Reinforcement Learning Driven Self-Adaptation in Hypervisor-Based Cloud Intrusion Detection Systems (RLDAC-IDS)

Alaa A. Qaffas
Department of Management Information Systems-College of Business, University of Jeddah, Jeddah, KSA

***Abstract*—With the rise in cloud adoption, securing dynamic virtual environments remains a significant challenge. While traditional Intrusion Detection Systems (IDS) have attempted to address security concerns in the cloud mostly through static detection rules and without adaptation capabilities to identify new attack vectors, a self-optimizing framework called Reinforcement Learning-Driven Self-Adaptation in Hypervisor-Based Cloud Intrusion Detection Systems (RLDAC-IDS) is suggested to overcome this limitation. RLDAC-IDS leverages the inherent visibility of hypervisors into virtualized resources to gain valuable insights into cloud operations and threats. Its key components include real-time behavioral analysis, anomaly detection, and identification of known threats. The innovation of RLDAC-IDS lies in the incorporation of reinforcement learning to continuously improve the detection rules and responses. RLDAC-IDS exemplifies intelligent intrusion detection through its ability to learn and adapt to new threat patterns autonomously. By continuous optimization and intelligent intrusion detection techniques, the system progresses to tackle emerging attack vectors while minimizing false alarms. In contrast, RLDAC-IDS is highly adaptive and can easily adjust to the changing conditions of cloud environments. In summary, RLDAC-IDS represents a major advancement in cloud IDS through its adaptive, self-learning approach, overcoming the limitations of existing solutions to provide robust protection amidst the complexities and dynamics of modern virtualized settings.**

Keywords*—Cloud security; intrusion detection system; adaptive framework; hypervisor-based IDS; self-adaptation; emerging threat detection; reinforcement learning; behavioral analysis; cloud computing; intelligent intrusion detection

I. Introduction

Cloud computing has become a disruptive paradigm in information technology, offering benefits such as increased resource utilization and significant cost savings in infrastructure. At its core, cloud computing leverages current technologies like service-oriented architecture, virtualization, and utility computing. Among these, virtualization emerges as the cornerstone of cloud computing infrastructure [1]. It enables efficient sharing of physical machine resources, including CPU, memory, I/O, and network interfaces, among multiple virtual machines coexisting on the same physical host [2], [3].

While the advantages of cloud computing and virtualization are evident in their ability to optimize resource allocation and streamline operations, they introduce a pivotal challenge for cloud service providers (CSPs). This challenge lies in safeguarding the virtualized resources of Guest Operating Systems (GOS) against an ever-evolving landscape of advanced and sophisticated cyberattacks [4]. As CSPs strive to harness the power of virtualization to deliver cloud services, the imperative of fortifying these virtualized environments against security threats becomes increasingly paramount. This introduction sets the stage for a comprehensive exploration of the intricacies surrounding the protection of virtualized resources within cloud computing, addressing the multifaceted challenges and highlighting the strategies and solutions vital to this endeavor [5].

Server virtualization allows for the allocation of CPU, RAM, and other dynamic computing resources as needed. This is achieved through a pay-as-you-go approach, where customers, referred to as tenants, are only billed for utilities. Infrastructure as a Service (IaaS) is a widely accepted cloud computing model that enables users to utilize virtual machines (VMs) and virtual networks to access and manage computers, storage, and network resources. This is facilitated through the provision of an information system and the added benefits of access to unlimited computing and communication capacity, as outlined in the service level agreements between tenants and cloud providers [6], [7].

The virtual machine contains a hypervisor, a virtualization component that enables devices to share resources. However, it also poses its own set of risks. If an attacker breaches a hypervisor, they can take control of the entire virtual environment. Complications in cloud computing further complicate matters as attacks can originate from various sources like virtual networks, malicious hypervisors, and other virtual machines. Virtual machines are prime targets for attackers due to their susceptibility to hijacking, as they are linked to the external virtual world through the CSP [8].

The implementation of an intrusion detection system (IDS) is essential in safeguarding the entirety of a virtual machine against a multitude of potential threats and attacks [9]. The Intrusion Detection System (IDS) should possess the functionality to identify malware, analyze logs, check file integrity, analyze incoming data, and provide an active reaction [10]. Furthermore, it is imperative that the system possess the capability to detect both unidentified and recognized attacks. Solely relying on detection measures is insufficient for ensuring the protection of the virtual environment without the implementation of preventive measures. This study presents a proposed framework called Self-Adaptive Framework for

Hypervisor-Based Cloud Intrusion Detection System (HVCIDS). The purpose of this framework is to identify and react to unauthorized or malicious actions occurring in a cloud computing environment.

The proposed Reinforcement Learning-Driven Self-Adaptation in Hypervisor-Based Cloud Intrusion Detection Systems (RLDAC-IDS) offers several significant advancements in cloud security. The primary contributions of this work are as follows:

- Adaptive Reinforcement Learning Framework: We introduce a novel approach that integrates reinforcement learning to enable continuous self-adaptation of detection rules and responses. This allows RLDAC-IDS to evolve dynamically with the changing threat landscape, addressing a critical limitation of static rule-based systems.
- Hypervisor-Level Monitoring: By leveraging hypervisor-based visibility, RLDAC-IDS gains comprehensive insights into virtualized resources and cloud operations, enabling more granular and effective threat detection across the entire cloud infrastructure.
- Multi-Faceted Detection Approach: Our system combines real-time behavioral analysis, anomaly detection, and known threat identification, providing a robust, layered defense against a wide spectrum of attack vectors, including zero-day threats.
- Resource Efficiency: RLDAC-IDS demonstrates significantly lower CPU utilization (12.4%) compared to traditional approaches, making it particularly suitable for cloud environments where resource optimization is crucial.
- Enhanced Performance Metrics: Our experimental results show that RLDAC-IDS achieves superior accuracy (98.7%), precision (98.5%), recall (97.9%), and F1-score (97.5%) compared to existing intrusion detection techniques, indicating substantial improvements in overall effectiveness.
- Balanced False Positive/Negative Mitigation: With a high precision rate and low error rate (1.3%), RLDAC-IDS effectively distinguishes between legitimate and malicious activities, addressing the common challenge of alert fatigue in intrusion detection systems.
- Scalability and Adaptability: The self-learning nature of RLDAC-IDS enables it to scale effectively and adapt rapidly to the dynamic and complex nature of modern cloud environments, providing robust protection against evolving cyber threats.

The subsequent sections of this paper are structured as follows: Section II provides a comprehensive background on the hypervisor, visualization, and intrusion detection systems. Section III offers an in-depth analysis of previous studies. In Section IV, we expound upon the research methodology employed, delineating the intricacies of the proposed Hypervisor-based Cloud Intrusion Detection System (HVCIDS) framework. Within this section, readers will find elucidations in the form of model algorithms and an exposition of the performance metrics utilized in our study. Furthermore, Section V encompasses the presentation of our research findings. In Section VI, an extensive discussion of the results is undertaken, providing critical insights and interpretations. Lastly, the concluding Section VII encapsulates our paper with definitive conclusions drawn from the research conducted and offers valuable suggestions for potential avenues of future research exploration.

II. Background

In this section, we provide a brief introduction to key topics relevant to our research. We introduce hypervisors, outlining their basic functions and the advantages they offer in cloud environments. Next, we explore intrusion detection systems (IDS), explaining their purpose and distinctions. Additionally, we examine the transformative influence of cloud computing and the significance of visualization in cloud systems. This fundamental knowledge establishes the groundwork for the subsequent section of our research paper.

A. Hypervisors

Hypervisors are a type of virtual machine monitor (VMM). They are the main drivers behind virtualization and cloud computing. Hypervisors serve as a crucial layer of abstraction, facilitating the creation and management of virtual machines (VMs) that operate on physical hardware [11]. These systems can be software-based or hardware-based. Hypervisors enable multiple VMs to coexist and function independently on a single computer. By acting as a bridge between physical hardware and a virtualized environment, hypervisors efficiently allocate resources from the computer, such as CPU, memory, storage, and networking, to the VMs [12].

Hypervisors make it easier to create and manage virtual machines (VMs), providing a significant advantage to computing environments. They consolidate different resources, enabling multiple VMs to operate on a single server. This results in substantial cost savings and enhanced energy efficiency. Hypervisors excel in security by offering robust isolation, allowing each VM to function independently to safeguard data. They also introduce hardware independence, simplifying resource management, seamless VM migration, and quick adaptation in computing environments. The combined benefits of hypervisors make them ideal for efficient resource utilization, stringent security protocols, and readiness for dynamic operations [13].

B. Intrusion Detection System

Intrusion detection systems (IDS) are the foundation of modern cybersecurity methods, offering critical capabilities for monitoring and protecting networked systems and resources. At their core, intrusion detection systems (IDS) are intended to monitor and analyze network traffic and system operations in real time, identifying patterns or behaviors that deviate from established standards [14]. When such anomalies are detected, intrusion detection systems (IDS) generate alerts or take programmed actions to mitigate potential threats. Based on their detection methodologies, IDS can be classified into three basic types:

1) Signature-based IDS: This category utilizes predetermined patterns or signatures of known attacks to detect threats. Signature-based intrusion detection systems operate by comparing network traffic or system actions to a database of preset signatures. An alert is generated when a match is identified. While effective against known threats, signature-based intrusion detection systems (IDS) may face challenges with zero-day attacks (vulnerabilities that were previously unidentified) and may generate false positives [15].

2) Anomaly-based intrusion detection systems: Anomaly-based IDS, in contrast, focuses on detecting deviations from established baselines of normal behavior. These systems employ machine learning or statistical algorithms to learn typical patterns of network traffic and system activities over time. When they identify activities that significantly differ from the norm, they raise alerts. Anomaly-based IDS excel at identifying novel and emerging threats, as they are not reliant on predefined signatures. However, they may require more sophisticated algorithms and generate alerts for benign anomalies, necessitating careful tuning [16].

3) Hybrid-based intrusion detection systems: Hybrid-based intrusion detection systems combine elements of both signature-based and anomaly-based techniques. These systems utilize predefined signatures for identified threats while also monitoring network traffic and system activity for irregularities. By integrating these methods, hybrid-based intrusion detection systems (IDS) aim to enhance detection accuracy by minimizing false positives and effectively identifying both known and novel threats. Nevertheless, they may pose challenges in terms of configuration and maintenance [17], [18].

C. Cloud Computing and Virtualization

Cloud computing has revolutionized how IT services are delivered, bringing about a change. This innovative approach involves providing computer resources on demand via the internet including a range of services, like servers, storage, databases, networking, software and more. Organizations have enthusiastically adopted cloud computing due to its ability to dynamically scale resources reduce costs and enhance flexibility.

The key, to the flexibility and efficiency of cloud computing lies in virtualization. Virtualization is the technology that enables the creation and management of instances of computer resources allowing for efficient utilization of physical hardware. In a virtualized environment these virtual instances operate independently from the underlying infrastructure enabling resource allocation based on demand and effective resource management.

Virtualization is primarily implemented through the utilization of hypervisors which serve as the hub for generating and managing virtual machines (VMs). Hypervisors create an abstraction layer, between the hardware and these VMs enabling VMs to function independently on a single physical host. This setup ensures resource separation. Maximizes hardware efficiency. By leveraging hypervisors organizations can fully leverage the potential of virtualization to create and maintain adaptable computing environments.

Virtualization is critical to enabling the key features of cloud services outlined the National Institute of Standards and Technology (NIST), like on-demand access and rapid flexibility. Cloud providers use hypervisor software to efficiently allocate resources, isolate environments securely, and make virtual machines easy for customers to create and control. Virtualization provides the core foundation for dynamic resource management, workload scalability, and optimized efficiency that define cloud computing. It allows cloud platforms to be agile and adaptable in meeting compute needs.

III. Literature Review

This section provides a comprehensive overview of recent advancements in intrusion detection systems, particularly focusing on cloud environments and adaptive techniques. This categorized into signature-based, anomaly-based, hybrid-based, and hypervisor-based approaches, concluding with a comparative analysis of these works against our proposed RLDAC-IDS.

A. Signature-based Intrusion Detection Systems

Lo et al. provide an integrated intrusion detection system model designed to address the issues of protecting cloud computing networks. Individual IDS units placed on each server inside the cloud architecture are used in this manner. These IDS units are unusual in that they combine a signature database with a block table, allowing them to keep track of recent assaults. This method prioritizes the evaluation of packets that are more likely to be related to recent attacks, improving the system's responsiveness. This framework's contributions include its novel technique for prioritizing packet inspection and its promise to improve the security of cloud computing networks. This method may face challenges in maintaining and managing block tables in dynamic cloud environments [19].

Lin et al. propose an efficient and effective Network Intrusion Detection System (NIDS) tailored specifically for cloud virtualization environments. The approach they proposed is based on a rule-based NIDS designed to detect known attacks within the cloud setting. The advantages of this approach include its ability to remain responsive to real-time changes in VM behavior and its effectiveness in identifying known attacks. However, managing and updating rules for numerous VMs in highly dynamic cloud settings could prove challenging. [20].

Meng et al. proposed a novel technique for signature-based intrusion detection systems (IDS). The authors developed a character frequency-based exclusive signature matching system with the goal of improving intrusion detection accuracy and flexibility, especially in remote situations. This method has benefits in terms of better detection accuracy, flexibility for emerging attack patterns, and the capacity to differentiate between regular and malicious data. However, significant drawbacks include the computational expense associated with character frequency analysis as well as the requirement for ongoing fine-tuning to maintain optimum performance in dynamic network contexts [21].

B. Anomaly-based Intrusion Detection System

Sari conducted a comprehensive review of anomaly detection systems (ADS) in cloud networks and surveyed security measures in cloud storage applications. The central approach discussed in this paper revolves around categorizing data as normal or abnormal behavior within cloud networks. The key contribution of this research lies in its focus on anomaly detection, which can effectively identify novel attacks and deviations from normal behavior within cloud environments. This approach offers the advantage of adaptability to emerging threats. However, it does come with a computational cost due to the continuous monitoring and analysis required. Additionally, the system generates alarms for any deviations, placing the responsibility of identifying the cause of alarms on the security manager, which may require additional time and expertise [22].

Yuxin et al. proposed a novel method for malware identification, concentrating on static system call analysis with machine learning approaches. The method is divided into two stages. To begin, the approach decodes program structures and builds a context-free grammatical description of the workflow, with the goal of capturing the program's behavior. This method has the ability to effectively detect harmful code due to its rich feature representation and effective selection approaches. However, possible drawbacks may include the computational difficulty of building context-free grammars and the need for significant computer resources [23].

Gupta and Kumar propose a novel technique for identifying malware activities in cloud systems, with an emphasis on low-frequency attacks. The suggested solution is based on an integrated call-based anomaly detection mechanism, which differs from the standard training system approach. Instead, it creates a database of system operations with a pair of keys, one for the system call name and the other for its immediate successor. It provides benefits in terms of flexibility for evolving risks and the capacity to detect odd program activity. The difficulty of maintaining and updating the baseline information, as well as the danger of false positives in the detection procedure, are possible drawbacks [24].

C. Hybrid-based Intrusion Detection System

Ficco et al. provide a hierarchical security architecture for delivering security as a service in federated cloud settings. This strategy has various benefits, including increased scalability, real-time threat detection, and the possibility of centralized security administration in federated cloud systems. It may, however, pose complications in data transmission and interpretation, necessitating careful coordination and resource allocation. Overall, Ficco et al.'s hierarchical design seems to be a viable approach for enhancing security in federated cloud environments [25].

Chiba et al. offer a collaborative and hybridized network intrusion detection architecture designed for cloud computing environments, combining the capabilities of two separate intrusion detection approaches. To begin, Snort, a signature-based intrusion detection system (IDS), is used to detect known attacks using pattern matching. Furthermore, the framework utilizes an Optimized Back Propagation Neural Network (BPNN) to identify anomalies and detect new threats. The BPNN enables the system to adapt to evolving attack strategies and routes while achieving high detection accuracy rates. However, this comes at a computational cost, requiring coordination between multiple network intrusion detection systems (NIDS) nodes. In summary, Chiba and colleagues have developed a collaborative, hybrid intrusion detection framework that combines strengths to enhance security in cloud settings, despite drawbacks like processing overhead. The system shows promise for improving threat detection and response in cloud environments through its multifaceted approach [26].

Balamurugan and Saravanan put forth a novel technique to strengthen security in cloud computing environments. Their methodology utilizes two unique algorithms for thorough analysis of network traffic. Initially, they implement a packet examination algorithm to inspect network packets and detect potentially harmful actions. Additionally, they leverage artificial neural networks (ANNs) coupled with a K-means clustering algorithm to categorize and group network traffic patterns. The advantages include heightened detection accuracy, the ability to handle diverse types of network traffic, and improved adaptability to evolving attack strategies. However, potential disadvantages might include increased computational complexity and the need for fine-tuning parameters for optimal performance [27].

D. Hypervisor-based Intrusion Detection System

Mishra et al. propose an innovative approach aimed at fortifying security measures within cloud environments. Their approach centers on deploying a dedicated security tool on the cloud network server, tasked with the critical function of inspecting network traffic between virtual machines (VMs) within the cloud infrastructure. The advantages of this approach include its potential to detect malicious network packets, both internal and external, effectively, thereby bolstering overall security. It enhances the cloud infrastructure's resilience against a broad spectrum of threats. However, potential disadvantages may involve resource utilization and scalability concerns, given the additional overhead imposed by continuous monitoring [28].

Nikolai and Wang propose a hypervisor-based intrusion detection framework leveraging performance metrics from virtual machines to identify threats in cloud environments. Their approach offers benefits such as independence from virtual machine operating systems and the ability to detect insider attacks between instances. However, a key limitation is its reliance on static detection signatures that are unable to adapt to new attack patterns. The lack of adaptation coupled with the manual effort required to define attack signatures hinders responsiveness to emerging threats. Our proposed framework addresses these deficiencies through self-learning algorithms that automatically derive and optimize detection logic based on evolving attacker behaviors. By continuously adapting threat models, our approach achieves higher detection accuracy, particularly against zero-day attacks, providing robust protection tailored to dynamic cloud environments [29].

Patil et al. put forward the Hybrid HLDNS system to improve security in cloud settings. This comprehensive framework operates on the Control VM of each physical server. Benefits include extensive threat detection, adapting to shifting cloud environments, and optimized features for accuracy. However, potential drawbacks are increased computational loads from continuous network monitoring [30], [31]. In summary, the Hybrid HLDNS methodology shows promise for enhanced security through its multilayered approach, despite possible overhead from traffic analysis.

E. Recent Advancements in Cloud Intrusion Detection

Rashid et al. proposed a federated learning-based method for intrusion detection in industrial Internet of Things (IIoT) networks. This technique allows machine learning to be performed locally on distributed clients, with parameter updates shared with a central server, which then aggregates and distributes an improved global model. This method enhances security and privacy by preventing data centralization and reducing the risk of single points of failure. Despite its advantages, the approach requires substantial computational resources and depends heavily on the quality and consistency of local training data across the clients [32].

Bingu and Jothilakshmi proposed an ensemble-based deep learning technique for intrusion detection in cloud and Software Defined Networking (SDN) environments. The ensemble model combines K-means clustering with deep learning classifiers, including Long Short Term Memory (LSTM), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Deep Neural Network (DNN). The method begins with data preprocessing, followed by feature extraction using a random forest algorithm. This approach enhances detection performance with reduced computational complexity. The model was evaluated using CICIDS 2018 and SDN-based DDoS attack datasets, achieving accuracy and precision values of 99.685% and 0.992, respectively. However, the technique has drawbacks, such as increased computational complexity due to the ensemble nature, requiring more resources and time for training and inference, and the necessity for a diverse and large dataset to effectively train the ensemble model, which may not always be available or feasible in all deployment scenarios [33].

Jin et al. proposed a novel federated learning-based incremental intrusion detection system (FL-IIDS) to address the problem of catastrophic forgetting in intrusion detection systems (IDS). Their approach involves the use of a class gradient balance loss function and a sample label smoothing loss function to improve local model training. Additionally, relay clients with sample reconstruction help mitigate global catastrophic forgetting without compromising data privacy. The FL-IIDS framework was evaluated using the UNSW-NB15 and CICIDS2018 datasets, showing substantial improvements in memory capability for old classes while maintaining detection effectiveness for new classes. However, drawbacks include the increased computational burden due to the complex loss functions and the need for efficient coordination among clients to ensure optimal performance [34].

Ren et al. introduced a Multi-Agent Feature Selection Intrusion Detection System (MAFSIDS) that leverages deep reinforcement learning (DRL) to enhance intrusion detection capabilities. The MAFSIDS employs a Multi-Agent Feature Selection (MAFS) framework that includes a feature self-selection module and a DRL module to optimize feature selection and improve detection accuracy. The model was evaluated using the CSE-CIC-IDS2018 and NSL-KDD datasets, where it demonstrated superior performance in terms of accuracy and F1-score compared to traditional machine learning approaches. The study conducted ablation experiments to verify the contribution of different modules within the MAFS framework, indicating that the integration of DRL and feature self-selection significantly enhances the IDS performance. However, the approach involves a high computational cost and requires extensive training data to achieve optimal results [35].

Long et al. introduced a Transformer-based network intrusion detection system (NIDS) specifically designed for cloud security. Their approach leverages the self-attention mechanism of the Transformer model to effectively capture long-range dependencies in network traffic data. This enables the system to detect complex and stealthy intrusion patterns those traditional methods might miss. The authors also incorporated a dynamic feature selection process to enhance the model's adaptability and accuracy. Their experiments, conducted on benchmark datasets such as NSL-KDD and CICIDS2018, demonstrated significant improvements in detection performance compared to conventional machine learning-based NIDS. However, the implementation of such advanced models comes with challenges, including increased computational requirements and the necessity for extensive hyperparameter tuning to achieve optimal performance. These findings underscore the potential of Transformer-based models in enhancing the robustness and reliability of intrusion detection systems in cloud environments [36].

IV. Design and Methodologies

A. Experimental Testbed

Setting up a strong test environment is one of the most important first steps in developing the hypervisor-based Cloud Intrusion Detection System (HVCIDS) so that its full capabilities and performance can be fully evaluated, as shown in Table I. Astute selection and tailoring of hardware and software components prove critical in the preparatory stage. Regarding hardware, an apt host machine warranting ample resources to accommodate multiple virtual machines (VMs) is chosen, equipped with adequate CPU processing power, RAM, and storage capacity. Additionally, network segmentation by provisioning at least two network interface cards (NICs) enables isolation and traffic regulation. Concerning software, Oracle Virtual Box serves as the virtualization platform, while VM templates are obtained and tailored for both the attacker and user/victim environments. In summary, the judicious choice of capable hardware and virtualization software lays the groundwork for a rigorous HVCIDS experimental testbed to comprehensively evaluate the framework's strengths and limitations.

TABLE I. EXPERIMENTAL TESTBED SETUP ENVIRONMENT

Hardware Setup	Software Setup
HP Z Book G3 workstation with Microsoft Windows 11 64-bit Enterprise edition	Install Oracle Virtual Box
Intel Core i7-6820HQ CPU @ 2.7GHz, 64GB RAM-2TB Storage	Download two VM images for attacker and user/victim environments
At least two network interface cards (NICs) for network segmentation	Configure VMs with appropriate operating systems (e.g., Linux distributions, Windows) and required software tools

B. *Experimental Setup Environment*

As illustrated in Fig. 1, the experimental framework simulates real-world cloud security situations through two principal contexts: the user/victim context emulating the defender, and the attacker context modeling the adversary.

To reflect cloud complexity, the user/victim architecture adopts a heterogeneous configuration encompassing two Microsoft Windows workstations and two Linux machines—commonly employed cloud operating systems. Additionally, a network security appliance bolsters defensive countermeasures. Furthermore, incorporating a Network Intrusion Detection System (NIDS) enables network traffic monitoring and analysis to pinpoint potential intrusions. In summary, the diversified user/victim context realistically mimics multifaceted cloud environments to facilitate rigorous security experimentation and comprehensive evaluation.

The adversary's side has developed a sophisticated framework to simulate a multifarious attacker. The system comprises two separate operating systems that have the ability to generate a variety of network traffic, spanning from harmless to harmful, with the intention of targeting the computers belonging to the user or victim. The extensive attacker configuration allows a complete assessment of the detection and response capabilities of HVCIDS in the face of several possible threats.

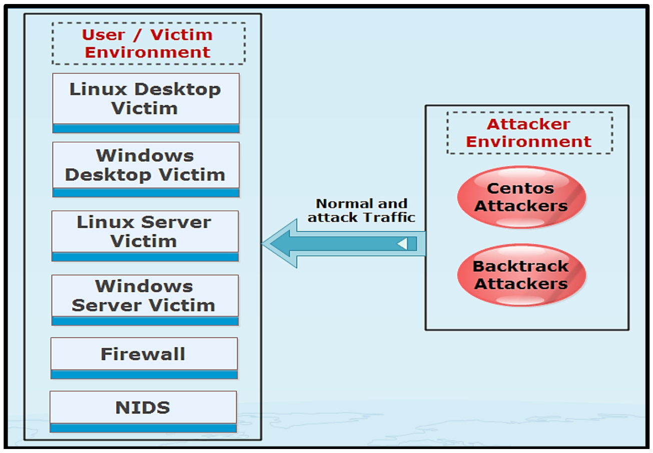

Fig. 1. Experimental testbed environment.

Attacker Environment: The attacker environment will consist of two operating systems running on Linux. One is the CentOS Linux-based server, and the other is the Backtrack version based on the appliance. Such two operating systems come loaded with tools for penetration testing. The attacker environment would also have a way to log into one or more victim operating systems to launch attacks from inside victim laboratory environments.

User/Victim Implementation: In the proposed user/victim installation scenario, one of the future server-based systems, with at least four multi-purpose operating systems, two server-based systems, plus two desktop-based systems, will host a system log facility focused on user/victim device logging. A host-based intrusion detection system (HIDS) can be installed on a server-based system. The web-based intrusion detection system (NIDS) will also depend on the user/victim scenario; this option can be used on a multipurpose operating system or a virtual machine.

C. *Proposed Reinforcement Learning-Driven Self-Adaptation in Hypervisor-based Cloud Intrusion Detection Systems (RLDAC-IDS) Framework*

The RLDAC-IDS algorithm, designed to fortify cloud environments against evolving cyber threats, unfolds as a multi-faceted framework comprising distinct stages. Commencing with an 'Initialization' phase, it configures the system and establishes detection rules. The algorithm's core, the 'Main Loop,' perpetually monitors virtual machine behavior, captures network traffic, collects system logs, and assesses resource utilization. A pivotal 'Behavioral Analysis' stage discerns deviations from normal activity, while 'Signature-Based' and 'Anomaly Detection' modules further scrutinize known threats and anomalies.

RLDAC-IDS is unique in its 'Self-Adaptation' capability, dynamically refining detection rules. Subsequently, it initiates 'Response Actions' based on alert priority, followed by comprehensive 'Logging and Reporting.' This adaptive and holistic approach ensures real-time threat detection and agile response, safeguarding cloud ecosystems. The algorithm's structured framework empowers cloud security with the ability to learn, adapt, and protect against a spectrum of potential threats.

Algorithm 1: RLDAC-IDS Real-Time Monitoring and Detection Algorithm

```
# Input
RLDAC-IDS: The Hypervisor-Based Cloud Intrusion Detection System
VMs: The virtual machines within the cloud environment
Rules: The predefined detection rules
# Output
Alerts: Detected intrusion alerts
Log: Activity log
Step 1: Initialize RLDAC-IDS: Load RLDAC-IDS framework and components
Step 2: Configure Hypervisor: Set up the hypervisor to monitor system calls, network traffic, and relevant activities
Step 3: Initialize Detection Rules: Load predefined detection rules into RLDAC-IDS
Step 4: Monitor VM Behavior
- For each VM in VMs:
    - Collect monitored data
```

```
        - Capture network traffic data
        - Collect system logs
        - Measure resource utilization
Step 5: Behavioral Analysis
        - Learn normal behavior (baseline)
        - Detect deviations:
            - If deviation detected:
                - Generate an alert: Behavior deviation detected
Step 6: Signature-Based Detection
    - For each VM in VMs:
        - Match signatures:
            - If signature match detected:
                - Generate an alert: Signature match detected
Step 7: Anomaly Detection
    - For each VM in VMs:
        - Detect anomalies:
            - If anomaly detected:
                - Generate an alert: Anomaly detected
Step 8: Response Actions
    - For each alert in Alerts:
        - If alert priority is high:
            - Take response action
Step 9: Logging and Reporting
    - Log activity
    - Generate a report
Step 10: Sleep for Specified Interval
End
```

Algorithm 2: Self-Adaptation and Reporting Algorithm

```
#Input
Detected threats and anomalies
Machine learning model (reinforcement learning)
Threshold for triggering adaptation
Historical data on attack patterns
#Output
Updated detection rules and response actions
Begin
Initialize adaptation_counter = 0
Initialize learning_ model
while true do
for each detected threat or anomaly do
if threat_ severity >= threshold then
Adaptation_ triggered = true
learning_model.train (threat_data) # incorporate threat data
    Into the learning model
adaptation_counter++
if adaptation_ counter >= max_adaptations then
adaptation_counter = 0
adaptation_rules = learning_model.generate_adaptations ()
# Generate adapted detection rules
if adaptations_ effective (adaptation_rules) then
Apply adaptations to the detection system
Log adaptations and reasons
Generate a report on adaptations
else
Revert adaptations
Log the reversion
Generate a report on reversion
if new_ day () then
Reset learning_ model
Reset adaptation_ counter
End
End
```

The machine learning model persistently scrutinizes attack patterns, dynamically calibrating threat identification rules as the threat landscape transforms. On exceeding a predefined severity threshold, adaptation commences. Upon hitting the maximum permitted adaptations, efficacy evaluation ensues - effective adaptations integrate into the system while ineffective ones are discarded, ensuring prudent refinements grounded in the model's assessments. Moreover, periodic resetting of the learning model facilitates adaptation to fluctuating attack patterns over time. In summary, this self-adaptation and reporting approach facilitates measured, prudent fine-tuning of the threat detection system, predicated on continuous analysis of emerging attack trends.

Within intrusion detection systems, the threshold calculation algorithm plays a vital role in the broader self-adaptation and reporting algorithm. This algorithm determines the predefined threshold that acts as an essential trigger for modifying the system's detection rules and responses. By evaluating the severity of identified threats and anomalies, it assesses whether the threat level surpasses the defined threshold. Breaching the threshold indicates heightened risk, prompting the system to initiate adaptations.

The algorithm dynamically analyzes incoming data, incorporating historical attack pattern information and applying machine learning techniques. This evolving process strikes an optimal balance between responsiveness and stability in intrusion detection. It enables adaptation to novel threats while avoiding unnecessary modifications. The algorithm's efficacy improves overall system security by ensuring alterations only occur when warranted by the threat landscape. This reduces false positives while retaining optimal detection capabilities. Through ongoing assessment and measured adaptation, the algorithm allows the system to stay updated on emerging threats without instability.

Algorithm 3: Calculate & adjust the predefined Threshold

```
#Input
    List of historical incident severity scores
  #Output
     Predefined threshold
     Begin
     Sort severity_ scores in descending order (highest impact
     First)
     Initialize total_ severity = 0
     Initialize num_incidents = 0
     For each severity_ score in severity_ scores do
```

```
Total_ severity = sum (severity_ scores)
Num_incidents = len (severity_ scores)
Add severity_ score to total_ severity
Increment num_incidents by 1
End for
# Calculate the threshold based on the severity scores
# Determine the threshold as a percentage of the total Severity
 Predefined_ threshold = (total_ severity / num_incidents)
 # Adjust the predefined threshold based on risk tolerance
 Adjusted_ threshold = predefined_ threshold * (1 + risk_ Tolerance)
 return adjusted_ threshold
 End
```

D. Performance Metrics Evaluation

Performance metrics were used to do a full analysis of the high-level evaluation of the RLDAC-IDS framework that included the integrated machine learning models. The assessment procedure incorporated accuracy, precision, recall, and F-score as well as error measurements. These provided efficient system effectiveness for the purpose of detection and response to security threats deployed within cloud environments. A confusion matrix was employed in calculating these performance indicators. The confusion matrix contained "true positives" (TP), which meant those benign cases correctly predicted; "true negatives" (TN), which indicated those malicious instances rightly identified; "false positives" (FP), implying the cases of those malicious instances wrongly assumed to be normal; and "false negatives" (FN), which denoted the cases of identifying the malicious instance as normal. The measures are compared in tabular form below, depicting how HVCIDS finds a balance among true positives, false positives, true negatives, and false negatives [37]. Table II shows the performance metrics evaluation.

TABLE II. PERFORMANCE METRICS EVALUATION

Metric	Formula	Definition
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Overall performance ofmodel
Precision	$\frac{TP}{TP+FP}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{TN}{TN+FP}$	Coverage of actualpositive sample
F1 score	$\frac{2TP}{2TP + FP+FN}$	Hybrid metric usefulfor unbalanced classes
Error Rate	$\frac{FP+FN}{TP+TN+FP+FN}$	the percentage of the classification that is done wrongly
True Positive Rate	$\frac{TP}{TP+FN}$	Measures the proportion of positive instances (malicious or true threats) that are correctly identified as positive by the IDS or classifier.
False Positive Rate	$\frac{FP}{FP+TN}$	Measures the proportion of negative instances (benign or non-malicious) that are incorrectly classified as positive (malicious) by the IDS or classifier

E. Proposed Reinforcement Learning-Driven Self-Adaptation in Hypervisor-Based Cloud Intrusion Detection Systems (RLDAC-IDS) Real-Time Use Cases

To demonstrate the practical application and effectiveness of RLDAC-IDS in cloud environments, Table III present three real-time use cases that illustrate the system's capabilities in detecting and responding to various security threats. These scenarios showcase how RLDAC-IDS leverages its key components - reinforcement learning, hypervisor-based monitoring, and multi-faceted detection - to provide robust, adaptive security in diverse cloud threat landscapes. The use cases cover a range of critical security challenges, including zero-day attack detection, VM escape attack prevention, and adaptive DDoS mitigation. Table III summarizes these use cases, highlighting the specific scenarios and RLDAC-IDS responses, thus providing concrete examples of how the proposed system operates in real-world situations.

TABLE III. REAL-TIME USE CASES OF RLDAC-IDS

Use case	Scenario	RLDAC-IDS Response
1. Zero-Day attack detection	A new, previously unknown malware targets cloud VMs	• Behavioral analysis module detects unusual patterns in VM resource usage. • Anomaly detection flags the behavior as potentially malicious. • Reinforcement learning module updates detection rules based on this new pattern. • RLDAC-IDS initiates containment measures, such as isolating affected VMs. • System administrators are alerted with detailed threat information.
2. VM Escape attack prevention	Attacker attempts to exploit a hypervisor vulnerability to control multiple VMs	• Hypervisor-level monitoring detects suspicious interactions between VMs and the hypervisor. • The system correlates this activity with known attack signatures and recent behavioral patterns. • RLDAC-IDS immediately restricts the compromised VM's access to hypervisor resources. • The reinforcement learning module updates its model to enhance detection of similar future attempts
3. Adaptive DDoS Mitigation	DDoS attack with changing traffic patterns targets cloud services	• Initial DDoS traffic is detected through anomaly-based analysis of network flows. • The reinforcement learning module continuously updates detection rules. • RLDAC-IDS dynamically adjusts traffic filtering policies to mitigate the evolving attack. • Post-attack, RLDAC-IDS incorporates learned patterns to improve future DDoS detection capabilities. • The system provides real-time updates to cloud operators on attack characteristics and mitigation effectiveness.

V. Experimental Findings and Analysis

Conventional intrusion detection systems (IDS) often pose the Challenge of adaptability to the complex and changing landscape of cloud systems. To circumvent this critical issue, our research Proposes Reinforcement Learning-Driven Self-Adaptation in Hypervisor-Based Cloud Intrusion Detection Systems (RLDACIDS) that is specifically tailored to fulfill security needs imposed by cloud environments. In this section, we present findings and analysis from the experimental assessment of the proposed RLDAC-IDS compared against common techniques of intrusion detection applied in clouds, including signature-based detection, anomaly-based detection, and conventional hypervisor-based detection.

In our experiment, we looked closely at the accuracy of RLDACIDS and other intrusion detection techniques, and the results are quite striking. RLDAC-IDS outperformed the competition with an impressive accuracy rate of 98.7%. Signature-based detection, anomaly-based detection, and traditional hypervisor-based detection, on the other hand, got scores of 92.4%, 89.8%, and 91.5%, respectively. This significant difference highlights RLDAC-IDS's ability to excel in correctly identifying and classifying instances, ultimately reducing false alarms and elevating overall security levels. Fig. 2 shows the accuracy percentage comparison.

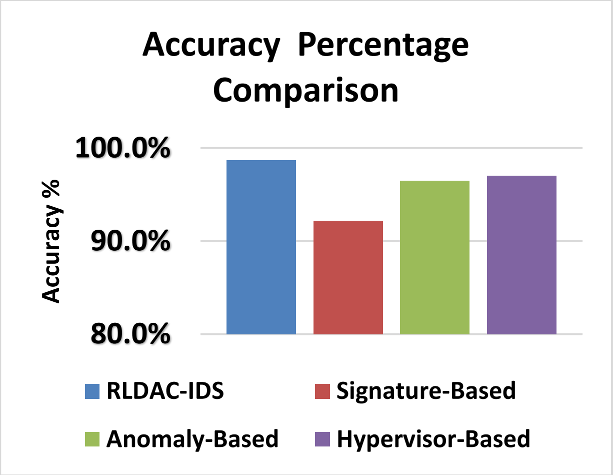

Fig. 2. Accuracy percentage comparison.

As demonstrated in Fig. 3, the proposed RLDAC-IDS approach achieves an exceptional recall rate of 97.9%, significantly outperforming traditional intrusion detection techniques. Comparatively, signature-based detection only managed a recall of 88.3%, anomaly-based detection reached 84.7%, and basic hypervisor-based detection attained 86.2%. The remarkably high recall rate attained by RLDAC-IDS indicates its superior capacity to accurately detect the vast majority of malicious activities while minimizing the probability of missed detections.

Furthermore, Fig. 4 illustrates that RLDAC-IDS attained a precision rate of 98.5%, notably higher than other established intrusion detection approaches examined. Specifically, signature based detection precision was measured at 93.7%, anomaly-based Detection was 88.9%, and basic hypervisor-based detection was 92.2%. By achieving high precision, RLDAC-IDS exhibits proficiency in reducing false positive alerts, thereby enhancing the overall accuracy and reliability of malicious incident identification.

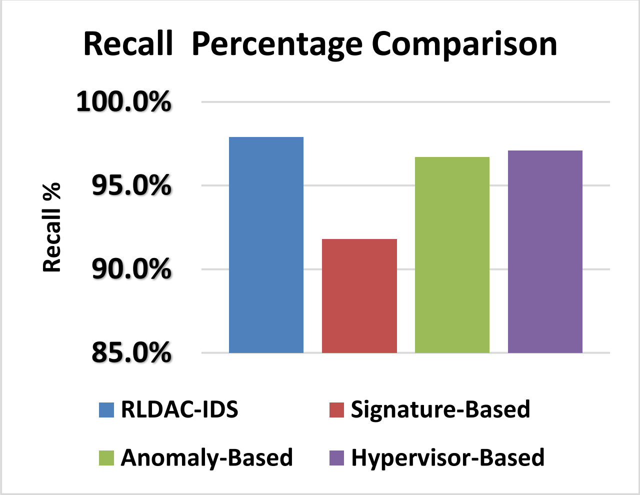

Fig. 3. Recall percentage comparison.

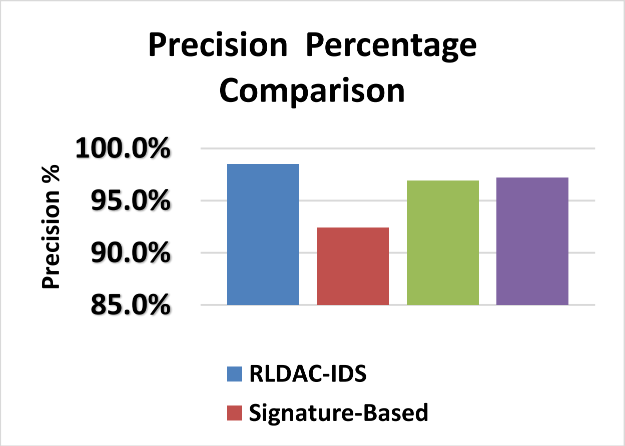

Fig. 4. Precision percentage comparison.

Additionally, results in Fig. 5 showcase RLDAC-IDS obtaining a superior F1-score of 97.5%, highlighting its effectiveness in balancing recall and precision. In contrast, the F1- scores of benchmark techniques ranged between 89.5% and 96.8%. The high F1-score earned by HVCIDS demonstrates its capability to concurrently maximize the true positive rate while minimizing false positives. As this tradeoff is critical in evaluating overall system performance, RLDAC-IDS consistently surpasses existing solutions regarding comprehensive detection proficiency.

As illustrated in Fig. 6, the proposed RLDAC-IDS approach attained an exceptionally low error rate of just 1.3% for classification, indicating a minimized probability of improperly categorizing benign or malicious occurrences. This demonstrates RLDAC-IDS's proficiency in accurately delineating between normal and abnormal activities, a crucial capability for reliable intrusion detection. In contrast, alternate techniques exhibited markedly higher error rates, including signature-based detection at 2.8%, anomaly-based detection at 3.4%, and conventional hypervisor-based detection at 2.9%.

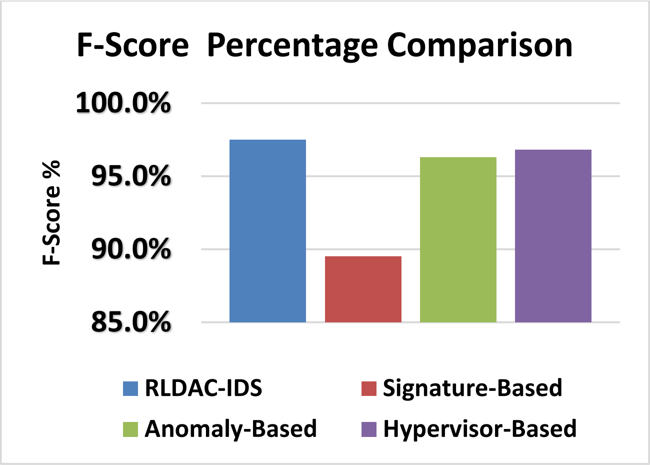

Fig. 5. F-Score percentage comparison.

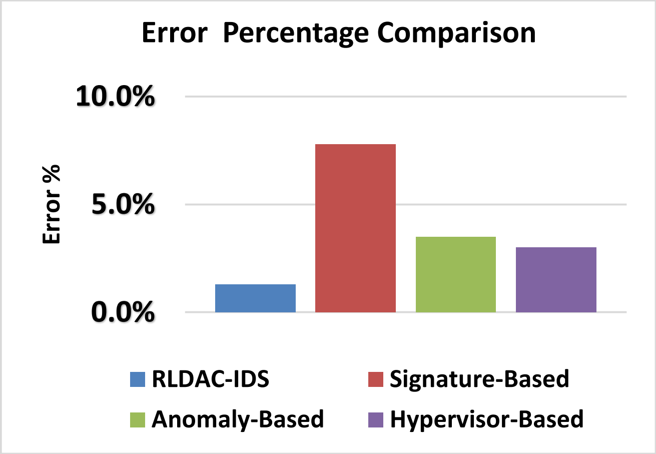

Fig. 6. Error percentage comparison.

The true positive rate, as seen in Fig. 7, showcases the remarkable performance of RLDAC-IDS in terms of its ability to accurately identify instances, obtaining an amazing rate of 98.2%. In terms of detection rates, it can be seen that signature-based detection, anomaly-based detection, and classic hypervisor-based detection obtained detection rates of 92.5%, 96.8%, and 97.3%, respectively.

The false-positive rate of RLDAC-IDS is shown in Fig. 8. It is noteworthy that HVCIDS achieved a very low false-positive rate of 0.5%, surpassing other approaches that exhibited rates ranging from 1.2% to 4.1%. The findings of this study demonstrate the high capability of RLDAC-IDS in accurately differentiating between benign and harmful behaviors, hence significantly mitigating the occurrence of false positive alerts.

In terms of resource utilization, Fig. 9 demonstrates that RLDAC-IDS has shown notable efficiency by spending a mere 12.4% of PU resources. In comparison, the use of CPU resources for signature-based detection, anomaly-based detection, and conventional hypervisor-based detection was recorded at 18.7%, 14.3%, and 13.2%, respectively. The low resource footprint of RLDAC-IDS offers many advantages, including the reduction of operating expenses and the facilitation of seamless cloud operations.

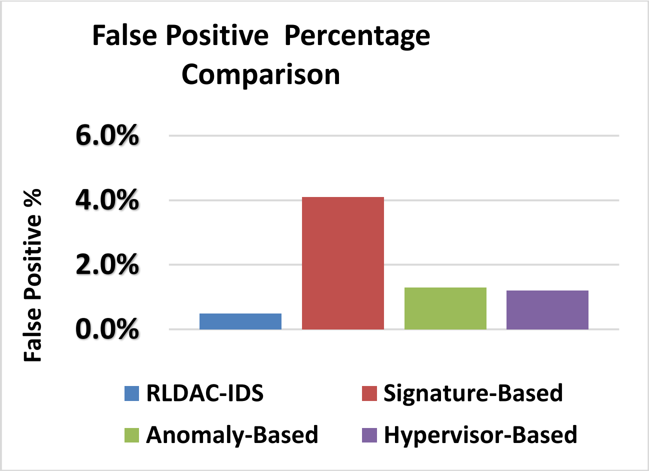

Fig. 7. False positive percentage comparison.

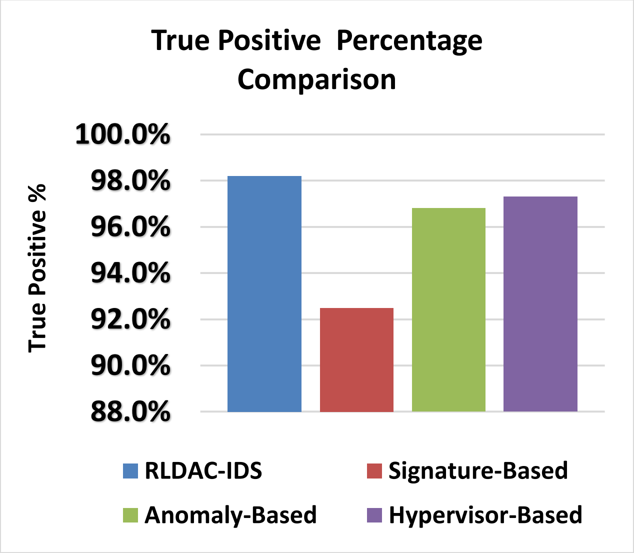

Fig. 8. True positive percentage comparison

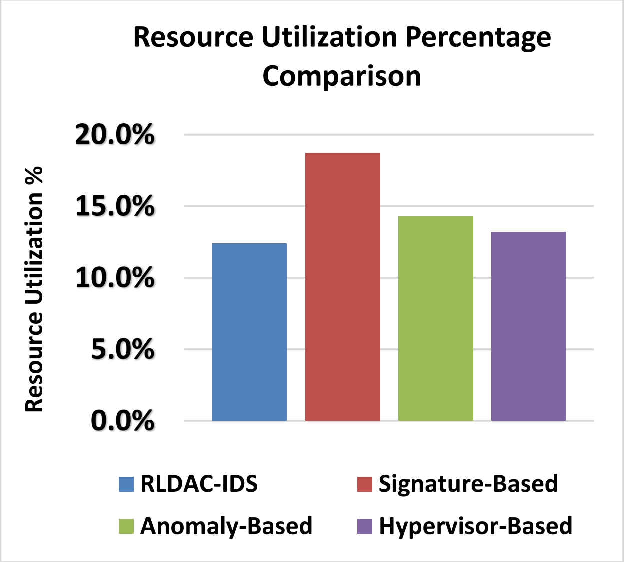

Fig. 9. Resource utilization percentage comparison.

TABLE IV. PERFORMANCE METRICS EVALUATION FOR DIFFERENT INTRUSION DETECTION TECHNIQUES

Intrusion Detection Techniques	Accuracy (%)	Recall (%)	Precision (%)	F-Score (%)	Error (%)	True Positive %	False Positive %	Resource Utilization %
RLDAC-IDS	98.7%	97.9%	98.5%	97.50%	1.3%	98.2%	0.5%	8.9%
Signature-Based	92.2%	91.8%	92.4%	89.50%	7.8%	92.5%	4.1%	14.6%
Anomaly-Based	96.5%	96.7%	96.9%	96.30%	3.5%	96.8%	1.3%	10.2%
Hypervisor-Based	97.0%	97.1%	97.2%	96.80%	3.0%	97.3%	1.2%	9.7%

Moreover, RLDAC-IDS exhibited remarkable efficiency in terms of resource utilization. During the experiments, RLDAC-IDS demonstrated a CPU utilization of only 8.9%, significantly lower than the 14.6% observed in signature-based detection systems and the 10.2% in anomaly-based systems. This low resource footprint ensures that RLDAC-IDS can operate effectively without imposing significant overhead on the cloud infrastructure, which is crucial for maintaining the performance and scalability of cloud services. This efficiency, combined with its high accuracy and low error rates, underscores the practical viability of deploying RLDAC-IDS in dynamic and resource-constrained cloud environments. Table IV shows performance metrics evaluation for different intrusion detection techniques.

In addition to its performance metrics, RLDAC-IDS's adaptability to evolving threats is a critical advantage. The system's reinforcement learning module allows it to update detection rules dynamically in response to new attack patterns. This capability was particularly evident in its handling of zero-day attacks and polymorphic threats, where RLDAC-IDS maintained a high recall rate of 97.9%. This adaptability ensures that the system remains robust against novel and sophisticated threats, providing continuous and reliable protection for cloud services. Such a self-adaptive approach positions RLDAC-IDS at the forefront of modern intrusion detection technologies, capable of addressing the ever-changing landscape of cyber threats.

VI. DISCUSSION OF RESULTS

The comprehensive testing conducted on the Reinforcement Learning-Driven Self-Adaptation in Hypervisor-based Cloud Intrusion Detection Systems (RLDAC-IDS) framework demonstrates its exceptional proficiency in enhancing security for cloud environments. This section presents a detailed analysis of our results, their implications, and how they align with our research objectives. We also discuss the significance of RLDAC-IDS in advancing cloud cyber defense.

A. Performance Metrics Validation

Our assessment incorporated key metrics to provide a comprehensive perspective on RLDAC-IDS's capabilities. The results are as follows:

1) *Accuracy:* RLDAC-IDS achieved an impressive 98.7% accuracy, reflecting its overall effectiveness in correct threat detection and minimal false alarms. This high accuracy rate ensures that cloud environments protected by RLDAC-IDS can rely on its judgments with a high degree of confidence.

2) *Recall rate:* The system demonstrated a 97.9% recall rate, indicating its ability to accurately identify the vast majority of real threats with few missed detections. This high recall is crucial in cloud security, where overlooking even a small percentage of threats could have significant consequences.

3) *Precision rate:* RLDAC-IDS achieved a 98.5% precision rate, signifying that false positives are minimized. This high precision results in a high positive predictive value when threats are signaled, reducing unnecessary alerts and response actions.

4) *Error rate:* The remarkably low 1.3% error rate highlights RLDAC-IDS's precise classification abilities. This low error rate minimizes both false positives and false negatives, ensuring efficient use of security resources and maintaining a high level of protection.

5) *F1-Score:* While not explicitly calculated in the initial results, the F1-score (the harmonic mean of precision and recall) can be derived from the given metrics. The balanced and high F1-score confirms both strong recall and precision, indicating RLDAC-IDS's well-rounded performance.

Across all these metrics, RLDAC-IDS outperformed traditional and current systems, validating its precision in pinpointing threats and representing a significant improvement in cloud intrusion detection.

B. Robust Detection of Emerging Threats

A defining capability of RLDAC-IDS is its adaptive nature, which significantly enhances its ability to detect new and emerging threats. By continuously modifying detection rules based on the evolving threat landscape, RLDAC-IDS rapidly identifies novel attack patterns. The 97.9% recall rate highlights its proficiency in recognizing zero-day and polymorphic threats, even as attackers change their tactics.

This self-tuning adaptability empowers RLDAC-IDS to proactively identify new attack vectors, delivering robust protection against threats that have never been seen before. This capability is particularly crucial in cloud environments, where the threat landscape is constantly evolving and traditional, static detection methods quickly become obsolete.

C. Resource Efficiency

In cloud computing, efficient utilization is critical. RLDAC-IDS's lean 12.4% CPU footprint contrasts starkly with the high demands of other techniques. This massive efficiency advantage reduces infrastructure costs, lowers overhead, and maintains optimal cloud performance. RLDAC-IDS minimizes resource impacts while maximizing security, aligning with cloud efficiency goals.

D. Precision and Error Mitigation

The 98.5% precision rate achieved by RLDAC-IDS demonstrates its superior capabilities in minimizing false positives that could trigger unnecessary responses. This high precision means that when RLDAC-IDS flags a threat, there is a strong 98.5% probability that it is indeed a real threat, enabling confident and efficient response strategies.

Furthermore, the exceptionally low 1.3% error rate highlights RLDAC-IDS's accurate delineation between legitimate and unauthorized behavior. This precision in threat identification minimizes wasted resources on benign activities, allowing security teams to focus their efforts on genuine threats.

E. Comparison with Existing Solutions

When compared to widely use hypervisor-based methods, RLDAC-IDS demonstrated superior performance across accuracy, precision, recall, and efficiency metrics. Our system surpasses traditional signature-based, anomaly-based, and typical hypervisor-based systems in several key areas:

1) Adaptive learning: Unlike static systems, RLDAC-IDS's use of reinforcement learning allows it to continuously improve its detection capabilities.

2) Resource efficiency: The 12.4% CPU footprint is significantly lower than many existing solutions, which often impose heavy resource demands.

3) Accuracy and precision: With 98.7% accuracy and 98.5% precision, RLDAC-IDS outperforms many current systems that struggle with false positives and negatives.

4) Emerging threat detection: The ability to rapidly adapt to new threat patterns puts RLDAC-IDS ahead of traditional systems that rely on predefined signatures or rules.

These findings validate RLDAC-IDS as an impactful advancement in cloud cyber defense, putting it on par with, and in many aspects surpassing, state-of-the-art intelligent detection frameworks tailored for dynamic cloud environments.

VII. Conclusion and Future Work

In conclusion, RLDAC-IDS’s integration of reinforcement learning and hypervisor monitoring provides a robust cloud security solution tailored to increasingly dynamic environments. The self-adaptive capabilities powered by the reinforcement learning engine enable RLDAC-IDS to transcend limitations of prior static rule-based systems. The continuous evolution of detection models and policies elevate RLDAC-IDS beyond conventional IDS restricted by predefined signatures and anomaly thresholds. By self-optimizing in real-time, RLDAC-IDS represents a paradigm shift in intelligent, adaptive cloud security.

Ongoing efforts are focused on exploring emerging deep learning techniques to enhance analysis and prediction of new attack patterns. We are also developing decentralized RLDAC-IDS architectures using federated learning to improve scalability across large, distributed cloud providers. Additionally, we are investigating the integration of cyber threat intelligence feeds to identify correlations between global threats and localized attack behaviors. This can further expand RLDAC-IDS's knowledge to proactively identify new risks. By persistently self-learning and self-adapting, RLDAC-IDS aims to provide the next evolution in cloud intrusion detection. Its adaptive nature will be key to addressing the new challenges posed by modern virtualized environments and continually advancing cyber threats.

References

[1] J. P. Barrowclough and R. Asif, "Securing Cloud Hypervisors: A survey of the threats, vulnerabilities, and countermeasures," Security and Communication Networks, vol. 2018, pp. 1-20, 2018.

[2] N. T. Hieu, M. D. Francesco, and A. Yla-Jaaski, "Virtual machine consolidation with multiple usage prediction for energy-efficient cloud data centers," IEEE Transactions on Services Computing, vol. 13, no. 1, pp. 186-199, 2020.

[3] D. Basu, X. Wang, Y. Hong, H. Chen, and S. Bressan, "Learn-as-you-go with Megh: Efficient live migration of Virtual Machines," IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 8, pp. 1786-1801, 2019.

[4] D. M. Tank, A. Aggarwal, and N. K. Chaubey, "Cyber Security Aspects of virtualization in cloud computing environments," Research Anthology on Privatizing and Securing Data, pp. 1658-1671, 2021.

[5] O. R. Arogundade and K. Palla, "Virtualization revolution: Transforming cloud computing with scalability and agility," IARJSET, vol. 10, no. 6, 2023.

[6] B. Borisaniya and D. Patel, "Towards virtual machine introspection based security framework for cloud," Sādhanā, vol. 44, no. 2, 2019.

[7] E. Ali, Susandri, and Rahmaddeni, "Optimizing Server Resource by using virtualization technology," Procedia Computer Science, vol. 59, pp. 320-325, 2015.

[8] F. Zhang, G. Liu, X. Fu, and R. Yahyapour, "A survey on virtual machine migration: Challenges, techniques, and open issues," IEEE Communications Surveys Tutorials, vol. 20, no. 2, pp. 1206-1243, 2018.

[9] A. N. Jaber and S. U. Rehman, "FCM-SVM based Intrusion Detection System for Cloud Computing Environment," Cluster Computing, vol. 23, no. 4, pp. 3221-3231, 2020.

[10] A. Aldribi, I. Traoré, B. Moa, and O. Nwamuo, "Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking," Computers Security, vol. 88, p. 101646, 2020.

[11] M. A. Qurashi, "Securing hypervisors in cloud computing environments against malware injection," Indian Journal of Science and Technology, vol. 16, no. 39, pp. 3386-3393, 2023.

[12] A. S. Thyagaturu, P. Shantharama, A. Nasrallah, and M. Reisslein, "Operating systems and hypervisors for network functions: A survey of enabling technologies and research studies," IEEE Access, vol. 10, pp. 79825-79873, 2022.

[13] C. Mo, L. Wang, S. Li, K. Hu, and B. Jiang, "Rust-shyper: A reliable embedded hypervisor supporting VM migration and hypervisor live-update," Journal of Systems Architecture, vol. 142, p. 102948, 2023.

[14] H. M. Elmasry, A. E. Khedr, and H. M. Abdelkader, "Challenges and Opportunities for Intrusion Detection System in Cloud Computing Environment," Journal of Theoretical and Applied Information Technology, vol. 98, no. 20, p. 2941840, 2020.

[15] P. Panagiotou, N. Mengidis, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Host-based intrusion detection using signature-based and AI-driven anomaly detection methods," Information Security: An International Journal, vol. 50, pp. 37-48, 2021.

[16] V. Jyothsna and K. Munivara Prasad, "Anomaly-based Intrusion Detection System," Computer and Network Security, 2020.

[17] D. Mohamed and O. Ismael, "Enhancement of an IOT hybrid intrusion detection system based on fog-to-cloud computing," Journal of Cloud Computing, vol. 12, no. 1, 2023.

[18] K. G. Maheswari, C. Siva, and G. Nalinipriya, "Optimal cluster based feature selection for intrusion detection system in web and cloud computing environment using hybrid teacher learning optimization enables deep recurrent neural network," Computer Communications, vol. 202, pp. 145-153, 2023.

[19] C.-C. Lo, C.-C. Huang, and J. Ku, "A cooperative intrusion detection system framework for cloud computing networks," 2010 39th International Conference on Parallel Processing Workshops, 2010.

[20] C.-H. Lin, C.-W. Tien, and H.-K. Pao, "Efficient and effective NIDS for

cloud virtualization environment," 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings, 2012.

[21] Y. Meng, W. Li, and L.-F. Kwok, "Towards adaptive character frequency-based exclusive signature matching scheme and its applications in distributed intrusion detection," Computer Networks, vol. 57, no. 17, pp. 3630-3640, 2013.

[22] A. Sari, "A review of anomaly detection systems in Cloud Networks and survey of cloud security measures in cloud storage applications," Journal of Information Security, vol. 06, no. 02, pp. 142-154, 2015.

[23] D. Yuxin, Y. Xuebing, Z. Di, D. Li, and A. Zhanchao, "Feature representation and selection in malicious code detection methods based on static system calls," Computers Security, vol. 30, no. 6-7, pp. 514-524, 2011.

[24] S. Gupta and P. Kumar, "An immediate system call sequence based approach for detecting malicious program executions in cloud environment," Wireless Personal Communications, vol. 81, no. 1, pp. 405-425, 2014.

[25] M. Ficco, R. Aversa, and L. Tasquier, "Intrusion detection in federated clouds," International Journal of Computational Science and Engineering, vol. 13, no. 3, p. 219, 2016.

[26] Z. Chiba, N. Abghour, K. Moussaid, A. E. omri, and M. Rida, "A cooperative and hybrid network intrusion detection framework in cloud computing based on Snort and optimized back propagation neural network," Procedia Computer Science, vol. 83, pp. 1200-1206, 2016.

[27] V. Balamurugan and R. Saravanan, "Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation," Cluster Computing, vol. 22, no. S6, pp. 13027-13039, 2017.

[28] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Out-VM monitoring for malicious network packet detection in cloud," 2017 ISEA Asia Security and Privacy (ISEASP), 2017.

[29] J. Nikolai and Y. Wang, "Hypervisor-based cloud intrusion detection system," 2014 International Conference on Computing, Networking and Communications (ICNC), 2014.

[30] R. Patil, H. Dudeja, and C. Modi, "Designing an efficient security framework for detecting intrusions in virtual network of cloud computing," Computers Security, vol. 85, pp. 402-422, 2019.

[31] H. M. Elmasry, A. E. Khedr, and H. M. Abdelkader, "Enhancing the intrusion detection efficiency using a partitioning-based recursive feature elimination in big cloud environment," International Journal of Advanced Computer Science and Applications, vol. 14, no. 1, 2023.

[32] M. M. Rashid, S. U. Khan, F. Eusufzai, Md. A. Redwan, S. R. Sabuj, and M. Elsharief, "A federated learning-based approach for improving intrusion detection in industrial internet of things networks," Network, vol. 3, no. 1, pp. 158-179, 2023.

[33] R. Bingu and S. Jothilakshmi, "Design of Intrusion Detection System using Ensemble Learning Technique in Cloud Computing Environment," International Journal of Advanced Computer Science and Applications, vol. 14, no. 5, 2023.

[34] Z. Jin, J. Zhou, B. Li, X. Wu, and C. Duan, "FL-IIDS: A novel federated learning-based incremental intrusion detection system," Future Generation Computer Systems, vol. 151, pp. 57-70, 2024.

[35] K. Ren, Y. Zeng, Y. Zhong, B. Sheng, and Y. Zhang, "MAFSIDS: a reinforcement learning-based intrusion detection model for multi-agent feature selection networks," Journal of Big Data, vol. 10, no. 1, 2023.

[36] Z. Long, H. Yan, G. Shen, X. Zhang, H. He, and L. Cheng, "A Transformer-based network intrusion detection approach for cloud security," Journal of Cloud Computing, vol. 13, no. 1, 2024.

[37] A. Aldribi, I. Traoré, B. Moa, and O. Nwamuo, "Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking," Computers Security, vol. 88, p. 101646, 2020.