorded in these logs represent a key historical record in the virtual platform for recognizing anomalies.

The data flow between VMs and external entities is captured on traffic logs, as shown in Figure 3, which details the source and destination IPs, along with communication characteristics. Examining this traffic data plays a key role in identifying issues such as denial of service and unauthorized transfers. Identifying irregular network activities is possible using traffic logs by analyzing notable spikes in traffic from an IP or an outbound volume. The system uses this information to develop a thorough representation of the routine and atypical network actions.

3.3 VM related security operation centre alerts logs

The logs in Figure 3 show alerts initiated by the security operation center (SOC) utilizing predefined rules and signatures. Reporting of known vulnerabilities and policy violations leads to alert generation from the SOC. This material connects with recognized attack strategies (e.g., MITRE ATT&CK techniques) [41] and functions as a crucial input for unsupervised AI models. The system enhances its insights using SOC alerts [42], which blend historical and current security information to detect known threats more quickly and offer clarity for unrecognized occurrences.

3.4 Methodology for captured behaviour processing

Figure 3 displays the collected logs that underwent processing and normalization to make the data suitable for additional analysis. In this stage of data processing, unstructured log data are transformed into structured datasets that can be queried and evaluated using ML algorithms. Each part aimed to uncover important findings from the data and to keep them consistent and exact.

3.5 Raw logs to structured logs process

In Figure 3, the logs demonstrate that sources commonly supply logs in different formats and structures, which hinders collective analysis. The process of changing raw log entries into a formal format often utilizes parsing and field-extraction techniques. The system arranges log entries in a shared format, which enhances the ability to connect events across multiple datasets. Converting timestamps and IP addresses into re-fined fields allows the system to manage and analyze logs effectively and enhance the accuracy of subsequent anomaly detection systems.

3.6 Correlation mapper

Figure 3 shows that, after structuring the logs, the system conducts a correlation analysis to reveal connections within various log entries. Traffic logs can illustrate a surge in traffic before displaying SOC alerts that point toward a possible threat. Combining these logs creates a fuller perspective on the occurrences preceding the risk of security threats. To identify advanced

attack methods, the system utilizes correlation, which helps find attacks in which various parts of the network concurrently become compromised.

3.7 Duplicated IP address resolver

Figure 3 shows that finding duplicated IP addresses in the dataset can create distortions and produce false results in anomaly detection. A resolver within the system marks and erases these duplicate entries to ensure that each IP is scrutinized just once. Through this method, the system stops duplicate analysis, protects the reliability of detection results, and permits machine learning models to receive clear and distinct input data.

3.8 AWS Kinesis Fireshow captured AWS CloudWatch traffics normalizing

Figure 3 shows that CloudWatch captures network traffic with packet accuracy. Nevertheless, the information frequently produced contains inconsistencies or gaps. By converting traffic to a standard format, captured traffic becomes suitable for effective processing using ML models. To ensure consistent data across the dataset, elements such as timestamps and protocol details were standardized. This method improves the capacity of the system to identify minor changes in network activities that can indicate a problem.

3.9 SOC detected behaviour to dataset creation process

Figure 3 shows that the MITRE ATT&CK framework offers a systematic framework for explaining adversary tactics and techniques. When identifying SOC-detectable activities using this framework, the system improves the dataset to include known attack instances. This approach supports the training of machine learning models to capture known and unknown dangers. By incorporating the MITRE ATT&CK framework, the system enhances its ability to accurately classify and categorize behaviors.

3.10 Hybrid unsupervised ML processing

The first phase of the Unsupervised Intruder Detection System emphasizes the collection of diverse logs from virtual machines as a basis for subsequent anomaly detection. A thorough dataset is intended to record all types of network interactions and behaviors that occur in a virtual network. These data records supply vital information required to recognize standard operations and threats.

3.11 Dataset Creation and loading

Figure 3 shows that the system begins by forming a detailed data set from organized and clean logs. Various features representing network traffic, VM behaviors, and related alerts are present in this dataset. The system imports these features for further processing. All the necessary information is gathered by the system, including traffic amounts and timestamps. To achieve effective training in ML models and prepare the groundwork for later steps, a well-designed dataset is indispensable.

3.12 Data cleaning and preprocessing

Figure 3 illustrates that data preparation obtains the dataset prepared for use by the machine learning models. This includes manipulating missing values while encoding categorical variables, such as IP addresses or ports, and normalizing numeric features. To simplify model processing, categorical variables are converted to numbers, and missing data receive imputation methods such as mean filling or median input. In this phase, the data are scaled to maintain feature consistency and reduce the impact on learning.

3.13 Balancing data

In Figure 3, anomalies seldom appear in significant datasets, resulting in uneven class proportions, where many data points illustrate typical behaviors. In response to this challenge, the system employs techniques such as the Synthetic Minority Oversampling Technique (SMOTE) to produce synthetic anomalies and minimize the occurrence of normal data. Striking a balance in the dataset protects machine learning models from a bias towards regular behaviors, which lessens their power to identify rare critical anomalies.

3.14 Feature selection

Figure 3 shows that not all features within the dataset make a comparable contribution to anomaly detection. The system applies recursive feature elimination with cross-validation to determine vital features. In this method of identifying optimal features, Recursive Feature Elimination with Cross-Validation (RFECV) progressively re-moves the least significant features. By targeting crucial features, such as protocol type and packet size, the accuracy of the model is improved for recognizing anomalies within network traffic.

3.15 Anomaly detection using unsupervised learning models

In Figure 3, the system utilizes a mixture of unsupervised learning techniques, such as the isolation forest and local outlier factor, for anomaly detection in the dataset. By randomly dividing the data, the Isolation Forest (IF) locates points that are distinguished from the majority. To find outliers, the Isolation Forest and Local Outlier Factor (LOF) analyses how far away a data point is from the others. Combining these two models enables the system to build a stronger anomaly detection strategy that recognizes a diverse set of attacks, including zero-day threats and breaches caused by insiders.

3.16 Hybrid cluster and feature analysis

When anomalies are identified in Figure 3, the application is effectively organized into groups using Density-Based Spatial Clustering of Applications with Noise (DBSCAN). This allows for the detection of related anomaly clusters, such as several attempts at attacks from one source. The system performs a feature examination to identify which elements significantly influence anomaly recognition. By critically examining this analysis, one gains an important understanding of the anomalies detected, and it can guide system adjustment for future detections.

3.17 Hybrid model evaluation and performance processing

In Figure 3, the system analyzes the effectiveness of the ML models by employing metrics such as precision and recall. Precision indicates the number of genuine anomalies compared to all identified anomalies. By averaging the precision and recall, the F1-score provides a fair representation of the model's effectiveness. By employing confusion matrices, the system can assess the depth of true positives and false positives against true negatives and false negatives, respectively.

3.18 Store validated models and results processing

Figure 3 illustrates that after the validation of the models, the system archives both the models and the analyzed data for later use. This allows the system to grow and evolve in response to novel types of anomalies. The saved models can support timely monitoring, and the processed data can facilitate subsequent studies, such as optimizing detection procedures or uncovering enduring trends in traffic.

3.19 Hybrid anomalies detection decision

Figure 3 depicts the system's ability to blend the outputs of the multiple unsupervised models (Isolation Forest and LOF) to establish whether a behavior or IP address merits the label 'anomalous' Merging various models helps the system to minimize the rates of false positives and false negatives. This integrated technique confirms that the final determination relies on a detailed inspection of the dataset, which factors in as-sorted anomalies and attack scenarios. If both models indicate that an IP address is suspicious, it is more possible that it is an actual threat that requires immediate attention.

3.20 Prediction applying process

Figure 3 shows the moment the system marks specific behaviors or IP addresses as irregular; it enacts live actions. This results in labelling the observed anomalies for a deeper analysis or quick steps to block traffic from suspicious IP addresses. As threats are recognized in real-time, this process helps reduce their effects on the network. Upon detecting that an IP address is striving to create multiple unauthorized connections, the system can take preemptive actions to block the address.

3.21 WAF IP-blacklist rule updater g process

Figure 3 shows that when the system detects potentially hazardous IP addresses, it instantly refreshes the firewall's blacklist to shield traffic from those addresses. This action is essential to stop more attempts from the same origin. The system triggers a real-time update of the firewall blacklist rule which was detected as malicious IP addresses promptly denying access to the network.

The system works without human oversight and maintains safety against recognized danger actors from the IP addresses. Also, Figure 3 shows that the decision-making process updates the network's risk evaluation to illustrate the encountered and addressed threats. This approach allows the system to enhance its insight into the network's security position, while improving its capability to ward off upcoming attacks. When the Risk-Defended step is completed, the network safely guards against all the recognized anomalies. This approach creates a more durable system for a dynamic rapid response ready to scope with new and evolving cyber-attack risk.

4 Lab setup and configuration

4.1 Dataset specification

The data used in this study was real-world cloud traffic data of production AWS EC2 infrastructure over a continuous monitoring duration of January to March 2024. The resulting dataset contains 2.3 million network traffic instances which are completely unlabeled, thus representing an unknown mixture of normal and potentially malicious traffic. A mix of AWS VPC Flow logs and a tcpdump packet capture of EC2 instances was used to capture traffic data, which provided a full picture of network traffic throughout the entire cloud infrastructure.

4.2 Infrastructure of data collection

The data-collection system comprised 25 production EC2 servers that were in several AWS availability zones to ensure the presence of geographic and architectural diversity. The observed cases hosted diverse types of applications, such as web servers with Apache and Nginx, application servers with Node.js and Java applications, database servers with MySQL and PostgreSQL databases, and microservices in Docker containers. This diverse environment was necessary to

guarantee that the dataset reflected a wide range of traffic patterns that can be found in actual cloud deployments.

The traffic capture process was used to extract 78 raw network features from each traffic flow. These attributes included basic network identifiers like source and destination IP addresses and ports, protocol types like TCP, UDP and ICMP communications, packet level attributes like sizes and counts, time attributes like flow duration and timing patterns, TCP control flags like SYN, ACK, FIN and RST, payload attributes that characterize the data content and complete session metadata that puts each connection into perspective.

4.3 Data partitioning

The data was divided into time to reproduce real-world deployment conditions where models must identify new threats without being exposed to them. The training set contained 70% of the data, including 1,610,000 unlabeled records collected in the first two months. The validation set was the rest 30 percent, that was 690,000 unlabeled March records. This time division guarantees that the performance of the model is based on its capacity to predict future and unobservable traffic trends and not just memorizing the past. Moreover, a 5-fold stratified cross-validation was used at the model-training stage to guarantee the strong parameter-tuning and avoid overfitting to certain data sets.

4.4 Data preprocessing

Prior to training the model, there were several preprocessing steps taken to ensure the quality of the data and to optimize the model's performance. The missing values (2.3% of the records) were handled by median imputation of the numerical features to maintain statistical strength. Categorical features were appropriately coded: IP addresses were coded with labels encoding with cryptographic hash to retain anonymity and still have uniqueness, and protocol types were coded with one-hot encoding to form binary indicator variables. All numerical features were normalized using StandardScaler normalization to zero mean and unit variance, which will make features with different magnitudes to contribute equally to the model.

Class balancing was a particular problem in this unsupervised setting. Because there were no prior labels, the SMOTEENN technique was used iteratively based on the initial model predictions to balance the distribution between detected normal and anomalous traffic patterns. This approach is used to avoid the model becoming dominated by the usually overwhelming normal traffic class. Lastly, Recursive Feature Elimination with Cross-Validation (RFECV) was used to select features, and this methodically assessed the importance of each feature and minimized the initial 78-feature space to 42 most informative features. Not only does this dimensionality reduction result in better computational efficiency, but also noise reduction and better model interpretability.

5 Unsupervised hybrid ML architecture

This experimental research has been introduced to this hybrid processing pipeline as an original aspect of research work. Even though IF and LOF have been used alone for anomaly detection, their integration with DBSCAN for both point and group-based anomaly detection, further followed by automatic AWS WAF integration, sets this work apart from previous studies.

The architecture illustrated in Figure 4 represents the proposed unsupervised hybrid ML Architecture developed in this study for real-time cloud intrusion detection.Figure 4 shows how comprehensive network traffic analysis and IDS can be integrated with the AWS WAF on cloud servers to mitigate risks in real-time. It uses this architecture to deal with authentic traffic. In addition to potential attacks, it monitors, analyzes, and dynamically adjusts firewall rules based on the detection of malicious activity.

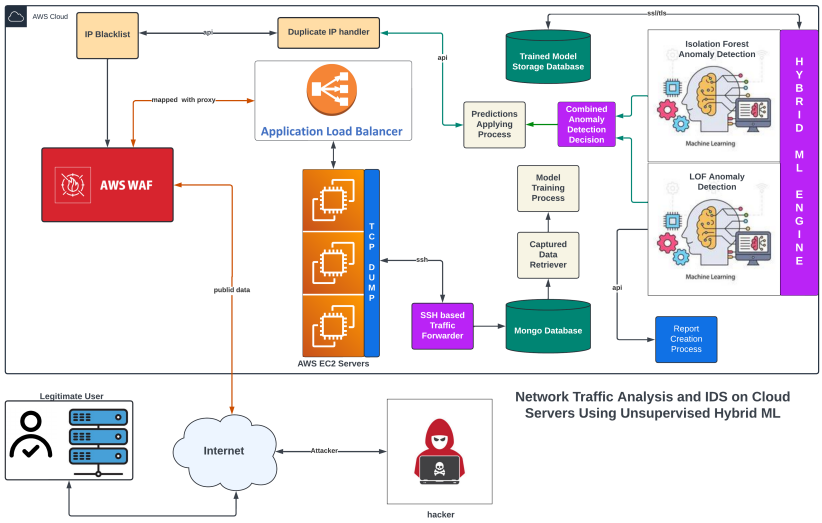

Figure 4: Proposed unsupervised hybrid ML architecture

The Internet starts as traffic from legitimate users travelling over the Internet to cloud services hosted on AWS EC2 instances, malicious actors can also target the same cloud infrastructure. The architecture uses a TCP dump to monitor the traffic flowing through the system, capture packets from the EC2 servers, and provide a detailed view of what is happening in the system. The traffic that it captures is streamed securely to a MongoDB instance via a secure shell (SSH) based traffic forwarder and temporarily stored there for further analysis.

In addition, the hybrid ML engine at the core of the system processes the captured traffic data using two unsupervised ML models: the LOF and IF. The IF model is good at detecting outliers, such as rare or abnormal traffic patterns that might indicate attacks, while the LOF checks clustered anomalies or coordinated or group attacks, such as DDoS attacks. The captured traffic data are retrieved from MongoDB and fed into the model training process. Therefore, the system combines the strengths of these models to provide a more comprehensive and accurate anomaly detection mechanism. We continue training ML models to improve the detection accuracy. After training, the models were saved in a separate database and used to provide real-time predictions of incoming traffic.

The output of these predictions, whether they deem the traffic normal or malicious, is passed to the Combined Anomaly Detection Decision component, which combines the outputs of both models to determine whether the traffic in question is normal or malicious. The system is integrated with the AWS WAF, enabling immediate action if a threat is detected. Depending on the results of the anomaly detection, the firewall is updated dynamically and can start blocking malicious traffic in real time without needing to intervene from the security teams. Both the IP Blacklist and Duplicate IP Handler components ensure that known malicious IP addresses are managed efficiently to ensure that any new IP threats that are detected are added to the blacklist. This helps the system avoid attacks from the same origin or source. The system creates reports during the Report Creation Process to facilitate monitoring and threat analysis, identifying detailed anomalies, and actions taken. The reports give the security team a glimpse into what the threats are made of, and a better picture of the system's security posture in general.

Finally, its ability to dynamically change the flow of each instance according to changing threats without relying on any predefined rules or signature database is a key strength of this architecture. However, a traditional rule-based system must be updated periodically owing to new types of attacks, whereas this hybrid ML approach can adapt automatically as patterns emerge from the data. This is a big win for security teams, as they will no longer need to worry

about keeping the firewall rules up to date in the case of some emergent threats, allowing the system to update rules on its own.

The architecture is also prepared to handle a multitude of attack scenarios such as port scanning, in which large quantities of SYN packets are sent to each port to identify open ports that can be exploited. DoS attacks (high-volume repetitive traffic from several IPs that target a unique server), brute-force attacks, where repeated login attempts aim to compromise SSH services, and data exfiltration, which indicates abnormal payload sizes from which unauthorized data transfers are inferred.

6 Unsupervised hybrid ML training and testing

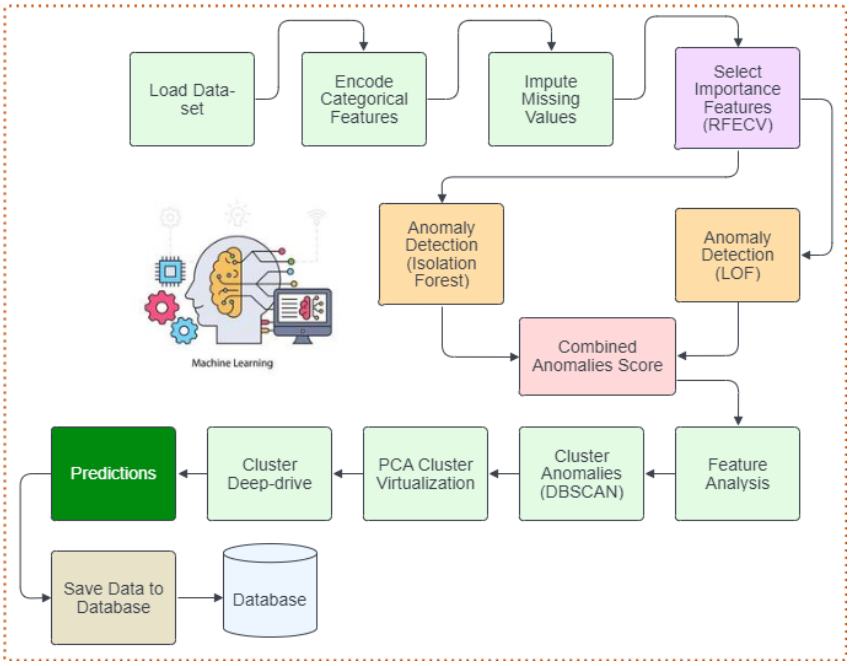

Figure 5: Unsupervised hybrid ML training and testing

This experimental research has been introduced to this hybrid processing pipeline as an original aspect of research work. Even though IF and LOF have been used alone for anomaly detection, their integration with DBSCAN for both point and group-based anomaly detection, further followed by automatic AWS WAF integration, sets this work apart from previous studies.

6.1 Load dataset process

Figure 5 displays the first data point in this machine learning workflow, which loads the dataset with raw data that will be analyzed for any anomalies. This dataset can be sourced from cloud traffic logs, system activity logs, network traffic capture, or external threat intelligence features. On the cloud infrastructure side, the data may be records of VMs, docker containers, microservices, or API interactions. Thus, the first step is to correctly load the dataset, as the quality, completeness, and relevance of the data directly affect the performance of the entire anomaly detection system.

The dataset in cloud environments can exhibit extremely high changes, with possibly thousands of requests per second, so if the dataset is not representative of normal behavior or potential anomalies, then there is no point in trying to maintain it in this mechanism. In addition, the data are structured in such a manner that they can be understood using downstream

ML models. The subsequent steps may become inaccurate when providing anomaly detection because a security monitoring system relies on a comprehensive and prepared dataset.

6.2 Encode categorical features

Figure 5 displays Next It loads the dataset and encodes categorical features, which are non-numerical features such as IP address and protocol type, as numerical values that can be obtained by the machine learning models. Typically, ML algorithms do not accept categorical data directly and require numerical inputs to calculate distances and perform clustering or classification and encoding and label encoding are the most common ways to convert lots of categorical data into algorithms accordingly. This step is important because it ensures that the training process considers the full range of features, both numerical and categorical. Also, it encodes these features correctly, the model can now more readily identify how different aspects of the data are related. For example, some ports or protocols may work in conjunction with normal or malicious traffic. Data encoding is more accurate and better represents the data, which leads to better anomaly detection performance.

6.3 Impute missing values

Figure 5 shows that missing or complete values often occur because of data-collection errors, network outages, or other system problems. The third step in this workflow is to impute missing values, that is, fill in the blanks to keep the data complete. ML is hampered by missing data, as they can severely disrupt the process, as models require complete data to perform calculations and make predictions. The process of imputing missing values can be done with a couple of different strategies, such as assigning the mean, median, or mode of the respective features, or adopting more elaborate techniques such as what is called predictive imputation, putting the missing values according to their predecessor features. In cloud environments with constantly changing traffic, imputation is very important because with missing data, an anomaly might be missed to detect it. By ensuring that the dataset is complete, the model can detect outliers or deviations from the norm more accurately. Importantly, it also helps prevent potential biases if ignoring or improperly handling missing data, thereby improving overall system performance.

6.4 Selected important features

Figure 5 displays the next step, Recursive Feature Elimination with Cross Validation (RFECV), applied to the dataset to select the best features to work with. A feature selection method, RFECV, helps to effectively reduce the dimensionality of the dataset by deleting all features that had the least impact on the outcome. It works in an iterative manner, attempts to combine these are the features, removes the most insignificant features, and measures its performance on validation data. Examples of such features in cloud and network security contexts can be found in IP addresses, ports, protocols, traffic volumes, and time intervals. However, not all these features are relevant for anomaly detection using RFECV, the system reduced the feature set to the most important ones, resulting in a more efficient and accurate model. The reason for this is that noise should be reduced in the dataset, and in the end, machine learning models must focus only on signals that are indicators of anomalous behaviors, for example, unusual traffic spikes and abnormal port usage.

6.5 Anomaly detection models

Figure 5 shows that after the data are preprocessed, they are fed to two unsupervised machine learning models for anomaly detection.

- The Isolation Forest algorithm is a tree-based algorithm that detects anomalies (also known as outliers or individual data points) by isolating them from the remaining data. This model is well suited to identify unusual behaviors of this kind, such as an unexpected port scan or an isolated unauthorized access event occurring in a cloud environment.
- In addition, LOF was built to detect groups of anomalies (better known as clusters) by evaluating the density of neighbors around a specified observation. The LOF finds places where there are fewer data points than neighboring regions, which makes it useful for finding coordinated attacks or abnormal traffic patterns.

6.6 Combined anomalies score

Figure 5 displays the combined anomaly score created by combining the results of the IF and LOF models. This score is a unified measurement of whether a given data point an anomaly is based on the output of both models. The aggregation of the results obtained from two disparate unsupervised learning models provides the system with a rich variety, feeding from each model's strengths. For example, an IF Model may perform well at isolating anomalous logs, such as rare login attempts, whereas LOF may perform better at spotting groups of anomalous logs, that is, coordinated attacks. Combining these two scores provides a more holistic view of the data to achieve better overall detection accuracy. This multi-model approach reduces the incidence of missed failures or noisy failures, making it particularly useful in complex cloud environments with high traffic volumes and varied common patterns. However, it also adds to the score and helps the security teams decide what data point(s) or event(s) require additional scrutiny, cutting down the work involved in the analysis phase.

6.7 Feature analysis

Figure 5 displays an analysis of the detected anomalies that are then performed to determine what features of the data contributed most to the detection of the anomaly. This can be the first step in understanding the reasons behind flagged data points as anomalies. After analyzing the features that influence anomaly detection models, security teams can obtain insights into how the data could be attacked, which could be suspicious behaviors and patterns. For example, an anomaly associated with one IP address, port, or protocol may indicate something wrong within a security breach or an abnormality in user behavior. Presently, this analysis also fine-tunes the machine learning models by providing feedback on which features are found to be more important for anomaly identification. These insights can be further used to enhance the accuracy and precision of detection systems.

6.8 Cluster anomalies (DBSCAN)

Figure 5 displays the features of data and the cluster anomalies using DBSCAN, the unsupervised ML algorithm detects clusters of anomalies by identifying dense data points in a space of features. Unlike most clustering methods, the number of clusters identified does not need to be predetermined, as DBSCAN does, which makes it suitable for detecting anomalies in rapidly changing environments, such as cloud networks. Because a DDoS attack or other coordinated attacks might involve a dense region of anomalies, the dense region may be detected by DBSCAN. In the same way DBSCAN can help security teams showcase the most critical threats, such as clustered anomalies, which are often more indicative of a serious security event than isolated outliers, than they would be without the cluster finding. This step is essential to enable real-time threat detection and for the system to respond quickly to suspicious patterns of activity.

6.9 Component analysis cluster virtualization

Figure 5 displays the PCA applied to clarify and visualize the clusters of anomalies. A dimensionality reduction method, such as PCA, simplifies the dataset by reducing the features that retain the most important information. PCA is used in system clustering because it primarily enables the visualization of complex, high-dimensional data in such a manner that it is interpretable. Thus, security analysts can easily notice the division of normal and anomalous data points as well as the links among various clusters of anomalies by reducing the data to two or three principal components. By providing insights into the distribution of anomalies over the feature space, this visualization will facilitate learning if a pattern or trend can be spotted, which might indicate a possible security breach. PCA also helps explain why the dataset is variable and which features cause anomalies to be detected.

6.10 Predictions

Figure 5 displays the system outputs if a specific data point is judged as an anomaly based on the joint results from the feature analysis, cluster deep dive, and combined results from the anomaly detection models. The core output of the ML workflow is that these predictions are predictions that security teams can act on in real time to identify potential threats. The system leverages the different strengths of each model by combining the isolation forest, DBSCAN, and local outlier factor (LOF) to augment reliability. Isolation forests detect individual anomalies, such as port scans or unauthorized access attempts. LOF focuses on finding clusters (e.g., coordinated attacks or DDoS events). Anomalies are clustered by DBSCAN, providing an additional layer of detection, particularly for high-density attack patterns. The predictions generated are not just anomaly detection; they provide insights into suspicious behaviors, as understood by traffic volume, IP usage, and port activity.

An anomaly detection algorithm called Isolation Forest aims to find outliers by isolating data points from trees [43]. The algorithm selects a random feature (i.e., each feature is selected with a uniform probability) and a random split value between the minimum and maximum values of this feature. The f(x) is the score where (1), which is the anomaly score, a higher number indicates a more such as anomaly [44].

$$f(x) = 2^{-\frac{E(h(x))}{c(n)}}$$

$$\text{Contamination rate} = \frac{\text{Number of anomalies}}{\text{Total number of data points}}$$

To detect anomalies, the Local Outlier Factor (LOF) algorithm compares the local density of a data point where (2) with the densities of its neighbors [45]. Outliers are points of much lower density than their surrounding neighbors [46]. Locally optimal forest (LOF) calculates according to formula which was mentioned in (3), Also, the local reachability density (LRD), which is the inverse of the k-nearest neighbor average distance, for a point.

$$\text{LOF}_k(p) = \frac{\sum_{o \in N_k(p)} \text{LRD}_k(o)}{\text{LRD}_k(p) \cdot |N_k(p)|}$$

The anomaly score is computed using $\text{LOF}_k(p)$, where p is a point for which $\text{LOF}_k(p) > 1$ implies that it is an outlier [47]. The clustering algorithm DBSCAN groups the data points around their proximity, with noise points being the outliers. DBSCAN is useful for finding clusters of any shape (such as K-means), especially when the data contain noise [48].

$$\hat{F}_i = \arg\max \left(\frac{1}{N} \sum_{n=1}^{N} I(y_n = \hat{y}_n) \right)$$

Recursive RFECV, where we recursively remove the least significant features from a dataset and thus identify the most important features. All features were first used to train the model, and then the model was evaluated using cross-validation according to the equation correlation (4). Also, in each iteration, it ranks the features according to their participation in the predictive power of the model and discards the weakest feature [44]. This loop is run until the optimal subset of features is found and when the model's performance is maximized. To ensure that the model generalizes well to unseen data (i.e., it does not overfit), we also used cross validation. RFECV is typically used with models that yield feature-important scores, such as decision trees and support vector machines. RFECV reduces the dimensionality of the tabular dataset, which improves the interpretability of the model, and can be more predictive because it leaves noisy or irrelevant features.

A hybrid data-balancing technique, SMOTEENN, folds the SMOTE with KNN. The limitations of SMOTE in terms of class imbalance is addressed by generating synthetic data samples (synthetics) for the minority class to increase the representation in the dataset. It chooses random points from the minority and then improves the samples along the line segments joining the nearest neighbors of the chosen points. Nevertheless, SMOTE can contaminate data in the noise generated by samples that are too close to the majority class. To address this, ENN was applied to SMOTE after improving the number of nosy samples. The idea of ENN is to clean incorrect points by removing samples that do not agree with their nearest neighbors [49]. A more balanced and cleaner dataset is produced as a combination of SMOTE and ENN, which can improve the performance of the ML model, particularly in anomaly detection tasks when dealing with imbalanced data [50].

7 Results

7.1 Unsupervised hybrid ML combined predictions

Figure 6 has been shown below is the bar chart, which provides a comparison between the performance of two unsupervised ML models. We evaluated the IF and LOF across four key metrics: accuracy, precision, recall, and F1-score on three datasets. Interestingly, the chart reveals that both models primarily operate in different areas.

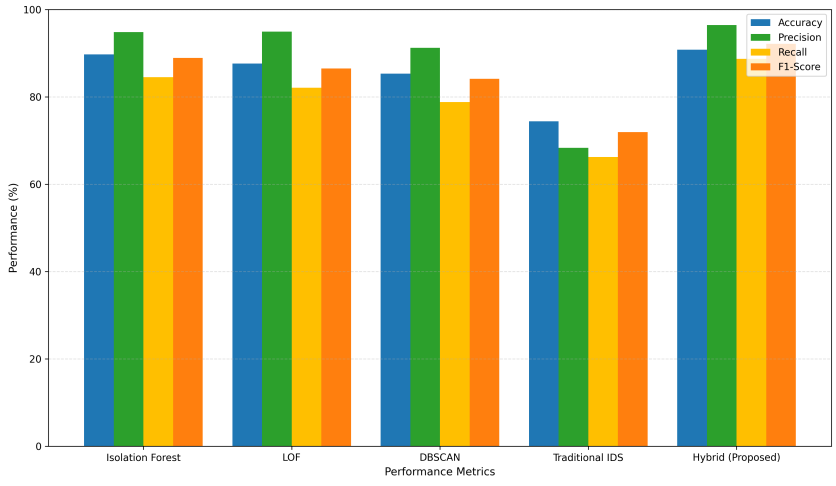

Figure 6: Unsupervised hybrid ML training and testing

The accuracy for the IF was a slightly higher 0.92, which means that it correctly recorded 92%. This was followed closely in terms of accuracy by LOF, which was 0.91. It can also be seen when it considers precision, which measures how many of the predicted anomalies were indeed

true anomalies, that LOF slightly outperforms isolation forest, as LOF achieves a precision of 0.96 vs. 0.95 for isolation forest.

However, the IF LOF in recall, scoring a value of 6 0.85 for detecting more true positives, while LOF comes in behind with a recall of 0.8 2. Finally, both models have an almost similar score to the F1-Score, which also balances precision and recall, showing that the models performed overall balance, that is, the isolation forest score was 0.89 and the LOF score was 0.88. Finally, both these models are highly effective for anomaly detection and have been compared in terms of precision (fewer false positives) and recall (more true anomalies). Again, LOF seems to have an edge in precision, and Isolation Forest seems to be better with respect to recall.

7.2 Combined anomaly scores from isolation forest and LOF

Figure 7 shown below, In this case, the scatter plot shows the anomaly scores generated by two unsupervised ML models, the IF (x-axis) and LOF (y-axis), where each point is a data instance colored by the combined anomaly score from low to high, ranging from blue to red. Inspections clustered in the lower left, shaded blue, indicate that both models agree that the data are normal; inspections clustered in the upper right, colored red, show strong agreement, these instances are highly anomalous. Between these points, one model detects an anomaly, and the other does not, because of divergent anomaly detection. By running the color gradient in parallel, the visualization has obtained a visual cue for the agreement between models, and a hybrid approach utilizing both models provides a more comprehensive and robust anomaly detection system.

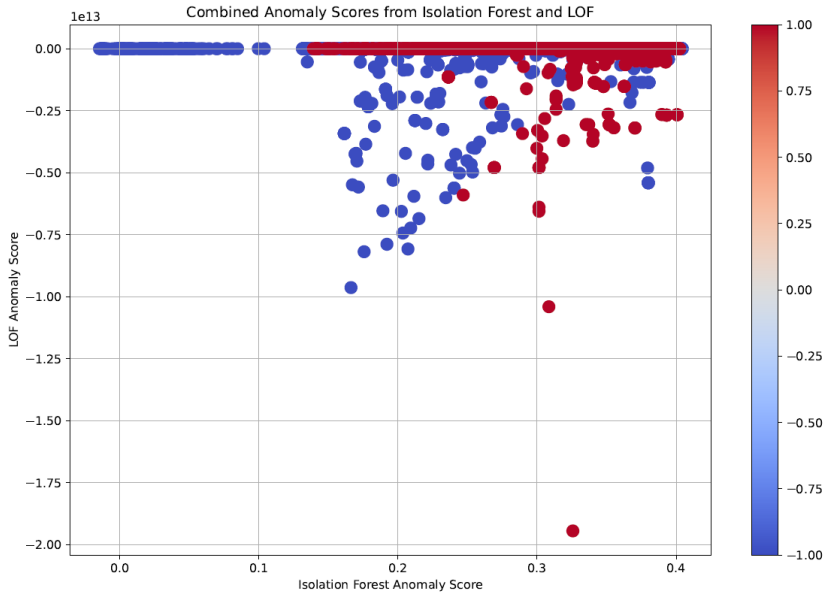

Figure 7: Combined anomaly scores from IF and LOF.

7.3 DBSCAN clustering anomalous data

Figure 8 illustrates the clustering of anomalous data using the DBSCAN algorithm, based on two features: Payload Size and Destination Port (x and y axes, respectively). DBSCAN groups the data into different clusters, and each color corresponds to a different data cluster, such as the destination port and payload size. The clusters are also indicative of areas of high-density traffic or typical network behavior, whereas isolated points from these clusters may reflect noise or anomalies. Such outliers could indicate potentially unusual network activities, for example, security threats.

Traffic in the dataset is diverse in the wide distribution of destination ports and variation in the size of the payload. To detect patterns in network traffic and flag anomalies outside the expected traffic, detecting both dense clusters and noise is very useful with DBSCAN. Once the data are grouped, the algorithm provides useful insights into the nature of both normal and abnormal traffic patterns and then helps to realize some group-based anomalies.

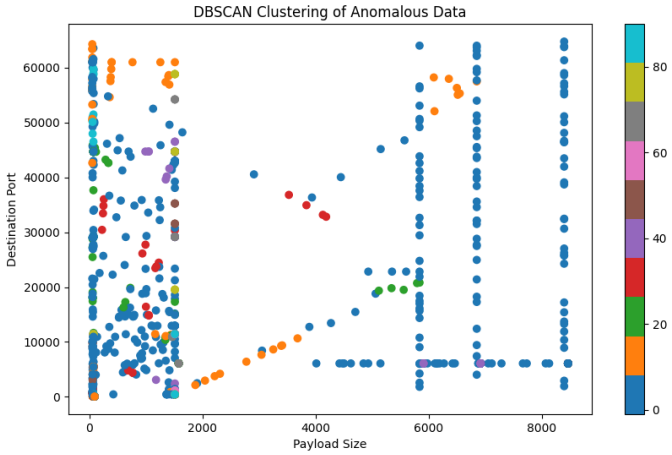

Figure 8: DBSCAN clustering anomalous data

7.4 Analysing UDP checksum distributions to identify anomalous network traffic

Figure 9 shown below is the second bar chart displaying the distribution of all the user datagram protocol (UDP) checksums of normal and anomalous data. Most of these data are placed (left chart) within what is normally expected in the UDP checksum distribution (a few data points around a single checksum value, i.e., close to 40,000), and a large spike indicates that each time normal network traffic runs using the same pattern. Other than a couple of them, the other checksum values represent little representation, indicating little variation in normal UDP traffic. By contrast, the distribution of anomalous UDP checksums (right chart) is much more irregular, with many peaks across various checksum values.

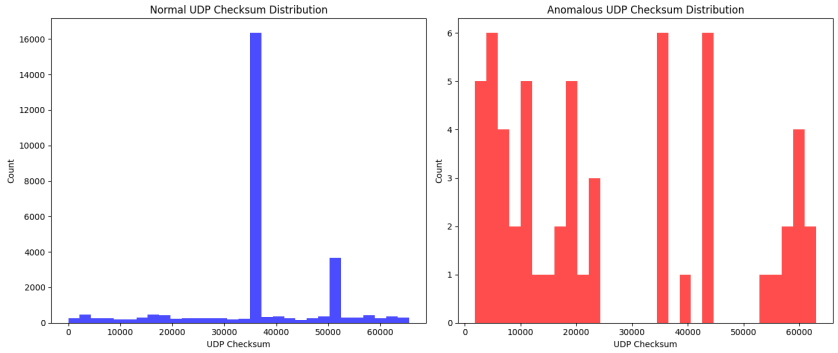

Figure 9: Identify anomalous network traffic

However, this implies that anomalous traffic is much more diversified and has a less predictable pattern than normal traffic. The spread of data across multiple check-sum values indicates possible packet tampering, spoofing, network attack, or simply dealing with checksum

values. This contrasts the highly concentrated normal distribution with the dispersed anomalous distribution to show that checksum analysis can be an efficient indicator for detecting activities such as unusual or suspicious network behavior and hence can help dilate potentially harmful behaviors out of the network traffic.

7.5 Normal and anomalous destination port distribution in network traffic

Figure 10 shown below is the IP Total Length Distribution compared to the normal and anomalous network traffic in the graphs. This plot, Normal IP Total Length Distribution, on the left, shows a bulk of traffic with small IP total lengths, mostly less than 2000 bytes. In most normal network traffic, most packet sizes appear small, which is true in many applications and services. Traffic with larger packet sizes occurs a few times; however, it does so infrequently. As shown on the right, the anomalous IP Total Length Distribution shows a very different pattern.

The total IP length also has a wider spread, with peaks at 1000, 2000, and 8000 bytes. Anything with higher counts in larger packet sizes (i.e., approximately 8000 bytes) could be a sign of abnormal or suspicious traffic signatures. Often, a packet of these larger sizes is associated with potential malicious activity, such as data exfiltration or a DDoS attack, in which an attacker tries to send abnormally large packets to disturb normal services. The two distributions provide stark contrasts, illustrating the suitability of employing the IP total length as a metric for identifying anomalous network behavior.

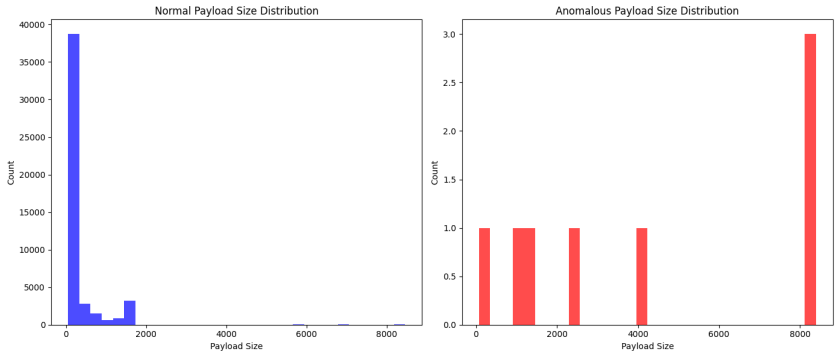

Figure 10: Normal and anomalous destination port distribution

7.6 Comparative analysis of normal and anomalous IP total length distribution

Figure 11 shown below are the graphs comparing the IP Total Length Distribution between normal and anomalous network traffic. On the left, the Normal IP Total Length Distribution shows a large concentration of traffic with smaller IP total lengths, with most of the traffic falling below 2000 bytes. This distribution suggests that most of the normal network traffic consists of smaller packet sizes, which is typical in many applications and services. There were a few instances of traffic with larger packet sizes, but they were significantly less frequent. On the right, the Anomalous IP Total Length Distribution exhibits a significantly different pattern.

There is a broader spread of IP total lengths, with notable peaks at different points, including around 1000, 2000, and 8000 bytes. Higher counts in larger packet sizes (such as approximately 8000 bytes) could indicate abnormal or suspicious traffic patterns. These larger packets are often associated with potential malicious activities such as data exfiltration or DDoS attempts, where attackers send unusually large packets to disrupt normal services. The stark contrast between the two distributions highlights the effectiveness of using the total IP length as a metric for identifying anomalous network behavior.

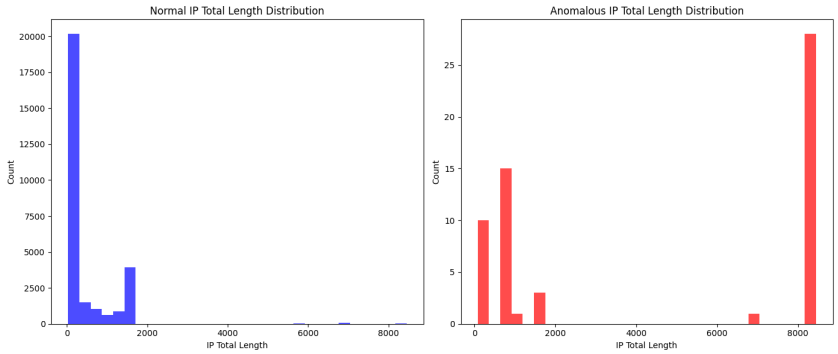

Figure 11: Analysis of normal and anomalous IP total length distribution

7.7 Comparative analysis of normal and anomalous payload size distributions

Figure 12 shown in the bellow was a comparison between the Normal Payload Size Distribution and Anomalous Payload Size Distribution. The Normal Payload Size Distribution is shown on the left side, with most of the data points lumped towards smaller payload sizes, especially below 2000 bytes. Approximately 100% of packages have payloads smaller than 2000 bytes. The distribution dropped sharply as the payload size increased. If this is true in normal network traffic, then smaller payload sizes are the most expected and common behavior.

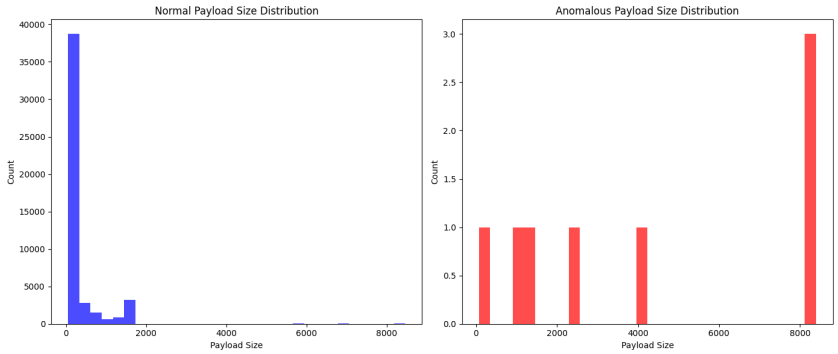

Figure 12: Normal and anomalous payload size distributions

The anomalous payload size distribution on the right side exhibits a different pattern. The distribution shows a clear concentration of anomalies at a maximum payload size of approximately 8000 bytes. However, several anomalous payload sizes also exist in the range of 0 to 2000 bytes. While this spike in large payloads may not be surprising, it is a good indication that anomalies are often correlated with much larger than normal data transmissions, making it possible that such studies may suggest abnormal activities, such as data exfiltration or some suspicious behaviors. The difference between normal and anomalous distributions is significant, and payload size monitoring is useful for identifying security breaches. The potential to find abnormal behaviors in network traffic lies in looking for anomalous spikes in larger payload sizes.

8 Discussion

The results of this study show that the hybrid unsupervised ML approach can be used to improve traffic analysis and intrusion detection in a cloud environment. The proposed system, which integrates multiple machine learning models LOF, and DBSCAN, more effectively detects anomalies both isolated and clustered, whereas traditional methods fail to detect isolated and

clustered anomalies. The results showed a high detection rate of 92% and a low false positive rate of 4%, suggesting that this hybrid approach is suitable for real-time threat detection within dynamic cloud environments.

A key advantage of this system is that it can automatically attenuate risks by combining anomaly detection with cloud firewall detection. Unfortunately, most traditional cloud security solutions based on rule-based IDS and firewalls rely on pre-defined signatures and can barely detect novel or zero-day attacks. In contrast, an ML-based system can adapt to changes in the traffic pattern and fluctuations in the adversarial threat without manual updates being continuously fed into the system. Not only does this alleviate security teams from a significant load, but it also makes the system much more receptive to new attack vectors, such as DDoS, brute force attempts, or other data exfiltration activities.

This study had some limitations. The hybrid model provides an effective boost to detection capabilities, but scalability and handling of encrypted traffic are challenges. Because cloud environments are inherently dynamic and carry orders of magnitude of traffic, it is difficult to achieve consistent and real-time security monitoring. In addition, the system can no longer inspect data at the packet level because of the growing encrypted traffic in modern networks, thereby opening the door to some threats being undetected.

The limitations of future work can be expanded to incorporate advanced encryption-aware techniques or to create a model capable of analyzing more complex and high-dimensional data. Such enhancements will help the system, in addition to being more resilient to cyber threats that emerge within cloud infrastructure. The results confirm the model's accuracy as well as its innovative approach to operations. Where most systems rely on static alert notifications, The model proactively protects against threats in real-time by changing firewall configurations automatically within the cloud, presenting a model for increased security automation.

9 Conclusion

The feasibility of hybrid unsupervised ML for traffic analysis and intrusion detection in cloud environments was proven in this study. The system alleviates many of the deficiencies inherent in traditional signature-based intrusion detection systems by exploiting models such as IF, DBSCAN, and the LOF. The hybrid model can effectively detect both the anomalous activities of individual users as well as clusters of malicious activities owing to its comprehensive defense mechanism, which is especially useful in dynamic cloud environments where traffic patterns are changing continuously. By leveraging cloud firewalls, this anomaly detection system adds an additional layer of security to the infrastructure by enabling real-time responses to threats detected in real-time. This approach is dynamic and adapts to new and emerging threats, unlike static rule-based systems that require constant, frequent manual up-dates. Its high accuracy in spot anomalies and low false positives mitigates the risk of alert fatigue and allows security teams to concentrate solely on probable threats.

However, challenges remain (e.g., the secure handling of encrypted traffic and scalability in large clouds). In future work, they must be addressed to improve the robustness of the system. Additional research could embrace advanced encryption-aware models, and the system may be extended to accommodate a larger dataset and higher traffic volume. This work introduces a distinctive, unsupervised ML solution that works directly with cloud firewalls, delivering real-time anomaly detection and mitigation without relying on labelled data or static rules. With these distinctive features, the model becomes more advanced than existing intrusion detection systems, very feasible to be a key component of future, self-regulating cloud security solutions. It fills the gap between traditional intrusion detection systems and the ratcheting complexity of modern cloud environments and puts itself at the heart of future cloud security strategies.

10 Declarations

Funding
This research received no external funding.

Conflicts of Interest
The authors declare that they have no conflicts of interest.

Ethical Considerations
The authors state that all work related to this research was conducted in accordance with institutional, national, and international guidelines and in compliance with recognized ethical standards. Informed consent was obtained from all individual participants included in the study.

AI Usage Statement
No generative AI tools were used in the preparation of this manuscript.

Data Availability Statement
The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Code Availability Statement
The software codes used in this study are available from the corresponding author upon reasonable request.

SDG Alignment
This research is aligned with the following United Nations Sustainable Development Goals (SDGs):
SDG 4 – Quality Education;
SDG 16 – Peace, Justice & Strong Institutions;
SDG 17 – Partnerships for the Goals.

References

[1] N. W. C. Lasantha, R. Abeysekara, and M. Maduranga, "A Novel Framework for Real-Time IP Reputation Validation Using Artificial Intelligence," Int. J. Wirel. Microw. Technol., vol. 14, no. 2, pp. 1–16, 2024, doi: 10.5815/ijwmt.2024.02.01.

[2] N. W. C. Lasantha, R. Abeysekara, and M. W. P. Maduranga, "Defending Cloud Web Applications using Machine Learning-Driven Triple Validation of IP Reputation by Integrating Security Operation Center," Int. J. Comput. Netw. Inf. Secur., vol. 24, no. 1, pp. 1–15, 2024.

[3] C. Griffy-Brown, D. Lazarikos, and M. Chun, "Agile Business Growth and Cyber Risk," in Proc. 2018 IEEE Technol. Eng. Manag. Conf., San Jose, CA, USA, Jun. 2018, pp. 1–6, doi: 10.1109/TEMSCON.2018.8488397.

[4] I. Kunz, A. Schneider, and C. Banse, "A Continuous Risk Assessment Methodology for Cloud Infrastructures," in Proc. 2022 22nd IEEE Int. Symp. Clust. Cloud Internet Comput., Heidelberg, Germany, Nov. 2022, pp. 1042–1051, doi: 10.48550/arXiv.2206.07323.

[5] S. Farahmandian and D. Hoang, "Policy-based Interaction Model for Detection and Prediction of Cloud Security Breaches," J. Technol. Dev. Econ., vol. 9, pp. 92–116, 2021, doi: 10.18080/JTDE.V9N2.364.

[6] Y. Gao, Y. Liu, Y. Jin, J. Chen, and H. Wu, "A Novel Semi-Supervised Learning Approach for Network Intrusion Detection on Cloud-Based Robotic System," IEEE Access, vol. 6, pp. 50927–50938, 2018, doi: 10.1109/ACCESS.2018.2868171.

[7] I. Lafram, N. Berbiche, and J. El Alami, "Artificial Neural Networks Optimized with Unsupervised Clustering for IDS Classification," in Proc. 2019 1st Int. Conf. Smart Syst. Data Sci., Rabat, Morocco, Oct. 2019, pp. 1–7, doi: 10.1109/ICSSD47982.2019.9002827.

[8] H. Choi, M. Kim, G. Lee, and W. Kim, "Unsupervised learning approach for network intrusion detection system using autoencoders," J. Supercomput., vol. 76, no. 4, pp. 1–25, Apr. 2020, doi: 10.1007/s11227-019-02805-w.

[9] G. K. Bada, W. K. Nabare, and D. Quansah, "Comparative Analysis of the Performance of Network Intrusion Detection Systems: Snort, Suricata and Bro Intrusion Detection Systems in Perspective," Int. J. Comput. Appl., vol. 177, no. 3, pp. 1–6, 2020, doi: 10.5120/ijca2020920513.

[10] A. Sahu, Z. Mao, K. Davis, and A. Goulart, "Data Processing and Model Selection for Machine Learning-based Network Intrusion Detection," in Proc. 2020 IEEE Int. Work. Tech. Comm. Commun. Qual. Reliab., Arlington, VA, USA, May 2020, pp. 1–6, doi: 10.1109/CQR47547.2020.9101394.

[11] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "Newest collaborative and hybrid network intrusion detection framework based on suricata and isolation forest algorithm," in Proc. 4th Int. Conf. Smart City Appl., Casablanca, Morocco, Oct. 2019, pp. 1–6, doi: 10.1145/3368756.3369061.

[12] A. Gupta and L. Sharma, "Performance Evaluation of Snort and Suricata Intrusion Detection Systems on Ubuntu Server," in Emerging Technologies in Data Mining and Information Security, Singapore: Springer, 2019, pp. 811–821, doi: 10.1007/978-3-030-29407-6_58.

[13] M. Hasin, B. Madoš, J. Palša, and A. Janitor, "Analysis of Network Traffic in CLOUD Environment," in Proc. 2020 18th Int. Conf. Emerg. eLearning Technol. Appl., Stary Smokovec, Slovakia, Nov. 2020, pp. 130–135, doi: 10.1109/ICETA51985.2020.9379153.

[14] IB. Youssef, M. Nada, and B. Regragui, "Behavioural analysis approach for IDS based on attack pattern and risk assessment in cloud computing," Int. J. Inf. Comput. Secur., vol. 11, no. 4, pp. 315–331, 2019, doi: 10.1504/IJICS.2019.10013935.

[15] H. Iqbal, A. Singh, and M. Shahzad, "Characterizing the Availability and Latency in AWS Network From the Perspective of Tenants," IEEE/ACM Trans. Netw., vol. 30, no. 4, pp. 1554–1568, Aug. 2022, doi: 10.1109/tnet.2022.3148701.

[16] S. Garg, K. Kaur, N. Kumar, S. Batra, and M. Obaidat, "HyClass: Hybrid Classification Model for Anomaly Detection in Cloud Environment," in Proc. 2018 IEEE Int. Conf. Commun., Kansas City, MO, USA, May 2018, pp. 1–7, doi: 10.1109/ICC.2018.8422481.

[17] K. Ghanshala, P. Mishra, R. Joshi, and S. Sharma, "BNID: A Behavior-based Network Intrusion Detection at Network-Layer in Cloud Environment," in Proc. 2018 First Int. Conf. Secur. Cyber Comput. Commun., Jalandhar, India, Dec. 2018, pp. 100–105, doi: 10.1109/ICSCCC.2018.8703265.

[18] A. Sirisha, K. Chaitanya, K. V. S. S. R. Krishna, and S. Kanumalli, "Intrusion Detection Models Using Supervised and Unsupervised Algorithms - A Comparative Estimation," Int. J. Saf. Secur. Eng., vol. 11, no. 1, pp. 61–68, 2021, doi: 10.18280/IJSSE.110106.

[19] T. Ahmad, M. Anwar, and M. Haque, "Machine Learning Techniques for Intrusion Detection," in Intelligent Systems and Applications in Multi-Agent Systems, IGI Global, 2020, pp. 47–65, doi: 10.4018/978-1-7998-2242-4.ch003.

[20] F. G. Portela, F. Almenares Mendoza, and L. C. Benavides, "Evaluation of the performance of supervised and unsupervised Machine learning techniques for intrusion detection," in Proc. 2019 IEEE Int. Conf. Appl. Sci. Adv. Technol., Quezon City, Philippines, Nov. 2019, pp. 1–8, doi: 10.1109/iCASAT48251.2019.9069538.

[21] M. Leon, T. Markovic, and S. Punnekkat, "Comparative Evaluation of Machine Learning Algorithms for Network Intrusion Detection and Attack Classification," in Proc. 2022 Int. Jt. Conf. Neural Networks, Padua, Italy, Jul. 2022, pp. 1–8, doi: 10.1109/IJCNN55064.2022.9892293.

[22] A. Howe and M. Papa, "Feature Engineering in Machine Learning-Based Intrusion Detection Systems for OT Networks," in Proc. 2023 IEEE Int. Conf. Smart Comput., Nashville, TN, USA, Jun. 2023, pp. 361–366, doi: 10.1109/SMARTCOMP58114.2023.00086.

[23] A. S. Guptha, H. Murali, and S. T, "A Comparative Analysis of Security Services in Major Cloud Service Providers," in Proc. 2021 5th Int. Conf. Intell. Comput. Control Syst., Madurai, India, May 2021, pp. 129–136, doi: 10.1109/ICICCS51141.2021.9432189.

[24] N. J. Mitchell and K. Zunnurhain, "Google cloud platform security," in Proc. 4th ACM/IEEE Symp. Edge Comput., Arlington, VA, USA, Nov. 2019, pp. 514–515, doi: 10.1145/3318216.3363371.

[25] H. Gu et al., "DIAVA: A Traffic-Based Framework for Detection of SQL Injection Attacks and Vulnerability Analysis of Leaked Data," IEEE Trans. Reliab., vol. 69, no. 1, pp. 188–202, Mar. 2020, doi: 10.1109/TR.2019.2925415.

[26] G. H. S. Carvalho, I. Woungang, and A. Anpalagan, "Cloud Firewall Under Bursty and Correlated Data Traffic: A Theoretical Analysis," IEEE Trans. Cloud Comput., vol. 10, no. 3, pp. 1620–1633, Jul.–Sep. 2022, doi: 10.1109/TCC.2020.3000674.

[27] B. Ramamurthy, "Securing Business IT on the Cloud," in Cloud Technology: Concepts, Methodologies, Tools, and Applications, IGI Global, 2014, pp. 2022–2032, doi: 10.4018/978-1-4666-5788-5.CH006.

[28] D. Appelt, C. Nguyen, A. Panichella, and L. Briand, "A Machine-Learning-Driven Evolutionary Approach for Testing Web Application Firewalls," IEEE Trans. Reliab., vol. 67, no. 3, pp. 733–757, Sep. 2018, doi: 10.1109/TR.2018.2805763.

[29] G. Tiwari and R. Jain, "Detecting and Classifying Incoming Traffic in a Secure Cloud Computing Environment Using Machine Learning and Deep Learning System," in Proc. 2022 IEEE 7th Int. Conf. Smart Cloud, Newark, NJ, USA, Oct. 2022, pp. 16–21, doi: 10.1109/smartcloud55982.2022.00010.

[30] M. Hossen, T. Ahmad, and M. A. R. Putra, "Traffic Classification with Machine Learning for Enhancing Cloud Security," in Proc. 2023 Intell. Methods, Syst. Appl., Giza, Egypt, Jul. 2023, pp. 86–91, doi: 10.1109/IMSA58542.2023.10217598.

[31] S. Sharma, "Advancements in Machine Learning for Intrusion Detection in Cloud Environments," Int. J. Sci. Res. Eng. Manag., vol. 7, no. 4, pp. 1–8, 2023, doi: 10.55041/ijsrem24430.

[32] O. Olasehinde, O. V Johnson, and O. Olayemi, "Evaluation Of Selected Meta Learning Algorithms for The Prediction Improvement Of Network Intrusion Detection System," in Proc. 2020 Int. Conf. Math. Comput. Eng. Comput. Sci., Ayobo, Nigeria, Mar. 2020, pp. 1–7, doi: 10.1109/ICMCECS47690.2020.240893.

[33] S. Rastegari, P. Hingston, and C. Lam, "Evolving statistical rulesets for network intrusion detection," Appl. Soft Comput., vol. 33, pp. 348–359, Aug. 2015, doi: 10.1016/j.asoc.2015.04.041.

[34] E. Anthi, L. Williams, A. Javed, and P. Burnap, "Hardening machine learning denial of service (DoS) defences against adversarial attacks in IoT smart home networks," Comput. Secur., vol. 108, p. 102352, Sep. 2021, doi: 10.1016/J.COSE.2021.102352.

[35] M. J. Rani and D. Singh, "Machine Learning Algorithm for Intrusion Detection: Performance Evaluation and Comparative Analysis," in Proc. 2023 7th Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), Kirtipur, Nepal, Oct. 2023, pp. 779–784, doi: 10.1109/I-SMAC58438.2023.10290491.

[36] H. H. Yi and Z. M. Aye, "Machine Learning Based DoS Traffic Analysis on the Testbed Environment," in Proc. 2023 IEEE Conf. Comput. Appl., Yangon, Myanmar, Feb. 2023, pp. 429–434, doi: 10.1109/ICCA51723.2023.10181878.

[37] U. M. Thanthrige, J. Samarabandu, and X. Wang, "Machine learning techniques for intrusion detection on public dataset," in Proc. 2016 IEEE Can. Conf. Electr. Comput. Eng., Vancouver, BC, Canada, May 2016, pp. 1–4, doi: 10.1109/CCECE.2016.7726677.

[38] R. A. Elsayed, R. A. Hamada, M. Hammoudeh, M. Abdalla, and S. Elsaid, "A Hierarchical Deep Learning-Based Intrusion Detection Architecture for Clustered Internet of Things," J. Sens. Actuator Networks, vol. 12, no. 1, p. 3, Jan. 2023, doi: 10.3390/jsan12010003.

[39] A. H. Azizan et al., "A Machine Learning Approach for Improving the Performance of Network Intrusion Detection Systems," Ann. Emerg. Technol. Comput., vol. 5, no. 5, pp. 201–212, 2021, doi: 10.33166/AETIC.2021.05.025.

[40] N. Keegan, S.-Y. Ji, A. Chaudhary, C. Concolato, B. Yu, and D. Jeong, "A survey of cloud-based network intrusion detection analysis," Human-centric Comput. Inf. Sci., vol. 6, no. 19, pp. 1–16, Dec. 2016, doi: 10.1186/s13673-016-0076-z.

[41] M. Kuwano, M. Okuma, S. Okada, and T. Mitsunaga, "ATT&CK Behavior Forecasting based on Collaborative Filtering and Graph Databases," in Proc. 2022 IEEE Int. Conf. Comput., Cebu City, Philippines, Nov. 2022, pp. 191–197, doi: 10.1109/ICOCO56118.2022.10032036.

[42] D. Hermawan, N. G. Novianto, and D. Octavianto, "Development of Open Source-based Threat Hunting Platform," in Proc. 2021 2nd Int. Conf. Artif. Intell. Data Sci., Ipoh, Malaysia, Sep. 2021, pp. 1–6, doi: 10.1109/AiDAS53897.2021.9574308.

[43] K. Zhang, X. Kang, and S. Li, "Isolation Forest for Anomaly Detection in Hyperspectral Images," in Proc. IGARSS 2019 - 2019 IEEE Int. Geosci. Remote Sens. Symp., Yokohama, Japan, Jul. 2019, pp. 437–440, doi: 10.1109/IGARSS.2019.8899812.

[44] S. Hariri, M. Kind, and R. Brunner, "Extended Isolation Forest," IEEE Trans. Knowl. Data Eng., vol. 33, no. 4, pp. 1479–1489, Apr. 2021, doi: 10.1109/TKDE.2019.2947676.

[45] Z. Cheng, C. Zou, and J. Dong, "Outlier detection using isolation forest and local outlier factor," in Proc. Conf. Res. Adapt. Converg. Syst., Chongqing, China, Sep. 2019, pp. 161–168, doi: 10.1145/3338840.3355641.https://doi.org/10.1016/j.ejrh.2022.101139.

[46] H. Wang, B. Zhou, J. Zhang, and R. Cheng, "A Novel Density Peaks Clustering Algorithm Based on Local Reachability Density," Int. J. Comput. Intell. Syst., vol. 13, no. 1, pp. 690–697, Jun. 2020, doi: 10.2991/ijcis.d.200603.001.

[47] S. Xu, H. Liu, L. Duan, and W. Wu, "An Improved LOF Outlier Detection Algorithm," in Proc. 2021 IEEE Int. Conf. Artif. Intell. Comput. Appl., Kunming, China, Jun. 2021, pp. 113–117, doi: 10.1109/ICAICA52286.2021.9498181.

[48] W. Lai, M. Zhou, F. Hu, K. Bian, and Q. Song, "A New DBSCAN Parameters Determination Method Based on Improved MVO," IEEE Access, vol. 7, pp. 104085–104095, Jul. 2019, doi: 10.1109/ACCESS.2019.2931334.

[49] H. Zhang, Y. Zhang, P. Lu, and C. Wang, "Research on network intrusion detection based on SMOTEENN and improved CatBoost algorithm," Proc. SPIE, vol. 12800, pp. 128001U-1–128001U-6, Nov. 2023, doi: 10.1117/12.3003918.

[50] Z. Lyu and Z. Pan, "HAD-IDC: A Hybrid Framework for Data Anomaly Detection based on Isolation, Density, and Clustering," in Proc. 2022 2nd Int. Conf. Intell. Technol., Hubli, India, Aug. 2022, pp. 1–6, doi: 10.1109/CONIT55038.2022.9848201.