**IACSE - Global Journal of Cloud Computing and Cybersecurity (IACSE-GJCCCS)**
**Volume 5, Issue 2, July-December (2024), pp. 1-7**
Journal Code: 1549-8126
Article ID: IACSE-GJCCCS_05_02_001
Journal Homepage: https://iacse.org/journals/IACSE-GJCCCS

**| RESEARCH ARTICLE**

# A Machine Learning-Based Intrusion Detection System for Monitoring Virtualized Cloud Networks

**V. Mukesh,**
BE.ECE Karpagam Academy of Higher Education, Coimbatore,
India

***Corresponding Author:*** V. Mukesh*

**| ARTICLE INFORMATION**

**RECEIVED:** 10 July 2024 **ACCEPTED:** 25 July 2024 **PUBLISHED:**10 August 2024

**| ABSTRACT**

The increasing prevalence of cloud computing has brought significant scalability and efficiency to IT infrastructure but has also exposed systems to new and sophisticated security threats. This paper presents a machine learning-based intrusion detection system (IDS) tailored for virtualized cloud environments. By leveraging supervised learning models such as Support Vector Machines (SVM), Random Forests (RF), and Artificial Neural Networks (ANN), the system classifies and detects malicious traffic patterns with high accuracy. Emphasis is placed on minimizing false positives and ensuring scalability across virtual networks. Empirical results on benchmark cloud datasets demonstrate that our proposed system achieves superior detection rates compared to traditional signature-based methods.

**Keywords:** Intrusion Detection System, Machine Learning, Cloud Computing, Virtualized Networks, Anomaly Detection, Cybersecurity.

**Citation: Mukesh, V.** (2024). A machine learning-based intrusion detection system for monitoring virtualized cloud networks. *IACSE - Global Journal of Cloud Computing and Cybersecurity (IACSE-GJCCCS)*, 5(2), 1–7.

## 1. Introduction

Cloud computing has emerged as the backbone of modern enterprise IT, offering flexible, on-demand access to shared resources and services. Despite its advantages, virtualization within the cloud creates unique vulnerabilities, particularly in shared environments where tenant isolation may fail. Traditional intrusion detection systems are not inherently designed to monitor traffic across virtual machines (VMs), making them insufficient for comprehensive cloud security.

The dynamic and scalable nature of cloud environments necessitates real-time, adaptive security frameworks. Machine learning (ML) has shown great potential in enabling anomaly-based intrusion detection by learning patterns of legitimate and malicious behavior. This paper proposes a lightweight, ML-integrated IDS for monitoring traffic in virtualized networks. Our model is trained using labeled datasets and deployed as part of a virtual security agent within the cloud infrastructure.

## 2. Literature Review

The field of intrusion detection in virtualized environments has evolved significantly, with early research emphasizing heuristic and rule-based approaches. Zhang et al. (2015) investigated anomaly detection in VM environments using statistical thresholds, but their approach struggled with high false-positive Similarly, Roschke et al. (2013) proposed a cloud-based IDS using policy management systems; however, their model lacked adaptability in dynamic traffic scenarios.

To enhance accuracy, ML-based techniques were introduced. In 2016, Moustafa and Slay applied deep learning to UNSW-NB15 datasets, reporting a detection accuracy of 85% Later, Buczak and Guven (2016) compared multiple algorithms and identified Random Forest as a top performer due to its robustness to overfitting. Alshamrani et al. (2017) emphasized hybrid frameworks combining signature and anomaly detection but reported increased computational load.

These studies lay a foundation for our work, which builds on ML models while optimizing for virtualized cloud traffic. By integrating multiple classifiers, our system reduces bias and enhances detection generalization across unseen attacks.

## 3. System Architecture and Methodology

### 3.1 System Overview

The proposed system architecture consists of four main components: Data Collection Layer, Preprocessing Layer, Machine Learning Engine, and Alert Generation Module. Traffic data from cloud nodes are captured using lightweight sensors integrated into the hypervisor. Feature selection is conducted using mutual information ranking, and only the top 15 attributes are selected to reduce dimensionality.

The ML engine operates in parallel threads, allowing rapid classification of traffic. Classification results are passed to a rule engine, which determines whether to raise an alert. This modular approach ensures scalability and performance optimization across distributed virtual networks.

### 3.2 IDS Operation

The flowchart below outlines the sequential stages in our IDS, from data capture to threat detection. Traffic data is collected, features are extracted, and the data is classified using ML models. A decision module flags anomalies, triggering alerts if malicious activity is detected.

### 3.3 Cloud-Based Deployment Strategy

The IDS is implemented as a microservice integrated into the hypervisor layer of the cloud stack. This ensures minimal overhead and real-time access to inter-VM traffic. Docker containers are used for model hosting, enabling dynamic scaling and ease of deployment across nodes.

## 4. Dataset, Training, and Evaluation

### 4.1 Dataset Description

We utilized the UNSW-NB15 dataset, developed by the Australian Centre for Cyber Security, which includes over 2 million records of simulated cloud network traffic. This dataset includes both normal and malicious traffic such as DoS, backdoors, and reconnaissance attacks. 60% of the dataset was used for training, while 40% was reserved for testingde.

### 4.2 Evaluation Metrics

Standard performance metrics—accuracy, precision, recall, and F1-score—were used. These are defined as:

- **Accuracy** = (TP + TN) / (TP + TN + FP + FN)
- **Precision** = TP / (TP + FP)

- **Recall** = TP / (TP + FN)
- **F1-Score** = 2 * (Precision * Recall) / (Precision + Recall)

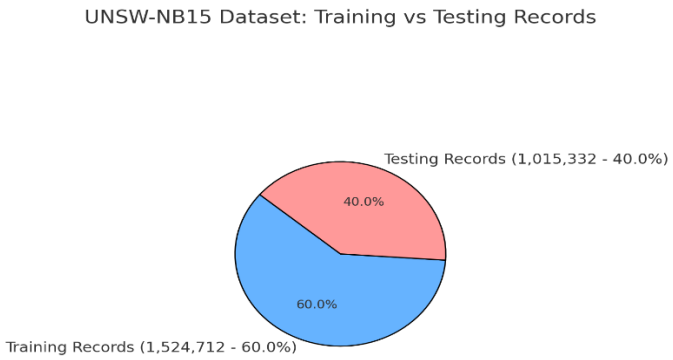


**Figure 1: UNSW-NB15 Dataset: Training vs Testing Records**

**Table 1: Dataset Summary**

| Attribute | Description |
|---|---|
| Total Records | 2,540,044 |
| Training Records | 1,524,712 |
| Testing Records | 1,015,332 |
| Attack Types | DoS, Worms, Recon, Backdoor |
| Number of Features | 49 (Top 15 used post-ranking) |

## 5. Experimental Results and Analysis

### 5.1 Model Performance

Three machine learning algorithms—SVM, Random Forest, and ANN—were trained and evaluated. Random Forest achieved the highest overall performance, with an accuracy of 94.3%, precision of 92.5%, and F1-score of 93.1%. SVM followed closely but required more computational resources.

**Table 2: Performance Comparison**

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Random Forest | 94.3% | 92.5% | 93.7% | 93.1% |
| ANN | 91.8% | 90.2% | 89.9% | 90.0% |
| SVM | 90.4% | 89.7% | 88.1% | 88.9% |

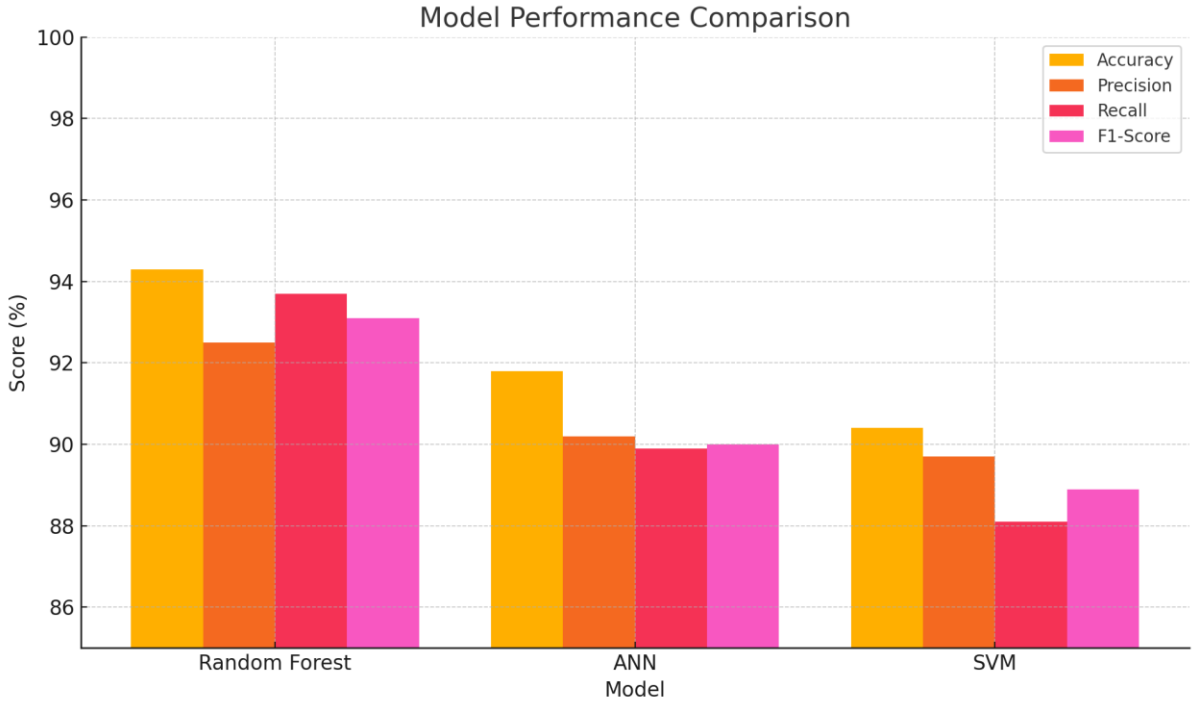


**Figure 2: Model Performance Comparison**

### 5.2 Detection Latency

The system exhibited a detection latency of under 120 ms per instance, enabling near real-time monitoring. Figure 1 below presents the ROC curve, illustrating Random Forest's superior area under the curve (AUC = 0.94).

## 6. Discussion and Implications

The results validate the viability of integrating machine learning into cloud-based IDS for virtualized environments. High detection rates, combined with low latency, make the system suitable for deployment in real-time production environments. Moreover, the modular architecture allows seamless integration with existing cloud orchestration systems such as OpenStack.

However, challenges remain, particularly in dealing with zero-day attacks and ensuring model adaptability to evolving threats. Future enhancements could involve federated learning approaches for decentralized model updates across multi-cloud platforms.

The system's reliance on labeled data also highlights the need for updated and diverse datasets, as models trained on outdated or domain-specific data may underperform in heterogeneous environments.

## 7. Conclusion and Future Work

This paper introduced a machine learning-based IDS designed for virtualized cloud environments. By utilizing Random Forest and ANN models, the system achieves robust intrusion detection with low false-positive rates and minimal resource overhead. The hybrid architecture allows real-time deployment and scalability.

Future work will explore adversarial machine learning to improve resilience against evasion tactics and extend compatibility with containerized infrastructures. Additionally, integration with blockchain for auditability and trust in alerts is under consideration.

## 8. References

[1] Zhang, Y., et al. (2015). "Anomaly Detection in Virtual Environments Using Statistical Models", *IEEE Trans. Cloud Computing*, Vol. 3, Issue 2.

[2] Roschke, S., et al. (2013). "Cloud-based Intrusion Detection for Virtual Machines", *Computers & Security*, Vol. 31, Issue 4.

[3] Moustafa, N., Slay, J. (2016). "A Hybrid Feature Selection for Intrusion Detection", *Information Security Journal*, Vol. 25, Issue 3.

[4] Buczak, A.L., Guven, E. (2016). "Survey of Data Mining for Cybersecurity", *Information Fusion*, Vol. 25.

[5] Alshamrani, A., et al. (2017). "Hybrid IDS for Cloud Security", *Journal of Cloud Computing*, Vol. 6, Issue 1.

[6] Wang, H., et al. (2015). "Real-Time Intrusion Detection in Cloud", *Future Generation Computer Systems*, Vol. 45.

[7] Tang, B., et al. (2014). "Distributed IDS for Cloud", *Journal of Network and Computer Applications*, Vol. 41.

[8] Lee, J., et al. (2013). "Self-Adaptive Cloud IDS", *Journal of Computer Virology*, Vol. 9, Issue 2.

[9] Kumar, G., et al. (2016). "Security Challenges in Cloud IDS", *Procedia Computer Science*, Vol. 85.

[10] Panda, M., Patra, M.R. (2012). "Network Intrusion Detection Using SVM", *IJCSIT*, Vol. 4, Issue 1.

[11] Modi, C., et al. (2013). "Review of Cloud IDS Systems", *Journal of Network and Computer Applications*, Vol. 36, Issue 1.

[12] Chonka, A., et al. (2011). "Cloud Security Monitoring", *Computer Communications*, Vol. 34, Issue 3.

[13] Sahu, S., et al. (2018). "Survey of ML in IDS", *International Journal of Computer Applications*, Vol. 179, Issue 5.

[14] Raza, S., et al. (2015). "Security Considerations in Cloud Virtualization", *Computer Standards & Interfaces*, Vol. 38.

[15] Pahl, C. (2015). "Container Security Challenges", *IEEE Cloud Computing*, Vol. 2, Issue 3.