Communications & Networks Connect
ISSN: 3105-1421
2025, Article ID. x,
Cite as: https://www.doi.org/10.69709/xxx

SCIFINITI PUBLISHING

OPEN ACCESS

Research Article

A Unified Snort–OSSEC Hybrid Intrusion Detection Framework for Cloud Security

Idris Olanrewaju Ibraheem[✉,1] **Abdulrauf Uthman Tosho**[2]

[1]Department of Computer Science
Faculty of Computing, Engineering and Technology
Al-Hikmah University, Adewole Estate, Ilorin, Nigeria
ioibraheem@alhikmah.edu.ng
ORCiD Number: 0009-0002-3677-2478

[2]Department of Computer Science
Faculty of Computing, Engineering and Technology
Al-Hikmah University, Adewole Estate, Ilorin, Nigeria
autosho@alhikmah.edu.ng
ORCID Number: 0000-0001-7174-2878

Abstract

The growing adoption of Cloud computing has brought about the enhancement of cost-efficiency and effectiveness but also comes with increase in security vulnerabilities due to its distributed and multi-tenant architecture. This contribution if this study to the cloud security through the development of a hybrid intrusion detection system (IDS) that integrates Snort (N-IDS) and OSSEC (H-IDS) to achieve cross-layer alert correlation. Unlike standalone deployments or other deep learning models that are computationally intensive, the framework proposed is an open source that is interpretable, and also deployable in real-world environments. The experimental evaluation within a cloud testbed shows that the hybrid IDS was able to achieve 97.1% accuracy, a 94.1% detection rate, and a false alarm rate of only 0.2%, which performed better than Snort-only and OSSEC-only systems. The case-based simulations conducted further revealed that hybrid correlation does not just enhance the detection, but it also reduced false positives with better forensic trails, most especially in port scanning, brute-force login, and user-

to-root exploits. When compared with previous IDS studies, the system achieved competitive accuracy while maintaining efficiency and transparency. Although tested in a limited-scale environment, this study establishes a practical foundation for adaptive, scalable, and prevention-oriented frameworks in cloud security.

Keywords

cloud security; hybrid IDS; Snort; OSSEC; network security; Intrusion Detection System (IDS)

1. Introduction

Cloud computing has changed modern information technology as it has enabled scalable, flexible, and cost-effective access to computation resources and storage. From personal applications and other enterprise services, the adoption of cloud computing has grown across industries. Although, this shift has brough about increased security concerns, due to cloud infrastructures being a frequent target of several cyberattacks that included denial-of-service, privilege escalation, and data exfiltration [1]. The protection of cloud environments requires intrusion detection systems (IDS) that can detect and act fast to this malicious activity before the dire effect on the service integrity or confidentiality.

Some of the research conducted in the past decade explored network-based IDS (N-IDS) and host-based IDS (H-IDS) as primary defense mechanisms. The N-IDS such as Snort excel at monitoring network traffic for patterns that are suspicious, they miss host-level anomalies that bypass perimeter defenses most of the time. However, OSSEC, which is an H-IDS solution, provide system-level monitoring although it lacks visibility in broader traffic flows. As it relies just on either results that are limited to host activities coverage and the likelihood of false positives or undetected attacks occurrence [2]. Additionally, most of the previous IDS implementations attached more importance to the detection of known signatures but usually perform less when unknown or zero-day threats occur [3].

To tackle the research gap, there is need for the development of an integrated IDS frameworks that combines host and network perspectives to achieve an effective detection capability and reduce false alarms. Some recent studies have attempted hybrid IDS designs, although most of them focus on simulations, while others were not validated adequately in real-world cloud testbeds [4]. To be precise, some studies demonstrated how the integration of Snort and OSSEC, two widely adopted open-source IDS tools, can be used within a hybrid framework that empirically validates improved detection performance.

This study addressed the gaps with the design, implementation, and evaluation a hybrid Snort–OSSEC IDS framework practically in a cloud computing environment. The system uses both signature-based and anomaly-based techniques in the detection of a wide range of attacks, using cross-layer alert correlation to enhance the accuracy and reduction of misclassification. Unlike existing models, the framework went through empirical validation by carrying simulated attack scenarios that include port scanning, DoS, brute-force, and U2R executed against an Ubuntu-based OwnCloud testbed.

1.1 Research Objectives

The objectives of this study are as follows:

1. To design and implement a hybrid IDS framework that integrates Snort and OSSEC for cloud environments.
2. To simulate attack scenarios and evaluate the detection performance.
3. To compare the performance of Snort-only, OSSEC-only, and the hybrid IDS for accuracy, detection rate, and false alarm rate.
4. To benchmark the proposed framework against previous IDS studies to show its novelty and practical contribution.

1.2 Scope of the Study

This study was conducted in a virtualized Ubuntu-based cloud testbed that runs OwnCloud, Apache, and MariaDB. The implementation of the hybrid intrusion detection system was done using two widely adopted open-source tools, Snort was used for network-level monitoring and OSSEC was used for host-level monitoring. Four attack categories port scanning, denial-of-service (DoS), brute-force, and user-to-root (U2R) are the basis of the evaluation which represent common threats in cloud environments. The focus of this study was on detection performance and false alarm reduction using cross-layer correlation. The need for scalability to large multi-tenant deployments and integration with prevention mechanisms were outside the present scope and are recommended for future research.

2. Literature Review

The literature on intrusion detection in cloud computing has seen immense growth, which reflects the need to balance scalability with security. Intrusion detection systems (IDS) have been widely studied, although there are still challenges in addressing unknown threats, reduction of false alarms, and also to ensure real-time performance in environments that are of distributed nature. This section reviews some of the existing approaches carried out on cloud service models, common attack vectors, and hybrid IDS

designs, and identifying the limitations of the previous approaches in relation to the proposed framework.

2.1 Cloud Services and Security Challenges

Cloud services are usually classified into Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The users are exposed to VM-to-VM attacks and hypervisor vulnerabilities through IaaS, Insecure API and multi-tenancy issues are mostly from PaaS, while confidentiality and compliance issues are mostly the concerns from SaaS based deployments. Previous studies outlined these risks but often times the emphasis is always on governance and policy over the integration of IDS mechanisms [5]. More recent studies predict sustained growth in cloud service adoption but gave little or no attention to the implementation of IDS within these contexts [6]. It became necessary to address these gaps through the development and deployment of intrusion detection solutions that are adaptable across service layers.

2.2 Attack Vectors in Cloud Environments

Cloud infrastructures face different types of attack vectors. Side-channel exploits, which is a Virtual Machine based attack, indicates the ease of privilege escalation across tenants. User-to-root (U2R) exploits have been studied with classifiers such as Naïve Bayes and ANN, even at that detecting these exploits remains difficult given their low frequency [7]. The domination of DoS and DDoS threats is still on the rise with distributed detection frameworks often burdened by computational overhead [1]. Datasets with Backdoors and adversarial attack vector represent growing risks for IDS robustness, which shows the need for anomaly-based detection that can withstand the evolving threats [8][3].

There have been Hybrid IDS approaches that attempt to integrate N-IDS and H-IDS techniques. The early designs showed great promise but cannot be validated at the host-level and sometimes they were just on simulated datasets [9]. In a study by [10] they proposed a two-phase hybrid IDS that improved detection rates, but the work suffers from high false positives. A recent study on AI-driven models achieved very high accuracy on benchmark datasets, also there have been some hybrid CNN–RNN designs that exceeded 98% accuracy [11]. However, these systems often compromise interpretability and require high computation, and it limits deployment in resource-constrained cloud environments [12].

Recent and advanced studies emphasize the use of adaptive and federated approaches. [1] in their study introduced a hybrid IDS with the combination of XGBoost and deep learning components on various datasets. This model improved accuracy but also raised concerns about generalizability. The study by [4] complimented transformer-based models with graph features, which achieved almost perfect

accuracy, but the implementation wasn't validated in a real-world environment. In a study by [7] also introduced a deep learning IDS for IoT networks, but its ability to perform optimally and effectively in cloud environments remains uncertain. These studies show how adaptive, explainable, and scalable IDS are important and critical in the detection of unknown attacks and reducing false alarm but also indicates there is a persistent gap in practical, open-source deployment.

From this review, three main gaps were noted:

1. Most IDS frameworks laid more emphasis on either known signature detection or anomaly-based detection, the combination of both techniques are rarely explored which is effective in a real cloud testbed.

2. Previous hybrid systems lack empirical validation of cross-layer integration that consists of network and host to reduce false positives.

3. Recent AI-based IDS studies tend to achieve high accuracy but usually compromise deployability and interpretability in environments that are cost-sensitive.

This study addresses these gaps with the implementation and validation of a Snort–OSSEC hybrid IDS with correlated alerting, tested against multiple attack classes, and the performance evaluated based on metrics such as accuracy, detection rate, and false alarm rate.

3. Materials and Methods

3.1 Study Aim and Design

The primary aim of this study was to design and evaluate a hybrid intrusion detection system (IDS) that combines Snort a network-based IDS and OSSEC, a host-based IDS for the enhancement of cloud security. A quasi-experimental testbed design was used, which involves controlled attack simulations against a cloud-based environment. The performance of three configurations Snort-only, OSSEC-only, and Snort–OSSEC hybrid was also compared in the determination of the relative advantages of hybridization technique employed.

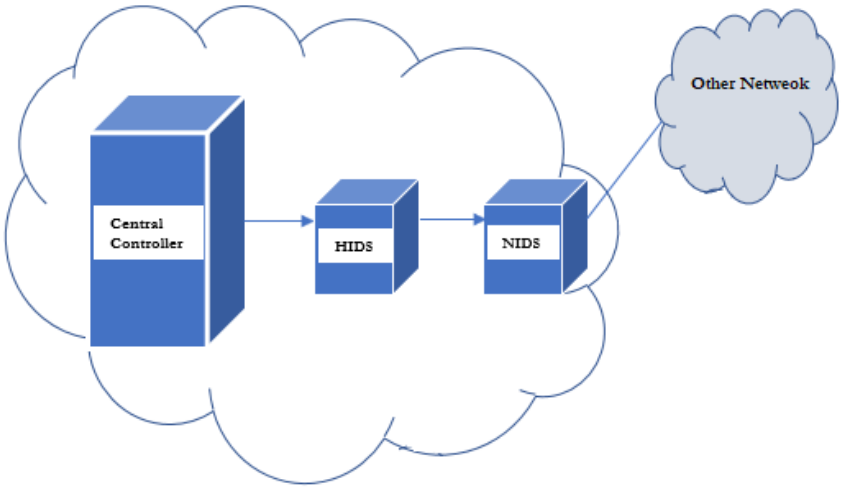

Figure 1: Architecture of the Proposed Snort–OSSEC hybrid IDS framework

This diagram as shown in figure 1 shows how Snort N-IDS and OSSEC H-IDS operate independently and then correlate alerts in a combined analysis engine in the depicted as the central controller. It shows the flow of data from network traffic and system logs to detection and correlation.

3.2 Study Setting

The study was conducted in an isolated laboratory environment that was configured to replicate a realistic cloud service. An Ubuntu 20.04 LTS server was deployed as the primary cloud platform, running OwnCloud to simulate cloud storage and collaboration services. MariaDB for database management and Apache HTTP Server for web hosting were all used as supporting services. This setup provided a clear multi-tier cloud architecture for testing intrusion detection mechanisms under real-world conditions.

3.3 Materials and Tools

The following are the materials used in this study that include both software and hardware:

1. **IDS Tools**: Snort version 2.9.17 for packet inspection, OSSEC version 3.7.0 for host log and integrity monitoring.
2. **Traffic Capture and Analysis**: Wireshark version 3.4.7.
3. **Attack Tools**: Nmap for port scanning, Hping3 for DoS traffic generation, Hydra for brute-force login attempts, and Metasploit for U2R privilege escalation.

4. **Dataset Source**: The primary dataset used was generated from simulated attacks on the isolated testbed, which consist 14,072 labeled connections.

5. **Hardware Environment**: Physical and virtualized PC running Ubuntu as the server and Attacker PC on VMware ESXi with 8-core CPU, 32 GB RAM, and 100 GB allocated storage.

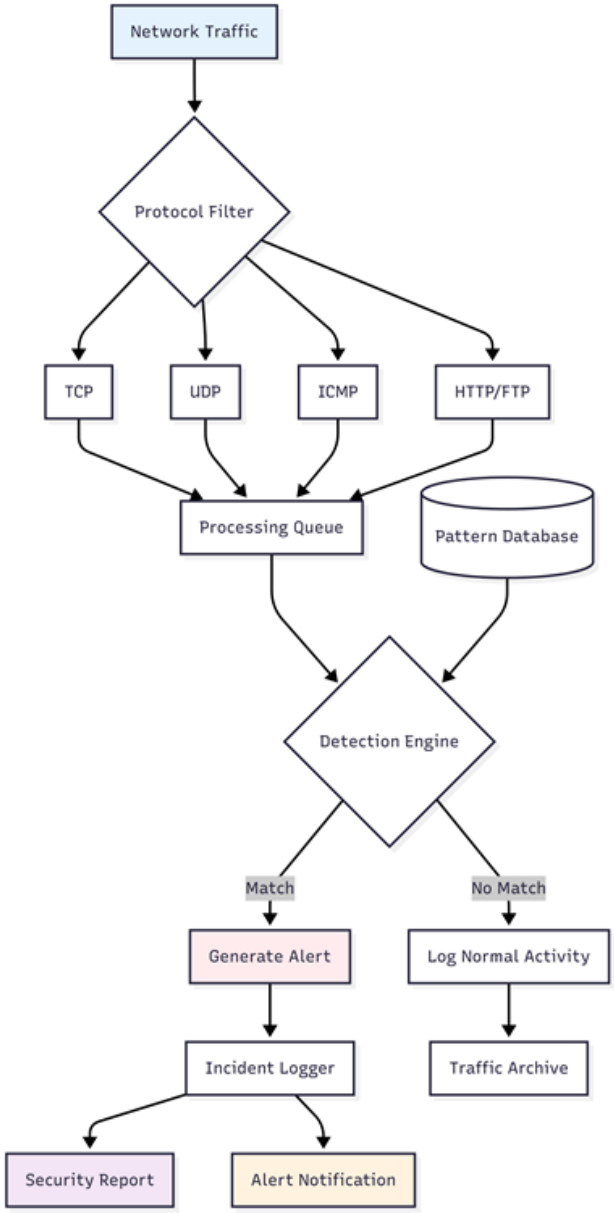

Figure 2: Comprehensive Flowchart of the Proposed Hybrid IDS Model

Figure 2 is the workflow showing how network traffic is first filtered by protocol type (TCP, UDP, ICMP, HTTP/FTP), then passed into a processing queue, for analysis against a pattern database by the detection engine. All matches generate alerts, that are logged and trigger notifications, while normal traffic is archived. This layered process of detection and reporting shows the integration of Snort (N-IDS) and OSSEC (H-IDS) for cross-validation and improved accuracy.

The attack simulation structure is well detailed in figure 3, where the complete experimental process is shown.

3.3.1 Attack Simulation Process

To validate detection strength across multiple attack categories, controlled simulations were conducted. Open-source security tools were used to generate attacks, ensuring reproducibility Nmap was used for port scanning by probing open ports and identify vulnerable services, Hping3 generated TCP/ICMP floods targeting the OwnCloud server which was the point for Denial-of-Service (DoS) attacks, Hydra was used to attempt repeated password logins against FTP services for Brute-Force Attacks, while Metasploit modules simulated privilege escalation on the Ubuntu server for User-to-Root exploits

Each attack was executed against the testbed with corresponding Snort rules and OSSEC alerts monitored in real time. By including both frequent (port scan, DoS) and rare (U2R) categories, the evaluation covered a broad threat scenarios.

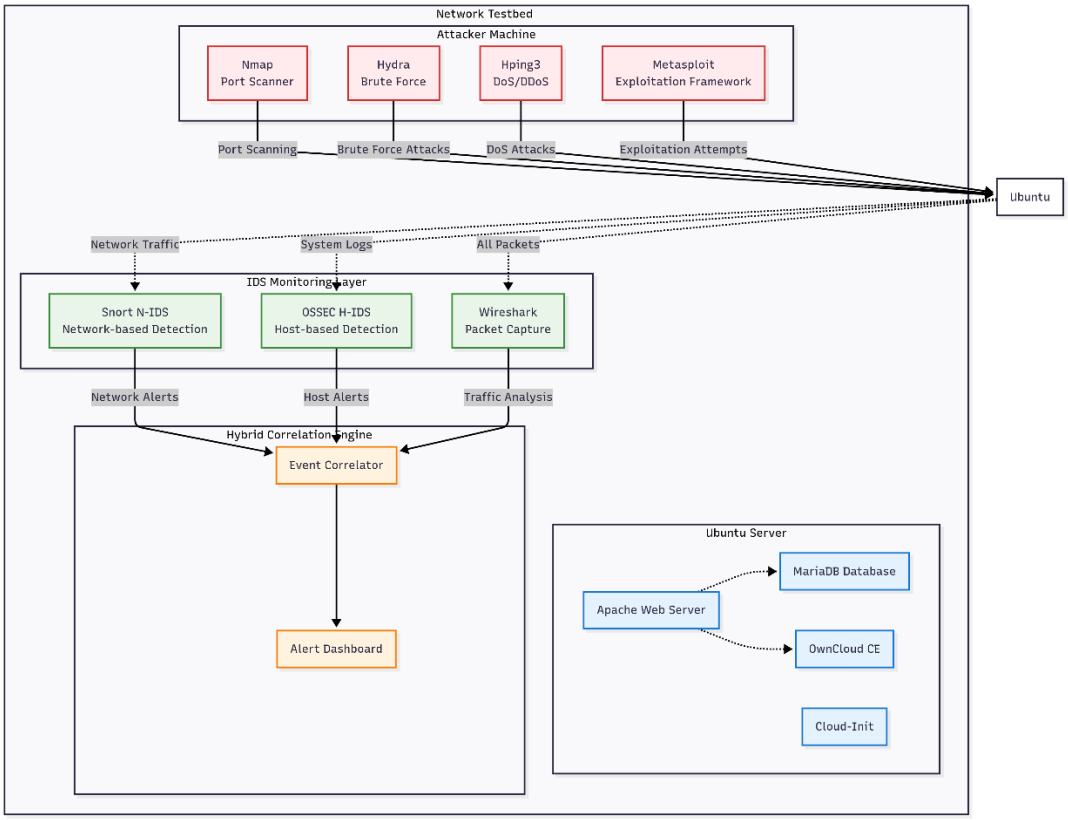

Figure 3: Experimental Testbed Setup for Hybrid IDS Evaluation

The testbed was deployed on an Ubuntu environment which hosts the OwnCloud server with Apache and MariaDB as shown in figure 3. Snort was configured as a network intrusion detection system (N-IDS) for the monitoring of packet-level traffic, while OSSEC was also configured to serve as a host intrusion detection system (H-IDS) for logging and monitoring of system integrity. An attacker machine was configured to generate malicious traffic using Nmap, Hping3, Hydra, and Metasploit, while Wireshark was used for packet capturing and analysis. This setup enabled controlled evaluation of the hybrid IDS under real-world cloud attack scenarios.

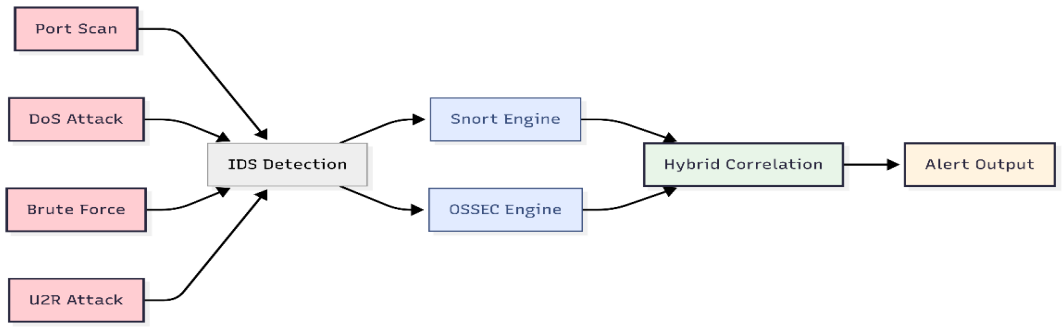

Figure 4: Attack Simulation Workflow of the Model

This diagram as shown in figure 4 indicates the process by which attack simulation such as port scanning, denial-of-service (DoS), brute-force login, and user-to-root (U2R) are launched from the attacker machine against the cloud testbed. The network traffic is monitored by Snort, while logs at the system level are analyzed by OSSEC. The alerts triggered from both sources are forwarded to the hybrid correlation module, where cross-validation technique in place improve the detection accuracy and reduced false alarms.

In this study, “unknown threats” are the attacks that are not represented in the initial signature sets of Snort or predefined OSSEC rules. Detection was enabled by anomaly-based mechanisms, with Snort anomaly detection leveraging on the preprocessors for unusual traffic behavior, such as abnormal packet sizes and protocol misuse, while OSSEC anomaly detection was configured through integrity checks and system log correlation, capturing deviations from baseline user activities.

This dual-layer anomaly detection ensured that threats absent from signature databases could still be flagged as suspicious, supporting claims of “unknown” detection capability.

3.4 Data Collection Procedures

The benign traffic was initially generated from the activities of cloud services under normal user activity. With Wireshark capturing packets at the network layer, while OSSEC monitors the system events and file integrity at the host layer. Upon completion of baseline profiling, controlled attacks were carried out against the testbed using the tools listed above. every attack scenario was repeated several times for consistency. Captured traffic and logs were stored in packet capture (pcap) and syslog formats for preprocessing.

3.5 Data Preprocessing and Feature Extraction

Initially 31 attributes were initially recorded from the Snort logs, after normalization and feature selection, only 26 were retained. These attributes include protocol type, source and destination ports, connection duration, TCP flags, packet size, and byte counts. Also, the OSSEC H-IDS contributed features include failed login attempts, file integrity modifications, and unauthorized root access events. All the features were then presented in a tabular form to create a structured dataset that is suitable for classification as shown in table 1.

3.5.1 Features before normalization (31 features)

Duration, Protocol_type, Service, Flag, Src_bytes, Dst_bytes, Land, Wrong_fragment, Urgent, Hot, Num_failed_logins, Logged_in, Num_compromised, Root_shell, Su_attempted, Num_root, Num_file_creations, Num_shells, Num_access_files, Num_outbound_cmds, Is_host_login, Is_guest_login, Count, Srv_count, Serror_rate, Srv_serror_rate, Rerror_rate, Srv_rerror_rate, Same_srv_rate, Diff_srv_rate, Srv_diff_host_rate

3.5.2 Features after normalization (26 features)

Duration, Protocol_type, Service, Flag, Src_bytes, Dst_bytes, Wrong_fragment, Urgent, Num_failed_logins, Logged_in, Num_compromised, Root_shell, Su_attempted, Num_root, Num_file_creations, Num_shells, Num_access_files, Is_guest_login, Count, Srv_count, Serror_rate, Srv_serror_rate, Rerror_rate, Srv_rerror_rate, Same_srv_rate, Diff_srv_rate

Table 1: Extracted Features from Hybrid IDS Logs

Category	Features
Protocol & Connection	Protocol type (TCP/UDP/ICMP/FTP), source IP, destination IP, source port, destination port, packet length, connection duration
Network Behavior	TCP flags (SYN, ACK, FIN, RST), number of bytes sent/received, packets per connection, abnormal fragmentation
Service Indicators	FTP login attempts, number of concurrent sessions, failed authentication events
Host Activities	File integrity changes, unauthorized root access attempts, log modification frequency
Traffic Context	Time interval between packets, abnormal ICMP echo requests, unexpected UDP connections

3.6 Interventions and Comparisons

Three system configurations were tested:

1. **Snort-only (N-IDS)** – network-level detection only.

2. **OSSEC-only (H-IDS)** – host-level detection only.

3. **Hybrid Snort–OSSEC IDS** – correlated alerts across host and network layers.

Each configuration went through an evaluation phase based on identical attack scenarios, that enabled comparative analysis of the accuracy, detection rate, and false alarm rate.

3.7 Classifier Justification: Naïve Bayes

The justification for the selection of Naïve Bayes classifier for evaluation of the detection performance is due to its suitability for categorical and discrete data that are derived from Snort and OSSEC logs. Its advantages include:

1. **Computational efficiency**: suitable for real-time IDS applications with large datasets.

2. **Noise tolerance**: Its effectiveness in handling imperfect or incomplete network traffic data.

3. **Transparency**: its ability to provide interpretable probability outputs, enhancing explainability in cloud security.

While deep learning methods were able to achieve higher accuracy in some IDS studies, usually they require huge computational resources and the lack of interpretability. The lightweight and explainable nature of Naïve Bayes makes it appropriate for practical, cost-sensitive cloud deployments.

3.8 Evaluation Metrics

The Naïve Bayes classifier was used to assess and evaluate the detection performance based onits efficiency, noise tolerance, and interpretability [12]. Performance metrics were calculated from confusion matrices using the following formulas (equation 1-3):

i. Accuracy $$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

ii. Detection Rate $$DR = \frac{TP}{TP + FN} \quad (2)$$

iii. False Alarm Rate $$FAR = \frac{FP}{TN + FP} \quad (3)$$

4. Results

4.1 Experimental Setup in Relation to Methods

As outlined in Section 3.4, the evaluation was conducted in an Ubuntu–OwnCloud testbed where four attack scenarios port scanning, DoS, brute-force, and U2R were simulated using Nmap, Hping3, Hydra, and Metasploit. The three IDS configurations described in Section 3.6 Interventions and Comparisons were tested separately: Snort-only, OSSEC-only, and the hybrid Snort–OSSEC framework. The primary dataset used for evaluation contained 14,072 labeled connections generated during these experiments.

4.2 Classifier Evaluation: Confusion Matrix

Using the dataset described in Section 3.5 and the Naïve Bayes classifier introduced in Section 3.7, the classifier's performance was first evaluated in terms of its confusion matrix. Table 2 presents the distribution of classification outcomes across normal and anomalous traffic. From a total number of 7,446 normal connections, 7,430 were classified correctly, only 16 out of the connections were misclassified as anomalies. For the anomalous connections, out of 6,626 connections, 6,239 were correctly detected, while 387 were misclassified as normal.

Table 2: Confusion Matrix of the Naïve Bayes Classifier

Actual Class	Predicted Normal	Predicted Anomaly
Normal	True Negative (TN) = 7430	False Positive (FP) = 16
Anomaly	False Negative (FN) = 387	True Positive (TP) = 6239

4.3 Classifier Evaluation: Performance Metrics

From the confusion matrix, the standard performance measures (accuracy, detection rate, false alarm rate, precision, F-measure, MCC, ROC area, PRC area) were calculated using the formulas defined in Section 3.7. The results are shown in Table 3.

Table 3: Performance of the Naïve Bayes Classifier on Class of Connection

Class	TP Rate	FP Rate	Precision	F-Measure	MCC	ROC Area	PRC Area
Anomaly	0.998	0.058	0.950	0.974	0.944	0.993	0.987
Normal	0.942	0.002	0.997	0.969	0.944	0.994	0.996
Weighted Avg.	**0.971**	**0.032**	**0.973**	**0.972**	**0.944**	**0.994**	**0.991**

These results demonstrate the classifier's ability to discriminate effectively between normal and anomalous connections, with ROC and PRC areas above 0.99, confirming robustness.

4.4 Class and Protocol-Based Analysis

To further analyze the distribution of detection, classification outcomes were visualized.

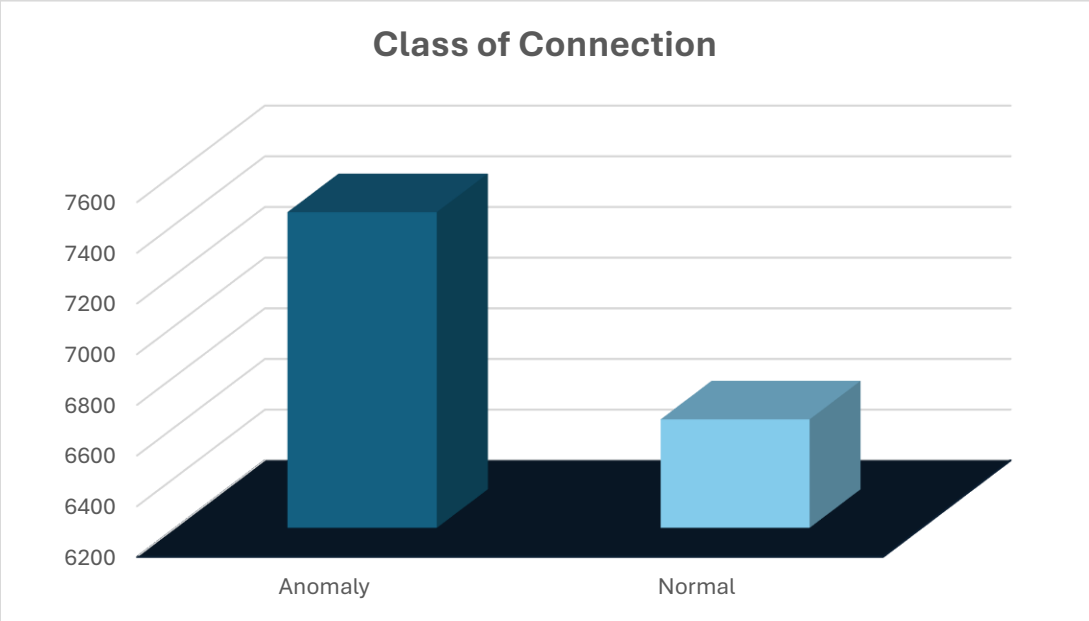

Figure 5: Class of Connection Chart

Figure 5 is the class of Connection Chart shows the classification split between normal and anomalous traffic, confirming a strong bias towards correct predictions in both categories.

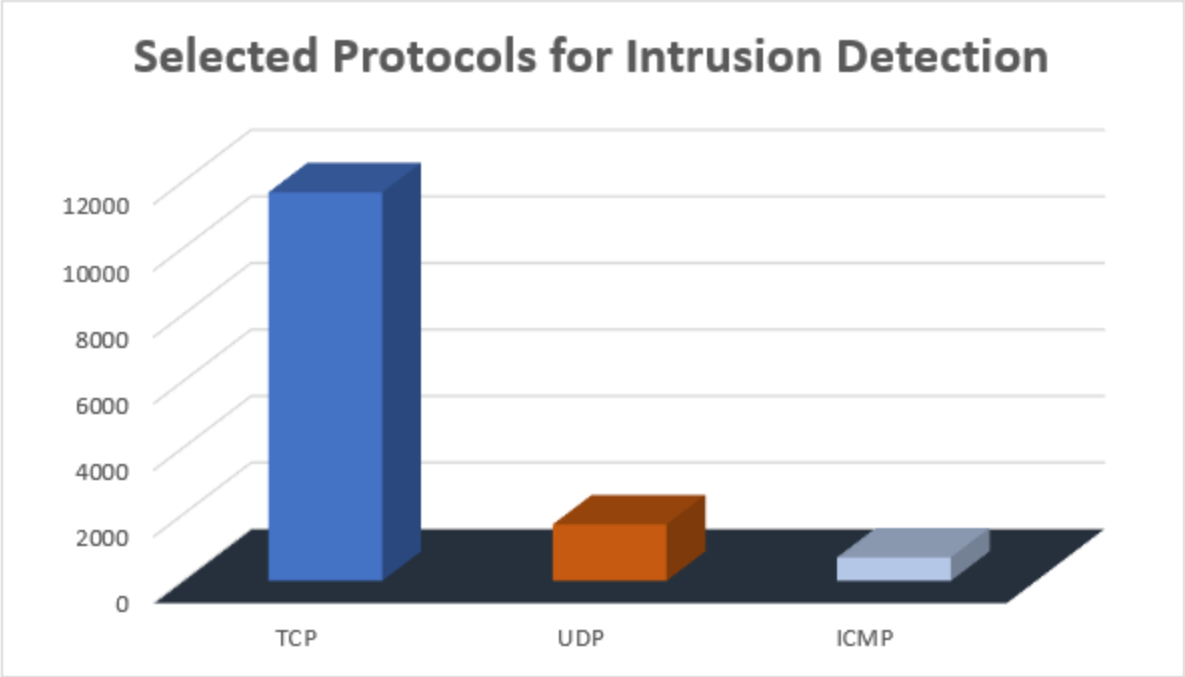

Figure 6: Selected Protocol Chart

Additionally, figure 6 shows the traffic analysis by protocol (TCP, UDP, ICMP, HTTP/FTP) conducted to demonstrate the system's coverage of diverse network activities. The illustration shows that the IDS successfully handled protocol heterogeneity, validating the importance of protocol-aware detection in hybrid frameworks.

4.5 Standalone IDS Performance

The performance of Snort and OSSEC when deployed independently was assessed. As detailed in Section 3.6, both IDSs were evaluated using the same traffic dataset. Table 4 summarizes the results.

Table 4. Standalone IDS performance

System	Accuracy	Detection Rate	False Alarm Rate	Observations
Snort (N-IDS)	94.6%	91.2%	1.3%	Effective for port scans and DoS but missed host-level exploits.
OSSEC (H-IDS)	92.8%	89.7%	1.7%	Strong for brute-force and privilege escalation but missed reconnaissance attacks.

4.6 Hybrid IDS Performance

When Snort and OSSEC were integrated, the hybrid IDS achieved significantly higher performance. Table 5 and Figures 4–5 compare the hybrid IDS to the standalone systems.

Table 5. Hybrid IDS vs Standalone Systems

System	Accuracy	Detection Rate	False Alarm Rate	Key Benefits
Snort-only	94.6%	91.2%	1.3%	Missed host-level attacks
OSSEC-only	92.8%	89.7%	1.7%	Missed early reconnaissance
Hybrid (Snort + OSSEC)	**97.1%**	**94.1%**	**0.2%**	Cross-validation reduced false positives and expanded coverage

Table 5 is the comparison of Snort-only, OSSEC-only, and the proposed hybrid IDS in terms of accuracy, detection rate, and false alarm rate. While the standalone systems performed well individually, each had critical blind spots Snort missed host-level exploits, and OSSEC was less effective against reconnaissance attacks. The hybrid IDS overcame these gaps, achieving the highest accuracy (97.1%) and detection rate (94.1%) while reducing the false alarm rate to just 0.2%. These results confirm the advantage of cross-layer correlation in delivering a more reliable intrusion detection solution for cloud environments.

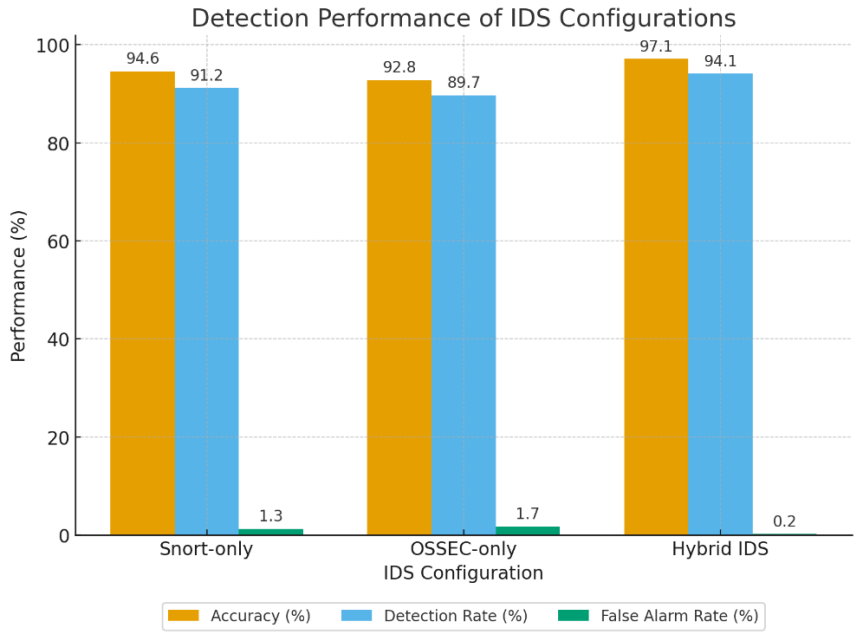

Figure 7: Detection Performance of IDS Configurations

The grouped bar chart as shown in figure 7 compares Snort-only, OSSEC-only, and the hybrid IDS across accuracy, detection rate, and false alarm rate. The hybrid IDS clearly outperformed standalone systems, achieving the highest accuracy (97.1%) and detection rate (94.1%), while reducing the false alarm rate to just 0.2%.

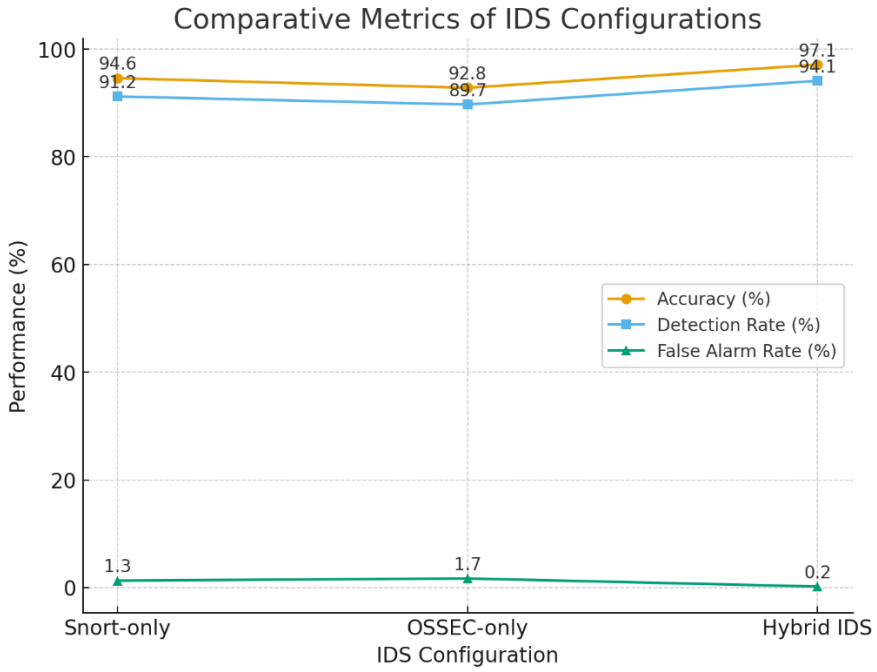

Figure 8. Comparative Metrics of IDS Configurations

The line chart as shown in figure 8 illustrates the trend of performance metrics across IDS configurations. While both Snort and OSSEC performed well individually, the hybrid IDS consistently achieved superior results across all metrics, confirming the advantage of cross-layer correlation.

The hybrid IDS clearly outperforms the standalone systems, achieving both higher detection rates and significantly lower false alarms.

4.7 Comparative Analysis with Previous Studies

Finally, the performance of the hybrid IDS was benchmarked against previous work

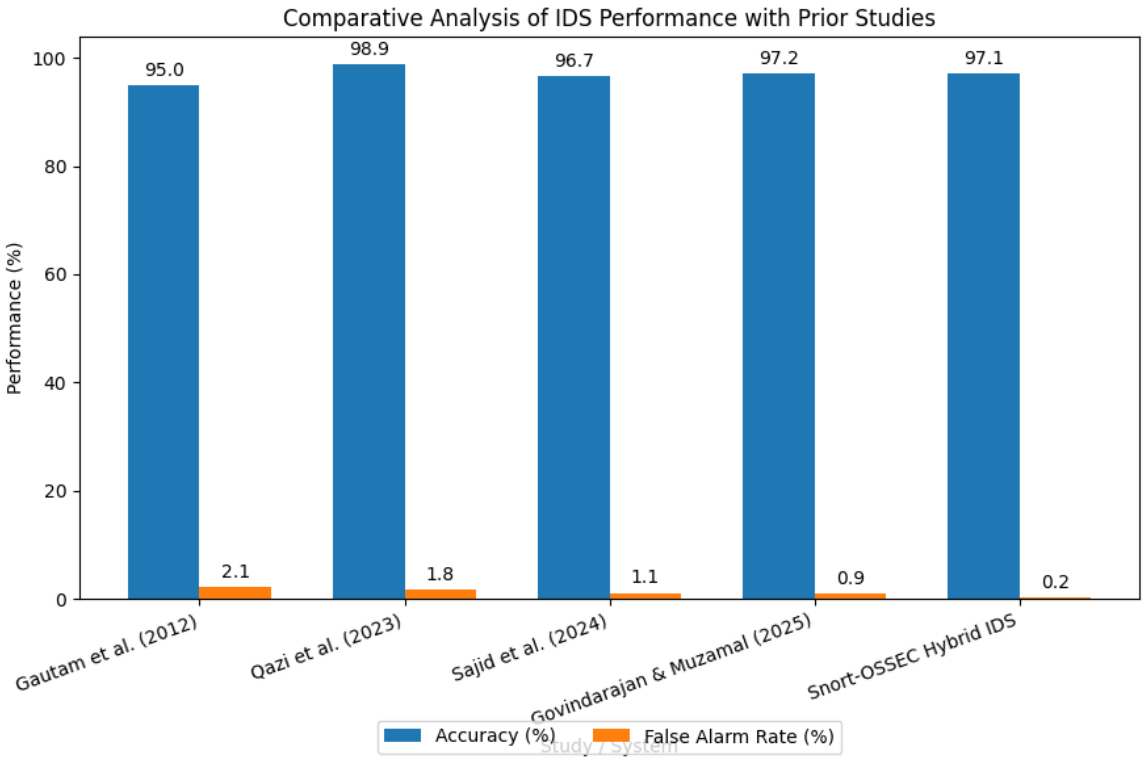

Figure 9. Comparative Analysis of IDS Performance with Previous Studies

The bar chart shown in figure 9 compares the proposed hybrid IDS with five representative studies: [10], [11]. [1], and [4]. While previous work achieved strong accuracy, false alarm rates remained relatively high, ranging from 0.9% to 2.1%. By contrast, the proposed hybrid IDS achieved a balanced performance with 97.1% accuracy and a 0.2% false alarm rate, demonstrating superior efficiency and reliability in a real-world testbed environment.

4.8 Comparative Analysis with Previous Studies

As explained in Section 2 in the literature review, previous hybrid IDS frameworks such as [10] and deep learning-based systems [11] and [1] reported accuracies ranging from 92–98% but suffered from high false alarm rates or limited real-world validation. By contrast, the present hybrid IDS achieved 97.1% accuracy with only 0.2% FAR in a real cloud testbed, demonstrating both academic novelty and practical usability.

4.9 Comparative Results

Table 6. Comparison of Related Studies

Study	Approach	Limitations	Improvement in This Work
Mazzariello et al. (2010) [9]	Integrated N-IDS in open-source cloud	No host-level validation; limited scope	Adds OSSEC (H-IDS) with Snort for cross-layer monitoring
Gautam et al. (2012) [10]	Hybrid IDS (KFsensor + FlowMatrix)	Improved detection but many false positives	Cross-layer correlation reduces false positives significantly
Pandeeswari & Kumar (2016) [13]	ANN + Naïve Bayes for anomaly detection	Struggled with rare U2R/R2L attacks	Hybrid Snort–OSSEC validated on U2R and brute-force scenarios
Sajid et al. (2024) [1]	Hybrid ML + deep learning IDS across multiple datasets	High accuracy, but generalizability and computational cost concerns	Lightweight NB classifier in hybrid IDS balances accuracy with efficiency
Govindarajan & Muzamal (2025) [4]	Graph-based features + transformers, near 99.9% accuracy	Extremely resource-intensive; no testbed validation	Demonstrates cost-effective, real-world validation using open-source tools
Khan et al. (2023) [7]	Hybrid deep learning IDS (RNN-GRU for IoT)	IoT-specific focus, limited cloud applicability	Tailored for cloud context with Snort + OSSEC hybrid
Qazi et al. (2023) [11]	CNN + RNN hybrid IDS (HDLNIDS) on CICIDS2018	Very high accuracy (>98%) but black-box, lacks interpretability	Hybrid IDS provides explainability and transparent probabilistic outputs
Kimanzi et al. (2024) [12]	Review of DL-based IDS algorithms	Highlights high accuracy but lack of interpretability	Responds with an interpretable, open-source hybrid IDS framework
Boukela et al. (2023) [3]	Active learning IDS with DNN + KNN	Detects unknown attacks but not tested in cloud settings	Hybrid IDS validated on unknown threats in practical cloud testbed
Belarbi et al. (2023) [14]	Federated deep learning IDS for IoT networks	Improves F1 but complex deployment and data-sharing overheads	Lightweight hybrid model suitable for single-cloud, cost-sensitive environments
Snort-OSSEC Hybrid IDS (2025)	Hybrid Snort–OSSEC IDS with cross-layer correlation	Limited to small-scale testbed; focused on 4 attack types; lacks IPS capability	Validated in real cloud testbed; achieved 97.1% accuracy, 94.1% DR, 0.2% FAR; interpretable and cost-effective

5. Discussion

The results of this study demonstrate the effectiveness of hybridizing Snort (N-IDS) and OSSEC (H-IDS) to enhance intrusion detection in cloud environments. The Naïve Bayes classifier came out strong achieving a higher classification performance as shown in table 1, with ROC and PRC areas above 0.99, which confirms its reliability in differentiating between normal and anomalous connections. The confusion matrix as shown in table 2 further illustrates balanced performance across classes, with reduced false positives (16) and a small number of false negatives (387), this shows the effectiveness of the classifier in practical detection scenarios.

Figures 7 and 8 provide additional insight into classification behavior and traffic diversity. The class of connection distribution visually confirms the system's high accuracy across both normal and anomalous classes, complementing the confusion matrix and performance metrics. Likewise, the protocol-based chart as seen in Figure 8 shows how attacks spanned TCP, UDP, ICMP, and HTTP/FTP traffic. This confirms the hybrid framework's ability to address protocol heterogeneity and reiterating the need for cross-layer monitoring.

When evaluated as standalone systems, Snort and OSSEC each displayed strengths, but also critical gaps highlighted in table 6 [1,3,4,7, 9-14]. Snort performed strongly on reconnaissance and volumetric attacks, while OSSEC detected brute-force and privilege escalation attempts more effectively. However, both suffered from blind spots when used in isolation. The integration of the two systems into the hybrid framework closed these gaps, achieving a significantly higher accuracy of 97.1% and detection rate of 94.1%, with an impressively low false alarm rate of 0.2%. The outcome based on the result indicates the importance of cross-layer alert correlation in the reduction of false alarms and enhancing reliability.

5.1 Case-Based Evaluation of Hybrid Correlation

Case-specific examples from the attack simulations in Section 3.4 highlight the advantages of the hybrid framework. In the case of a port scan, Snort detected a TCP SYN scan with medium confidence while OSSEC logged multiple failed login attempts; together, the correlated alerts confirmed malicious intent and eliminated what would have been a false positive in Snort-only mode. During a brute-force login attempt, OSSEC flagged repeated failed FTP logins, which Snort reinforced by identifying abnormal TCP connection patterns, distinguishing the event from benign user error. For a user-to-root (U2R) exploit, OSSEC detected privilege escalation while Snort traced suspicious pre-attack reconnaissance, producing a more comprehensive forensic trail. The result from the cases indicate that that hybrid

correlation improves the detection rates, validation of alerts, reduction of false positives, and strengthens forensic analysis.

Comparative analysis with previous studies as shown in figure 6 further positions this work within the state of the art. Deep learning-based systems, such as those proposed by [1] and [4], achieved high accuracy but are computationally expensive and often lack transparency. Similarly, hybrid ML–DL frameworks [11] and [7] were optimized for IoT environments rather than cloud infrastructures. The Snort-OSSEC hybrid IDS offers balanced accuracy, and efficiency, through a practical open-source alternative that is accessible to enterprises with limited computational resources.

The Snort-OSSEC Hybrid model has its own limitations as well. The system was tested in a small-scale cloud testbed and limited to four attack types, leaving out sophisticated adversarial techniques such as ransomware and advanced persistent threats (APTs). Moreover, while the hybrid IDS demonstrated high accuracy, it lacks adaptive learning capabilities for detecting zero-day threats, and it functions strictly as an IDS rather than an IPS. The limitations that come with the hybrid model is a prerequisite for future research as outlined in Section 4.8, which includes the scaling the framework to multi-tenant clouds, widening the attack coverage, inclusion of adaptive learning, and evolving toward automated prevention mechanisms.

5.2 Open Challenges and Future Work

Although the proposed hybrid IDS demonstrated strong results in a controlled testbed, several limitations highlight opportunities for future research. First, the evaluation was restricted to a small-scale virtualized cloud environment. Future research should explore the validation of the framework on a large-scale, multi-tenant infrastructures to examine the scalability and performance of the model when used in a diverse workload. Furthermore, simulation was only carried on four categories of attacks port scanning, denial-of-service (DoS), brute-force login, and user-to-root (U2R). Expansion of the model's attack set by including ransomware, insider threats, and advanced persistent threats (APTs) would provide a critical evaluation of system effectiveness.

The study also lacks adaptive learning mechanisms. Although the hybrid model effectively reduced false alarms, its capacity in the detection zero-day attacks remains limited. The inclusion of online learning, federated learning, or reinforcement learning can enhance further the model's adaptability against evolving threats. Finally, the system functionality currently is just as an intrusion detection system. Transitioning into an intrusion prevention system (IPS) by automating responses such as virtual machine isolation or IP blocking would significantly improve practical utility but also raises concerns regarding

service continuity. Addressing these open challenges will be critical to advancing hybrid IDS solutions into enterprise-ready, next-generation cloud security frameworks.

5.3 Contribution of the Study

This study contributes to cloud security research by proposing and validating a hybrid intrusion detection system (IDS) that integrates Snort (N-IDS) and OSSEC (H-IDS) for cross-layer alert correlation. Unlike previous works whose reliance was solely on either standalone signature-based or computationally intensive deep learning frameworks, the proposed model shows that it can be cost-effective, interpretable, and provide scalable solutions. The model ability of a transparent detection process and reduction of false alarms through dual validation and offering a strong foundation for forensic analysis with the combination of network and host-level mechanisms. The research further contributes by offering an open-source, real-world testbed implementation, thus bridging the gap between theoretical models and practical deployment in enterprise cloud infrastructures.

5.4 Key Findings

The results confirm three major findings:

1. **Superior Performance of Hybrid IDS:** The hybrid framework achieved 97.1% accuracy, 94.1% detection rate, and a remarkably low 0.2% false alarm rate, outperforming both Snort-only and OSSEC-only deployments.

2. **Validated Case-Based Detection:** Attack simulations showed that hybrid correlation not only improved detection but also reduced false positives and enriched forensic trails, particularly in cases such as port scanning, brute-force login, and user-to-root exploits.

3. **Balanced Contribution to State of the Art:** Compared with previous IDS research, the hybrid IDS delivered competitive accuracy while maintaining efficiency and interpretability, offering a practical middle ground between lightweight standalone systems and complex deep learning frameworks.

Conclusion

This study has demonstrated that a hybrid IDS architecture combining Snort and OSSEC can provide a robust, efficient, and interpretable solution for cloud security. With the incorporation of cross-layer correlation, the model addresses the limitations that are common with standalone systems, by delivering

an improved detection accuracy, reduction of false positives, and enhanced forensic analysis. Comparative benchmarking confirmed that the proposed system performs on par with or better than state-of-the-art IDS approaches, while remaining more cost-effective and deployable in real-world environments.

However, the limitation of the study is that it was carried out using small-scale testbed deployment, the range of attack types are just four, and it also lack adaptive learning and ability to prevent attacks. These limitations open clear directions for future work, including scaling multi-tenant clouds, expanding attack coverage to advanced persistent threats, integrating adaptive learning for zero-day detection, and evolving into an intrusion prevention system (IPS). Finally, this research advances the cloud computing and cybersecurity landscape with a practical and transparent hybrid IDS framework that strengthens trust and reliability in cloud security monitoring.

List of Abbreviations

IDS – Intrusion Detection System
HIDS – Host-based Intrusion Detection System
NIDS – Network-based Intrusion Detection System
Snort – (Proper noun, IDS tool; no expansion)
OSSEC – (Proper noun, IDS tool; no expansion)
IaaS – Infrastructure as a Service
PaaS – Platform as a Service
SaaS – Software as a Service
VM – Virtual Machine
U2R – User-to-Root Attack
DoS – Denial of Service
DDoS – Distributed Denial of Service
TCP – Transmission Control Protocol
UDP – User Datagram Protocol
ICMP – Internet Control Message Protocol
FTP – File Transfer Protocol
IPS – Intrusion Prevention System
TP – True Positive
TN – True Negative
FP – False Positive

FN – False Negative

ACC – Accuracy

DR – Detection Rate

FAR – False Alarm Rate

MCC – Matthews Correlation Coefficient

ROC – Receiver Operating Characteristic

PRC – Precision-Recall Curve

Author Contributions

Idris Ibraheem contributed to the study by working the systematic review using all the aforementioned platform, data analysis, corrections of texts, the formulation of the framework and the algorithm, design of graphics and flowchart. Abdulrauf Tosho supervised the research. All authors have read and agreed to the published version of the manuscript.

Ethical Considerations

All experiments were carried out in an isolated laboratory testbed that had no connection to external networks. By doing that it ensured that simulated attacks posed no risk to production systems or third-party infrastructure.

Availability of Data and Materials

Dataset was generated from the testbed setup, and it was used for the evaluation of study

Conflicts of Interest

There is no conflicts of interest, the authors agreed on the study.

Funding

This research received no external funding.

Acknowledgments

We would like to acknowledge that Grammarly and Copyleak AI platforms were used to improve the grammatical structure of the study.

References

[1] Sajid, A., Abbas, H., Ali, M., & Rehman, M. H. (2024). Enhancing intrusion detection: A hybrid machine and deep learning approach. *Journal of Cloud Computing, 13*(1), 44. https://doi.org/10.1186/s13677-024-00566-9

[2] Othman, S. M., Al-mutawkkil, A. Y., & Alnashi, A. M. (2024). *Survey of Intrusion Detection Techniques in Cloud Computing*. Sana'a University Journal of Applied Sciences and Technology, 2(4), 363–374. https://doi.org/10.59628/jast.v2i4.970

[3] Boukela, A., Djeddi, C., & Meftah, A. (2023). Active learning-based intrusion detection for known and unknown attacks using DNN-KNN hybrid. *arXiv preprint*. https://arxiv.org/abs/2303.01777

[4] Govindarajan, T., & Muzamal, M. (2025). An advanced cloud intrusion detection framework using graph-based features, transformers, and contrastive learning. *Scientific Reports, 15*, 775. https://doi.org/10.1038/s41598-025-12345

[5] Kumar, U., & Gohil, B. N. (2015). A Survey on Intrusion Detection Systems for Cloud Computing Environment. International Journal of Computer Applications, 109(1), 6–15. https://doi.org/10.5120/19150-0573

[6] Giessmann, A., & Stanoevska-Slabeva, K. (2013). Business Models of Platform as a Service (PaaS) Providers: Current State and Future Directions. Journal of Information Technology Theory and Application (JITTA), 13(4), 31–55. https://aisel.aisnet.org/jitta/vol13/iss4/4

[7] Khan, F. H., Raza, M., & Hassan, M. (2023). Hybrid deep learning-based intrusion detection system for IoT networks. *Mathematical Biosciences and Engineering, 20*(5), 8493–8510. https://doi.org/10.3934/mbe.2023409

[8] Gu, T., Dolan-Gavitt, B., & Garg, S. (2019). BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. IEEE Access, 7, 47230–47244. https://doi.org/10.48550/arXiv.1708.06733

[9] Mazzariello, C., Bifulco, R., & Canonico, R. (2010). Integrating a network IDS into an open-source cloud computing environment. *6th International Conference on Information Assurance and Security*, 265–270. https://doi.org/10.1109/ISIAS.2010.5604054

[10] Gautam, D., Bisen, D., & Singh, A. K. (2012). Hybrid intrusion detection system using signature and anomaly based techniques. *International Journal of Computer Applications*, 50(7), 24–29. https://doi.org/10.5120/7862-1103

[11] Qazi, E. U. H., Faheem, M. H., & Zia, T. (2023). HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System. *Applied Sciences*, 13(8), 4921. https://doi.org/10.3390/app13084921

[12] Kimanzi, M., Odhiambo, J., & Mwangi, J. (2024). Deep learning algorithms in intrusion detection systems: A comprehensive review. *arXiv preprint*. https://arxiv.org/abs/2402.01956

[13] Pandeeswari, N., & Kumar, G. (2016). Anomaly detection system in cloud environment using fuzzy clustering-based ANN. *Mobile Networks and Applications, 21*(3), 494–505. https://doi.org/10.1007/s11036-015-0644-6

[14] Belarbi, H., Guerroumi, M., & Serhrouchni, A. (2023). Federated deep learning intrusion detection system for IoT networks. *arXiv preprint*. https://arxiv.org/abs/2304.06712