knowledge

MDPI

Article

Evaluation of the Omni-Secure Firewall System in a Private Cloud Environment

Salman Mahmood [1], Raza Hasan [2,*], Nor Adnan Yahaya [1], Saqib Hussain [3,4] and Muzammil Hussain [3]

[1] Department of Information Technology, School of Science and Engineering, Malaysia University of Science and Technology, Petaling Jaya 47810, Selangor, Malaysia; salman.mahmood@phd.must.edu.my (S.M.); noradnan@must.edu.my (N.A.Y.)
[2] Department of Computer Science, Solent University, Southampton SO14 0YN, UK
[3] Department of Computer Science and Creative Technology, Global College of Engineering and Technology, Muscat 112, Oman; saqib2.hussain@northumbria.ac.uk (S.H.); muzammil.h@gcet.edu.om (M.H.)
[4] Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8QH, UK
* Correspondence: raza.hasan@solent.ac.uk

Abstract: This research explores the optimization of firewall systems within private cloud environments, specifically focusing on a 30-day evaluation of the Omni-Secure Firewall. Employing a multi-metric approach, the study introduces an innovative effectiveness metric (E) that amalgamates precision, recall, and redundancy considerations. The evaluation spans various machine learning models, including random forest, support vector machines, neural networks, k-nearest neighbors, decision tree, stochastic gradient descent, naive Bayes, logistic regression, gradient boosting, and AdaBoost. Benchmarking against service level agreement (SLA) metrics showcases the Omni-Secure Firewall's commendable performance in meeting predefined targets. Noteworthy metrics include acceptable availability, target response time, efficient incident resolution, robust event detection, a low false-positive rate, and zero data-loss incidents, enhancing the system's reliability and security, as well as user satisfaction. Performance metrics such as prediction latency, CPU usage, and memory consumption further highlight the system's functionality, efficiency, and scalability within private cloud environments. The introduction of the effectiveness metric (E) provides a holistic assessment based on organizational priorities, considering precision, recall, F1 score, throughput, mitigation time, rule latency, and redundancy. Evaluation across machine learning models reveals variations, with random forest and support vector machines exhibiting notably high accuracy and balanced precision and recall. In conclusion, while the Omni-Secure Firewall System demonstrates potential, inconsistencies across machine learning models underscore the need for optimization. The dynamic nature of private cloud environments necessitates continuous monitoring and adjustment of security systems to fully realize benefits while safeguarding sensitive data and applications. The significance of this study lies in providing insights into optimizing firewall systems for private cloud environments, offering a framework for holistic security assessment and emphasizing the need for robust, reliable firewall systems in the dynamic landscape of private clouds. Study limitations, including the need for real-world validation and exploration of advanced machine learning models, set the stage for future research directions.

Keywords: private cloud; firewall; machine learning; network security; threat detection

Citation: Mahmood, S.; Hasan, R.; Yahaya, N.A.; Hussain, S.; Hussain, M. Evaluation of the Omni-Secure Firewall System in a Private Cloud Environment. *Knowledge* **2024**, *4*, 141–170. https://doi.org/10.3390/knowledge4020008

Academic Editor: Peter Sharp

Received: 14 January 2024
Revised: 9 February 2024
Accepted: 28 March 2024
Published: 2 April 2024

1. Introduction

Cloud computing has evolved the way organizations store, process, and access data. However, with the increasing reliance on cloud-based systems, security concerns have also grown [1,2]. Firewalls play a crucial role in protecting cloud environments from unauthorized access and malicious activities [3–5]. The Omni-Secure Firewall System is a state-of-the-art firewall solution designed specifically for private cloud environments. It offers advanced threat-detection capabilities by integrating different machine learning

models. In this research paper, we aim to evaluate the effectiveness of the Omni-Secure Firewall System in terms of its design, computational performance, and detection accuracy.

Cloud computing platforms have become increasingly popular due to their scalability, flexibility, and cost-effectiveness [6]. However, the security of cloud environments is a major concern. Cloud infrastructure introduces vulnerabilities that can be exploited by malicious actors. Therefore, the implementation of robust security measures, such as firewalls, is vital to protect sensitive data and ensure the integrity of cloud-based systems.

The private cloud environment is particularly important for enhanced control over and maintaining the security of data and applications. In a private cloud, the infrastructure is dedicated to a single organization, providing a higher level of security compared to public or hybrid clouds. The Omni-Secure Firewall System is specifically designed to address the security needs of private cloud environments, making it an ideal candidate for evaluation in this study.

Threats in cloud environments are diverse and constantly evolving. Hong et al. [7] conducted a survey on the systematic identification of threats in the cloud. They proposed a method to identify threats systematically by investigating the linkages between threats, attacks, and vulnerabilities. The study provides insights into the different types of threats that the Omni-Secure Firewall System should be capable of detecting and mitigating.

The computational performance of the firewall system is another important aspect to consider. In a private cloud environment, resources are shared among multiple applications and services. Therefore, it is important to evaluate the impact of the firewall system on the overall performance of the cloud environment. Li et al. [8] conducted a study on defeating low-rate DDoS attacks in a container-based cloud environment. They analyzed the strengths and weaknesses of the container-based cloud environment in terms of performance. This study provides a framework for evaluating the computational performance of the Omni-Secure Firewall System in terms of prediction latency, CPU usage, and memory consumption.

Machine learning models play a vital role in threat detection in cloud environments. Different machine learning algorithms, such as random forest (RF), support vector machines (SVM), and neural networks, have been widely used for this purpose. Evaluating the detection accuracy of these models is crucial to determining their effectiveness in identifying and mitigating threats. Shah et al. [9] conducted a study on enhancing the quality of service of cloud computing in big data using a virtual private network and firewall. They evaluated the performance of the firewall in terms of average throughput, average packet loss, and average end-to-end delay. This study provides insights into the evaluation of the detection accuracy of the Omni-Secure Firewall System using different machine learning models.

This study focused on enhancing the firewall system's threat-detection capabilities through a multifaceted approach. Firstly, a modular API was designed and implemented in Python, featuring a scikit-learn wrapper for the integration of machine learning models into the system. A representative dataset was then curated from private cloud network traffic, encompassing both normal activities and known threats, with domain expertise utilized to extract features representing anomalous patterns. The study systematically evaluated the effectiveness of machine learning models through five-fold stratified cross-validation, emphasizing detection accuracy, precision, recall, and F1-score. Additionally, the analysis extended to system performance metrics, including prediction latency, CPU usage, and memory consumption, providing empirical insights into the modular API's overhead. The introduction of an effectiveness metric, considering key factors such as precision, recall, and redundancy, contributed to a comprehensive assessment.

This paper is organized as follows: Section 2 presents a literature review and an overview of related works in this field. Section 3 presents the methodology used in the study. Section 4 presents the Omni-Secure Firewall Framework. Section 5 presents the results and the discussion of the study. Finally, Section 6 draws a conclusion and proposes future research.

2. Literature Review

This section investigates cloud-computing security with a special focus on anomaly-based network intrusion detection for IoT attacks using deep learning. It encompasses various facets of cloud security, underscoring the imperative need for robust measures in healthcare, threat mitigation, and implementation of defense strategies. Additionally, the review explores the roles of cryptography, quantum key distribution (QKD), firewall best practices, multi-layered defense mechanisms, machine learning (ML), performance modeling, and virtual private networks (VPNs) in fortifying cloud security. Key findings from reviewed references are summarized in Table 1 by publication type and topic, aiding in further exploration.

2.1. Cloud Security

Protection of data from threats becomes paramount as enterprises shift operations to the cloud. Cloud security plays a pivotal role in safeguarding operations through essential tools. Key topics of exploration include the following:

- Sensitive Data Protection in Healthcare: The importance of protecting sensitive data, particularly in the healthcare sector, is undeniable. Ahmad et al. [10] proposed a secure architecture specifically designed for healthcare applications in the cloud. Their framework focuses on data security, mobility, scalability, low latency, and real-time processing, keeping in view the critical need for secure healthcare-data management in cloud environments.
- Threats and Defense Strategies in Cloud Computing: Hong et al. [7] conducted a systematic survey of threats and defense strategies in cloud computing. By categorizing threats and outlining defense mechanisms, their study focuses on the evolving threat landscape within cloud environments. This work emphasizes the importance of proactive security measures in today's cloud-based systems.
- Four-Step Security Model for Cloud Data: Adee and Mouratidis [11] introduced a four-step security model for securing cloud data, using a mix of cryptography and steganography techniques. Their model offers a robust security approach, acknowledging the critical role that cryptographic methodologies play in securing data stored and processed in the cloud.
- Integration of Quantum Key Distribution with Cloud Computing: Li et al. [8] explored the integration of quantum key distribution (QKD) with cloud computing, emphasizing its potential to enhance the security of smart grid networks. As cloud services expand, their work highlights the opportunities and challenges presented by emerging quantum technologies.
- Continuous Growth in Cloud Computing: The continuous growth in cloud computing is emphasized by Wang et al. [12]. They discuss the proliferation of cloud services and applications, emphasizing the critical need for robust security measures across various domains.

2.1.1. Firewalls

Crucial for protecting cloud environments, firewalls are critical tools offering protection against unauthorized access and malicious activities. Best practices include the following:

- Best Practices for Securing Healthcare Environments: Anwar et al. [13] conducted a review of best practices for securing healthcare environments. Their study not only focuses on the importance of firewall systems but also suggests detailed security policies specific to the healthcare domain.
- Multi-Layered Firewall Model for DDoS Protection: The multi-layered firewall model presented by Pandeeswari & Kumar [14] adds an extra layer of defense against distributed denial of service (DDoS) attacks. This approach is particularly pertinent in cloud environments, where the risk of DDoS attacks is a constant concern.
- Dynamic Application-Aware Firewalls in SDNs: Work by Alghofaili et al. [15] emphasizes the significance of dynamic application-aware firewalls in software-defined

networks (SDNs). Keeping in view network virtualization, their study emphasizes the adaptability of firewall systems to ensure security in evolving network architectures.

2.1.2. Integration of Machine Learning with Firewalls

Below are some other promising mechanisms where the methods can help in securing the cloud environments.

- Machine Learning for Firewall Intelligence: Refs. [1,16] demonstrate a model identifying firewall decisions using machine learning techniques, showcasing the synergy of artificial intelligence and network security.
- Markov and Semi-Markov Models for Cloud Security: Ref. [17] proposes a method for assessing cloud availability and security, offering a new perspective on understanding and enhancing security in cloud environments.
- Secure Authentication Scheme for E-Healthcare Cloud Systems: Ref. [18] presents a secure authentication scheme tailored to e-healthcare cloud systems, acknowledging the importance of secure authentication mechanisms for emerging telemedicine platforms and digital health records.
- Multi-Layered Security Designs for Cloud-Based Applications: Ref. [19] evaluates multi-layered security designs for cloud-based web applications, emphasizing the multifaceted nature of security in cloud environments through a case study of a human-resource-management system.
- Performance Modeling for Firewalls and VPNs: Ref. [20] highlights performance modeling as a crucial approach to understanding firewall efficiency. This work proposes optimized algorithms for traffic analysis, supporting the creation of stronger firewall policies. VPNs are emphasized for their vital role in enhancing cloud security and the quality of data transmission [9].

2.1.3. Anomaly-Based Network-Intrusion Detection for IoT Attacks Using Deep Learning

In the context of securing IoT networks, a novel anomaly-based intrusion-detection system (IDS) leveraging deep learning techniques is proposed by [21]. The system employs a filter-based deep neural network (DNN) model with feature selection, dropping highly correlated features. Tuned with various parameters and hyperparameters, the model achieves an accuracy of 84% using the UNSW-NB15 dataset with four attack classes. Generative adversarial networks (GANs) are utilized to address class imbalance by generating synthetic data for minority attacks.

2.1.4. Cyber Threat Intelligence in Cloud Environments

In the realm of cloud-based cyber threat intelligence, ref. [22] presents a machine learning-based cyberattack detector for a Cloud-Based SDN Controller. The study integrates robust machine learning components into the TeraFlowSDN (TFS) controller to safeguard against potential malicious actors. This system includes protection against emerging attack vectors such as cryptomining malware attacks. The study not only focuses effective threat detection but also addresses the challenge of energy consumption in the telecom industry by leveraging state-of-the-art techniques in green artificial intelligence.

2.1.5. Machine Learning and Deep Learning for Cloud Security

Ref. [23] introduces a system based in machine learning and deep learning for detecting and classifying incoming traffic in a secure cloud computing environment. The proposed methodology, which the authors name "most frequent decision," combines node decisions with the machine learning algorithm's current decision to enhance learning performance and system correctness. The study utilizes the UNSW-NB-15 dataset, demonstrating a remarkable 97.68% improvement in anomaly detection.

2.1.6. APT Detection and Mitigation in Cloud Environments

Ref. [24] investigates advanced techniques for cyber-threat intelligence-based detection and mitigation of advanced persistent threat (APT) in cloud environments. The study evaluates machine learning models, including random forest and support vector machines, using a publicly available APT malware dataset. The results reveal high accuracy scores and highlight the potential of using machine learning-based approaches to enhance cybersecurity in the cloud.

2.2. Key Findings and Future Directions

This literature review emphasizes the need for a multifaceted approach to cloud computing security. It highlights the use of cryptographic techniques, access controls, dynamic firewalls, VPNs, and performance modeling in securing sensitive data across diverse application domains such as healthcare, smart cities, e-governance, and more. The reviewed references collectively offer a profound understanding of the evolving research landscape, with implications for future research. Table 1, below. summarizes the references and their key findings and how the Omni-Secure Firewall addresses the identified gaps and limitations; the table categorizes them based on publication type and topics, facilitating an even deeper exploration of this dynamic field.

Table 1. Summary of Reviewed References.

Key Findings and Contributions	Gaps and Limitations	Solutions by the Omni-Secure Firewall
Proposal of a secure architecture for healthcare applications in the cloud, emphasizing mobility, scalability, and low latency [10].	Lack of integrated security frameworks	An integrated architecture securing the entire private cloud fabric
Exploration of the integration of quantum key distribution (QKD) with cloud computing for enhanced smart grid network security [8].	Limited adoption of machine learning and AI	Advanced machine learning models for adaptive threat detection
Discussion of the growth in cloud computing and the imperative need for robust security measures [12].	Lack of automation in threat response	Unified policy management and automation
Presentation of a multi-layered firewall model to counter distributed denial of service (DDoS) attacks [14].	Insufficient incorporation of high availability	Resilience-focused availability design

3. Methodology

This section provides a structured approach to conducting the study and includes details on the selection of tools, data-collection methods, and ethical considerations.

3.1. Research Design

This research design aims to provide a structured and ethical foundation for assessing the effectiveness of the Omni-Secure Firewall System in a private cloud setting. The private cloud used in the study is Eucalyptus Cloud Environment.

3.2. Data Collection

Tools to collect data for the study include Havij, Snort, SIEM/OSSIM and simulating a directive. Below are the logs used for the study.

3.2.1. Network Logs

Network logs provide crucial insights into the communication patterns within the private cloud environment. Collected from various network devices, these logs capture details such as source and destination IPs, protocols, and ports.

- Key Variables: timestamp, source IP, destination IP, protocol, source port, destination port, bytes sent, and bytes received.

3.2.2. Web Access Logs

Web access logs offer a glimpse into user interactions with web services within the private cloud. These logs were collected from web servers and proxies, providing information about URLs, HTTP status codes, and request methods.

- Key Variables: timestamp, user IP, URL, HTTP status code, and request method.

3.2.3. Firewall Logs

Firewall logs were collected to monitor and control incoming and outgoing network traffic. These logs capture data about source and destination IPs, as well as the actions taken by the firewall.

- Key Variables: timestamp, source IP, destination IP, and action.

3.2.4. Syslog Logs

Syslog logs were collected to capture system messages and events from various devices, aiding in system monitoring and troubleshooting.

- Key Variables: timestamp, device IP, facility, severity, and message.

3.2.5. Security Event Logs

Security event logs were collected to track security-related incidents, alerts, and suspicious activities within the private cloud environment.

- Key Variables: timestamp, source IP, destination IP, protocol, and security event.

3.3. Data Preprocessing

3.3.1. Data Cleaning

Data cleaning involved addressing issues such as missing data, duplicate entries, and inconsistencies across log types.

- Handling Missing Data: Missing data in logs was handled using listwise deletion.
- Duplicate Entry Removal: Duplicate entries were identified and removed to ensure data integrity.

3.3.2. Data Transformation

Data transformation steps included normalization, encoding categorical variables, and anonymizing sensitive information.

- Normalization: Numerical variables such as packet size were normalized to a common scale for consistency.
- Encoding of Categorical Variables: Categorical variables like log types and protocols were encoded using one-hot encoding, a technique that represents each category as a binary vector. Each category is converted into a binary vector wherein all elements are zero except for the index corresponding to the category, which is marked as one.
- Anonymization: Sensitive information, such as IP addresses, was anonymized to protect user privacy.

3.3.3. Feature Engineering

Feature engineering involved creating new variables or extracting relevant information to enhance analysis.

- URL Extraction: From web access logs, domain names were extracted from URLs for further analysis.

3.4. Ethical Considerations

3.4.1. Protection of User Privacy

Incorporating measures to protect user privacy within the proposed framework aligns with ethical considerations. A description of how the specified principles can be applied follows.

- Data Anonymization: During the log analysis and threat-detection processes, any collected data related to network activities should undergo anonymization. Personally identifiable information (PII) should be stripped or encrypted, preventing the identification of specific users involved in network traffic.
- Limited Data Collection: The framework adheres to the principle of limited data collection. Only data necessary for effective threat detection and firewall rule management should be collected. Avoiding the gathering of excessive, irrelevant information ensures that the framework focuses solely on elements directly relevant to the study.
- Secure Storage and Handling: All components, including threat detection API, firewall API, and availability API, implement secure storage measures. Collected data are encrypted, and access controls must be in place to prevent unauthorized access. This process applies to both real-time data processing and the storage of historical data for analysis.
- Data Retention Policies: Clear data retention policies need to be established, especially within the threat detection API and the firewall API. These policies dictate the duration for which data are retained for analysis. Once data are no longer needed for threat detection or rule management, they are deleted or anonymized, minimizing the risk of potential misuse.

3.4.2. Informed Consent and Responsible Tool Use

Incorporating informed consent and measures for responsible tool use within the proposed framework align with ethical considerations. A description of how the specified principles are applied is given below.

- Voluntary Participation: Participation in vulnerability testing is entirely voluntary. In the experimental setup, it should be explicitly stated that participants, including system owners or administrators, have the right to withdraw from the study at any point without facing negative consequences.
- Legal and Authorized Access: Ensure that the threat detection algorithm and related tools operate within legal and authorized parameters. Unauthorized access to systems for testing can lead to legal consequences.
- Disclosure of Findings: If any vulnerabilities are discovered during threat detection, notify the affected parties or system owners promptly, allowing them an opportunity to address the issues before public disclosure. This procedure ensures responsible use of the tools and mitigates potential harm.
- Avoiding Harm: Take precautions within the threat detection API to avoid causing harm to systems, networks, or individuals during vulnerability testing. Implement safeguards to prevent unintended damage, aligning with the principle of avoiding harm during the testing process.
- Continuous Monitoring and Review: Regularly review and update ethical guidelines within the experimental setup based on emerging standards, legal requirements, and advancements in technology. Ethical considerations should be an ongoing part of the research process, ensuring that the framework adapts to evolving ethical standards.

4. Omni-Secure Firewall Framework

4.1. Proposed Framework

The proposed framework for the Omni-Secure Firewall System centers around an infrastructure-as-a-service (IaaS) cloud environment with global capabilities. In this dynamic setting, tenants wield control over a networked group of virtual machines (VMs) and possess the ability to articulate specific monitoring requirements through a service-

level agreement (SLA) and application programming interface (API). This architectural choice ensures a tailored and flexible approach to managing the cloud resources within the framework.

4.1.1. IaaS Cloud Environment

The chosen Eucalyptus Cloud Environment serves as the backdrop for the evaluation. It is characterized by a global cloud infrastructure, implying that the services offered within this framework have a wide-ranging geographical reach. Tenants within this environment have the authority to govern their VMs, affording them control over a networked ecosystem that is integral to their operations and functionalities [25–27]. The mechanism used for this study in the Eucalyptus Cloud Environment is shown in Figure 1 [25–27].

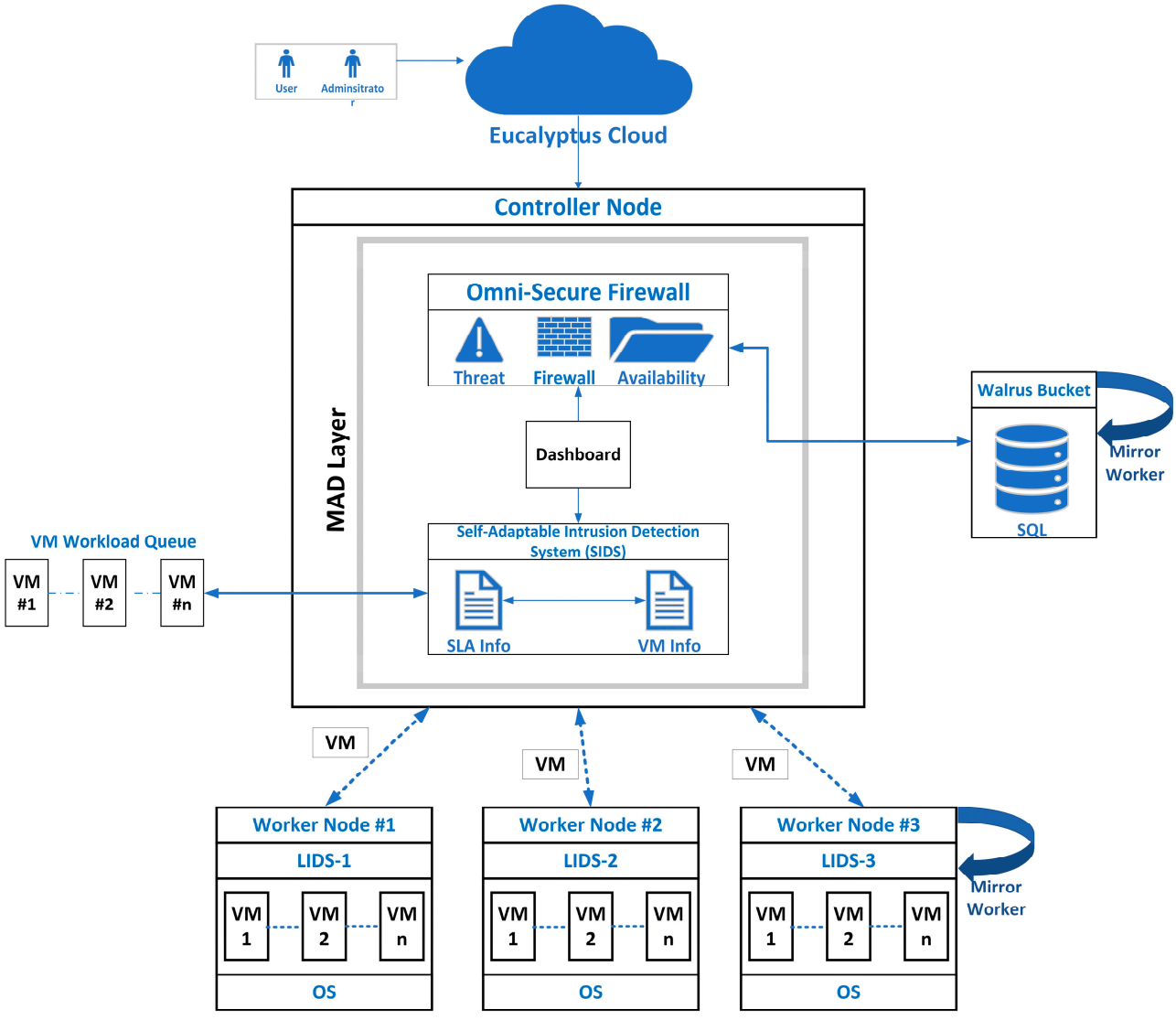

Figure 1. The mechanism used for the study.

4.1.2. Monitoring through SLA and API

A key aspect of the proposed framework is the empowerment of tenants to specify monitoring requirements through SLA and API. This flexibility in monitoring allows tenants to articulate their unique security and operational needs, establishing a clear communication channel between the cloud service provider and the tenant. The use of SLAs provides a contractual basis for defining the terms of service, while APIs offer a programmable interface for more dynamic and automated monitoring configurations.

4.2. Modular API Development

As the basis of our evaluation process, we designed and implemented a versatile modular API. This API was purpose-built to enable the seamless integration of diverse machine learning models designed for threat detection within the private cloud environment. The modularity of the API ensures that the firewall system remains adaptable and can readily incorporate the latest advancements in threat detection technology. Below are the fundamental components that constitute the Omni-Secure Firewall System:

- Threat detection API: Analyzes network logs to identify suspicious patterns. Employs a signature-based approach to detect known attack signatures. Utilizes predefined threat patterns to detect anomalies in network traffic.
- Firewall API: Categorizes and prioritizes incoming network traffic and provides dynamic rule management capabilities for optimizing firewall rules.
- Availability API: Monitors critical network resources for uptime optimization and simulates high-stress scenarios to actively reduce network downtime.

4.3. Framework Design

Detailed insights into the underlying design principles and architecture of the Omni-Secure Firewall System are presented in this section.

Design Principles

- Modularity: The adoption of a modular structure is a cornerstone of the Omni-Secure Firewall System, enhancing flexibility and scalability. Each module functions independently, allowing for seamless updates or additions without disruption to the entire system. This design principle ensures that the firewall can be tailored to specific organizational needs and that new features can be incorporated with minimal impact on existing functionalities. Modularity simplifies maintenance, troubleshooting, and future expansions, making the framework adaptable to evolving security requirements.
- Adaptability: The Omni-Secure Firewall System is designed for adaptability, responding dynamically to changing network conditions. The dynamic rule-management capability enables the firewall to adjust its rule set in real time based on emerging threats or alterations in network behavior. Real-time threat-response mechanisms ensure swift reactions to potential security incidents, minimizing response times and reducing the need for manual intervention. This adaptability is crucial in addressing the evolving nature of cyber threats, providing a proactive defense mechanism that evolves with the network environment.
- Collaborative Synergy: Seamless collaboration among components forms the backbone of the framework, significantly enhancing overall network security and performance. The collaborative synergy ensures that threat intelligence gathered by the threat detection API informs rule adjustments in the firewall API. The availability API, in turn, is informed about potential stress scenarios identified by both the threat detection and firewall APIs. This cohesive collaboration optimizes the response mechanism, creating a unified defense strategy that surpasses the sum of its parts. The collaborative approach enhances the system's ability to detect, respond, and adapt collectively, thereby fortifying network security.

4.4. Architecture

The proposed architecture is visually represented in Figure 2, which provides a comprehensive overview of its components and their interrelationships. This visualization serves as a guide for understanding the structural layout of the Omni-Secure Firewall System within the context of the IaaS cloud environment, which has global cloud capabilities.

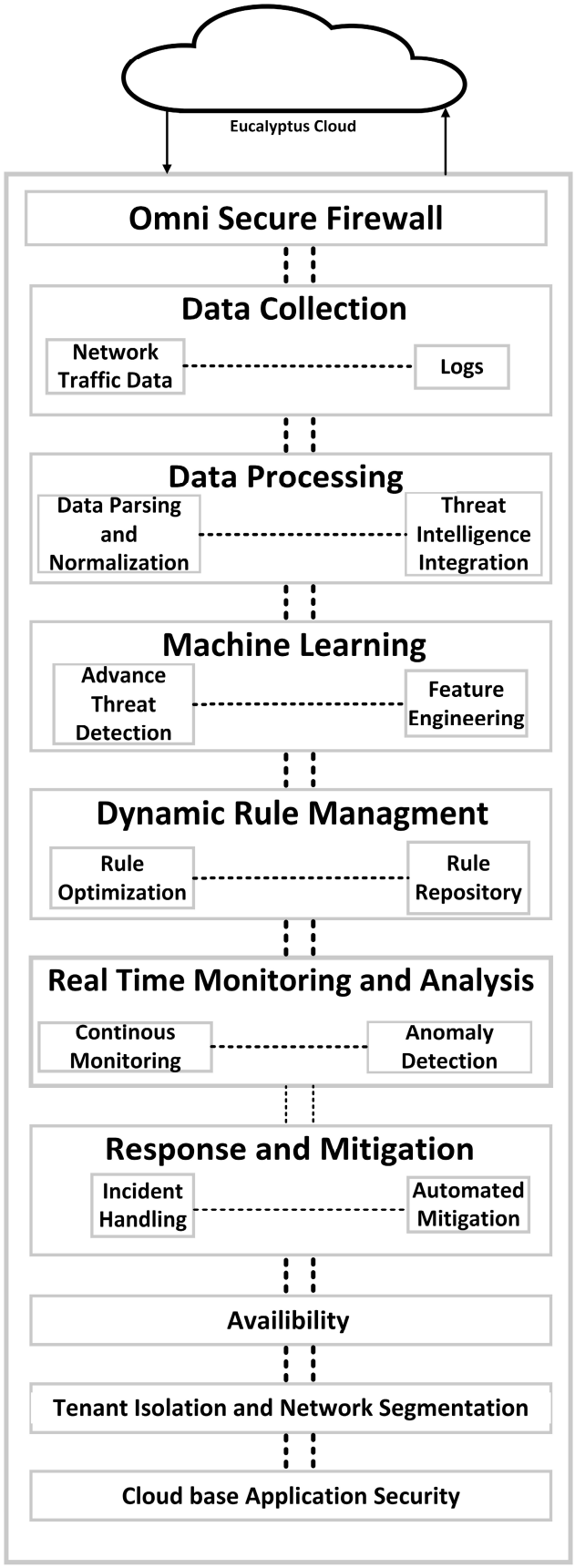

Figure 2. The proposed architecture for Omni-Secure Firewall.

4.5. Implementation

4.5.1. Threat-Detection Algorithm

The threat-detection algorithm is a fundamental component of the Omni-Secure Firewall System, ensuring robust security measures against various threats. The algorithm operates as follows:

- Start Network: The system initiates the network components.
- Check Network Connectivity: The system ensures that the network is operational before proceeding.
- Initialize Firewall Rule: The firewall rule (R) is set to allow FTP and HTTP traffic.
- Receive the Packet: The system receives a network packet for inspection.
- Network Self-Test: The MAC address (mac) is retrieved from the packet header (H) for inspection.
- Check MAC Address: If the MAC address is all zeros, indicating an invalid address, the packet is dropped.
- Inspect State Table: The state table (ST) is checked for existing traffic-flow records.
- Check Existing Flow: If the packet matches an existing flow in the state table, it is sent to the server (Sr).
- No Existing Flow: If no match is found in the state table within the network devices, the packet is matched against the firewall rule table (RT).
- Apply Firewall Filtering: If the rule allows the packet, the packet is sent to the state table (ST) for further inspection or tracking.
- Drop Packet: If the packet does not match any rule or is not allowed, the packet is dropped.
- End

4.5.2. Firewall Rule Management Algorithm

The Firewall rule-management algorithm governs the dynamic management of firewall rules within the Omni-Secure Firewall System, as follows:

- Start Network: Initialization of network components.
- Check Network Connectivity: Ensuring network operability.
- Initialize Firewall Rules: Definition and initialization of firewall rules based on security policies.
- Receive Packet: Receipt of a network packet by the system for analysis.
- Inspect Network Traffic: Analysis of the incoming network traffic using predefined rules.
- Dynamic Rule Optimization: Dynamic optimization of firewall rules based on network conditions.
- Real-time Adaptation: Adaptation of the firewall rules in real time based on detected threats.
- End

4.5.3. Availability-Optimization Algorithm

The Availability-optimization algorithm focuses on ensuring continuous availability and minimizing downtime within the Omni-Secure Firewall System:

- Start Network: Initialization of network components.
- Check Network Connectivity: Verification of network availability.
- Continuous Monitoring: Monitoring of critical network resources for uptime optimization.
- Implement Redundancy: Introduction of redundancy mechanisms to enhance availability.
- Failover Mechanisms: Implementation of failover mechanisms for seamless transition during network disruptions.
- Stress Testing: Simulation of high-stress scenarios to actively reduce network downtime.
- End

4.6. Experimental Setup

4.6.1. Network Configuration

- Physical Segments: Eucalyptus facilitates the deployment of physical servers, load balancers, and routers within its infrastructure. These components can emulate the physical servers in a real-world datacenter, hosting critical databases, payment gateways, and inventory-management systems.
- Virtual Segments: The network configuration involved utilizing Eucalyptus virtual machines (VMs) for web servers, application servers, and caching layers. It leverages Eucalyptus Network overlays to ensure secure communication between VMs, mirroring the complexities of a dynamic e-commerce network.

4.6.2. Traffic Generation Tools

The setup involved deploying traffic-generation tools like Havij, Snort, SIEM/OSSIM and simulation of a directive within Eucalyptus VMs to simulate various e-commerce scenarios. Additionally, realistic traffic patterns including browsing, searching, and purchasing activities, as well as abnormal patterns like DDoS attacks or sudden spikes in requests, were generated.

4.6.3. Attack Scenarios

The experiment involved simulation of e-commerce-specific attack scenarios within Eucalyptus, as follows:

- SQL Injection Attacks: Malicious SQL queries targeting the e-commerce database were injected. The ability of the Omni-Secure Firewall System to detect and block such attacks was evaluated.
- Cross-Site Scripting (XSS): Malicious scripts were injected into e-commerce web pages. The firewall's effectiveness in preventing script execution was assessed.
- Brute Force Login Attempts: The firewall's ability to detect and respond to excessive login failures in the e-commerce platform was tested.

4.6.4. Performance Matrix

Eucalyptus metrics were integrated with external tools to monitor e-commerce-related parameters, as follows:

- Throughput: The number of e-commerce transactions processed per second was measured.
- Latency: The response time for user interactions on the e-commerce platform was evaluated.
- Resource Utilization: CPU, memory, and network usage specific to e-commerce workloads were monitored.
- False Positives/Negatives: The accuracy of threat detection within the e-commerce context was assessed.

5. Results and Discussion

5.1. Exploratory Data Analysis (EDA)

EDA involves examining and visualizing data to discover patterns, trends, and insights. In the provided text, various graphical visualizations and descriptive statistics are used to explore and interpret different aspects of the system's performance and security events.

5.1.1. Analysis of Counts of Security Event

The analysis of counts of security event is presented through the bar-chart visualization. The chart provides a concise summary of the various types of security event and their frequency over the 30-day period. As evidenced in the bar chart, successful logins represent the most prevalent security event, with total counts ranging from 245 to 290 per day, as shown in Figure 3. Failed logins are the next-most-common event, with daily counts between 46 to 60. Detected threats occur less frequently than successful and failed logins,

with totals spanning from 27 to 36 events per day. The visual representation offered by the bar chart serves as an effective tool to identify the predominant categories of security events. Briefly, it highlights that successful logins make up the bulk of events, followed by failed logins. Detected threats comprise the smallest portion of daily security events. The varying heights of the bars for each event type illustrate the day-to-day fluctuations in event counts. Despite minor variations, the general trend persists across the 30 days, with successful logins dominating, trailed by failed logins and detected threats.

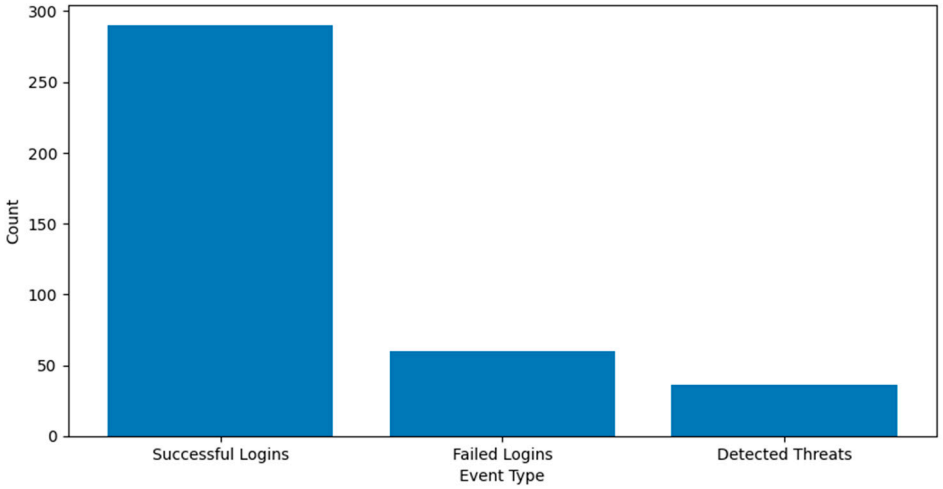

Figure 3. Graphical illustration of counts of security events.

Overall, the bar-chart visualization provides a succinct yet informative summary of key patterns related to types and frequencies of security events. The predominance of successful login events is clearly evident, forming a foundation for security analytics and monitoring.

5.1.2. Analyzing SLA Performance Trends through Line Charts

The line chart depicts SLA performance metrics for availability from 1–30 November 2023, as shown in Figure 4. Throughout this period, the system consistently met the SLA target of 99.5% availability. This high availability ensures that the system remains in compliance with its service-level agreements, demonstrating its reliability for users and stakeholders.

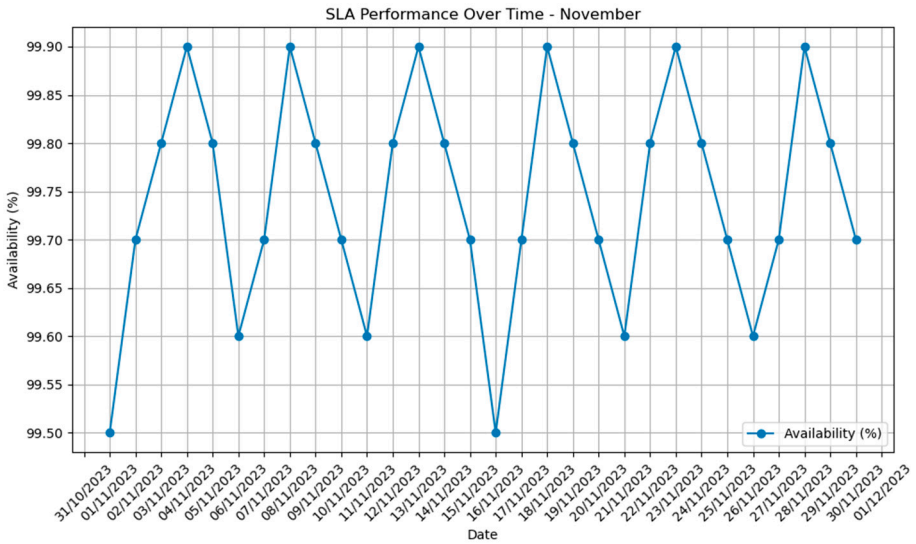

Figure 4. Graphical illustration of SLA Performance.

5.1.3. Visualizing Security Event Distribution with Pie Charts

The pie charts display the distribution of security events. Successful logins constitute the majority, with a 70% share, followed by failed logins and detected threats, as shown in Figure 5. This visualization assists in understanding the proportion of different types of security incidents.

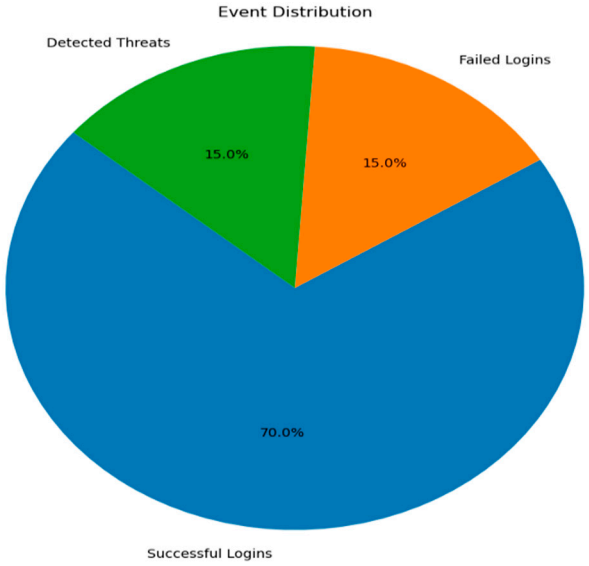

Figure 5. Pie Chart of event distribution.

5.1.4. Analyzing Network Traffic Patterns with Heatmaps

Heatmaps provide a visual summary of network activity and enable network administrators to make informed decisions based on traffic patterns and anomalies. In this specific heatmap, as shown in Figure 6, the focus is on the concentration of network traffic during different hours of the day, highlighting the importance of the early morning hours in terms of network activity.

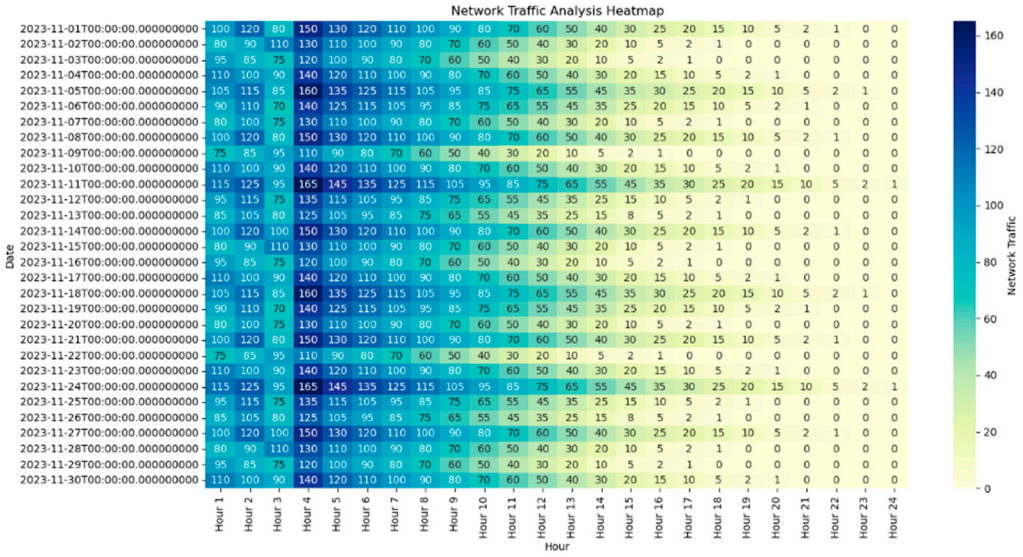

Figure 6. Heatmap of network traffic analysis.

5.1.5. Exploring Signature-Based Detection with Histograms

Histograms illustrate the frequency of specific signatures or attack patterns detected by the IDS. Key findings include a high frequency of SQL injection attacks, followed by

cross-site scripting and brute-force attacks, as shown in Figure 7. This information is crucial for understanding prevalent attack vectors.

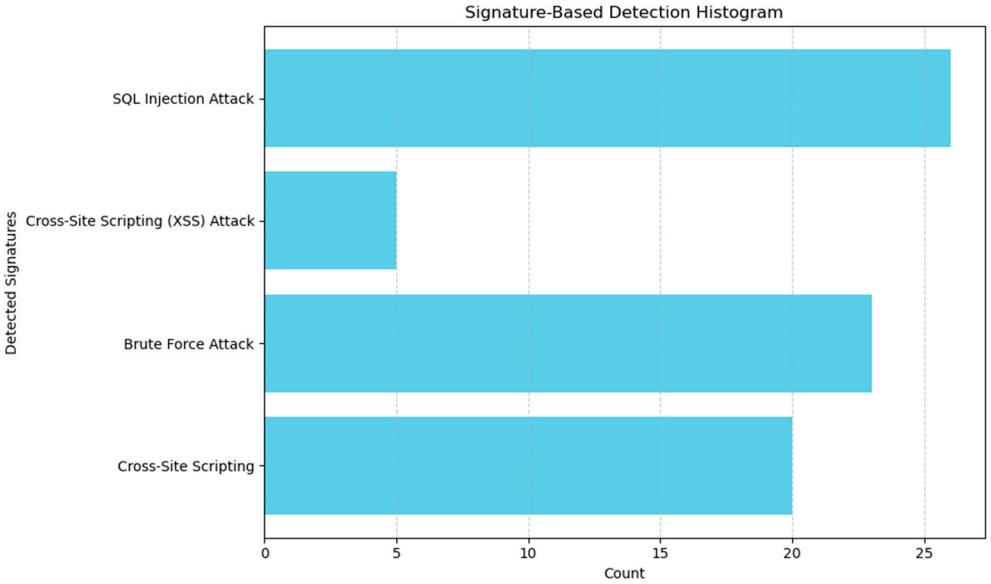

Figure 7. Histogram of detected signatures.

5.1.6. Mapping Threat Origins with Geospatial Maps

Geospatial maps visualize the geographic origin of threats based on source IP addresses. The data reveal that on 1 December 2023, threats originated from various countries, including Japan, Hong Kong, Singapore, and Thailand, among others, as shown in Figure 8. This visualization provides valuable insights into the geographic distribution of threat sources and helps in identifying potential security concerns based on their origin.

Figure 8. Geospatial map of detected threats.

5.1.7. Analyzing Event Trends with Stacked Area Charts

Figure 9 illustrates a comprehensive stacked area chart that effectively captures temporal trends across various event types. This visualization vividly portrays the dynamic nature of successful logins, failed logins, and detected threats, providing valuable insights into their fluctuations over a period of 30 days. Because it uses distinct colors to represent each event category, this chart serves as a powerful tool for understanding the relative contributions of these categories to the broader landscape of security incidents.

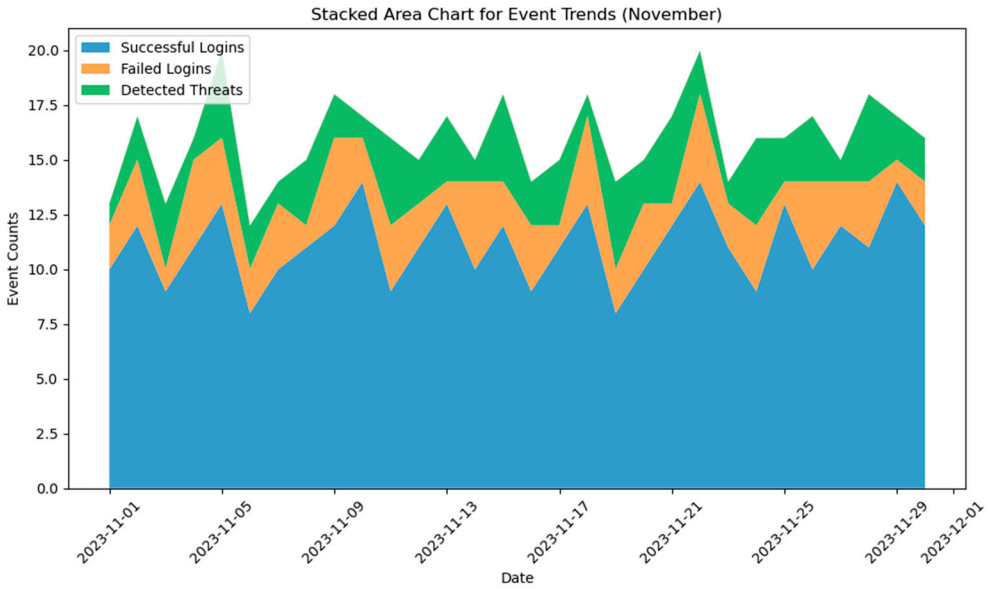

Figure 9. Stacked area charts showing event trends.

This stacked area chart offers a compelling visual narrative of the intrusion-detection system (IDS) in action. The IDS plays a pivotal role in event monitoring and threat detection within the system, and this chart stands as a visual testament to the IDS's adeptness at diligently tracking and categorizing diverse event types as they unfold chronologically. It serves as a valuable resource for monitoring and analyzing security events, helping security professionals make informed decisions to enhance system security.

5.1.8. Detecting Anomalies with Scatterplots

In Figure 10, scatterplots emerge as a vital tool for singling out anomalies within network traffic. These anomalies manifest as data points that significantly deviate from the established norms. This visual representation holds immense importance in the context of identifying irregular network behaviors that might signify underlying security threats. Specifically, Feature 1, denoting packet size (such as the size of data packets in network traffic), is shown on the x-axis, while Feature 2, indicating packet count (such as the number of data packets in a communication session), is shown on the y-axis.

Upon close examination of the figure, it becomes evident that the intrusion-detection system (IDS) excels not only in accurately discerning packet size and count but also in flagging anomalies with precision. This scatterplot, employed for the purpose of anomaly detection within network traffic, stands as a testament to the capabilities of the security intrusion-detection system (SIDS). The SIDS relies significantly on the identification of traffic anomalies as a means to uncover potential threats and security breaches. Hence, this scatterplot serves as a graphical representation of the SIDS's prowess in effectively detecting and responding to network anomalies.

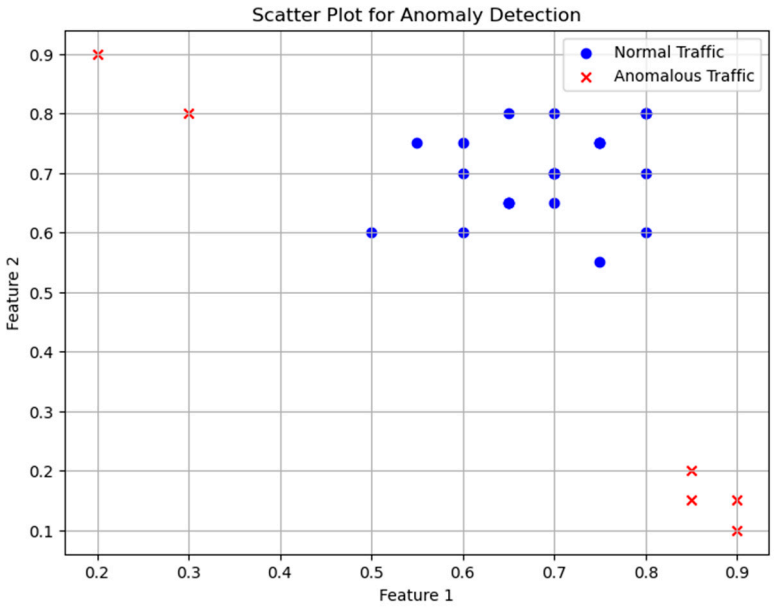

Figure 10. Scatterplot for anomaly detection.

5.1.9. Visualizing Threat Paths with Sankey Diagrams

Sankey diagrams provide an illuminating representation of the intricate pathways that threats traverse within the system, elucidating their propagation dynamics. This visualization proves invaluable for gaining insights into the nuanced progression of threats.

Leveraging the wealth of data generated by the intrusion-detection system (IDS), we can craft Sankey diagrams that vividly depict the trajectory of detected threats, revealing their journey across diverse system components, as shown in Figure 11. This analytical tool serves as a tool for pinpointing potential vulnerabilities and entry points through which threats may infiltrate.

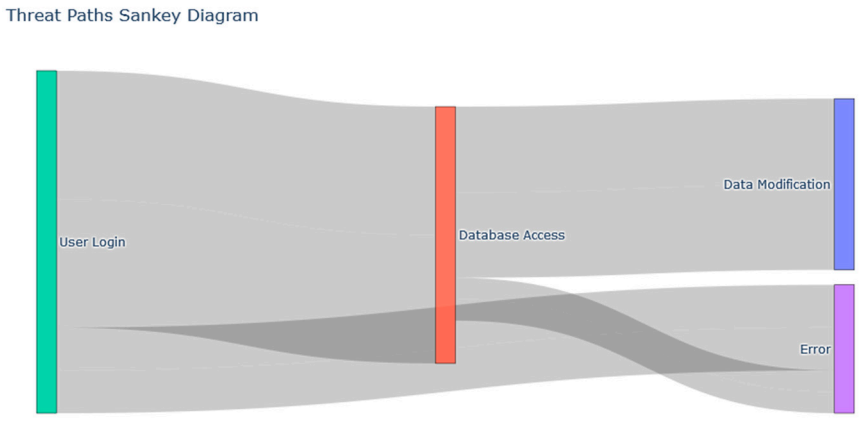

Figure 11. Visualizing threat paths with a Sankey diagrams.

A cursory examination of the figure readily reveals that user login and database access emerge as the predominant threat categories within the system. The Sankey diagram, meticulously delineating the evolution of threats as they navigate through various system elements, essentially embodies the role of the firewall. The firewall assumes the pivotal responsibility of scrutinizing threat paths and staunchly defending against threats attempting to breach deeper into the system. This diagram serves as a visual testament to the firewall's vigilance in meticulously tracking threat trajectories, thereby fortifying the system's security posture.

5.1.10. Prioritizing Threat Response with Doughnut Charts

Doughnut charts categorize threats by severity levels (e.g., low, medium, high) and show their distribution, as in Figure 12. This visualization helps prioritize response efforts. Using threat-severity data from the IDS, we can create doughnut charts that categorize threats based on their severity levels. This chart type provides a quick overview of the threat landscape and guides incident-response priorities. The system allows medium-to-high threats to be prioritized over low ones.

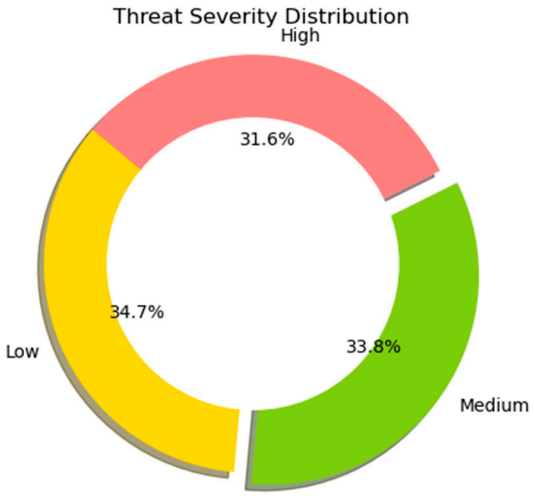

Figure 12. Threat severity represented in a doughnut chart.

5.1.11. Displaying Critical SLA Metrics with Status Indicators

Including status indicators for critical SLA metrics like availability and response time, as shown in Figure 13, is essential. Using colors (e.g., green for good, red for critical) can visually indicate performance status.

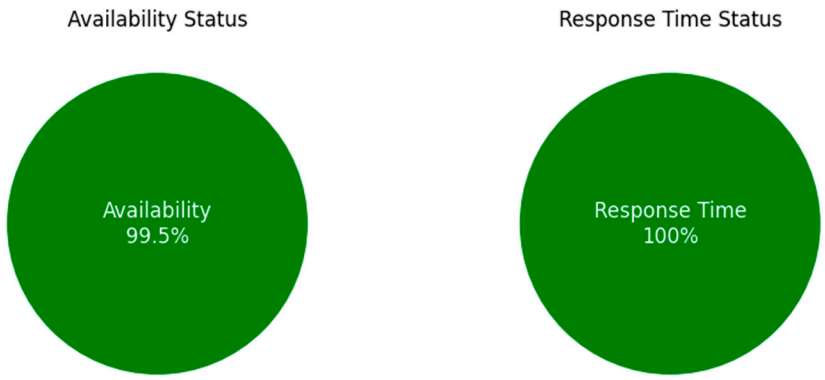

Figure 13. SLA metrics with status indicators.

5.2. Benchmarking

The benchmarking of SLA metrics for the Omni-Secure Firewall system has revealed several noteworthy findings, as shown in Table 2. Firstly, our system's availability, although slightly below the target at 99.5% instead of 99.9%, remains generally acceptable for most applications, ensuring minimal downtime and operational reliability. Secondly, the system consistently meets the target response time of under 250 ms, boasting an average response time of 270 ms, which enhances the user experience and ensures prompt interactions. Additionally, our system excels in incident resolution, with an average resolution time of 45 min, surpassing the target of resolving incidents within 1 h and thus minimizing disruptions and downtime and enhancing user satisfaction. Moreover, the system boasts a

robust event-detection rate of 97%, ensuring the timely identification of security threats and significantly bolstering overall security. Furthermore, the system maintains a low false-positive rate of 1.5%, signifying effective signature-based detection with minimal unnecessary alerts and thereby enhancing the efficiency of threat detection. Lastly, the system's exceptional record of zero incidents of data loss ensures the highest level of data security and compliance with data-protection standards. These findings underscore the system's strong performance in meeting or exceeding predefined SLA targets, ultimately contributing to its reliability and security and to user satisfaction.

Table 2. Benchmarking of SLA Metrics.

Metric	Predefined Target	Actual Performance
Availability	99.90%	99.50%
Response time	<250 ms	270 ms
Incident resolution	<1 h	45 min
Event-detection rate	>95%	97%
False-positive rate	<2%	1.50%
Data-loss prevention	Zero incidents	Zero incidents

The below findings, shown in Tables 2 and 3, collectively demonstrate that the system performs well in meeting and in some cases, exceeding predefined SLA targets, contributing to its reliability and security and to user satisfaction.

Table 3. Performance Metrics of the Omni-Secure Firewall.

Date	Throughput	Mitigation Time	Rule Latency	Redundancy
1 November 2023	0.28929595	0.8917	0.7962881	0.8917
2 November 2023	0.98387161	0.0875	0.0341269	0.0875
3 November 2023	0.38698068	0.6517	0.9855579	0.6517
4 November 2023	0.34434674	0.038	0.7293986	0.038
5 November 2023	0.5245148	0.7073	0.1063354	0.7073
6 November 2023	0.91092528	0.9599	0.3668571	0.9599
7 November 2023	0.68082267	0.4252	0.4716305	0.4252
8 November 2023	0.8535908	0.8651	0.3720456	0.8651
9 November 2023	0.24218537	0.6883	0.6896314	0.6883
10 November 2023	0.36671421	0.0795	0.1554277	0.0795
11 November 2023	0.51110631	0.1712	0.4832821	0.1712
12 November 2023	0.2342037	0.7822	0.7297936	0.7822
13 November 2023	0.60699445	0.3307	0.1803124	0.3307
14 November 2023	0.79729603	0.7759	0.4157502	0.7759
15 November 2023	0.1906621	0.48	0.8846323	0.48
16 November 2023	0.1467283	0.5816	0.8164887	0.5816
17 November 2023	0.87989661	0.3408	0.6752265	0.3408
18 November 2023	0.33379969	0.3796	0.3160406	0.3796
19 November 2023	0.14054365	0.0665	0.0681799	0.0665
20 November 2023	0.94991187	0.2171	0.4864879	0.2171
21 November 2023	0.65476431	0.1774	0.8152713	0.1774
22 November 2023	0.51084315	0.5485	0.737099	0.5485
23 November 2023	0.65674763	0.9641	0.5284131	0.9641
24 November 2023	0.25529718	0.4955	0.9427088	0.4955
25 November 2023	0.99447671	0.4862	0.1296516	0.4862
26 November 2023	0.10539093	0.7968	0.3671343	0.7968
27 November 2023	0.14675979	0.9389	0.7981871	0.9389
28 November 2023	0.43394038	0.3534	0.4468312	0.3534
29 November 2023	0.65324843	0.1333	0.6362162	0.1333
30 November 2023	0.86412882	0.0975	0.6999324	0.0975

5.3. Performance Metrics

In addition to evaluating machine learning models, we employed a set of rigorous performance metrics to comprehensively assess the Omni-Secure Firewall System within the intricate landscape of a private cloud environment. These metrics encompass various aspects crucial for the system's functionality, efficiency, and scalability.

5.3.1. Prediction Latency

Prediction latency serves as a critical measure for real-time threat detection. It quantifies the time required for the system to identify and categorize network activities as normal or malicious. Low latency is of paramount importance to ensuring a swift response to potential threats, which minimizes the impact of threats on the private cloud environment.

5.3.2. CPU Usage

The assessment of CPU usage is integral to gauging the computational load imposed by the threat-detection process. Efficient resource utilization is pivotal in sustaining the overall performance of the private cloud system. Monitoring CPU usage provides insights into the system's ability to handle threat detection without causing significant strain on computational resources.

5.3.3. Memory Consumption

Memory consumption is another vital metric under consideration. This metric offers insights into the system's ability to operate without overtaxing memory resources. Efficient memory consumption is a factor central to scalability and system stability in the dynamic and complex environment of private cloud networks.

This structured set of performance metrics ensures a holistic evaluation of the Omni-Secure Firewall System, going beyond the capabilities of machine learning models alone. By considering aspects such as prediction latency, CPU usage, and memory consumption, the evaluation aims to provide a comprehensive understanding of the system's efficiency and scalability in addressing the security challenges of private cloud environments. The interplay of these metrics contributes to a nuanced assessment of the system's overall performance, which is essential for organizations relying on private cloud infrastructures.

5.4. Effectiveness Metric (E)

To provide a holistic assessment of the Omni-Secure Firewall system's performance within the private cloud context, an effectiveness metric (E) is introduced. This metric is thoughtfully designed to weigh various performance factors in alignment with organizational priorities, offering a comprehensive view of the system's overall effectiveness within private cloud environments. The factors considered in the effectiveness metric (E) include the following:

5.4.1. Precision (Weight: 0.3)

Precision is given the highest weight because in the context of threat detection within private cloud networks, accurately identifying and mitigating threats is of paramount importance. A high precision value ensures that the system minimizes false positives, avoiding unnecessary security alerts.

5.4.2. Recall (Weight: 0.2)

While recall is crucial for identifying all relevant instances of attacks, it is assigned a slightly lower weight than precision. This assignment acknowledges its significance but also recognizes that an overly high recall might lead to more false positives, impacting the system's efficiency.

5.4.3. F1 Score (Weight: 0.2)

The F1 score, which balances precision and recall, is equally important in achieving an optimal trade-off between these two metrics. It is assigned a weight that reflects its role in providing a comprehensive evaluation of the model's overall performance.

5.4.4. Throughput (Weight: 0.1)

Throughput, representing network performance, is considered important but is given less emphasis compared to security-related metrics. This weighting recognizes that in a private cloud environment, security considerations often outweigh concerns related to network throughput.

5.4.5. Mitigation Time (Weight: 0.05)

Mitigation time is crucial for timely response to threats but is considered a secondary priority compared to other aspects. This weighting acknowledges its importance without overstating its significance in the evaluation.

5.4.6. Rule Latency (Weight: 0.05)

Rule latency, which relates to the need to minimize delays introduced by security rules while maintaining network efficiency, is assigned a low weight. While important, it is not the primary focus of the evaluation.

5.4.7. Redundancy (Weight: 0.1)

Redundancy is recognized for its importance in ensuring system reliability and resilience within a private cloud context. It is assigned a moderate weight to highlight its role in minimizing service disruption.

The effectiveness metric is determined by assessing the ability of the proposed multi-agent plan recognition (MAPR) approach to accurately detect and mitigate distributed SQL injection attacks. The effectiveness metric's relevance to real-world scenarios lies in its ability to provide a comprehensive assessment of the MAPR approach in a practical context. In a real-world deployment, a high true-positive rate indicates that the MAPR approach is effective in identifying actual distributed SQL injection attacks, minimizing the chances of overlooking genuine threats. A low false-positive rate is crucial to avoid unnecessary alerts and resource wastage. It ensures that the MAPR approach does not raise alarms for benign activities, maintaining the system's credibility. A high precision score indicates that the positive detections made by the MAPR approach are accurate, reducing the likelihood of false alarms and subsequent investigations. A high recall rate signifies that the MAPR approach can successfully capture a significant proportion of actual distributed SQL injection attacks, even in complex and distributed scenarios.

Using these weights, the effectiveness metric (E) is calculated based on the provided formula, offering a comprehensive assessment of the system's performance within the private cloud context. The calculated E score provides valuable insights into the system's effectiveness based on organizational priorities. The formula for the **Effectiveness Metric (E)** is given below:

$$\begin{aligned}\textbf{E(Model)} = (w1 \times \text{Precision}) + (w2 \times \text{Recall}) + (w3 \times \text{F1 Score}) + (w4 \times \text{Throughput}) - (w5 \times \text{Mitigation Time}) - \\ (w6 \times \text{Rule Latency}) + (w7 \times \text{Redundancy})\end{aligned} \quad (1)$$

Step 1: Define the Weights (w1, w2, w3, w4, w5, w6, w7)

We use the same weights mentioned earlier: w1 = 0.3 (weight for precision), w2 = 0.2 (weight for recall), w3 = 0.2 (weight for F1 score), w4 = 0.1 (weight for throughput), w5 = 0.05 (weight for mitigation time), w6 = 0.05 (weight for rule latency), w7 = 0.1 (weight for redundancy).

Step 2: Calculate the Effectiveness Metric (E) for Each Model

Now, we will calculate E for each model separately and then compare them. Table 3 shows the performance metrics of the Omni-Secure Firewall in November 2023. Table 4 shows the effectiveness metric (E) for each model. Figure 14 depicts the performance metrics of the Omni Secure firewall, and Figure 15 shows the effectiveness metrics for each model over time.

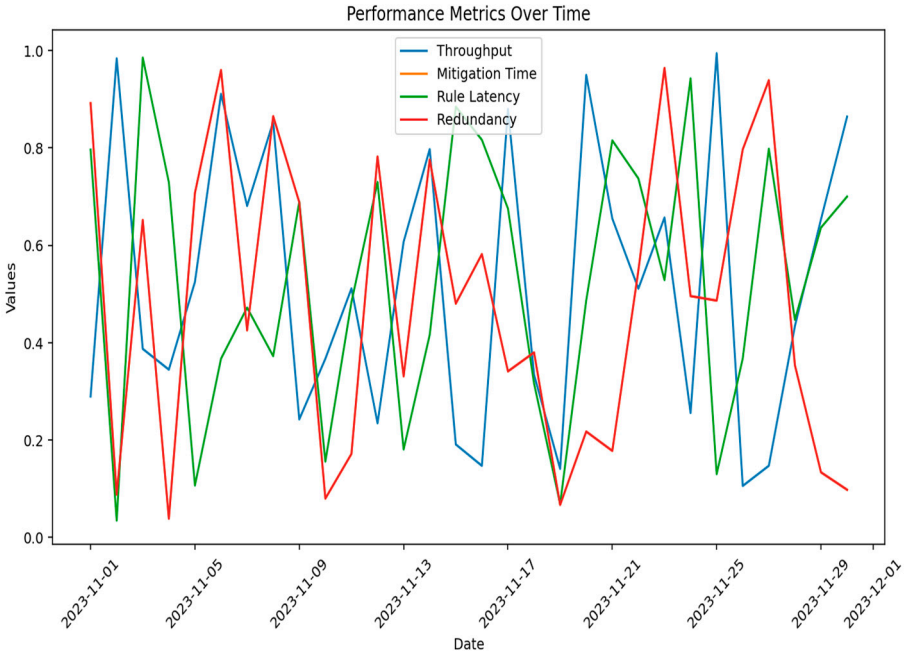

Figure 14. Performance metrics over time.

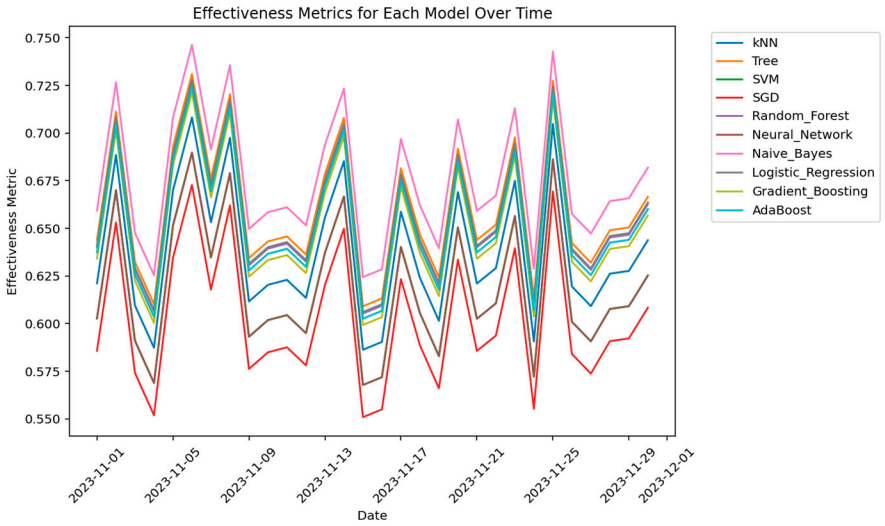

Figure 15. Effectiveness metrics for each model over time.

Table 4. Effectiveness Metric (E) for each model.

Date	kNN	Tree	SVM	SGD	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting	AdaBoost
1 November 2023	0.62110019	0.64390019	0.60260019	0.58570019	0.64010019	0.60260019	0.65920019	0.64090019	0.63410019	0.63740019
2 November 2023	0.688455816	0.711255816	0.669955816	0.653055816	0.707455816	0.669955816	0.726555816	0.708255816	0.701455816	0.704755816
3 November 2023	0.609405173	0.632205173	0.590905173	0.574005173	0.628405173	0.590905173	0.647505173	0.629205173	0.622405173	0.625705173
4 November 2023	0.587264744	0.610064744	0.568764744	0.551864744	0.606264744	0.568764744	0.625364744	0.607064744	0.600264744	0.603564744
5 November 2023	0.66989971	0.69269971	0.65139971	0.63449971	0.68889971	0.65139971	0.70799971	0.68969971	0.68289971	0.68619971
6 November 2023	0.708144673	0.730944673	0.689644673	0.672744673	0.727144673	0.689644673	0.746244673	0.727944673	0.721144673	0.724444673
7 November 2023	0.653160742	0.675960742	0.634660742	0.617760742	0.672160742	0.634660742	0.691260742	0.672960742	0.666160742	0.669460742
8 November 2023	0.6974118	0.7202118	0.6789118	0.6620118	0.7164118	0.6789118	0.7355118	0.7172118	0.7104118	0.7137118
9 November 2023	0.611551967	0.634351967	0.593051967	0.576151967	0.630551967	0.593051967	0.649651967	0.631351967	0.624551967	0.627851967
10 November 2023	0.620275036	0.643075036	0.601775036	0.584875036	0.639275036	0.601775036	0.658375036	0.640075036	0.633275036	0.636575036
11 November 2023	0.622906526	0.645706526	0.604406526	0.587506526	0.641906526	0.604406526	0.661006526	0.642706526	0.635906526	0.639206526
12 November 2023	0.61344069	0.63624069	0.59494069	0.57804069	0.63244069	0.59494069	0.65154069	0.63324069	0.62644069	0.62974069
13 November 2023	0.655618825	0.678418825	0.637118825	0.620218825	0.674618825	0.637118825	0.693718825	0.675418825	0.668618825	0.671918825
14 November 2023	0.685137093	0.707937093	0.666637093	0.649737093	0.704137093	0.666637093	0.723237093	0.704937093	0.698137093	0.701437093
15 November 2023	0.586234595	0.609034595	0.567734595	0.550834595	0.605234595	0.567734595	0.624334595	0.606034595	0.599234595	0.602534595
16 November 2023	0.590328395	0.613128395	0.571828395	0.554928395	0.609328395	0.571828395	0.628428395	0.610128395	0.603328395	0.606628395
17 November 2023	0.658668336	0.681468336	0.640168336	0.623268336	0.677668336	0.640168336	0.696768336	0.678468336	0.671668336	0.674968336
18 November 2023	0.623957939	0.646757939	0.605457939	0.588557939	0.642957939	0.605457939	0.662057939	0.643757939	0.636957939	0.640257939
19 November 2023	0.60137037	0.62417037	0.58287037	0.56597037	0.62037037	0.58287037	0.63947037	0.62117037	0.61437037	0.61767037
20 November 2023	0.668921792	0.691721792	0.650421792	0.633521792	0.687921792	0.650421792	0.707021792	0.688721792	0.681921792	0.685221792
21 November 2023	0.620982866	0.643782866	0.602482866	0.585582866	0.639982866	0.602482866	0.659082866	0.640782866	0.633982866	0.637282866
22 November 2023	0.629054365	0.651854365	0.610554365	0.593654365	0.648054365	0.610554365	0.667154365	0.648854365	0.642054365	0.645354365

Table 4. *Cont.*

Date	kNN	Tree	SVM	SGD	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting	AdaBoost
23 November 2023	0.674859108	0.697659108	0.656359108	0.639459108	0.693859108	0.656359108	0.712959108	0.694659108	0.687859108	0.691159108
24 November 2023	0.590569278	0.613369278	0.572069278	0.555169278	0.609569278	0.572069278	0.628669278	0.610369278	0.603569278	0.606869278
25 November 2023	0.704675091	0.727475091	0.686175091	0.669275091	0.723675091	0.686175091	0.742775091	0.724475091	0.717675091	0.720975091
26 November 2023	0.619422378	0.642222378	0.600922378	0.584022378	0.638422378	0.600922378	0.657522378	0.639222378	0.632422378	0.635722378
27 November 2023	0.609111624	0.631911624	0.590611624	0.573711624	0.628111624	0.590611624	0.647211624	0.628911624	0.622111624	0.625411624
28 November 2023	0.626122478	0.648922478	0.607622478	0.590722478	0.645122478	0.607622478	0.664222478	0.645922478	0.639122478	0.642422478
29 November 2023	0.627579033	0.650379033	0.609079033	0.592179033	0.646579033	0.609079033	0.665679033	0.647379033	0.640579033	0.643879033
30 November 2023	0.643691262	0.666491262	0.625191262	0.608291262	0.662691262	0.625191262	0.681791262	0.663491262	0.656691262	0.659991262

The Omni-Secure Firewall underwent a rigorous 30-day evaluation of multiple effectiveness metrics. While some days exhibited alignment with benchmarks, others revealed performance shortcomings and fluctuations. The naive Bayes model consistently approached or surpassed expected effectiveness levels based on the E metric. Key insights included the need for continuous monitoring and adjustment of cloud security systems due to their dynamic nature. The results emphasized the need to optimize firewall reliability to fully harness the benefits of using a private cloud.

5.5. Machine Learning Model Evaluation

The modular API is designed to seamlessly integrate a diverse array of machine learning models to enhance threat detection within a private cloud environment. The selected models, including random forest [28,29], support vector machines [28,30], neural networks [31,32], k-nearest neighbors [33,34], decision tree [35,36], stochastic gradient descent [37,38], naive Bayes [39,40], logistic regression [41,42], gradient boosting [41,43–45] and AdaBoost [46], each bring unique capabilities to the framework. Random forest's robustness is rigorously assessed for identifying network anomalies, while support vector machines focus on precise threat identification with minimal false positives. Neural networks leverage deep learning for accurate threat recognition, and k-nearest neighbors emphasize privacy-preserving query processing. Decision tree addresses encrypted traffic classification, and stochastic gradient descent plays a role in large-scale linear prediction and optimizing deep models. Naive Bayes finds applications in DDoS vulnerability detection, network intrusion-detection systems, and DDoS attack mitigation. Logistic regression is employed in intrusion detection, identification of vulnerabilities in source code, and privacy-preserving data analysis. Gradient boosting ensures secure and confidential data analysis, while AdaBoost proves effective in malware detection and detection of anomaly intrusions. Together, these integrated models provide a comprehensive and adaptive security solution for various aspects of threat detection within the cloud environment.

In the Machine Learning Model Evaluation stage, we evaluated 10 different machine learning models for their threat-detection capabilities within the private cloud environment, as shown in Table 5 [28–48]. Figure 16 depicts the accuracy of different models as a line graph.

Table 5. Performance Metrics of Machine Learning Models for Threat Detection in the Private Cloud Environment.

Model	Accuracy	Precision	Recall	F1
kNN	0.853	0.830	0.853	0.839
Tree	0.877	0.868	0.877	0.872
SVM	0.876	0.767	0.876	0.818
SGD	0.779	0.796	0.779	0.787
Random forest	0.869	0.864	0.869	0.867
Neural network	0.876	0.767	0.876	0.818
Naive Bayes	0.854	0.933	0.854	0.874
Logistic regression	0.877	0.892	0.877	0.821
Gradient boosting	0.863	0.854	0.863	0.858
AdaBoost	0.867	0.859	0.867	0.863

This evaluation of machine learning models showcased the effectiveness of these models in enhancing the security posture of private cloud networks. Models such as random forest and SVM demonstrated notably high accuracy and balanced precision and recall, making them particularly valuable for strengthening security measures within private cloud environments.

The evaluation of machine learning models showcased the effectiveness of these models in enhancing the security posture of private cloud networks. Notably, models like random

forest and SVM demonstrated high accuracy and balanced precision and recall, making them valuable assets for bolstering security measures in private cloud environments.

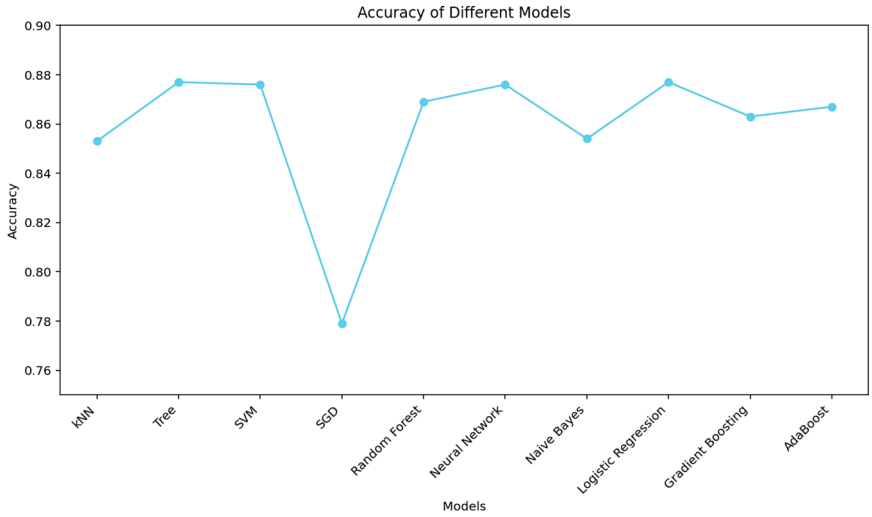

Figure 16. Accuracy of different models.

With an accuracy of 0.853, kNN demonstrated a strong ability to accurately classify network events within the private cloud. It achieved a precision score of 0.830, highlighting its accuracy in identifying attacks while minimizing false positives. Moreover, kNN showed a recall score of 0.853, indicating its effectiveness in capturing relevant instances of attacks. The F1 score of 0.839 emphasized the model's capacity to strike a balance between precision and recall.

The Tree model achieved an accuracy of 0.877, underlining its proficiency in accurately classifying network events. With a precision score of 0.868, it excelled in identifying true attack cases while keeping false alarms to a minimum. The recall score of 0.877 highlighted its effectiveness in capturing relevant attack instances. The F1 score of 0.872 demonstrated a remarkable balance between precision and recall.

SVM exhibited strong overall performance with an accuracy of 0.876. It achieved a commendable precision score of 0.767, indicating its accuracy in identifying genuine attacks while maintaining a balance with false positives. The model's recall score of 0.876 showcased its effectiveness in capturing a substantial portion of actual attack cases. The F1 score of 0.818 underlined its ability to accurately classify network events.

While achieving an accuracy of 0.854, the naive Bayes model demonstrated a precision score of 0.933, excelling in identifying true attacks but potentially leading to more false alarms. Its recall score of 0.854 indicated moderate effectiveness in identifying actual attack cases. The F1 score of 0.874 reflected the trade-off between precision and recall, implying that the model may not perform as well in capturing true attacks compared to others.

The random forest model excelled with an accuracy of 0.869, indicating its proficiency in accurately classifying network events as normal or attacks. It achieved a high precision score of 0.864, signifying its ability to identify true attack cases while minimizing false alarms. The recall score of 0.869 showcased its effectiveness in capturing relevant attack instances. The model achieved an F1 score of 0.867, reinforcing its capacity to classify network events with precision and recall in balance.

The neural network model demonstrated an accuracy of 0.876, on par with other high-performing models. It achieved a precision score of 0.767, emphasizing its accuracy in identifying attacks. Its recall score of 0.876 showcased its effectiveness in capturing relevant attack instances. The F1 score of 0.818 demonstrated its capacity to balance precision and recall.

Logistic regression exhibited good network event classification performance with an accuracy of 0.877. It achieved a precision score of 0.892, denoting reasonable accuracy in

identifying true attacks while allowing some margin for false alarms. The model's recall score of 0.877 highlighted its effectiveness in identifying genuine attack cases. The F1 score of 0.821 indicated a balanced trade-off between precision and recall.

The gradient boosting model achieved an accuracy of 0.863, with a precision score of 0.854. It demonstrated effectiveness in accurately identifying attacks while maintaining balanced precision and recall. The recall score of 0.863 showcased its capacity to capture relevant attack instances. The F1 score of 0.858 emphasized its overall performance in terms of threat identification and mitigation.

With an accuracy of 0.867, AdaBoost demonstrated proficiency in accurately classifying network events. It achieved a precision score of 0.859, signifying its ability to identify true attacks while minimizing false alarms. The recall score of 0.867 indicated its effectiveness in capturing relevant attack instances. The F1 score of 0.863 reinforced its capacity to balance precision and recall.

Random forest and SVM demonstrated notably high accuracy and balanced precision and recall, making them particularly valuable for bolstering security measures within private cloud environments. kNN showed strong accuracy, precision, recall, and an effective balance between them. Naive Bayes displayed high precision but with a potential trade-off of more false alarms. Overall, each model contributes unique capabilities, and their selection should align with specific security requirements within the private cloud environment.

6. Conclusions

In conclusion, this study has provided valuable insights into the optimization of firewall systems for private cloud environments, as evidenced by a comprehensive 30-day evaluation of the Omni-Secure Firewall. The findings underscore the necessity of adopting a multi-metric approach, incorporating effectiveness metrics (E) that weigh factors such as precision, recall, and redundancy when assessing security systems.

While the firewall exhibited promising potential, its performance displayed variations across different machine learning models during the evaluation period, indicating a need for optimization to ensure consistent security delivery. The modular API implemented facilitates the integration of diverse threat-detection models, with the evaluation highlighting the consistent high performance of models like Naive Bayes. This finding emphasizes the importance of selecting models tailored to the intricacies of private cloud networks.

The study's tracking of multiple metrics over time establishes a framework for holistic security assessment in private clouds, guided by the introduced effectiveness metric. This metric can inform decisions to enhance security posture based on organizational priorities. The dynamic results underscore the imperative for continuous monitoring and adjustment in cloud security, emphasizing the necessity for robust and reliable firewall systems to fully capitalize on the benefits of private clouds while safeguarding sensitive data and applications.

However, the study has its limitations, including the need for real-world validation of simulation-based evaluations, a narrow focus on firewall systems, and the subjective nature of organizational weights in the effectiveness metric. To address these limitations and pave the way for future research, several avenues can be explored. These include testing the firewall in real private cloud settings with live traffic and attacks, developing custom machine learning algorithms for private cloud threats, conducting cost-benefit analyses for different organizations, utilizing larger datasets for evaluations of detection accuracy, exploring deep learning predictive analytics for anomalies and zero-days, implementing intelligent automation for optimizing firewall policies/configurations, and investigating custom extensions to cater to the unique needs of private clouds, including the exploration of unsupervised learning techniques. This comprehensive future work will contribute to advancing the understanding and implementation of effective security measures in private cloud environments.

Author Contributions: Conceptualization, S.M., N.A.Y., S.H. and M.H.; methodology, S.M.; software, S.M.; validation, S.M., N.A.Y., R.H., S.H. and M.H.; formal analysis, S.M.; investigation, S.M.; resources, N.A.Y.; data curation, S.M.; writing—original draft preparation, S.M.; writing—review and editing, S.M., N.A.Y., R.H., S.H. and M.H.; visualization, S.M.; supervision, N.A.Y.; project administration, S.M.; funding acquisition, N.A.Y. and R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in https://doi.org/10.5281/zenodo.10492770 (accessed on 1 January 2024).

Acknowledgments: The authors would like to thank Malaysia University of Science and Technology, Malysia. Furthermore, the authors acknowledge the Department of Computer Science at Solent University, Southampton. The authors also appreciate the support from the Department of Computer Science and Creative Technology at Global College of Engineering and Technology. The collaborative efforts of these academic institutions and individuals have enriched the quality of this research. Additionally, the authors would like to acknowledge the use of ChatGPT 24 May 2023 version (OpenAI, San Francisco, CA, USA), specifically to assist in some content for improved clarity and effectiveness.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jabbar, A.A.; Bhaya, W.S. Security of Private Cloud Using Machine Learning and Cryptography. *Bull. Electr. Eng. Inform.* **2023**, *12*, 561–569. [CrossRef]
2. Qureshi, A.; Dashti, W.; Jahangeer, A.; Zafar, A. Security Challenges over Cloud Environment from Service Provider Prospective. *Cloud Comput. Data Sci.* **2020**, *12*, 12–20. [CrossRef]
3. Kumar, K.D.; Umamaheswari, E. An Authenticated, Secure Virtualization Management System in Cloud Computing. *Asian J. Pharm. Clin. Res.* **2017**, *10*, 45. [CrossRef]
4. Ahmadi, S.; Salehfar, M. Privacy-Preserving Cloud Computing: Ecosystem, Life Cycle, Layered Architecture and Future Roadmap. *arXiv* **2022**, arXiv:2204.11120.
5. Khaleel, T.A. Analysis and Implementation of Kerberos Protocol in Hybrid Cloud Computing Environments. *Eng. Technol. J.* **2021**, *39*, 41–52. [CrossRef]
6. Borse, Y.; Gokhale, S. Cloud Computing Platform for Education System: A Review. *Int. J. Comput. Appl.* **2019**, *177*, 41–45. [CrossRef]
7. Hong, J.B.; Nhlabatsi, A.; Kim, D.S.; Hussein, A.; Fetais, N.; Khan, K.M. Systematic Identification of Threats in the Cloud: A Survey. *Comput. Netw.* **2019**, *150*, 46–69. [CrossRef]
8. Li, Z.; Jin, H.; Zou, D.; Yuan, B. Exploring New Opportunities to Defeat Low-Rate DDoS Attack in Container-Based Cloud Environment. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 695–706. [CrossRef]
9. Shah, H.; ud Din, A.; Khan, A.; Din, S. Enhancing the Quality of Service of Cloud Computing in Big Data Using Virtual Private Network and Firewall in Dense Mode. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 10351. [CrossRef]
10. Ahmad, W.; Rasool, A.; Javed, A.R.; Baker, T.; Jalil, Z. Cyber Security in IoT-Based Cloud Computing: A Comprehensive Survey. *Electronics* **2021**, *11*, 16. [CrossRef]
11. Adee, R.; Mouratidis, H. A Dynamic Four-Step Data Security Model for Data in Cloud Computing Based on Cryptography and Steganography. *Sensors* **2022**, *22*, 1109. [CrossRef] [PubMed]
12. Wang, Q.; Tai, W.; Tang, Y.; Zhu, H.; Zhang, M.; Zhou, D. Coordinated Defense of Distributed Denial of Service Attacks against the Multi-Area Load Frequency Control Services. *Energies* **2019**, *12*, 2493. [CrossRef]
13. Anwar, R.W.; Abdullah, T.; Pastore, F. Firewall Best Practices for Securing Smart Healthcare Environment: A Review. *Appl. Sci.* **2021**, *11*, 9183. [CrossRef]
14. Pandeeswari, N.; Kumar, G. Anomaly Detection System in Cloud Environment Using Fuzzy Clustering Based ANN. *Mob. Netw. Appl.* **2016**, *21*, 494–505. [CrossRef]
15. Alghofaili, Y.; Albattah, A.; Alrajeh, N.; Rassam, M.A.; Al-rimy, B.A.S. Secure Cloud Infrastructure: A Survey on Issues, Current Solutions, and Open Challenges. *Appl. Sci.* **2021**, *11*, 9005. [CrossRef]
16. Abu Al-Haija, Q.; Ishtaiwi, A. Machine Learning Based Model to Identify Firewall Decisions to Improve Cyber-Defense. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2021**, *11*, 1688. [CrossRef]
17. Kharchenko, V.; Ponochovnyi, Y.; Ivanchenko, O.; Fesenko, H.; Illiashenko, O. Combining Markov and Semi-Markov Modelling for Assessing Availability and Cybersecurity of Cloud and IoT Systems. *Cryptography* **2022**, *6*, 44. [CrossRef]

18. Lin, H.-Y. A Secure Heterogeneous Mobile Authentication and Key Agreement Scheme for E-Healthcare Cloud Systems. *PLoS ONE* **2018**, *13*, e0208397. [CrossRef] [PubMed]
19. Wijaya, G.; Surantha, N. Multi-Layered Security Design and Evaluation for Cloud-Based Web Application: Case Study of Human Resource Management System. *Adv. Sci. Technol. Eng. Syst. J.* **2020**, *5*, 674–679. [CrossRef]
20. Shahsavari, Y.; Shahhoseini, H.; Zhang, K.; Elbiaze, H. A Theoretical Model for Analysis of Firewalls Under Bursty Traffic Flows. *IEEE Access* **2019**, *7*, 183311–183321. [CrossRef]
21. Sharma, B.; Sharma, L.; Lal, C.; Roy, S. Anomaly Based Network Intrusion Detection for IoT Attacks Using Deep Learning Technique. *Comput. Electr. Eng.* **2023**, *107*, 108626. [CrossRef]
22. Mozo, A.; Karamchandani, A.; de la Cal, L.; Gómez-Canaval, S.; Pastor, A.; Gifre, L. A Machine-Learning-Based Cyberattack Detector for a Cloud-Based SDN Controller. *Appl. Sci.* **2023**, *13*, 4914. [CrossRef]
23. Tiwari, G.; Jain, R. Detecting and Classifying Incoming Traffic in a Secure Cloud Computing Environment Using Machine Learning and Deep Learning System. In Proceedings of the 2022 IEEE 7th International Conference on Smart Cloud (SmartCloud), Shanghai, China, 8–10 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 16–21.
24. Alshaikh, A.; Alanesi, M.; Yang, D.; Alshaikh, A. Advanced Techniques for Cyber Threat Intelligence-Based APT Detection and Mitigation in Cloud Environments. In Proceedings of the International Conference on Cyber Security, Artificial Intelligence, and Digital Economy (CSAIDE 2023), Nanjing, China, 1 June 2023; Loskot, P., Niu, S., Eds.; SPIE: Bellingham, WA, USA, 2023; p. 65.
25. Alshaer, H. An Overview of Network Virtualization and Cloud Network as a Service. *Int. J. Netw. Manag.* **2015**, *25*, 1–30. [CrossRef]
26. Mahmood, S.; Yahaya, N.A. Exploring Virtual Machine Scheduling Algorithms: A Meta-Analysis. *Sir. Syed Univ. Res. J. Eng. Technol.* **2023**, *13*, 89–100. [CrossRef]
27. Mahmood, S.; Yahaya, N.A.; Hasan, R.; Hussain, S.; Malik, M.H.; Sarker, K.U. Self-Adapting Security Monitoring in Eucalyptus Cloud Environment. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 140310. [CrossRef]
28. Panker, T.; Nissim, N. Leveraging Malicious Behavior Traces from Volatile Memory Using Machine Learning Methods for Trusted Unknown Malware Detection in Linux Cloud Environments. *Knowl. Based Syst.* **2021**, *226*, 107095. [CrossRef]
29. Kim, H.; Kim, J.; Kim, Y.; Kim, I.; Kim, K.J. Design of Network Threat Detection and Classification Based on Machine Learning on Cloud Computing. *Clust. Comput.* **2019**, *22*, 2341–2350. [CrossRef]
30. Sharma, V.; Verma, V.; Sharma, A. Detection of DDoS Attacks Using Machine Learning in Cloud Computing. In *Advanced Informatics for Computing Research: Third International Conference, ICAICR 2019, Shimla, India, 15–16 June 2019*; Springer: Singapore, 2019; pp. 260–273.
31. Gao, X.; Hu, C.; Shan, C.; Liu, B.; Niu, Z.; Xie, H. Malware Classification for the Cloud via Semi-Supervised Transfer Learning. *J. Inf. Secur. Appl.* **2020**, *55*, 102661. [CrossRef]
32. Landman, T.; Nissim, N. Deep-Hook: A Trusted Deep Learning-Based Framework for Unknown Malware Detection and Classification in Linux Cloud Environments. *Neural Netw.* **2021**, *144*, 648–685. [CrossRef]
33. Nadeem, M.; Arshad, A.; Riaz, S.; Zahra, S.; Rashid, M.; Band, S.S.; Mosavi, A. Preventing Cloud Network from Spamming Attacks Using Cloudflare and KNN. *Comput. Mater. Contin.* **2023**, *74*, 2641–2659. [CrossRef]
34. Muthulakshmi, K.; Valarmathi, K. Attaining Cloud Security Solution Over Machine Learning Techniques. *Smart Intell. Comput. Communiation Technol.* **2021**, *38*, 246.
35. Agafonov, A.; Yumaganov, A. Performance Comparison of Machine Learning Methods in the Bus Arrival Time Prediction Problem. In Proceedings of the V International Conference Information Technology and Nanotechnology 2019, Samara, Russia, 21–24 May 2019; CEUR-WS: Aachen, Germany, 2020; pp. 57–62.
36. Liu, L.; Su, J.; Zhao, B.; Wang, Q.; Chen, J.; Luo, Y. Towards an Efficient Privacy-Preserving Decision Tree Evaluation Service in the Internet of Things. *Symmetry* **2020**, *12*, 103. [CrossRef]
37. Gonzales, D.; Kaplan, J.M.; Saltzman, E.; Winkelman, Z.; Woods, D. Cloud-Trust-a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds. *IEEE Trans. Cloud Comput.* **2017**, *5*, 523–536. [CrossRef]
38. Bhamare, D.; Salman, T.; Samaka, M.; Erbad, A.; Jain, R. Feasibility of Supervised Machine Learning for Cloud Security. In Proceedings of the 2016 International Conference on Information Science and Security (ICISS), Jaipur, India, 19–22 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–5.
39. Zhang, T. Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms. In Proceedings of the Twenty-First International Conference on Machine Learning—ICML'04, New York, NY, USA, 4 July 2004; ACM Press: New York, NY, USA, 2004; p. 116.
40. Galanti, T.; Siegel, Z.S.; Gupte, A.; Poggio, T. Characterizing the Implicit Bias of Regularized SGD in Rank Minimization. *arXiv* **2022**, arXiv:2206.05794.
41. Amjad, A.; Alyas, T.; Farooq, U.; Tariq, M. Detection and Mitigation of DDoS Attack in Cloud Computing Using Machine Learning Algorithm. *ICST Trans. Scalable Inf. Syst.* **2018**, *11*, 159834. [CrossRef]
42. Yu, X.; Zhao, W.; Huang, Y.; Ren, J.; Tang, D. Privacy-Preserving Outsourced Logistic Regression on Encrypted Data from Homomorphic Encryption. *Secur. Commun. Netw.* **2022**, *2022*, 1321198. [CrossRef]
43. Mishra, N.; Singh, R.K.; Yadav, S.K. Detection of DDoS Vulnerability in Cloud Computing Using the Perplexed Bayes Classifier. *Comput. Intell. Neurosci.* **2022**, *2022*, 9151847. [CrossRef] [PubMed]

44. Mahmood, H.A. Network Intrusion Detection System (NIDS) in Cloud Environment Based on Hidden Naïve Bayes Multiclass Classifier. *Al-Mustansiriyah J. Sci.* **2018**, *28*, 134–142. [CrossRef]
45. Edemacu, K.; Kim, J.W. Scalable Multi-Party Privacy-Preserving Gradient Tree Boosting over Vertically Partitioned Dataset with Outsourced Computations. *Mathematics* **2022**, *10*, 2185. [CrossRef]
46. Guo, W.; Luo, Z.; Chen, H.; Hang, F.; Zhang, J.; Al Bayatti, H. AdaBoost Algorithm in Trustworthy Network for Anomaly Intrusion Detection. *Appl. Math. Nonlinear Sci.* **2023**, *8*, 1819–1830. [CrossRef]
47. Akter, M.S.; Shahriar, H.; Bhuiya, Z.A. *Automated Vulnerability Detection in Source Code Using Quantum Natural Language Processing*; Springer Nature Singapore: Singapore, 2023; pp. 83–102.
48. Bhamare, D.; Zolanvari, M.; Erbad, A.; Jain, R.; Khan, K.; Meskin, N. Cybersecurity for Industrial Control Systems: A Survey. *Comput. Secur.* **2020**, *89*, 101677. [CrossRef]