electronics

MDPI

Article

AIDS-Based Cyber Threat Detection Framework for Secure Cloud-Native Microservices

Heeji Park, Abir EL Azzaoui and Jong Hyuk Park *

Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea; heeji@seoultech.ac.kr (H.P.); abir.el@seoultech.ac.kr (A.E.A.)
* Correspondence: jhpark1@seoultech.ac.kr; Tel.: +82-2-970-6702

Abstract: Cloud-native architectures continue to redefine application development and deployment by offering enhanced scalability, performance, and resource efficiency. However, they present significant security challenges, particularly in securing inter-container communication and mitigating Distributed Denial of Service (DDoS) attacks in containerized microservices. This study proposes an Artificial Intelligence Intrusion Detection System (AIDS)-based cyber threat detection solution to address these critical security challenges inherent in cloud-native environments. By leveraging a Resilient Backpropagation Neural Network (RBN), the proposed solution enhances system security and resilience by effectively detecting and mitigating DDoS attacks in real time in both the network and application layers. The solution incorporates an Inter-Container Communication Bridge (ICCB) to ensure secure communication between containers. It also employs advanced technologies such as eXpress Data Path (XDP) and the Extended Berkeley Packet Filter (eBPF) for high-performance and low-latency security enforcement, thereby overcoming the limitations of existing research. This approach provides robust protection against evolving security threats while maintaining the dynamic scalability and efficiency of cloud-native architectures. Furthermore, the system enhances operational continuity through proactive monitoring and dynamic adaptability, ensuring effective protection against evolving threats while preserving the inherent scalability and efficiency of cloud-native environments.

Keywords: cloud-native; AI-based intrusion detection system; Resilient Backpropagation Neural Network; security

Academic Editor: Aryya Gangopadhyay

Received: 30 October 2024
Revised: 30 December 2024
Accepted: 31 December 2024
Published: 8 January 2025

Citation: Park, H.; EL Azzaoui, A.; Park, J.H. AIDS-Based Cyber Threat Detection Framework for Secure Cloud-Native Microservices. *Electronics* **2025**, *14*, 229. https://doi.org/10.3390/electronics14020229

1. Introduction

In the context of Information and Communication Technology (ICT), the adoption of cloud computing has been driven by several key factors. Cloud computing offers an innovative computing model that simplifies IT resource management while providing cost savings and scalable solutions for both private and public sectors [1]. Additionally, cloud computing allows organizations to reduce their IT-related costs and operational expenses while gaining competitive advantages, including energy savings and improved efficiency [2]. As cloud computing evolves, it continues to integrate various Information and Communication Technologies (ICTs), including fog computing, edge computing, containers, and virtual machines, to enhance its effectiveness [3,4]. Fog and edge computing distribute computational power closer to the location of the data, which reduces latency and improves response times. In addition, through the use of containers and virtual machines, cloud computing offers more cost-effective and reliable environments for application deployment. Containers provide a flexible execution context for applications, while virtual machines

host fully functional guest operating systems on emulated hardware, thereby increasing the portability and versatility of cloud services. These technological integrations illustrate that cloud computing has shifted away from a monolithic, centralized, and scalable model toward a more distributed, intelligent, and cognitive computing paradigm [5]. The emergence of cloud-native environments, despite the existence of traditional cloud environments, can be attributed to several key factors. First, cloud-native applications are designed to leverage microservice architecture, allowing for efficient resource distribution and flexibility. This adaptability to cloud architecture enables improved scalability, operational efficiency, and enhanced customer service within organizations [6]. Second, cloud-native technology emerged from the need to effectively utilize cloud computing infrastructure and maximize its capacity, driven by advancements in virtualization and cloud technologies [7]. Finally, while traditional cloud environments have predominantly focused on centralized models, cloud-native approaches introduce distributed structures and enhanced flexibility, leading to significant improvements in performance and scalability [8,9].

Cloud-native technology represents a new paradigm in the design and deployment of applications within cloud environments, optimizing efficiency, scalability, availability, and performance by leveraging the inherent characteristics of cloud architecture [10]. And cloud-native services represent a sophisticated amalgamation of ideas and methodologies aimed at creating highly portable, scalable, and robust applications. These services utilize containers, where each application component and its dependencies are isolated in separate containers, thereby enabling optimal resource utilization, rapid deployment, and consistent application behavior across diverse environments [11,12].

Cloud-native services are a transformative wave in software development and deployment, underpinned by the DevOps methodology, which emphasizes automation and close collaboration between development and operations teams. To ensure software quality and the rapid deployment of new features with minimal points of failure, modern development practices such as Continuous Deployment (CD) and Continuous Integration (CI) are employed. These practices facilitate the seamless integration, testing, and implementation of code changes. Additionally, Infrastructure as Code (IaC) enables the provisioning and management of infrastructure resources through code, ensuring regularity and consistency. Microservice architectures are integral to cloud-native services, involving the decomposition of applications into smaller, independently deployable services. Each microservice is designed, deployed, scaled, and updated independently, with a focus on specific business functions. This architecture's flexibility, agility, and scalability allow organizations to rapidly introduce new products and adapt to changing customer demands [13,14].

In the context of containers, it is essential to highlight the role of cloud-native orchestration tools, such as Kubernetes, which manage and automate the deployment, scaling, and maintenance of containers. Kubernetes enhances application performance by integrating features like load balancing, auto-healing, and service discovery. Cloud-native services are optimized for performance, extensibility, and durability, utilizing auto-scaling tools to dynamically adjust resource allocation based on demand. Load balancers distribute incoming traffic across multiple service instances, ensuring high resource utilization and reliability [15]. Given the necessity of storing large volumes of data with high accessibility and functional efficiency, cloud-native environments employ distributed storage and caching mechanisms. Monitoring and observability are also critical for cloud-native services to ensure optimal resource utilization and early detection of issues [16]. Commonly used monitoring tools collect metrics, logs, and traces from various components of the cloud-native environment, providing visibility into its behavior, health, and intended functionality. Observability aids in troubleshooting, performance optimization, and pre-incident health checks, highlighting the distributed nature of cloud-native architectures.

Security in cloud-native applications involves addressing the unique challenges associated with services and applications built on cloud-native architectures. Key security concerns include securing service meshes, containers, and dynamic deployments [17]. Continuing research is essential to secure cloud-native applications in real-world settings, as technology and cyber threats are constantly evolving and dynamic. Built on microservices, containers, and orchestration platforms, cloud-native applications present new difficulties that necessitate ongoing research and development in the field of cybersecurity. To keep ahead of new risks, one must constantly engage in research due to the quick pace of technological improvements.

Given the evolving nature of cyber threats, particularly in cloud-native ecosystems, malicious actors often exploit the dynamic and distributed characteristics of these environments to launch sophisticated attacks, such as Distributed Denial of Service (DDoS) and lateral movement attacks. Addressing these threats requires not only traditional security measures but also adaptive and intelligent frameworks capable of real-time threat detection and mitigation, as outlined in this study [18].

Modern technologies like containerization and orchestration tools like Kubernetes are used in cloud-native systems [19]. New ways to attack vulnerabilities may appear as these technologies grow, necessitating the analysis, comprehension, and development of alternatives by researchers to reduce potential threats. In addition, customized security solutions are necessary due to the dispersed and elastic nature of cloud-native systems. Applications capacity for dynamic scalability, frequently spanning many cloud providers and geographical locations, adds complexity to the management of identities, access restrictions, and inter-service communication security [20,21].

To provide effective security solutions that consider the unique characteristics of cloud-native applications, researchers must dive into these complexities. Moreover, cloud-native ecosystems' reliance on open-source components and the integration of third-party services adds more layers of complexity and possible security gaps. To assess the state of security of these external services, comprehend their possible influence on the application's overall security, and put strong security measures in place in a real-world setting, ongoing study is needed [22].

The proposed framework provides significant advancements over traditional security solutions by addressing the unique challenges of cloud-native environments. Traditional solutions often rely on static, rule-based approaches, which lack the adaptability required to respond to evolving threats in real time. Furthermore, these solutions typically focus on DDoS detection at the network layer, with limited capability to address sophisticated attacks targeting the application layer. The inability to adapt to the dynamic and distributed nature of cloud-native systems further limits their effectiveness in modern, scalable environments. Additionally, traditional mechanisms often introduce significant resource overheads and latency, making them suboptimal for high-performance cloud-native architectures.

In contrast, the AIDS-based framework introduces several key contributions that enhance security and efficiency. By leveraging a Resilient Backpropagation Neural Network (RBN), the framework enables real-time network traffic analysis and adaptive threat detection. This capability allows the system to distinguish between benign and malicious traffic patterns effectively, providing a proactive defense against evolving attack vectors. Furthermore, the integration of advanced technologies such as eXpress Data Path (XDP) and extended Berkeley Packet Filter (eBPF) ensures low-latency, high-performance security enforcement, addressing the performance limitations of traditional solutions.

The framework also incorporates a multi-layered defense mechanism that secures both the network and application layers, enabling comprehensive protection against DDoS attacks and other security threats. Its adaptability and scalability make it particularly suited

for the distributed and dynamic characteristics of cloud-native environments, overcoming the compatibility and performance challenges faced by legacy systems. Additionally, the proposed framework enhances inter-container communication security and addresses challenges such as dynamic IP changes and inter-service communication vulnerabilities, which are critical in containerized environments.

By addressing these limitations and introducing an intelligent, adaptive, and high-performance security solution, the AIDS-based framework represents a significant advancement over traditional approaches. It provides a robust, scalable, and efficient mechanism for ensuring security in dynamic and distributed cloud-native systems, making it a promising solution for emerging cybersecurity threats.

The rest of the paper is organized as follows: Section 2 provides a detailed review of the existing literature and highlights the technical distinctions between traditional cloud environments and cloud-native environments. Section 3 presents the proposed framework for securing containerized cloud-native microservices using an intelligent Intrusion Detection System (IDS). The conclusion of this paper is presented in Section 4.

2. Related Works

This section examines the technical distinctions between cloud and cloud-native environments, as well as the various security challenges and proposed solutions, drawing from a comprehensive review of state-of-the-art publications and academic research.

2.1. Cloud-Native Environments

Cloud-native technology represents a modern approach to building and running applications that fully exploit the advantages of cloud computing. This method maximizes the efficiency, scalability, performance, and availability of applications by utilizing the built-in features of cloud infrastructure [23]. Fundamentally, cloud-native services are a collection of ideas and procedures that facilitate the development of extremely scalable, portable, and resilient applications.

At its core, cloud-native services embody a sophisticated of concepts and methodologies designed to foster the creation of applications that are exceptionally scalable, portable, and robust [24]. The foundation of these services is containerization, which isolates and lightweights' individual application components along with their dependencies inside of separate containers. This containerization makes it possible to use resources more effectively, deploy applications more quickly, and guarantee consistent behavior in various contexts. These services represent a new paradigm in the creation and implementation of software that is firmly based on the use of DevOps principles, a methodical approach that emphasizes automation and collaboration between both operations and development teams [25].

To guarantee the quality of software and quick functionality deployment with fewer vulnerabilities, DevOps approaches like Continuous Deployment and Continuous Integration (CD/CI) are essential in optimizing the integration, examining, and implementation of code changes. Moreover, the fundamental element of DevOps, Infrastructure as Code (IaC) facilitates the delivery and administration of infrastructure resources via code, fostering regularity and consistency. From a scientific perspective, this method unites automation, version control, and infrastructure management to promote a more effective, dependable, reliable, and flexible software development and implementation process in the ever-changing world of cloud-native services.

Microservice architectures are used to build cloud-native services, which include breaking down programs into smaller, freely connected, self-contained deployable services. Every service may be independently designed, deployed, scaled, and updated, and each

one focuses on a certain business function [13]. Because of the flexibility, agility, and scalability that this architecture fosters, businesses can quickly roll out new products and adjust to shifting customer needs. Applications and their dependencies are packaged and isolated using containers in microservice architectures. Consistency across many computing environments is ensured via containers, which offer a lightweight and portable runtime environment. They make it simple to scale, deploy, and maintain apps, which enables businesses to make effective use of their resources and launch quickly [26].

In a cloud-native context, orchestration tools like Kubernetes are essential for controlling and automating the setting up, scaling, and maintenance of containers. They ensure that applications function efficiently and dependably by offering capabilities like balancing loads, self-repairing, and service exploration. Developers may concentrate on creating and implementing applications rather than worrying about the intricacies of the infrastructure through orchestration platforms, which take away the underlying infrastructure problems [19].

The scalability and robustness of cloud infrastructures are employed by cloud-native services [27]. To ensure optimum performance and cost-efficiency, they make use of autoscaling features to dynamically alter resources based on workload needs. To provide high availability and fault tolerance, load balancers split up incoming traffic among many service instances. To manage big data quantities and offer quick and effective access, dispersed storage of data and caching techniques are used.

Strong monitoring and observability skills are also necessary for cloud-native services to guarantee peak performance and identify problems. Monitoring tools provide information about the behavior, health, and functionality of services by gathering and analyzing methods, records, and traces from numerous cloud-native environment aspects. Troubleshooting, performance improvement, and proactive application management are made easier by observability. One characteristic of such architectural paradigms is their excessive distribution.

2.2. Security Issues in Cloud-Native Environments

Significant benefits including scalability, flexibility, and cost-efficiency are provided by cloud-native services. However, because there are so many inherent issues, building safe cloud-native applications is exceedingly challenging. Sensitive data processing and storage pose extra cybersecurity issues within the environment of cloud-native applications. Additionally, maintaining trust, adhering to privacy laws, and averting any legal, financial, and reputational repercussions all depend on protecting user details and confidential data. Furthermore, security in cloud-native applications refers to security issues with services and apps that are created and developed to make use of cloud-native architectures' features. The usage of serverless computing, microservices, containerization, and orchestration tools like Kubernetes define these services [28].

Specifically, security in cloud-native applications takes care of certain issues and demands like service mesh security, container security, and dynamic deployments that come with using cloud-native architectures. It includes the security of every cloud-native module, how those modules interact with one another, and the general security of the cloud-native ecosystem. To be more precise, every attempt to develop issues in cloud-native applications must include the following, in addition to some elements also critical to cloud security: management accessibility, data safety, security surveillance, and handling incidents.

- Containerization and Microservice Security: This entails protecting the cloud-native applications' containerized elements, including Docker containers. To reduce specific container threats like corrupted containerized images or containerized escape assaults, it incorporates techniques like runtime protection, secure container image

management, and vulnerability scanning. However, access control and authorization must be enhanced to eliminate any unauthorized access to the cloud-native data base. Moreover, the security of individual microservices and their interconnections is considered in cloud-native service security. This entails protecting the routes of communication between microservices, putting in place authentication and access restrictions for inter-service communication, and making sure that data are properly authorized and protected between microservices [29,30].

- Orchestration and Automation Security: The orchestration technologies used in cloud-native settings, like Kubernetes, are also subject to security constraints. This includes protecting the crucial components like the stored data, securing the Kubernetes control plane, and putting secure configuration procedures into place. To stop unwanted access or changes to deployments, it also entails protecting container orchestration and deployment procedures [31].
- DevSecOps Procedures: The incorporation of security principles across the creation, placement, and operating lifecycle is embraced by security issues in cloud-native applications. This highlights how security should be incorporated in the developing phase and how the DevSecOps method should merge automated security checks, ongoing security inspection, and security surveillance [32].

Containers and microservices often scale up or down depending on demand, which is an attribute of cloud-native settings. It is difficult to keep a stable and safe state because of its dynamic tendency. Cloud-native settings change quickly, and traditional security solutions intended for static systems might not be able to keep up, which could result in runtime vulnerabilities. Applications running in containers may be vulnerable to security risks due to untrusted host OS vulnerabilities [33].

The host OS kernel is shared by all containers; therefore, any weakness in the kernel may affect the security of every container operating on that host. The security of containers and the apps they host might be compromised by attackers using flaws in the underlying host operating system if appropriate runtime protections are not in place. Due to the distinct features of containerized apps, traditional malware detection techniques might not work well in cloud-native environments. Cybercriminals may use complex software, such as cryptocurrency mining malware, that is difficult for conventional detection techniques to find in a cloud-native setting. The system, including all microservices and containerization, operating on the host, is in danger of being compromised by kernel vulnerabilities. A hacked kernel puts the integrity of the whole cloud-native system at risk by allowing for illegal access, data breaches, or other security events [34,35].

Preventing unwanted access and data breaches requires ensuring secure communication between microservices and containers. Attackers may be able to intercept or alter communication between microservices in the absence of appropriate security controls, which might result in security breaches [36].

In the context of cloud-native environments, runtime security solutions such as Falco and AppArmor play a pivotal role in enhancing system security. Falco, a cloud-native runtime security tool, operates on Linux operating systems to allow for the real-time monitoring of system calls and other events, enabling the detection of anomalous behaviors and potential security threats through customizable rule-based mechanisms [37]. Similarly, AppArmor, a Linux security module, enforces granular access control policies for individual applications, thereby restricting their permissible actions and effectively reducing the system's attack surface [38].

Distributed Denial of Service (DDoS) attacks can affect the responsiveness and availability of cloud containers. DDoS assaults have the potential to overwhelm containerized applications, resulting in downtime and service interruption. To avoid unwanted access

and data leakage, it is crucial to ensure secure-access control methods for microservice communication. The security and integrity of data may be compromised by unapproved interactions between microservices caused by insufficient access control. Attackers may find centralized access control systems appealing due to their sensitivity to single points of failure. Widespread unauthorized access across microservices might occur from a breach in a centralized access control system.

2.3. Key Considerations

In cloud-native environments, security concerns revolve around four key pillars: Confidentiality, Integrity, Availability, and Privacy. These elements are essential for ensuring the secure operation of distributed systems, such as microservices and container-based architectures. Failure to adequately address any of these components can significantly compromise the security and reliability of the entire system.

- Integrity ensures that data remains unaltered and reliable during storage or transmission [39]. In distributed cloud-native environments, safeguarding data integrity is a critical challenge. Data validation should be rigorously enforced at service boundaries to prevent malicious or malformed data from entering the system [40]. Cryptographic hashing techniques can be employed to detect any data tampering during transmission or storage. Ensuring the integrity of container images is also crucial, requiring the verification of signatures and vulnerability scanning before deployment from secure registries.
- Availability guarantees that systems can operate continuously without disruption [41]. Cloud-native environments, due to their distributed architecture and reliance on microservices, are particularly vulnerable to threats such as system failures or Distributed Denial of Service (DDoS) attacks. Auto-scaling mechanisms, such as those provided by Kubernetes, dynamically allocate resources in response to traffic surges, ensuring service availability [42]. Load balancing distributes incoming traffic across multiple instances, preventing overloading and ensuring stable performance. DDoS protection requires the use of technologies like Intrusion Detection Systems (IDSs) and filtering mechanisms such as eXpress Data Path (XDP) and extended Berkeley Packet Filter (eBPF) to detect and block malicious traffic.
- The Scalability and Performance of cloud-native environments are inherently designed for dynamic scalability and high performance [43,44]. However, achieving this flexibility requires the implementation of robust load balancing, auto-scaling, and resource management mechanisms to efficiently manage varying workloads. Advanced monitoring and observability tools are necessary to sustain optimal performance and to ensure rapid response to emerging issues.
- Efficiency in cloud-native environments refers to the optimization of resource utilization, ensuring high performance while minimizing waste. Cloud-native architectures achieve this by leveraging automated resource allocation and optimizing data flow to maintain operational throughput. Additionally, the principle of data minimization is applied to restrict the collection of unnecessary data, while employing techniques such as encryption and pseudonymization to safeguard user privacy. These measures not only support compliance with regulations such as the General Data Protection Regulation (GDPR) but also ensure the efficient operation of the system [45,46]. Moreover, regular security audits of third-party services are essential for mitigating potential privacy risks associated with external components, thereby maintaining both security and operational efficiency.
- Management Complexity reflects the dynamic and distributed nature of cloud-native systems, and significantly increases the complexity of managing identities, access

controls, and the security of inter-service communications [47]. Organizations must adopt advanced management tools and practices capable of handling the intricate requirements of cloud-native deployments, particularly when these systems span multiple cloud providers and geographical locations [48].

Confidentiality, integrity, availability, and privacy are essential to ensuring the security of cloud-native microservices. These elements must work in tandem to address the evolving security challenges, and ongoing research and strategic adaptations are necessary to protect cloud-native applications against emerging threats.

2.4. Existing Solutions in Cloud-Native Environments

Increased use of microservice architectures and containerization drives the need for new security paradigms. There is a growing emphasis on integrating security into the DevOps cycle (DevSecOps), making security an integral part of the development and deployment process. Similarly, many companies are leveraging artificial intelligence and automation to enhance threat detection and response in dynamic cloud-native environments. On the other hand, the booming e-commerce and online services sector in South Korea, especially post-pandemic, requires secure cloud-native solutions to handle vast amounts of customer data and transactions. Moreover, in South Korea, the gaming and entertainment industry is very significant and it demands robust security for cloud-based platforms to protect intellectual property and user data.

Extensive research has been conducted to address the security dimensions of microservices architecture and containerization. Table 1 presents a synthesized overview of recent scholarly works, meticulously comparing their key considerations to elucidate emerging trends and critical insights in this domain.

Table 1. Comparison of key considerations in the literature with those in this study.

Reference	Integrity	Availability	Scalability and Performance	Efficiency	Management Complexity
[49]	○	○	○	○	
[50]	○	○			○
[51]	○	○	○		○
[52]	○	○	○	○	
[53]		○	○	○	
[54]		○	○	○	○
[55]	○	○			○
[56]	○	○	○		○
[57]		○	○	○	○
[58]		○	○	○	
Proposed Framework	○	○	○	○	○

○: The symbol is used to indicate if the referred paper has addressed one or multiple key consideration.

Huang, Hang et al. [49] present the design and implementation of PVM, a guest hypervisor for KVM that is transparent to the host hypervisor and assumes no hardware virtualization support. PVM addresses security isolation and privacy protection in cloud-native environments where containers are deployed within VMs. It features a minimal shared memory region and an efficient shadow page table design, significantly outper-

forming current nested virtualization in KVM for memory virtualization. PVM has been adopted by Alibaba Cloud for hosting secure containers.

Che, Kun et al. [50] investigate the genetic defects in the inherent mechanism of cloud-native systems, particularly in the service governance framework, from the perspective of zero trust. They identify endogenous security threats resulting from security flaws such as "default information authenticity" and "ubiquitous data visibility". The paper proposes employing "zero trust" to overcome "default trust" and "implicit mapping" to achieve "limited visibility" of user data. It constructs a novel zero-trust gateway architecture ecology and a corresponding load-balancing strategy, demonstrating its progressiveness through experimental results.

Chandramouli, R et al. [51] present a cloud-native software architecture for Verification of Request (VoR) utilizing standardized OPC UA PubSub communication over MQTT, as specified by NAMUR Open Architecture (NOA). NOA aims to securely decouple the life cycle of IT and OT components in the process industry. The architecture has been validated on a simulated system employing Kubernetes clusters and Virtual Machines hosting DCS and edge components.

Tomar, Manish et al. [52] introduce GrassHopper (GH), a cross-layer enforcement approach for network policies in cloud-native systems. GH automatically generates security group configurations from verified network policies, aiming to reduce the network attack surface between VMs by 75 to 99% without causing performance overhead. It has been evaluated on a Kubernetes cluster running on OpenStack, demonstrating its effectiveness in low-latency applications and cluster setups.

Dalila Ressi et al. [53] explore how artificial intelligence (AI) can enhance blockchain technology, focusing on the integration of AI techniques such as machine learning, deep learning, and natural language processing to improve blockchain's efficiency, security, and scalability. The paper categorizes AI's applications in blockchain, from optimizing consensus mechanisms and smart contracts to ensuring data privacy. It identifies synergies between AI and blockchain, highlighting potential advancements in security, transparency, and interoperability for industries ranging from finance to healthcare.

Suresh et al. [54] proposed a secure framework tailored for deploying microservices using cloud container technology. The framework emphasizes addressing the security challenges intrinsic to containerized environments, particularly those arising from the interactions between microservices. It leverages advanced cloud-native tools to create an environment that ensures isolation, scalability, and minimal inter-service interference. The authors demonstrate how the framework mitigates common vulnerabilities by incorporating strict security policies and efficient orchestration strategies, making it highly applicable for modern enterprise environments. Through a series of experimental evaluations, the framework is shown to significantly improve deployment security and operational efficiency.

Miller et al. [55] presented a novel approach to secure and leak-free workflows in microservices architecture by employing microservice isolation techniques. Their work focuses on mitigating data leakage and enforcing strict data owner policies within distributed workflows. The proposed methodology aligns with zero-trust principles, ensuring that each microservice operates within a confined and secure boundary. This is achieved through the implementation of isolation layers that minimize inter-service communication risks. Experimental results validate the framework's effectiveness in enhancing security and ensuring robust compliance with organizational data policies, marking a significant contribution to secure distributed systems design.

Kodakandla, et al. [56] examines the application of Zero Trust Architecture (ZTA) to enhance security in cloud-native infrastructures. The study emphasizes key ZTA principles,

such as continuous authentication, micro-segmentation, and least-privilege access, to mitigate threats like lateral movement and insider attacks. Using case studies, the paper demonstrates how ZTA improves resilience and compliance in multi-cloud environments while addressing challenges like unauthorized access and data breaches.

Ahmed, Mohammed Ilyas et al. [57] presents a new cross-layer enforcement mechanism called GrassHopper (GH) for network policies in cloud-native environments. GH automatically synchronizes security group policies from known consistent policies on the network with a goal of minimizing the network exposure between VMs within the rage of 75 to 99% without impacting performance. It has been tested in a Kubernetes cluster on OpenStack, and it can produce good results with low-latency applications and clustering.

Reddy, Amit Kumar et al. [58] explore the integration of security practices into the DevOps pipeline, referred to as DevSecOps, specifically in the context of cloud-native applications. The paper emphasizes leveraging shift-left security, continuous testing, and automated compliance to address the unique challenges posed by cloud-native architectures, such as container security, microservices vulnerabilities, and multi-cloud complexities. It highlights the importance of tools like Kubernetes and Docker in automating security measures, ensuring application resilience, and reducing management overhead. Additionally, the study examines how AI and ML can be incorporated to enhance threat detection and response within dynamic environments, providing insights into securing cloud-native systems efficiently.

Existing studies primarily focus on traditional static environments or address specific security issues, often employing static policy-based approaches such as predefined rules and basic access control mechanisms. While these methods [49–58] may be effective in fixed environments, they fail to adequately address the dynamic nature of cloud-native architectures. Furthermore, many existing works limit their DDoS mitigation strategies to the network layer or specific scenarios, lacking comprehensive and flexible defense mechanisms against distributed denial-of-service (DDoS) attacks.

In contrast, the proposed architecture in this study is specifically designed to accommodate the dynamic and distributed nature of cloud-native environments. It employs a Resilient Backpropagation Neural Network (RBN) for the real-time detection and mitigation of DDoS attacks across both the network and application layers, enabling adaptive and robust security even in highly volatile threat landscapes. Unlike the existing research, which often overlooks the complexity of secure inter-container communication, the proposed system integrates an Inter-Container Communication Bridge (ICCB) to ensure secure communication between containers. Additionally, it leverages advanced technologies such as the eXpress Data Path (XDP) and extended Berkeley Packet Filter (eBPF) to provide high-performance, low-latency security. These distinctive features allow the proposed architecture to address the multifaceted security demands of cloud-native environments more effectively than conventional approaches.

3. AIDS-Based Cyber Threat Detection Solution

The solution introduces an AIDS-based Cyber Threat Detection solution designed to enhance the security of cloud-native environments. This system leverages advanced techniques such as the Resilient Backpropagation Neural Network (RBN) to detect and mitigate Distributed Denial of Service (DDoS) attacks in containerized microservices. The solution is structured across three layers—the device, network, and cloud—which work together to monitor and secure inter-container communication, using technologies like the eXpress Data Path (XDP) and extended Berkeley Packet Filter (eBPF) to ensure minimal-latency and high-performance security enforcement.

This solution addresses key challenges, including the dynamic nature of containerized environments, by continuously monitoring network traffic and applying proactive threat detection and response mechanisms at both the network and application levels. Through this, the system offers a robust defense against security vulnerabilities while maintaining the operational efficiency of cloud-native microservices.

On the other hand, the proposed architecture demonstrates significant robustness against adversarial attacks through its multi-layered security mechanisms and adaptive learning capabilities. At the core of its resilience lies the Resilient Backpropagation Neural Network (RBN), which not only excels in detecting Distributed Denial of Service (DDoS) attacks but is also equipped to identify subtle patterns indicative of adversarial manipulations. By leveraging the real-time monitoring and dynamic adjustment of its parameters, the RBN can adapt to evolving attack vectors, minimizing the risk of successful exploitation.

Furthermore, the integration of technologies such as the eXpress Data Path (XDP) and extended Berkeley Packet Filter (eBPF) ensures that adversarial traffic is detected and mitigated with minimal latency. These technologies enhance the system's ability to maintain high throughput while applying rigorous security checks, even under adversarial conditions. The layered structure of the solution—comprising device, network, and cloud layers—provides an additional line of defense, enabling the isolation of malicious activities at various levels before they can propagate through the system.

The system's proactive monitoring of network traffic, combined with advanced anomaly detection, ensures that adversarial attacks, such as poisoning attacks or evasion techniques, are identified early. This is further reinforced by the use of customized rule-based mechanisms, which complement the neural network by enforcing strict access controls and filtering adversarial inputs. Together, these features enhance the overall robustness of the architecture, ensuring reliable operation and security in dynamic and distributed cloud-native environments.

3.1. System Overview

The violation detection and response solution is crucial for IT infrastructure integrity and security, especially in dynamic cloud computing and large-scale deployments. Containerization has advanced application deployment but also introduced new security challenges in container networking, with concerns over the efficacy of current security measures in cloud-native environments. In containerized environments, security risks arise when packet source association is lost, enabling malicious activities like lateral attacks and traffic poisoning. The dynamic nature of container IPs and the limitations of IP-based access controls further complicate security policy management. The architectural design of containerization, while efficient, increases susceptibility to DDoS attacks, disrupting microservice communication and leading to resource exhaustion and performance degradation. Decentralized microservice architecture introduces complex security vulnerabilities, especially in access control, posing risks of data breaches and unauthorized manipulation in cloud-native systems. The main requirements to solve the issues are to develop a method to efficiently deploy security functions across multiple hosts by utilizing kernel features, enabling direct end-to-end forwarding at the kernel level to enhance security in container networks, and to create a solution based on the Resilient Backpropagation Neural Network (RBN) to detect and counter DDoS attacks in containerized cloud environments, employing advanced machine learning for proactive threat detection and response.

Accordingly, this paper introduces an AIDS-based Cyber Threat Detection Framework, as shown in Figure 1. The figure consists of three layers: the device layer, network layer, and cloud layer, and it illustrates the interaction between the layers and components within the framework.

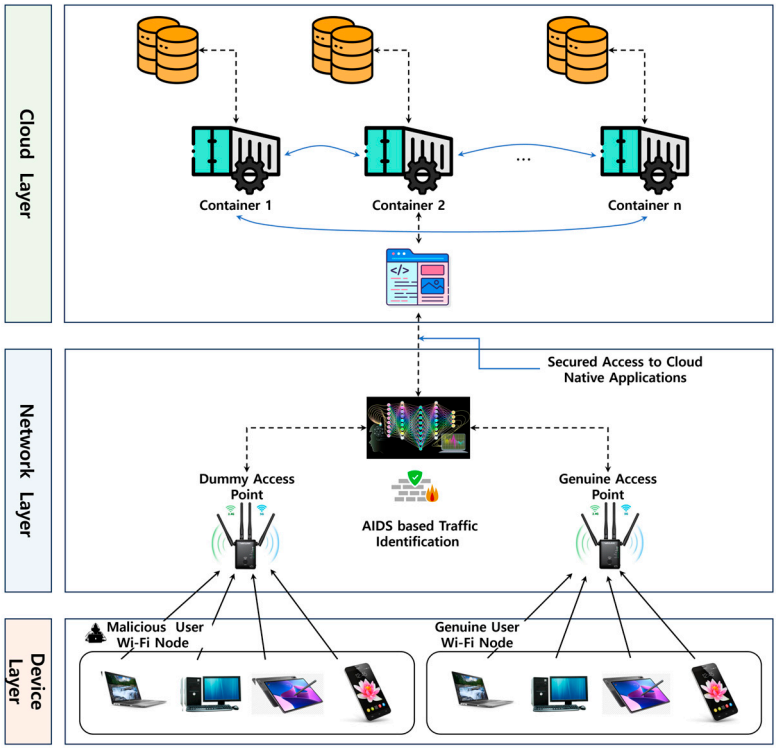

Figure 1. Proposed AIDS-based cyber threat detection framework.

The device layer comprises physical devices, sensors, and actuators that directly interact with the physical world in systems such as the Internet of Things (IoT). These devices are responsible for collecting environmental data and transmitting them to higher layers for further processing. Security at this layer is focused on safeguarding device integrity, implementing secure communication protocols, and mitigating threats such as physical tampering, malware, and unauthorized access. Ensuring an efficient and secure data flow from the device layer to the network layer is critical for minimizing latency and maintaining system reliability.

The network layer serves as the intermediary between devices and the cloud, securely transmitting data across various infrastructures, including Wi-Fi, cellular, and Ethernet networks. This layer employs advanced technologies such as the eXpress Data Path (XDP) and extended Berkeley Packet Filter (eBPF) to enhance data routing and security. These technologies enable real-time packet processing with minimal latency, ensuring optimal performance and robust security. However, their integration may pose challenges related to compatibility with legacy systems and the need for specialized expertise to manage and maintain these implementations. The network layer also addresses critical security concerns such as data interception, man-in-the-middle attacks, and the preservation of the reliability and confidentiality of transmitted information.

The Cloud Layer is tasked with processing, storing, and analyzing the data collected from devices via the network. This layer provides the computational power, data storage, and scalability required to handle large volumes of data in real time. The efficient and secure data flow facilitated by the network layer minimizes performance bottlenecks during transmission, ensuring seamless operation. Security mechanisms at the cloud layer emphasize data encryption, access control, compliance, and the maintenance of the confidentiality, integrity, and availability of stored data. These measures are essential for mitigating risks such as data breaches and unauthorized access, enabling the system to achieve high levels of scalability and performance without compromising security.

This architecture effectively integrates modern technologies to ensure low-latency data flow and robust security across all layers, addressing both performance and compatibility challenges inherent in cloud-native environments.

3.2. AIDS-Based RBN Model

The AIDS-based RBN model leverages a Resilient Backpropagation Neural Network (RBN) to achieve the precise detection of DDoS attacks within cloud-native environments. This model continuously monitors network traffic, accurately distinguishing between benign and malicious activities. Through an adaptive learning approach, the RBN model dynamically adjusts its parameters based on real-time data, enabling the effective identification of complex and evolving DDoS patterns. This proactive detection capability ensures robust security and operational continuity in containerized environments, facilitating rapid response to emerging threats.

In containerized environments, security risks arise when packet source association is lost, enabling malicious activities like lateral attacks and traffic poisoning. The dynamic nature of container IPs and the limitations of IP-based access controls further complicate security policy management. The architectural design of containerization, while efficient, increases susceptibility to DDoS attacks, disrupting microservice communication and leading to resource exhaustion and performance degradation. Decentralized microservice architecture introduces complex security vulnerabilities, especially in access control, posing risks of data breaches and unauthorized manipulation in cloud-native systems.

There are three primary approaches to enhancing security in containerized environments, securing the network environment of the container, securing the container host, and detecting configuration violation. The solution we propose is about monitoring and securing the network traffic to and from the containers, rather than focusing on the host machine's security. This approach aims to detect and prevent intrusions at the network level, ensuring the safety of the native containerized applications from external threats. Our proposed solution employs a network-based intelligent intrusion detection system (IDS). This system operates within the networking environment of containerized architectures, focusing on monitoring the network traffic between the internet and the container's environment, as shown in Figure 1. The goal is to identify and differentiate normal traffic from potential DDoS attacks.

At the core of our proposed system is the use of a sophisticated dataset, to train and test the neural network. The dataset needs to first be preprocessed, involving min–max normalization and feature selection, to ensure that the neural network receives high-quality, relevant data. Normalization is performed as per the min–max normalization method as follows:

$$Normalized\ Data = \frac{(Raw\ Data - Min)}{Max - Min} \times (Max_{new} - Min_{new}) + Min \quad (1)$$

Behavior Modeling and Classification: the behavior of both benign and malicious DDoS traffic will be modeled using a Resilient Backpropagation Neural Network (RBN). This network employs a gradient descent-based approach for error minimization, adjusting its parameters (weights) based on the difference between predicted and actual outputs. It utilizes the sum of squared differences as the error function, which is minimized using Stochastic Gradient Descent (SGD).

The training dataset is denoted as

$$Dataset = \left\{ \left(x^{(i)}, y^{(i)} \right) \mid i \in [1, N] \right\} \quad (2)$$

where $x^{(i)}$ is the *ith* input vector representing network traffic features, and $y^{(i)}$ is the corresponding output label. The output labels are binary, with '1' indicating DDoS attack traffic and '0' representing normal traffic.

The objective of neural network training is to minimize the error between its predicted output $\overline{y}^i$ and the actual label $y^{(i)}$ for each input in the dataset. This process is formally defined as an optimization problem:

$$\hat{\theta} = argmin_{\theta} \sum_{i=1}^{N} E\ \hat{y}^i, y^{(i)} \tag{3}$$

Here, $\hat{\theta}$ symbolizes the optimal parameters (like weights) of the neural network to be learned during training. The function $E\ \hat{y}^i, y^{(i)}$ quantifies the error or difference between the predicted and actual outputs. The training involves iteratively adjusting the neural network parameters to reduce this error across the entire training dataset, effectively enhancing the model's ability to accurately distinguish between normal and DDoS traffic. This approach is central to the development of an effective intrusion detection system for containerized cloud environments.

The RBN learning algorithm, a part of the local adaptive algorithm's family, is particularly designed for efficiently updating the weights of a neural network. The method is structured into two main steps.

- Weight Change Step: This step involves updating the weights based on a weight-specific update value. The equation for this process is as follows:

$$\Delta w_{ij}^{(t)} = -\gamma_{ij}^{(t)} \cdot sgn\left(\nabla_i E^{(t)}\right) \tag{4}$$

Here, $\Delta w_{ij}^{(t)}$ denotes the change in the weight w_{ij} at time (t). The term $\gamma_{ij}^{(t)}$ is the update value specific to that weight, and $sgn\left(\nabla_i E^{(t)}\right)$ represents the sign of the partial derivative of the error function EE with respect to the weight w_{ij} at time (t). This approach ensures that the weight adjustment is influenced by the direction of the error gradient, essentially moving the weight in the direction that reduces the error.

- Sign-Dependent Adaptation Step: In this step, the update value $\gamma_{ij}^{(t)}$ for the current epoch is adjusted based on the changes in the sign of the error gradient. The adjustment is governed by the following equation:

$$\begin{cases} min\left(\vartheta + \gamma_{ij}^{(t-1)}, \gamma_{max} \quad if \quad \nabla_i E^{(t)} \cdot \nabla_i E^{(t-1)} > 0\right) \\ min\left(\vartheta - \gamma_{ij}^{(t-1)}, \gamma_{min} \quad if \quad \nabla_i E^{(t)} \cdot \nabla_i E^{(t-1)} < 0\right) \\ \gamma_{ij}^{(t-1)} \quad otherwis \end{cases} \tag{5}$$

The constants $\vartheta+$ and $\vartheta-$ satisfy $\vartheta+ > 1$ and $0 < \vartheta- < 1$. This formula adjusts the learning rate γ_{ij} dynamically. If the error gradient continues in the same direction (sign), the learning rate is increased (up to a maximum of γ_{max}) to accelerate learning. Conversely, if the gradient sign changes, indicating potential overshooting, the learning rate is reduced (down to a minimum of γ_{min}) to prevent oscillations and ensure more stable convergence.

These two steps work to optimize the weight adjustments in the neural network, enhancing the efficiency and effectiveness of the learning process. The RBN algorithm's ability to adapt the learning rate for each weight independently based on the error gradient's sign makes it a robust and efficient choice for training neural networks, particularly in complex tasks like intrusion detection in containerized cloud environments.

3.3. Inter-Container Communication Bridge

The Inter-Container Communication Bridge (ICCB) is developed to secure inter-container communication within cloud-native environments by systematically managing and protecting network interactions between containers. The ICCB framework comprises per-container network stacks that enforce granular security policies and includes chained security functions, such as source verification and direct forwarding, to ensure data integrity and authenticity. Leveraging advanced technologies like XDP and eBPF, the ICCB achieves efficient, low-latency communication while maintaining a robust security layer that prevents unauthorized access and preserves data integrity across containerized services.

The XDP (eXpress Data Path) and eBPF (extended Berkeley Packet Filter) are advanced technologies used in Linux networking. They provide high-performance and programmable packet processing at the kernel level, which is particularly useful for tasks like networking, security, and performance monitoring. By using XDP/eBPF, the system can efficiently inspect and filter network packets at a very low level in the Linux network stack. This provides an effective way to enforce security policies directly on the network traffic entering or exiting containers. The programmability of eBPF allows for the creation of complex, fine-grained security policies that can be dynamically applied based on the context of the container's network traffic. Similarly, since XDP operates at the driver level and eBPF programs are executed within the kernel, they can process packets with minimal overhead, which is crucial in high-throughput container environments.

The proposed Inter-Container Communication Bridge is strategically structured into three integral components, as shown in Figure 2.

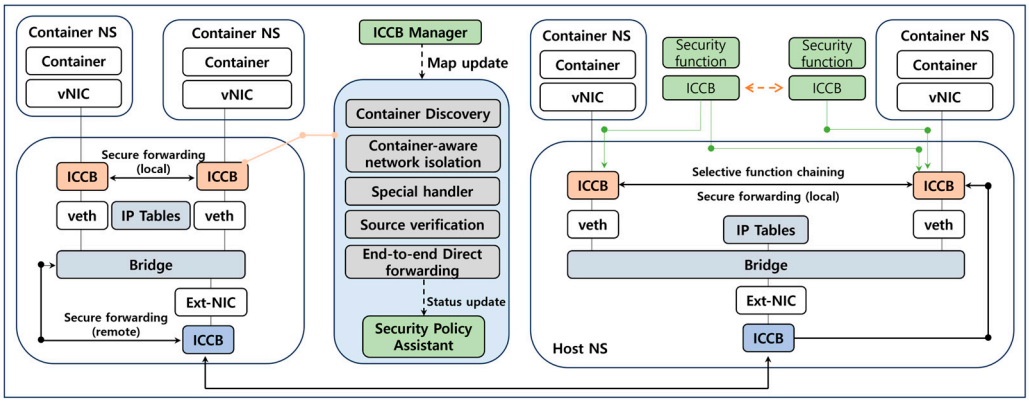

Figure 2. Inter-container communication bridge overview.

ICCB Manager: This component is crucial for maintaining a comprehensive network view of all containers, including their inter-container dependencies. It serves as the central point for overseeing the entire container network, ensuring cohesive communication and security management. The manager has partial components, including Container Network Information Collection, Network Stack Management, and Security Function Management. Container Network Information Collection gathers and maintains a detailed view of the container network, tracking the relationships and dependencies between different containers. This comprehensive understanding is vital for effective network management and security. And the Manager oversees the individual network stacks assigned to each container. It ensures that the security policies and protocols are correctly implemented, maintaining the integrity of container-to-container communications. In addition the Manager ensures these functions are effectively integrated and operational, providing an additional layer of security to the network.

Per-Container Network Stacks: These are dedicated to each container, where the actual enforcement of security policies occurs. Before a container's packets are delivered into the broader container network, they pass through Container Discovery, Container-Aware Network Isolation, Gateway and Service-IP Handling stacks, ensuring that security measures are applied at the most granular level. The Container Discovery process involves identifying and recognizing containers within the network, ensuring that each container is correctly accounted for and managed within the security framework. Container-Aware Network Isolation provides isolation mechanisms tailored to the unique network context of each container, ensuring secure and segregated communication channels. Gateway and Service-IP Handling manages the routing and addressing within the container network, effectively handling the gateway functionalities and service IP al-locations.

Chained Security Functions: These functions are deployed to provide additional layers of security, Source Verification, End-to-End Direct Forwarding functions enable thorough inspections of inter-container network traffic. They are tailored to execute content-based access controls and other security measures, ensuring that only authorized and secure communications occur between containers. Source Verification ensures the authenticity of the data packets, verifying the source of each packet to prevent unauthorized access or data breaches. End-to-End Direct Forwarding enables the direct forwarding of data packets from their source to their destination, bypassing potential security threats and ensuring the efficient and secure transmission of data.

3.4. Open Issues and Solutions

We contemplate the consequence and possible effect of the proposed Intrusion Detection System (IDS) framework for containerized cloud-native microservice security. Hence, this framework uses a Resilient Backpropagation Neural Network (RBN) to detect and prevent DDoS attacks. It is also necessary to discuss the importance of the proposed approach, identify the strengths and weaknesses of the developed approach as compared to traditional ones, and consider potential problems in its implementation.

Another strength of the proposed IDS framework is the identification of learning features or rather the overall learning ability of the same. The RBN help the IDS to monitor the traffic continually and to change the parameters of the system dependent on actual data; therefore, the system could work better in distinguishing normal and malicious traffic. This is a much better proposition than the deterministic rule-based systems, whose efficacy might plummet due to emergent threats. Furthermore, through its architecture, the framework works seamlessly in environments that require containerization, thus being able to scale up in line with the cloud-native infrastructure layout. This scalability assures end users with a possibility to receive a high level of protection for large amounts of traffic without having to compromise the performance level of IDS.

Yet another benefit applies to monitoring: the framework is comprehensive in following the development steps. In order to do this, the IDS is designed to monitor network traffic in and out of the containers that host containerized applications, which provides a complete solution to detecting DDoS attacks before they increase in size and magnitude, as opposed to monitoring specific applications within the containers, making them vulnerable to these types of attacks. In addition, traffic analysis enables the intelligent processing of data, so the load on the system is less compared with that in the more traditional IDS solutions, but the response remains as fast as that in cloud-native applications.

The proposed architecture effectively addresses several critical dimensions, including integrity, availability, efficiency, and the complexity of management. Nevertheless, the discourse appears to overlook crucial aspects of scalability and performance evaluation.

Table 2 provides a comprehensive explanation of the mechanisms by which the proposed architecture addresses each of these identified elements.

Table 2. Comparison of existing research and proposed architecture.

Key Consideration	Existing Research	Proposed Architecture
Integrity	PVM maintains the integrity of memory virtualization, while zero-trust architectures protect integrity through fine-grained policies and implicit mapping. AI-enhanced blockchain combines blockchain immutability with AI verification to strengthen integrity.	Utilizes cryptographic hashing and data validation to ensure data are untampered and reliable during storage and transmission. Essential for securing container images and communication in cloud-native environments against unauthorized modifications and external attacks.
Availability	Efficient shadow paging, zero-trust-based network security, and multi-cloud access control maximize system availability. AI-enhanced blockchain ensures high availability through the resilience of distributed networks.	Ensures continuous operation through auto-scaling, load balancing, and orchestration tools like Kubernetes. Includes DDoS detection and mitigation mechanisms to maintain service stability under heavy network traffic.
Scalability and Performance	PVM enhances scalability and performance through hardware-independent design, while zero-trust architectures offer scalable policy deployment with security that maintains performance. AI-enhanced blockchain optimizes both scalability and performance using adaptive AI mechanisms.	Employs dynamic resource allocation using orchestration tools to respond to traffic surges, minimizing downtime. Scalability is achieved through automated scaling and robust performance under varying workloads.
Efficiency	PVM streamlines guest-hypervisor state transitions, zero-trust architectures enhance the efficiency of security processes, and AI-enhanced blockchain boosts operational efficiency through transaction workflow optimization.	Implements data minimization techniques to protect user privacy and comply with regulations like GDPR. Conducts regular security audits for third-party services, reducing privacy risks while ensuring system efficiency.
Management Complexity	PVM simplifies management by eliminating the need for hardware support, while zero-trust approaches reduce complexity with integrated policy deployment. AI-enhanced blockchain simplifies complex network management using AI-driven tools.	Addresses challenges in managing dynamic and distributed cloud-native systems with advanced tools like IaC for automation. Streamlines identity and access control management, improving consistency across multi-cloud and geographically distributed environments.

Integrity plays a crucial role in the proposed architecture. Cryptographic hashing and data validation techniques are applied to ensure that data remain untampered and reliable during storage and transmission. These mechanisms are particularly essential for maintaining the integrity of container images, preventing unauthorized modifications, and ensuring that data transferred between services remain accurate in cloud-native environments. This ensures the protection of data from external attacks and secures communication between services.

In terms of availability, the architecture guarantees continuous operation through technologies like auto-scaling and load balancing. By leveraging orchestration tools such as Kubernetes, the architecture can dynamically allocate resources in response to traffic

surges, minimizing downtime and enhancing system reliability. Moreover, the inclusion of mechanisms for detecting and mitigating Distributed Denial of Service (DDoS) attacks ensures stable performance even under increased network traffic, maintaining the overall availability of the services.

Efficiency focuses on protecting user data and complying with legal regulations such as the General Data Protection Regulation (GDPR). Cloud-native environments, which heavily integrate open-source components and third-party services, amplify privacy concerns. The architecture implements data minimization techniques, ensuring that only the necessary data are collected and shared with the required services. Regular security audits of third-party services are also conducted to mitigate privacy risks, ensuring that the system remains efficient while preserving user privacy.

Management complexity is a significant challenge due to the dynamic and distributed nature of cloud-native systems. The proposed architecture adopts advanced management tools and practices to handle identities, access controls, and secure communication between services, particularly in environments spanning multiple cloud providers and geographical locations. The use of automation tools such as Infrastructure as Code (IaC) helps streamline management tasks, improving efficiency and ensuring consistency in operations.

However, scalability and performance are areas where the architecture is less developed. While it mentions load balancing and auto-scaling, there is a lack of detailed discussion on how the architecture handles extreme scalability or manages resource overhead as the environment expands. In terms of performance, granular optimizations, such as container orchestration efficiencies, routing enhancements, and network latency management, are not deeply covered beyond the general use of load balancing techniques.

In conclusion, the proposed system provides a robust foundation in terms of security and availability. Nevertheless, the architecture would benefit from additional detailed solutions to address scalability and performance more comprehensively, creating a more complete framework overall. In addition, these technologies can be applied to the environments of cloud service providers (AWS, Azure, Google Cloud, etc.), e-commerce platforms (Amazon, eBay, etc.), financial institutions (PayPal, Stripe, etc.), the healthcare industry (Teledoc, Cerner, etc.), and telecommunication companies (KT, SKT, LG, etc.).

4. Conclusions

The proposed AIDS-based Cyber Threat Detection Framework for containerized cloud-native microservices presents a robust and innovative approach to addressing security challenges in dynamic environments. The framework's use of the Resilient Back-propagation Neural Network (RBN) significantly improves the detection and prevention of Distributed Denial of Service (DDoS) attacks by leveraging continuous monitoring and adaptive learning. This neural network allows for the real-time adjustment of system parameters based on actual traffic data, offering a more flexible and proactive defense mechanism compared with traditional rule-based Intrusion Detection Systems (IDSs).

One of the primary strengths of the proposed architecture lies in its seamless integration with cloud-native infrastructure, which ensures high scalability and the ability to handle large traffic volumes without compromising performance. By utilizing technologies like the eXpress Data Path (XDP) and extended Berkeley Packet Filter (eBPF), the system provides low-latency, high-performance security enforcement at both the network and application levels, which is crucial in containerized environments where inter-container communication is highly dynamic.

The proposed framework has been meticulously developed to meet critical security requirements by ensuring system integrity, availability, operational efficiency, and streamlined management. Conceived as a comprehensive and robust solution, it seeks to address

the limitations identified in prior research. Future work will involve the rigorous validation and demonstration of the framework's scalability and performance across diverse operational conditions.

Moreover, an in-depth evaluation will be conducted within multi-cloud environments to assess its cross-platform adaptability and resilience. To further enhance the framework's applicability, encryption and anonymization techniques will be integrated to effectively address privacy concerns associated with the ICCB, ensuring robust protection of sensitive data while maintaining operational efficiency.

Comparative analyses with alternative machine learning models are anticipated to yield valuable insights for enhancing detection accuracy while minimizing computational overhead. These research directions are expected to strengthen the framework's comprehensiveness, ensuring its suitability for diverse cloud-native architectures and its efficacy in mitigating emerging cybersecurity threats.

Author Contributions: Conceptualization, H.P. and A.E.A.; methodology, H.P. and A.E.A.; writing—original draft preparation, H.P.; supervision, A.E.A. and J.H.P.; funding acquisition, J.H.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Seoul National University of Science and Technology.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest regarding the design of this study, and the analyses and writing of this manuscript.

References

1. Lim, J. Versatile Cloud Resource Scheduling Based on Artificial Intelligence in Cloud-Enabled Fog Computing Environments. *Hum.-Centric Comput. Inf. Sci.* **2023**, *13*, 54.
2. Modisane, P.; Jokonya, O. Evaluating the Benefits of Cloud Computing in Small. *Medium Micro-Sized Enterp. (SMMEs) Procedia Comput. Sci.* **2021**, *181*, 784–792. [CrossRef]
3. Costa, B.; Bachiega, J., Jr.; de Carvalho, L.R.; Araujo, A.P. Orchestration in Fog Computing: A Comprehensive Survey. *ACM Comput. Surv. (CSUR)* **2022**, *55*, 29. [CrossRef]
4. Laghari, A.A.; Jumani, A.K.; Laghari, R.A. Review and State of Art of Fog Computing. *Arch. Comput. Methods Eng.* **2021**, *28*, 3631–3643. [CrossRef]
5. Mansouri, Y.; Babar, M.A. A Review of Edge Computing: Features and Resource Virtualization. *J. Parallel Distrib. Comput.* **2021**, *150*, 155–183. [CrossRef]
6. Laroui, M.; Nour, B.; Moungla, H.; Cherif, M.A.; Afifi, H.; Guizani, M. Edge and Fog Computing for IoT: A Survey on Current Research Activities Future Directions. *Comput. Commun.* **2021**, *180*, 210–231. [CrossRef]
7. Malviya, A.; Dwivedi, R.K. A Comparative Analysis of Container Orchestration Tools in Cloud Computing. In Proceedings of the 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 23–25 March 2022; pp. 698–703.
8. Deng, Q.; Goudarzi, M.; Buyya, R. FogBus2: A Lightweight and Distributed Container-based Framework for Integration of IoT-enabled Systems with Edge and Cloud Computing. In Proceedings of the International Workshop on Big Data in Emergent Distributed Environments, Virtual Event, 20 June 2021; pp. 1–8.
9. Wang, W.; Tornatore, M.; Zhao, Y.; Chen, H.; Li, Y.; Gupta, A.; Zhang, J.; Mukherjee, B. Infrastructure-efficient Virtual-Machine Placement and Workload Assignment in Cooperative Edge-Cloud Computing over Backhaul Networks. *IEEE Trans. Cloud Comput.* **2023**, *11*, 653–665. [CrossRef]
10. He, T.; Buyya, R. A Taxonomy of Live Migration Management in Cloud Computing. *ACM Comput. Surv.* **2023**, *56*, 1–33. [CrossRef]
11. Alonso, J. Understanding the Challenges and Novel Architectural Models of Multi-Cloud Native Applications. *J. Cloud Comput.* **2023**, *12*, 6. [CrossRef]
12. Garg, S. On Continuous Integration/Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps. In Proceedings of the 2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering, Laguna Hills, CA, USA, 1–3 December 2021.
13. Kai, P.; Bohai, Z.; Muhammad, B.; Xiaolong, X.; Anand, N. QoS-Aware Cloud-Edge Collaborative Micro-Service Scheduling in the IIoT. *Hum.-Centric Comput. Inf. Sci.* **2023**, *13*, 28.

14. Rahaman, M.S. Static-Analysis-Based Solutions to Security Challenges in Cloud-Native Systems: Systematic Mapping Study. *Sensors* **2023**, *23*, 1755. [CrossRef] [PubMed]
15. Mohammed, C.M.; Zeebaree, S.R.M. Sufficient Comparison Among Cloud Computing Services: IaaS, PaaS, and SaaS: A Review. *Int. J. Sci. Bus.* **2021**, *5*, 17–30.
16. Ankit, K.; Turki, A.; Sun-Yuan, H.; Udham, S.K.; Teekam, S.; Linesh, R.; Kumar, S.J.; Kumar, M.R. A hybrid solution for secure privacy-preserving cloud storage information retrieval. *Hum.-Centric Comput. Inf. Sci.* **2023**, *13*, 11.
17. Deng, S.; Zhao, H.; Huang, B.; Zhang, C.; Chen, F.; Deng, Y. Cloud-Native Computing: A Survey from the Perspective of Services. *Proc. IEEE* **2024**, *112*, 12–46. [CrossRef]
18. Arora, S.; Khare, P.; Gupta, S. AI-Driven DDoS Mitigation at the Edge: Leveraging Machine Learning for Real-Time Threat Detection and Response. In Proceedings of the 2024 International Conference on Data Science and Network Security (ICDSNS), Tiptur, India, 26–27 July 2024; IEEE: Piscataway, NJ, USA, 2024.
19. Theodoropoulos, T.; Rosa, L.; Benzaid, C.; Gray, P.; Marin, E.; Makris, A.; Cordeiro, L.; Diego, F.; Sorokin, P.; Girolamo, M.D.; et al. Security in Cloud-Native Services: A Survey. *J. Cybersecur. Priv.* **2023**, *3*, 758–793. [CrossRef]
20. Ajay, A.; Ahmad, S. Cloud security: Emerging threats, solutions, and research gaps. In *Artificial Intelligence and Information Technologies*; CRC Press: Boca Raton, FL, USA, 2025; pp. 64–70.
21. Vardia, A.S.; Chaudhary, A.; Agarwal, S.; Sagar, A.K.; Shrivastava, G. Cloud Security Essentials: A Detailed Exploration. In *Emerging Threats and Countermeasures in Cybersecurity*; Scrivener Publishing: Wiley, NJ, USA, 2025; pp. 413–432.
22. Hayagreevan, H.; Khamaru, S. Security of and by Generative AI platforms. *arXiv* **2024**, arXiv:2410.13899.
23. Jeon, J.; Jeong, B.; Jeong, Y.-S. PreVA: Predictive Vertical Autoscaler Using Multi Bi-GRU for Sustainable Cloud-Native Computing. *Hum.-Centric Comput. Inf. Sci.* **2024**, *14*, 1–17.
24. Ahmed, M.I. Threat Analysis for Cloud-Native Deployments. In *Cloud-Native DevOps: Building Scalable and Reliable Applications*; Apress: Berkeley, CA, USA, 2024; pp. 355–387.
25. Liu, G.; Huang, B.; Liang, Z.; Qin, M.; Zhou, H.; Li, Z. Microservices: Architecture, container, and challenges. In Proceedings of the 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Macau, China, 11–14 December 2020.
26. Jones, R. The Impact of AI on Secure Cloud Computing: Opportunities and Challenges. *Indones. J. Comput. Sci.* **2024**, *13*, 1–17.
27. Meiran, G. Contribution to Information Security Continuous Audit in Cloud-Native Environments. Ph.D. Thesis, Singidunum University, Belgrade, Serbia, 2024.
28. Kim, T.W.; Azzaoui, A.E.L.; Koh, B.; Kim, J.; Park, J.H. A secret sharing-based distributed cloud system for privacy protection. *Hum.-Centric Comput. Inf. Sci.* **2022**, *12*, 20–36.
29. Admass, W.S.; Munaye, Y.Y.; Diro, A.A. Cyber security: State of the art, challenges and future directions. *Cyber Secur. Appl.* **2024**, *2*, 100031. [CrossRef]
30. Balantrapu, S.S. Current Trends and Future Directions Exploring Machine Learning Techniques for Cyber Threat Detection. *Int. J. Sustain. Dev. Through AI ML IoT* **2024**, *3*, 1–15.
31. Dhadhania, A.; Bhatia, J.; Mehta, R.; Tanwar, S.; Sharma, R.; Verma, A. Unleashing the power of SDN and GNN for network anomaly detection: State-of-the-art, challenges, and future directions. *Secur. Priv.* **2024**, *7*, e337. [CrossRef]
32. Rodriguez, G.; Yannibelli, V.; Rocha, F.G.; Barbara, D.; Azevedo, I.M.; Menezes, P.M. Understanding and addressing the allocation of microservices into containers: A review. *IETE J. Res.* **2024**, *70*, 3887–3900. [CrossRef]
33. Eyvazov, F.; Ali, T.E.; Ali, F.I.; Zoltan, A.D. Beyond Containers: Orchestrating Microservices with Minikube, Kubernetes, Docker, and Compose for Seamless Deployment and Scalability. In Proceedings of the 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 14–15 March 2024; IEEE: Piscataway, NJ, USA, 2024.
34. Christudas, B.A. Microservice Containers. In *Java Microservices and Containers in the Cloud: With Spring Boot, Kafka, PostgreSQL, Kubernetes, Helm, Terraform and AWS EKS*; Apress: Berkeley, CA, USA, 2024; pp. 345–404.
35. Lad, S. Cybersecurity Trends: Integrating AI to Combat Emerging Threats in the Cloud Era. *Integr. J. Sci. Technol.* **2024**, *1*, 8.
36. Thapa, P.; Arjunan, T. AI-Enhanced Cybersecurity: Machine Learning for Anomaly Detection in Cloud Computing. *Q. J. Emerg. Technol. Innov.* **2024**, *9*, 25–37.
37. Degioanni, L.; Grasso, L. *Practical Cloud Native Security with Falco: Risk and Threat Detection for Containers, Kubernetes, and Cloud*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2022.
38. Zhu, H.; Gehrmann, C.; Roth, P. Access security policy generation for containers as a cloud service. *SN Comput. Sci.* **2023**, *4*, 748. [CrossRef]
39. Anandharaj, N. AI-Powered Cloud Security: A Study on the Integration of Artificial Intelligence and Machine Learning for Improved Threat Detection and Prevention. *J. Recent Trends Comput. Sci. Eng. (JRTCSE)* **2024**, *12*, 21–30.
40. Rehan, H. AI-Driven Cloud Security: The Future of Safeguarding Sensitive Data in the Digital Age. *J. Artif. Intell. Gen. Sci. (JAIGS)* **2024**, *1*, 132–151.

41. Stutz, D.; Assis, J.T.; Laghari, A.A.; Khan, A.A.; Andreopoulos, N.; Terziev, A. Enhancing Security in Cloud Computing Using Artificial Intelligence (AI). In *Applying Artificial Intelligence in Cybersecurity Analytics and Cyber Threat Detection*; Wiley: New Jersey, NJ, USA, 2024; pp. 179–220.
42. AllahRakha, N. Demystifying the Network and Cloud Forensics' Legal, Ethical, and Practical Considerations. *Pak. J. Criminol.* **2024**, *16*, 1–14.
43. Padmanaban, H. Quantum Computing and AI in the Cloud. *J. Comput. Intell. Robot.* **2024**, *4*, 14–32.
44. Kettunen, J.P. Maintainability in Cloud-Native Architecture. Master's Thesis, University of Jyväskylä, Jyväskylä, Finland, 2024.
45. Oyeniran, O.C.; Adewusi, A.O.; Adeleke, A.G.; Akwawa, L.A.; Azubuko, C.F. Microservices architecture in cloud-native applications: Design patterns and scalability. *Comput. Sci. IT Res. J.* **2024**, *5*, 2107–2124. [CrossRef]
46. Branco, D.; D'Angelo, S.; Martino Bd Esposito, A.; Lisi, V.d.; Paravati, G. Cloud-Native Software Development Life Cycle: A Case Study with Italian Ministry of Justice. In *International Conference on Advanced Information Networking and Applications*; Springer Nature: Cham, Switzerland, 2024.
47. Wang, K.; Hu, C.; Shan, C. Evaluation of Application Layer DDoS Attack Effect in Cloud Native Applications. *IEEE Trans. Cloud Comput.* **2024**, *12*, 522–538. [CrossRef]
48. Tatineni, S.; Chakilam, N.V. Integrating Artificial Intelligence with DevOps for Intelligent Infrastructure Management: Optimizing Resource Allocation and Performance in Cloud-Native Applications. *J. Bioinform. Artif. Intell.* **2024**, *4*, 109–142.
49. Huang, H.; Lai, J.; Rao, J.; Lu, H.; Hou, W.; Su, H.; Xu, Q.; Zhong, J.; Zeng, J.; Wang, X.; et al. Pvm: Efficient shadow paging for deploying secure containers in cloud-native environment. In Proceedings of the 29th Symposium on Operating Systems Principles, Koblenz, Germany, 23–26 October 2023.
50. Che, K.; Shuo, S. Cloud Native Network Security Architecture Strategy under Zero Trust Scenario. In Proceedings of the 2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 15–17 September 2023; IEEE: Piscataway, NJ, USA, 2023; Volume 7.
51. Chandramouli, R.; Butcher, Z. *A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Cloud Environments*; No. NIST Special Publication (SP) 800-207A; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023.
52. Tomar, M.; Ramalingam, S.; Krishnaswamy, P. Cloud-Native Enterprise Platform Engineering: Building Scalable, Resilient, and Secure Cloud Architectures for Global Enterprises. *Aust. J. Mach. Learn. Res. Appl.* **2023**, *3*, 601–639.
53. Ressi, D.; Romanello, R.; Piazza, C.; Rossi, S. AI-enhanced blockchain technology: A review of advancements and opportunities. *J. Netw. Comput. Appl.* **2024**, *225*, 103858. [CrossRef]
54. Suresh, S.; Ramachandran, N.; Hanumanthappa, M.; Ravikumar, K.; Jain, A. A Secure Framework for the Deployment of Microservices Using Cloud Container Technology. In *Rising Threats in Expert Applications and Solutions*; Springer Nature: Singapore, 2022; pp. 77–85.
55. Miller, L.; Mérindol, P.; Gallais, A.; Pelsser, C. Towards secure and leak-free workflows using microservice isolation. In Proceedings of the 2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR), Paris, France, 7–10 June 2021.
56. Kodakandla, N. Securing Cloud-Native Infrastructure with Zero Trust Architecture. *J. Curr. Sci. Res. Rev.* **2024**, *2*, 18–28.
57. Ahmed, M.I. CI/CD Pipeline in Cloud-Native DevOps. In *Cloud-Native DevOps: Building Scalable and Reliable Applications*; Apress: Berkeley, CA, USA, 2024; pp. 135–177.
58. Reddy, A.K. DevSecOps: Integrating Security into the DevOps Pipeline for Cloud-Native Applications. *J. Artif. Intell. Res. Appl.* **2021**, *1*, 89–114.