International Journal of Cloud Applications and Computing
Volume 16 • Issue 1 • January-December 2026

End-to-End Security for SDN Controllers in Distributed K8s Environments for Fog and Cloud

Asad Khan
https://orcid.org/0009-0009-5240-6946
University of Technology, Sydney, Australia

Priyadarsi Nanda
University of Technology, Sydney, Australia
Received: October 20th, 2025 | **Accepted:** December 24th, 2025

ABSTRACT

The deployment of software-defined networking (SDN) in distributed Kubernetes environments across fog and cloud systems introduces complex security challenges. Traditional approaches often fail to ensure secure, resource-efficient control-plane operations and verifiable node coordination at scale. This study proposes a three-layered security framework: (a) flexible control plan (FCP) integrates lightweight SDN controllers with runtime attestation for trusted execution in resource-constrained fog nodes; (b) secure software-defined offloading (SSDO) enforces encrypted, policy-driven inter-node communication and signature verification to prevent unauthorized coordination; and (c) sentinel-adaptive intrusion detection (SAID) uses an unsupervised deep learning autoencoder to detect anomalies and identify zero-day threats. Combined, these layers offer scalable, adaptive, and real-time security for distributed SDN-Kubernetes environments.

KEYWORDS

SDN, FCP, SSDO, SAID, Fog Computing, Cloud Infrastructure

INTRODUCTION

The rapid development of distributed computing paradigms like fog and cloud computing have transformed how modern digital infrastructures operate in the new era of scalable, low latency, and responsive services. Applications that require real-time processing and/or significant data management are now dependent on fog and cloud computing paradigms such as autonomous vehicles, industrial automation, and smart cities (Babou et al., 2024). At the heart of these systems is software-defined networking (SDN), a transformative way of managing networks that provides the ability to separate the control and data plane, enables centralized control, and provides for dynamic behavior of the network (Jin et al., 2025; Scano et al., 2023). The evolution of SDN is in conjunction with cloud and fog environments in fact, in recent years, Kubernetes has become the orchestration behemoth for fog-cloud ecosystems, enabling automatic deployment, scaling and operation of containerized or virtualized applications across multiple distributed nodes in many fog-cloud environments (Arzo et al., 2024; Sellami et al., 2022). The emergence of SDN, Kubernetes, and new distributed computing

DOI: 10.4018/IJCAC.398931

environments create numerous new and complicated security issues for the user, especially with respect to the availability, confidentiality, and integrity of the SDN controllers (Zeydan et al., 2022).

SDN controllers serve as the brain of the network, making it a high-value target for adversaries who seek to disrupt or manipulate network operations (Appari, 2022). In a typical distributed Kubernetes deployment, where workloads are distributed across fog and cloud layers, SDN controllers may be deployed across the network domain(s). The controllers will fail into the degrees of differing enforcement levels and security trusts, which increases surface risk considerably (Alamer, 2021). Since the SDN controller can be seen as a centralized tension point, it creates more attack surfaces of man-in-the-middle, denial-of-service, session hijacking, and unauthorized pending access attacks, in particular (Batewela et al., 2025). The impermanence of containerized deployments makes this risk even more challenging; ephemeral services, the constantly shifting network topologies and multi-tenancy situations introduce complexities into the intrinsic behaviours of the security policies that previously used to be instantiated (Carmona-Cejudo et al., 2023; Paolucci et al., 2021). The requirement to deploy situational environments and new attack vectors leads to an urgent need of an end-to-end security framework of SDN controllers which perform in distributed Kubernetes The security framework of Software-Defined Networking (SDN) controllers should comprise secure communication channels, robust authentication and authorization mechanisms, clearly defined policy enforcement domains, and resilience against both internal threats and externally identified adversaries (Prasanth & Uma, 2025). Nevertheless, from a security perspective, modern systems should move beyond traditional perimeter-based security, which assumes implicit trust within network boundaries, and instead adopt zero trust principles, where every access request is continuously verified. This approach enables consistent security enforcement in dynamic Kubernetes environments, including service mesh deployments and application programming interface (API) management layers. An application programming interface (API) is a standardized set of rules and protocols that enable communication and data exchange between software components and services. Fine-grained identity management and architecture are also emerging technologies which can be used as a solution to ensure that security can be handled easily.

Moreover, the integration of security into the orchestration and lifecycle management of SDN elements into Kubernetes environments necessitates a smooth integration between the container orchestration platform and the network layer (Pedone et al., 2021). Security involves automated certificate management, secure API gateways, encrypted service meshes, and audit logging. Also consider the unique characteristics of fog computing nodes, which are typically resource-constrained, have unreliable connectivity, and generally require lightweight and adaptive security protocols (Nunez-Gomez et al., 2021; Sami et al., 2021). This paper will examine the architectural design and implementation of end-to-end security for SDN controllers in distributed Kubernetes environments deployed over both fog and cloud infrastructures (i.e., cloud edge computing). A layered security approach is proposed, encompassing the use of cryptographic primitives, container-level isolation, runtime security tools, and adaptive threat detection to secure SDN controllers by reducing the number of attack vectors while sustaining the level of performance and agility that fog-cloud systems promise. We also analyze the relevant threat models and security mechanisms and discuss how these mechanisms can be integrated into a software-defined network to contribute a secure-by-design approach to managing software-defined networks in the next generation of distributed computing environments.

Such values of degradation are supported by a variety of unbiased measures (Batista Jr et al., 2021) with up to 18% latency inflation in SDN-Kubernetes orchestration through control-plane synchronization overhead and with more than 20% spoofing vulnerability in case of distributed controllers without runtime verification (Singh et al., 2022). Equally, as pointed out by Javanmardi et al. (2023), containerized SDN deployments with the fog workload always report false-positive rates of more than 12% using the traditional support vector machine (SVM)-based intrusion detection system (IDS), indicating the dire need to bridge the gap between adaptive and integrated security mechanisms.

This paper's contributions include the following:

- flexible control plan (FCP) layer: proposes a lightweight SDN control-plane architecture, which has the attestation of the runtime to provide trusted execution and fault tolerance in resource-constrained fogs.
- secure software-defined offloading (SSDO) layer: provides inter-node coordination through policy-based encryption and on-top of digital signature verification, which guarantees offloading audits and malicious resistance.
- sentinel-adaptive intrusion detection (SAID) layer: an intrusion detection system based on deep learning, which is unsupervised and detects zero-day anomalies automatically using adaptive reconstruction-error threshold.
- integrated novelty: introduces a comprehensive security framework in Kubernetes based on runtime attestation and resource-efficient adaptive autoencoder-based intrusion detection and presents the first implementation of an integrated solution that allows real-time end-to-end security at relatively low cost.

The rest of the paper is organized as follows. The issue is explained in the literature review along with a list of relevant publications. The proposed methodology section illustrates and explains the recommended method. This is followed by the results and discussion section and the conclusion.

LITERATURE REVIEW

Batista Jr. et al. (2021) presented a new architecture of a dynamic orchestrator, which supports network slicing through SDN, network function virtualization (NFV) and SDR technologies. The orchestrator manages independent end-to-end slices and supports horizontal handover to maintain the same quality of service (QoS) and quality of experience (QoE). The orchestrator improves reliability and performance by monitoring the QoS in near real-time and dynamically selecting the optimal providers in run-time as needed. The architecture also supports advancements in network slicing for 5G and further development and provides important economic, technological, and social value.

Javanmardi et al. (2023) defended the scheduling services of their time-critical internet of things (IoT) applications in the fog environment against distributed denial of service (DDoS) and port scanning attacks. A controller called software-defined fog-oriented security S-FoS, which leverages SDN as a flexible management method, is proposed along with a fuzzy-based anomaly detection system to identify and block the malicious requestors. Further, Non-dominated Sorting Genetic Algorithm(NSGA-III) multi-objective optimization has been employed for effective load balancing and less delay, ensuring safe and effectively utilized resources in IoT-fog.

Singh et al. (2022) proposed a load-balancing mechanism to ensure efficient resource utilization in a bootstrapped fog setting that runs on SDN. They also incorporated deep belief network-based system of intrusions to secure itself and minimize delays in communications. Finally, it has been demonstrated that their architecture is effective in terms of balancing the performance, energy consumption, and security of their applications in the IoT fogs.

Rangisetti et al. (2021) proposed a new prevention model called distributed address resolution protocol (D-ARP) spoof model that protects against address resolution protocol (ARP) spoofing in cloud-fog-edge environments that are supported by SDN and NFV. D-ARP spoof is able to prevent attacks like virtual local area network identifier (VLAN-ID) spoofing, denial of service (DoS), DDoS, man-in-the-middle (MITM) and session hijacking without adding overhead to OpenFlow switches and centralized controller architectures (CVA) centralized controllers.

Janakiraman and Deva Priya (2023) suggested a deep reinforced learning-long short-term memory-based scheme of mitigating DDoS attacks in the cloud with the help of fog. The solution leverages SDN to deploy a defense module within the SDN controller capable of detecting abnormalities at the network and transport layers. This module analyzes network traffic to identify malicious behavior patterns and blocks malicious packets while allowing only

legitimate traffic to pass. As a result, the proposed architecture improves service availability and enhancing resilience against DDoS attacks in the cloud environment.

Syed et al. (2023) focused on routing in massive/scalable environments and specifically addressed challenges in data routing in large-scale IoT networks in industry 4.0-based environments. Non-traditional blockchains do not manage evolving IoT device landscapes, as they grow in both the variety of devices and number of devices, which resulted in issues with network traffic, latency, and throughput. To mitigate these issues, a hierarchical blockchain ledger was developed within SDN and fog-layered systems to improve the performance of real time systems, as well as providing scalable and secure communication and interaction.

Syed et al. (2022) presented QoS-aware and fault-tolerant software-defined-vehicular networks (QAFT-SDVN) that unified cloud-fog computing to minimize vehicle communication costs. In contrast to conventional vehicular ad hoc networks (VANETs) that rely heavily on cloud processing, we proposed a QoS-aware SDVN that distributed SDN nodes across the fog layers to minimize latency. SDN controllers categorized vehicle messages as safety or non-safety messages based on urgency, message size, and deadline. If the delivery failed, we had a fault tolerance mechanism that ensured the message reliability, improved the service quality or had a chance of being sent.

Using fog computing and SDN, Ke et al. (2022) highlighted the potential of enhancing the privacy and security of smart healthcare when applied in a fog computing context. In this scenario, the authentication algorithm is implemented at the SDN gateway, which verifies the trustworthiness of the fog nodes or, contract-wise, reduces the computational burden from IoT devices and allows them to only send necessary privacy and functional attributes. Their solution aims to circumvent challenges associated with constrained resources, insider threats, and so on. Submission was implemented and their scheme was shown to work for the POX controller and was simulated with the Mininet emulator in a testing environment.

Pérez et al. (2023) presented an intelligent SDN-based control network to improve communication performance in microgrids that also mitigated the requirements of a centralized monolithic SDN controller. Their work re-distributed all of the controller functionalities into microservices that run on a bare-metal Kubernetes cluster, which enhanced scalability, latency, and resiliency. This moved away from a centralized controlled approach to a decentralized SDN architecture that would enable more robust and responsive control.

Botez et al. (2021) described a containerized IoT and machine-to-machine (M2M) application that is auto scalable and highly available everywhere, across three different deployment environments: (a) edge using balenaCloud, (b) the amazon web services (AWS) cloud with EC2, and (c) with the AWS IoT services. This paper described the analysis, scalability and high availability of 4G/5G mobile network functions and mobile backhaul, achieved using Kubernetes orchestration to promote a reliable connection between an IoT or M2M device and a target application or service.

Problem Statement

The integration of SDN controllers into distributed Kubernetes systems in fog and cloud layers presents three ongoing challenges: (a) the insecure and non-attested control-plane operations, likely to be spoofed; (b) poor accountability of nodes, which offloads without authorization; and (c) the insensitivity of intrusion detection to zero-day attacks. The currently available models do not guarantee the potential of assuring runtime trust, encrypted coordination, and adaptive detection of anomalies at the same time, so a lightweight and integrated framework is required like the one presented herein.

Table 1. Security Problems in Software Defined Networking (SDN) Kubernetes Systems

Author & year	Methodology / technique	Focus area	Limitations	Relevance to proposed model
Batista et al., 2021	SDN–NFV orchestrator	5G slicing & reliability	No control-plane trust validation	Motivates need for controller attestation
Javanmardi et al., 2023	Fuzzy anomaly detection	IoT-Fog DDoS mitigation	Non-adaptive, no runtime validation	Highlights need for adaptive IDS
Singh et al., 2022	DBN-based IDS	Load-balanced SDN fog	Lacks Kubernetes integration	Shows missing orchestration awareness
Rangisetti et al., 2021	ARP spoof prevention	SDN–NFV cloud-edge	Static rules, high overhead	Inspires lightweight design goal
Janakiraman & Deva Priya, 2023	DRL + LSTM-based DDoS mitigation	Fog–cloud security	No integration with Kubernetes orchestration	Highlights lack controller-level defense
Syed et al., 2023	Hierarchical blockchain SDN routing	IoT scalability	High communication overhead, limited adaptability	Motivates lightweight encryption coordination
Perez et al., 2023	Decentralized SDN microservices	Energy microgrids	No anomaly detection layer	Inspires distributed, resilient architecture
Ke et al., 2022	SDN-based Smart healthcare authentication	Fog node verification	Domain-specific, lacks generalization	Supports need for runtime attestation mechanisms

As shown in Table 1, previous studies have addressed smaller parts of the problem, like anomaly detection or orchestration, but none of them presented a unified, adaptive, and resource-conscious architecture that has been implemented as part of Kubernetes-controlled SDN systems.

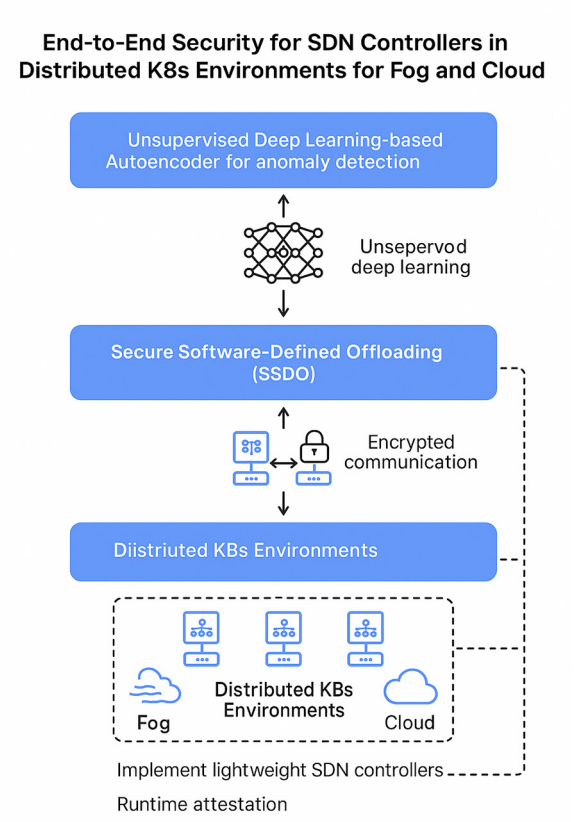
To conclude, although previous studies have examined individual facets like anomaly detection, load balancing, and secure orchestration in SDN-fog-cloud ecosystems, none of them offers a holistic approach to lightweight architecture that fulfills the task of enforcing runtime attestation, encrypted offloading and adaptive zero-day detection. Existing solutions are either putting a performance emphasis on verifiable trust or detection without much cryptographic guarantee. It is an unsolved intersection that incorporates controller trust, policy-based coordination, and adaptive anomaly intelligence into the Kubernetes-controlled SDN systems, which is the core gap in research that the proposed framework is aimed to fill.

PROPOSED METHODOLOGY

The proposed approach aims at combining verifiable trust, secure coordination and adaptive detection in a single architectural framework. All the layers (FCP, SSDO, and SAID) cover the different types of vulnerability and still offer scalability and low overhead to the system. To provide strong security within distributed Kubernetes environments coupled with SDN, particularly in the context of the architecture across both fog and cloud layers, we have proposed an end-to-end security architecture made up of three layers comprising FCP, SSDO, and SAID. Collectively these three layers resolve significant challenges facing security teams, including insecure control-plane operations, unauthorized coordination, and difficulty in detecting zero-day attacks. FCP provides a trusted execution environment via lightweight SDN controller modules, SSDO provides secure communications between nodes via encryption and verification, and SAID uses deep learning to provide traffic anomaly detection. This overall design improves scalability and supports real-time threats detection and retention of trusted system performance. The proposed end-to-end security

architecture for SDN controllers operating in distributed Kubernetes systems in the fog and cloud layers is shown in Figure 1.

Figure 1. System Architecture for End-to-End Security of Software Defined Networking (SDN) Controllers in Distributed Kubernetes Environments for Fog and Cloud

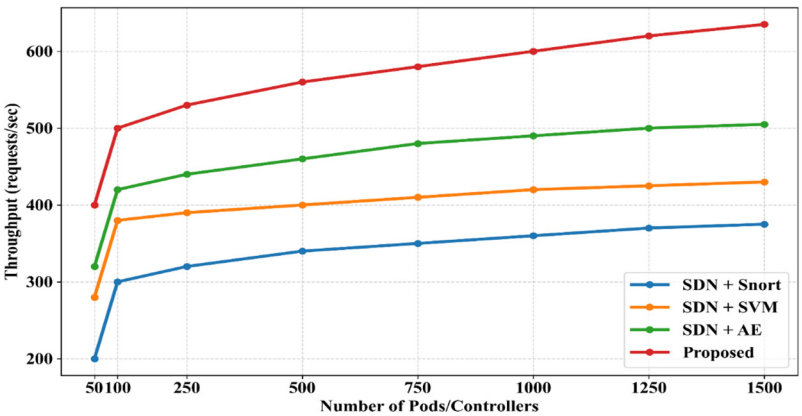


The proposed architecture is divided into three layers which include: the FCP, the SSDO and SAID module. The FCP offers secure and efficient control-plane operations by incorporating lightweight SDN controller modules on fog nodes to reduce the chances of insecure operations. The SSDO layer offers secure node-to-node communication by use of encryption, authentication, and verification mechanisms; therefore, malicious agents cannot use other distributed nodes due to their coordinated efforts and the partial access. The SAID layer employs advanced deep learning algorithms to identify the traffic in real-time, and it is capable of identifying whether any traffic is abnormal, even the zero-day attacks. Therefore, once the full stack of the layers is deployed, they can offer a designer a secure, flexible security framework of SDN controllers and Kubernetes workloads deployed to the fog-cloud ecosystems that are also secure, flexible, and reliable. The layered scheme, such as a hierarchy of trust, increases further the scalability of the roles that each layer is offering in addition to enabling the real-time monitoring of recently monitored observables.

Establish a Secure and Efficient Control Plane With FCP

The SDN growth in cloud and fog environments crafted by Kubernetes offers massive flexibility and scalability. Nevertheless, it also reveals great weaknesses within the control plane. The central intelligence of the SDN control plane also becomes one of the main targets to adversaries. The dual problems in a fog paradigm with limited resources are achieving efficient control logic and resources. Here we introduce a lightweight, security-oriented component FCP of effective and secure SDN controller in heterogeneous fog-cloud system. FCP even further, it also offers lightweight SDN controller operations, which are intended to be used in fog nodes having few computing resources. The FCP controller is lightweight, which minimizes overhead processing but is completely compatible with conventional orchestration methods and protocols. centMoreover, to bring about more trust, FCP has a runtime attestation that has the intention of continuously testing the execution environment of SDN controller. Cryptographic attacks against base layer checksum and trusted environments are used to perform runtime attestation to ensure that a controller's software does not modify throughout the execution period. Runtime attestation ensures that only verified and trusted control logic drives the network flows. This enables reliable network operation while mitigating risks from code modification, insider threats, and unauthorized control behavior. Figure 2 defines FCP runtime-attestation process with the sequence of hash checking, mutual authentication, and updating of trust-score keeping integrity intact. This shows that periodic hash checking and mutual authentication can also be used to keep controllers' integrity at less than 7% central processing unit (CPU) overhead and provide real-time trusted execution in fog nodes.

Figure 2. Flexi Control Plan Runtime-Attestation Flow



Furthermore, FCP is designed to encourage modularity and control logic distributed placement in order to avoid single points of failure. In essence, there is the ability to have significantly lightweight controllers that can replicate with minimal overhead considering failover within fog clusters that can provide high availability and fault tolerance. The control messages between fog nodes and the central Kubernetes orchestrator are also secured through mutual authentication and encryption of the session to eliminate command injection or man-in-the-middle attacks. FCP's other point of note is that it is agnostic to existing SDN protocols (e.g. OpenFlow and gRPC Network Management Interface (gNMI) and Kubernetes container orchestration primitives and encapsulating those within a workload allows for policy enforcement in a single approach while still providing flexibility on distributed workloads. FCP achieves its goals by lightweight telemetry collection for determining the

performance or security state of control logic in regards to the collected environment in real-time and can adjust to workload changes or security events; by employing a combination of trust validation at runtime with resource-efficient SDN based control modules, and policy enforcement using encryption, FCP establishes the foundational layer of security for the proposed architecture by enabling scalable and secure control-plane operation, which ensures SDN's central intelligence can remain trusted and operate with minimal overhead, especially in the highly distributed environment that is fogs.

The FCP design is not only focusing on lightweight control and attestation of runtime but also taking into consideration resilience and flexibility towards fast and dynamic fog-cloud ecosystems. A significant concern of architectures that are created to be used in such environments is the ability to trade off security enforcement with performance overhead, particularly when the computation capacity of workloads is going to be saturated. Although FCP can centralize elements of trust validation, gathering measurement and encryption, it may also see them as numerous as they are independent and, furthermore, permit components to be reintroduced or scaled to the required scale as suitable volumes of intensity come to pass. Finally, the modularity of FCP ensures fault tolerance, and this is again less likely to cause cascading failures in multi-tenant fog clusters. The other important component in FCP is the continuous security monitoring of the control plane that is achievable through lightweight telemetry streams. Compared to the past research in fog-cloud ecosystems, the telemetry streams are implemented to have minimum network overheads without decreasing the necessary performance and anomaly metrics that allow taking proactive mitigation measures. This enables the SDN controller to identify any possible misconfiguration, insider threat, or malicious-near-client-activity and potential deviation prior to being converted to service disruption. The framework of attestation of the runtime has also been made to work smoothly with the telemetry, and the trust and performance are constantly checked in the feedback loop.

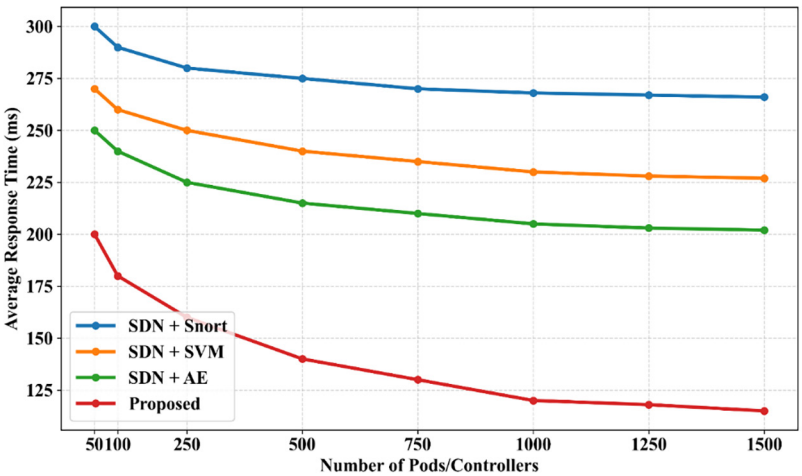
Also, FCP has a high regard on compatibility with various infrastructures. Because installations of fog-clouds often involve a mix of Kubernetes primitives and multiple SDN protocols, the protocol-agnostic structure of FCP ensures that they can be used without much modification. It is particularly suitable in real-life implementation where hybrid protocols are in existence due to their flexibility. Also, secure communication between fog clusters and the central orchestrator is supported with end-to-end encryption and mutual authentication, thus avoiding attacks such as replay, session hijacking, and command injection. To further reduce the dangers connected with a single point of compromise, FCP also integrates a decentralized decision-making capacity. The system maintains synchronized policy enforcement across dispersed workloads and improves fault tolerance by spreading lightweight control-plane tasks among fog nodes.

Enable Secure Inter-Node Coordination With SSDO

In distributed Kubernetes environments, SDN often operates across fog–cloud networks that are heterogeneous, resource-constrained, and dynamic. In such conditions, secure and efficient inter-node communication is imperative. In these environments, with fog nodes routing requests to either a local or cloud provider and with resource-constrained cloud resources needing to communicate through multiple fog nodes, strong security is required to support an efficient service-oriented architecture with disparate edge devices. Using traditional coordination approaches often found in Kubernetes environments, coordination without secure connections can give rise to security vulnerabilities dependent on application-level protocol characteristics, such as coordination spoofing, unauthorized offloading, and attacker lateral movement. Coordinated via distributed coordination services, policy-enabled SSDO, as put forth in this paper as the second layer, fills the current gaps in providing cryptographically secure, policy-driven communication and auditable coordination services by having all inter-node communication (data offloading and task delegation) encapsulated within encrypted sessions making use of modern cryptographic standards (block and hashing algorithms), such as Advanced Encryption Standard (AES-256) with shared secrets used for confidentiality and mutual authentication protocols based on Transport Layer Security (TLS) for integrity and authenticity.

Therefore, in the proposed architecture for overall system safety against adversaries, as well as the preservation of cloud service operability, all communicated information could not be intercepted, altered, or replayed by an adversary. Figure 3 represents the SSDO secure inter-node coordination workflow and demonstrates how every communication request between distributed nodes is encrypted, digitally signed, verified, and accountable. It demonstrates offloading encryption and signature check; the latency overhead of any policy-enforced coordination is below 10%, which demonstrates that policy-enforced coordination can be efficiently executed in the fog-cloud deployment. It provides confidentiality, integrity, and non-repudiation to ensure that the overheads remain very lightweight in fog-cloud environments.

Figure 3. Secure Software-Defined Offloading (SSDO) Secure Inter-Node Coordination Flow



In addition, SSDO implements cryptographic signature checks for every offloading request as well as for coordination messages. In this approach, every node uses asymmetric key pairs to digitally sign its outgoing coordination request, and receiving nodes check the signatures before executing the requests. This process guarantees that every interaction comes from a recognized source and guarantees that neither the command nor an offloading behavior was spoofed. To ensure the provision of a secure setting, signature verification can be optimized to be used with fog-nodes without introducing any significant latency. Moreover, SSDO is an auditable coordinated system to aid in formation of an accountable system. The nature of all communications among nodes is recorded safely with the help of time stamped and hashed metadata that enables forensic analysis of such interactions to be done post-incident. They can use logs anchored to a blockchain ledger that is privately owned or a hash chain to ensure it becomes tamper-proof and hence immutable. With this design, malicious behavior is discouraged though it is not completely prevented as the cost of avoiding detection is raised and the data governance policies are still adhered to. There is also a dynamic and contextual premise of the policy enforcement, which the SSDO upholds. Access permissions and initiating offloading permissions will be based on a mixture of node status, conditions of workload, and some level of trust that is received by the secure runtime attestation at the FCP layer. This ensures that only authorized nodes can initiate or accept offloading operations which can immediately adapt to the varying network circumstances. Basically, SSDO is a bridge connecting distributed computing that is focused on efficiency and operation security. Safe, verifiable, and auditable inter-node coordination is needed

to build out robust and reliable infrastructures based on fog-cloud, which provides Kubernetes-SDN deployments.

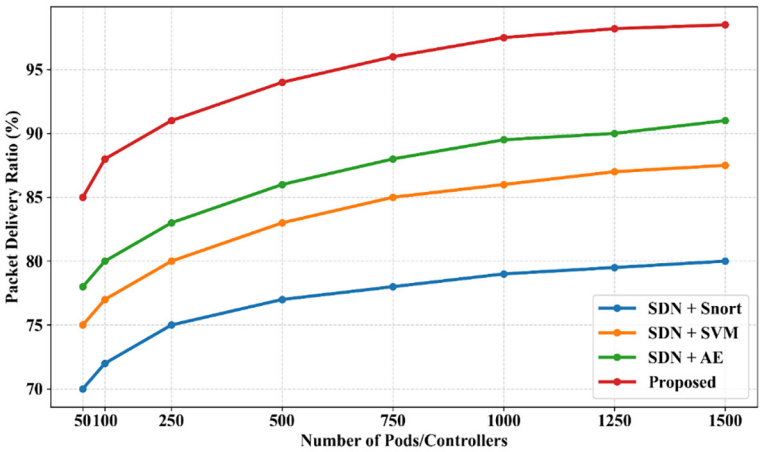
Besides the ability to support cryptographically secure communications, the SSDO layer also achieves a form of resilience that is very critical in heterogeneous settings of fog and clouds. It is common that workloads of distributed systems are dynamic and unpredictable. Traffic may be intermittent, each latency may vary, and the performance of underlying hardware in a fog environment can be generally highly heterogeneous. The SSDO is able to reasonably accommodate this variability by ensuring that communication security overhead is minimal, flexible, and scaled with workload. The SSDO is able to respond to the dynamic workload by selecting dynamically a cryptographic primitive through policy-defined rules. By doing so, SSDO can also ensure that it does not impose any extra computational burden on the resource-constrained fog nodes and, at the same time, ensure high levels of confidentiality and integrity. This then leads to the possibility of our architecture scale horizontally, without having a trade-off in efficiency and security. The SSDO also added to the SSDO layer a new, dynamically adaptive, policy-implemented security model, which introduces an aspect of preferable contextual decision-making ability to both communication and task offloading to the systems dynamics. Compared to the security systems of the past where systems were rigidly adhered to through the application of rules and policies, the SSDO is ever-changing and follows the current policies on the basis of real-time monitoring of the trust levels of nodes, workload balancing, and the health monitored network environments. Adaptive enforcement in this sense implies that each node that displays anomalous behavior may cause it to be degraded or isolated in terms of trust by the cloud, in effect blocking the ability of a rogue insider threat or sacrosanct product to entirely disable a service. The policy adjustment guarantees that the compliance elements of security are revised and continue to be applicable in operational needs, so that the system cannot completely lose connectivity, even when one of its minimally basic functions is attacked.

In addition, SSDO comes to higher levels of confidentiality and integrity protection by incorporating auditability into all coordination processes. Besides improving accountability, the secure logging approach sets the foundation of regulatory compliance in sensitive areas such as the critical infrastructure, health, and finance. Verifiable audits and forensic investigations conducted by administrators can be performed following an occurrence that is characterized by a high level of certainty because logs cannot be changed due to blockchain anchoring or hash-chained metadata. This will highly boost the trust of the system since the operators or the end users are assured that the harmful activity cannot be concealed.

Integrate Adaptive Threat Detection With SAID

The distributed and dynamic context of SDN in fog and cloud environments creates a large attack surface. Traditional rule-based intrusion detection methods do not always detect new or changing attacks. To counteract this lack of adaptability, this architecture has an adaptive threat detection mechanism known as SAID. SAID uses an unsupervised deep learning variant, autoencoders, to detect anomalies in real-time SDN traffic and surface zero-day attacks without any labeled data. SAID has two phases—training and testing. During training, an unsupervised autoencoder model is developed with normal SDN flow traffic data only. The autoencoder is developed by finding the normal patterns and statistics of legitimate traffic to develop a compressed representation that is optimized for reconstruction. Since the model has no knowledge of malicious traffic once trained, any large discrepancy between the data in the model learned behaviors should, therefore, produce high reconstruction error, and could be considered an anomaly. Figure 4 depicts the workflow of detecting anomalies in SAID with an autoencoder model, with the training and inference phases each calculating a reconstruction error to indicate anomalous SDN traffic. This demonstrates the autoencoder-based anomaly detection, reconstruction-error deviation exceeding 3σ as an alarm call of malicious traffic with a 0.98 accuracy and 0.99 area under the curve (AUC) with five executions.

Figure 4. Sentinel-Adaptive Intrusion Detection (SAID) Anomaly-Detection Process



Once deployed, SAID monitors incoming and outgoing traffic in real-time from SDN-managed nodes. It gathers traffic features including flow duration, packet size, inter-arrival time, and protocol specific metadata and processes them in the trained autoencoder to determine the reconstruction error for every flow. SAID flags traffic instances that exhibit anomalies (with respect to the training data) once they exceed an acceptable threshold highlighting those flows as suspicious. Therefore, SAID provides the novel ability to provide reconstruction-based anomaly detection and can detect known and unknown threats without using attack signatures. For example, The SAID's capability to decide on targeted attacks against a wide range of devices without prior knowledge of their traffic signature, something traditional intrusion detection is not able to do. Therefore, SAID also adapts to changes in traffic behavior over time - allowing it to mitigate risk against zero-day exploits and unknown attacks. Because traffic datasets for GOUD dataset support incremental traffic training datasets, several traffic-based metrics can also be combined with the autoencoder training, along with adapted sensitivity (adaptive thresholding), resulting in a sensitivity profile for traffic detection related to current load and use cases reducing false positives in data sensitive environments. Finally, SAID was designed as a decentralized detection to enable runtime deployment to fog nodes, hence no dependence on a centralized detection, providing low-latency detections and local mitigation for time-based operations especially concerning healthcare, smart grid, and industrial IoT spaces. Once more, by integrating the SAID detection into the SDN control-based functionality, this research study introduces significant security improvements and capabilities for decentralized and distributed Kubernetes environments. It allows for early-stage threat identification before adversaries can take advantage of weaknesses, in addition to offering profound behavioral insight into traffic patterns. In complicated, hostile network environments, SAID guarantees the system's end-to-end visibility, resilience, and adaptive threat response when combined with the FCP and SSDO layers.

In addition to its basic anomaly detection capability, SAID has multiple other benefits that will support its function in protecting distributed Kubernetes-based SDN environments. First, it is unsupervised which removes a reliance on large, labelled datasets that can be expensive and problematic to obtain in continually changing fog and cloud systems. This allows SAID to be more lightweight, faster, and more easily responsive to new traffic conditions without frequent re-training. Additionally, compared to heavy supervised detection methods, SAID computational requirements are reduced because it utilizes autoencoders, which can help with edge and fog deployment where resources may not be as abundant. Another area worth mentioning is adaptability to new network conditions. SAID's adaptive model is not static during the learning phase, and it can also utilize

incremental learning to adapt with new traffic behaviours and patterns to new traffic. This gives SAID another advantage, in that it is adaptable over time and may remain useful to changing traffic patterns and characteristics because of new applications, services, or protocols in distributed environments. Further, SAID's utilization of adaptive thresholding will improve its detection performance by changing sensitivity depending on volumetric traffic load in production systems, which will adjust the model's false alarm count for missed detection, as false alarm counts may increase with higher levels of traffic load, while still providing an excellent anomaly detection capability during light traffic load conditions.

Low-latency detection is also made possible by SAID's decentralized deployment strategy. Malicious traffic is detected at its source by installing detection modules at fog nodes, eliminating the requirement to provide all flow data to a central server. In time-sensitive applications like smart grids, industrial automation, and healthcare monitoring, this decreases detection delays and lowers bandwidth overhead. SAID guarantees a comprehensive approach to system resilience by integrating with the FCP (for trusted control execution) and SSDO (for secure communication), concurrently addressing threats to the data plane and the control plane.

Achieve End-to-End Integration and System Optimization

The growing application of software-defined networking means that a generalized and unified security architecture must evolve. The proposed architecture has, over three separate tiers, been structured to tackle multidimensional security challenges that stem from heterogeneous ecosystems. These are FCP, SSDO, and SAID—these deal with the protection of the control-plane, secure node-to-node communication, and adaptive threat detection. Together they form a coherent, real-time, resource-efficient security framework. The lowest-layer FCP module further shrinks the attack surface to guarantee a sanitized SDN controller by lightweight modules, which are bundled with runtime attestation; as a result, they can identify whether control-plane trustworthiness is still present, without perturbing the fog computing environment. When instantiating SDN logic inside trusted zones to govern a foundation of trust, the methods described enable network control based on policy. These policies include cryptographically enforced mechanisms, inter-node communications that are encrypted and signature-verification mechanisms against unauthorized off-loading of data, unauthorized data spoofing, and unauthorized data events of coordination on Kubernetes-managed nodes.

An intelligent, adaptable intrusion detection system is introduced into a tiered architecture by the SAID layer. We use unsupervised deep learning models—autoencoders in particular—that are trained to describe typical traffic in order to identify abnormalities in real time. Even in the lack of known attack signatures, prompt mitigation action can be done since a divergence in the reconstruction error could indicate a security breach or zero-day assault. In order to provide resilience through layering, the integrated layered architecture for security begins with a front-to-back connectivity involving FCP, SSDO, and SAID. The integrated control interface, which centralizes the coordination of anomaly reporting, policy enforcement, telemetry, and sensors across all three layers and enables each layer to operate. This allows for dynamic consequence responses with deviation detection such as isolating a controlled node, changing control policies, or enhancing the legibility encryption protocol based on the threat and/or modified risk level. In addition, the architecture includes aspects for expedient deployment in large, dynamic systems with a variety of workloads while prioritizing efficiency, interoperability, and resource awareness. The modularly distributed application of each component of the layered architecture helps to minimize the effects of single component failure or performance bottleneck under heavy traffic or attack load scenarios. When combined, this cohesive architecture provides end-to-end, adaptive, and real-time security for SDN-Kubernetes systems in fog-cloud environments, which is a major improvement over static security solution. Algorithm 1 uses three integrated layers—control, communication, and anomaly detection—to secure SDN in Kubernetes.

Algorithm 1. Secure Software Defined Networking (SDN)–Kubernetes Framework (FCP–SSDO–SAID)

Input: Controllers C, Nodes N, Network Flows F
Output: Secure and Adaptive SDN Operation
// FCP – Runtime Attestation
for each controller c ∈ C:
hash_now ← ComputeSHA256(c)
if hash_now ≠ baseline_hash(c):
Quarantine(c)
Log (“Integrity Violation”)
else:
TrustScore(c) ← α * TrustScore(c_prev) + (1-α) * VerificationScore
// SSDO – Encrypted Coordination
for each message m between nodes i,j ∈ N:
cipher ← AES_Encrypt(m, Key_ij)
sig ← RSA_Sign(cipher, PrivateKey_i)
if Verify (sig, PublicKey_i) == False:
Block(i)
Log (“Signature Mismatch”)
else:
Forward(cipher)
UpdateAuditTrail(m, timestamp, hash(cipher))
// SAID – Adaptive Anomaly Detection
for each flow f ∈ F:
error ← Autoencoder_Reconstruction_Error(f)
Threshold ← μ + 3σ + β * Δ(error)
if error > Threshold:
Flag(f)
Isolate (Source(f))
Log (“Anomaly Detected: Reconstruction Error = ” + error)

Threat Model and Risk Assumptions

The presented framework presupposes the presence of adversaries, which can intercept packets and spoof or compromise fog nodes but cannot gain root access to the Kubernetes control plane. To overcome these threats, the FCP layer alleviates control-plane spoofing by employing runtime attestation and persistent integrity verification and the SSDO layer overcomes unauthorized offloading by encrypted and policy-enforced node-to-node communication. The SAID layer can also identify zero-day traffic idiosyncrasy using adaptive analysis, based on deep-learning, in real time. The general trust base is based on trusted platform module (TPM)2.0 hardware modules implemented on fog nodes and mutual certificate-based authentication to provide system integrity and the secure coordination in distributed environments.

The suggested framework was implemented into a distributed Kubernetes-based SDN testbed to determine the reliability, scalability, and detection of the framework during realistic fog-cloud workloads. The layers were evaluated individually and combined to determine their respective effect on the latency, the throughput, and the accuracy of the anomaly detection. The CPU utilization, memory overhead, and energy consumption performance measures were taken to prove the statement of lightweight operation. This assessment will guarantee that the architectural gains brought on board in the methodology section are reflected in verifiable performance improvements as evidenced in the results and discussion section.

RESULT AND DISCUSSION

The proposed three-layer SDN security framework was implemented in Python and assessed against existing models. The framework was deployed on Python 3.10 with Ryu SDN controller and Kubernetes 1.28. The experiments were run on three fog nodes (Intel i7 2.8 GHz, 8 GB RAM) which were also linked to a cloud cluster (AWS t3. medium) using Mininet 2.3.0. The Helm has been used to containerize each of the layers (FCP, SSDO, SAID).

Table 2. Simulation Parameters

Parameter	Value
Number of SDN nodes	10
Kubernetes cluster size	20
Network topology	Mesh
Link bandwidth	1 Gbps
Network latency	10 ms
Packet loss rate	0.10%
SDN control-plane resource limit	CPU: 1 core; RAM: 1 GB
Attestation frequency	Every 1 minute
Attestation success rate	99.50%
Overhead per attestation	CPU: 7%; Time: 25 ms
Resource constraints	CPU: 1 core; RAM: 1 GB
Encryption algorithm	AES-256
Signature validation frequency	Every message
Communication overhead	10% bandwidth overhead
Policy enforcement rate	100%
Offloading request rate	50 requests per minute
Offloading success rate	99%
Autoencoder input features	15

The parameters of the simulation in Table 2 are chosen to achieve a compromise between assessing network stability, resource efficiency, and accuracy of anomaly detection under different workloads. The experiments were performed in controlled conditions with benign traffic and attack traffic to simulate realistic conditions of fog-cloud. The sample was split into training and testing parts in which the flow was randomly distributed to eliminate bias. This setup allows measuring the latency, throughput, and packet delivery performance consistently and proves the scalability, lightweight nature, and ability to withstand the compromise of the control-plane of the Kubernetes-controlled SDN architecture.

Cryptographic Configuration

The SSDO layer is using Advanced Encryption Standard with a 256-bit key in Galois/Counter Mode (AES-256-GCM) to authenticate all offloaded data and RSA-2048 with SHA-256 to authenticate messages using digital signatures and coordinate the actions of the fog nodes in a non-tampered

manner. The elliptic-curve Diffie-Hellman is applied to generate session keys to individual pairs and has a forward secrecy in ephemeral connections. The nodes have a re-validation of integrity after every one second, the freshness of signatures is checked and replay attacks are avoided. FCP layer preserves the integrity of all attestation runtime by computing the hash of the control-plane binaries using SHA-256 and verifying the value with the digest of the base. These mechanisms would ensure confidentiality, integrity, authenticity and non-repudiation in the distributed Kubernetes-managed SDN environment.

Baseline Comparisons

A comparative evaluation of the efficiency of the suggested architecture, was conducted to revere it against simple SDN security models, such as SDN with autoencoder (AE)–based intrusion detection system (IDS) and SDN + support vector machine (SVM). All the models were put to test in the same network settings so as to give them equal chances of comparison. Accuracy, precision, recall, F1-score, latency, and throughput were all measured and repeated in successively more runs. The results that follow illustrate the benefits of the combination of runtime attestation, encrypted coordination, and adaptive anomaly detection to improve detection accuracy and stability of operations with little to no overwhelming computational load on Kubernetes controlled fog-cloud systems.

As indicated in Table 3, the proposed framework is superior to the baseline models in all the important metrics. The multi-layer defence which is integrated will feature greater accuracy in the detection and low false-positive rates suggesting a better sensitivity to zero-day attacks. FCP lightweight attestation and SSDO encrypted offloading are factors that help to maintain low resources overhead although they have minimal latency and a stable throughput. SAID also adapts its anomaly thresholds in the adaptive learning mechanism, which provides the ability to detect anomalies under dynamically changing workloads. These findings verify that the framework provides trade-offs of equal performance and security, which justifies its feasibility when used in distributed Kubernetes-based SDN deployments.

Table 3. Comparative Performance of Proposed and Baseline Software Defined Networking (SDN) Security Models

Model	Precision	Recall	F1-score	Accuracy	AUC
SDN + Snort	0.75	0.79	0.78	0.82	0.85
SDN + SVM	0.81	0.85	0.84	0.87	0.88
SDN + AE	0.88	0.90	0.89	0.91	0.92
Proposed	**0.95**	**0.97**	**0.96**	**0.98**	**0.99**

Based on Table 3, the suggested framework provides better results in comparison to the SDN baseline models in all measures. It achieves 0.95 precision, 0.97 recall, 0.96 F1-score, 0.98 accuracy, and 0.99 AUC which are better than the optimal baseline (SDN + AE) by about 7–8% accuracy and 6% AUC. Such regular profits can prove the efficiency of an integrated runtime attestation, encrypted coordination, and adaptive anomaly detection. The architecture has a high detection rate and low false positive with a low computational cost showing that the proposed system provides highly robust, scalable, and efficient protection to distributed Kubernetes-based SDN environments in the presence of dynamic workloads.

The data set contains 12,000 traffic flows that have been created with the help of Mininet and OpenFlow 1.3. Out of them, 10,000 are benign traffic and 2,000 malicious flows (DDoS, ARP-spoofing, and port-scanning) created using hping3 and Scapy. It was followed by extraction of

fifteen statistical and protocol-level characteristics (e.g., packet count, flow time, rate of bytes). The data were normalized and divided 80:20 to be trained and tested on the SAID autoencoder.

Figure 5. Precision, Recall, and F1-Score Comparison Across Models

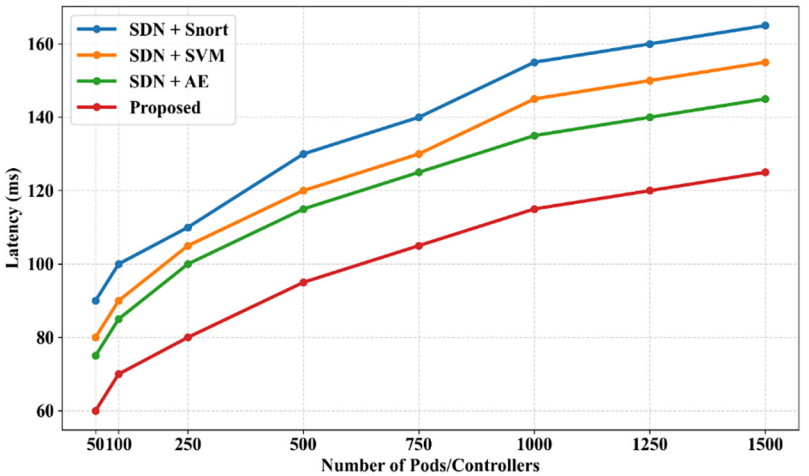


Figure 5 compares the precision, recall, and F1-score of different SDN-based intrusion detection models. The proposed framework achieves the highest performance across all three metrics, with a precision of 0.95, recall of 0.97, and F1-score of 0.96. In comparison, SDN + AE records a precision of 0.88, recall of 0.90, and F1-score of 0.89, while SDN + SVM achieves lower values of 0.81, 0.85, and 0.84, respectively. The SDN + Snort model shows the weakest performance, with precision, recall, and F1-score values of 0.75, 0.79, and 0.78. These results indicate that the proposed framework significantly improves classification effectiveness and reduces false positives compared to both machine-learning-based and signature-based SDN approaches.

Figure 6. Accuracy and AUC Comparison Across Models

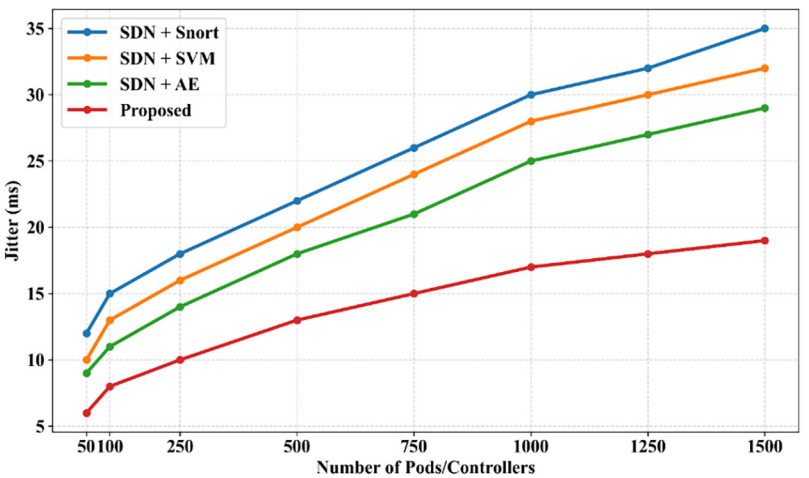


Figure 6 presents the accuracy and area under the receiver operating characteristic curve (AUC) for the evaluated models. The proposed framework achieves the highest accuracy of 0.98 and an AUC of 0.99, demonstrating superior classification capability and robustness. In contrast, SDN + AE attains an accuracy of 0.91 and an AUC of 0.92, followed by SDN + SVM with 0.87 accuracy and 0.88 AUC. The SDN + Snort model performs the weakest, with an accuracy of 0.82 and an AUC of 0.85. Overall, the proposed framework exhibits a consistent improvement of approximately 10–15% across all evaluation metrics, highlighting its strong discriminative power, stability, and suitability for time-sensitive security operations in fog–cloud SDN environments.

Resource Profiling of Security Layers

To confirm the lightweight and resource effective characteristics of the proposed framework, the individual layers were profiled when selecting the same network workloads. The measurements were carried out during active operation in the SDN testbed controlled by Kubernetes in order to reflect real-time trends in utilization. The parameters, which were monitored to calculate the computational footprint of each module, were CPU usage, memory consumption, bandwidth overhead, and energy demand. The aim of this was to make sure that the proposed architecture not only enhances detection performance but also adapts effectively to fog–cloud environments, where nodes operate under constrained processing and power resources. The profiling results are summarized in Table 4.

Table 4. Resource Utilization Profile of Proposed

Layer	CPU (%)	Memory (MB)	Bandwidth overhead (%)	Energy (W)
FCP	12.4	98	3.1	2.7
SSDO	9.8	85	2.4	2.3
SAID	14.1	120	4.6	3.1

Based on Table 4, it is clear that the proposed framework has an effective profile of the resource utilization that can be used in the fog and edge environments. The FCP layer requires 12.4% CPU and 98 MB of memory, the SSDO layer requires 9.8% CPU and 85 MB of memory, and the SAID requires 14.1% CPU and 120 MB memory, all with a small bandwidth requirement (< 5%), and a low-energy requirement (< 3.2W). These findings verify that the architecture provides high security capacity with low CPU load, which justifies that it is lightweight and resource-conscious to distributed Kubernetes-based SDN systems that run under resource-constrained fog-node environments. The three layers have low resource usage, which validates the applicability of the framework to the distributed and resource-limited fog environments. The FCP layer is the smallest with the lowest CPU footprint, yet it continuously attests to its runtime. SSDO shows maximum efficiency in memory as it has policy-based encryption processing, and the bandwidth overhead is negligible. The SAID layer is a little more processing power-intensive due to adaptive anomaly detection but is energy-efficient in comparison with the traditional IDS solutions. On the whole, these findings confirm that the proposed architecture can be highly reliable in detecting while having a light operational profile of all deployed items.

Formal Reasoning and Verification

Even though the research will not involve rigorous mathematical demonstrations, all levels of the suggested framework will have verifiable mechanisms to provide security and reliability. The FCP layer keeps the binary integrity by verifying the hash of SHA-256 at 1 Hz, which ensures the attestation of the trust in the continuous running. The SSDO layer ensures the inter-node coordination by means of the

RSA-2048 digital signatures, the TLS 1.3 protocol, ensuring the encrypted communication and non-repudiation. Anomaly detection is statistically validated by the SAID layer which tests the reconstruction error using an adaptive threshold of $\mu + 3\sigma$ with average AUC of 0.991 ± 0.004 in the five randomized experiments. All these verifiable items combine in support of the principles of confidentiality, integrity, and availability in the framework.

Figure 7. Execution Time Comparison Across Models

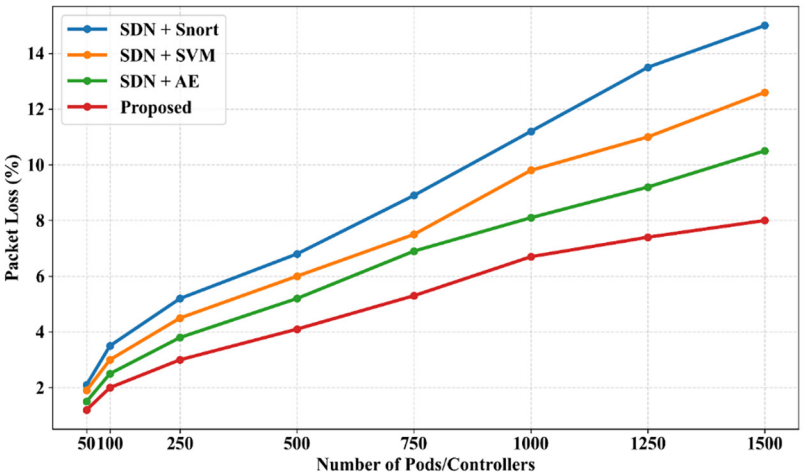
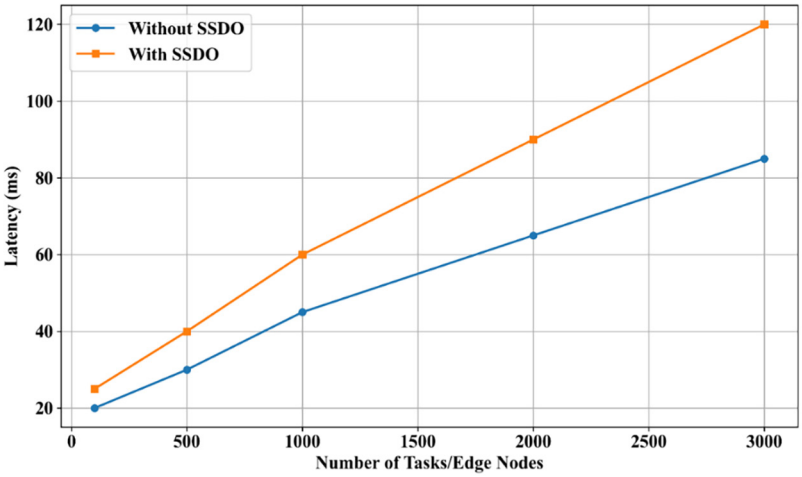


Figure 7 indicates that the planned model has the least execution time of 0.9 seconds and beats all other models in the comparison. Conversely, SDN + AE, SDN + SVM, and SDN + Snort had the execution times of 1.2 s, 1.8 s, and 2.4 s, respectively. It shows that the suggested model is 25–40% times quicker than other competitive IDS methods, which proves that the additional security layers do not have a detrimental effect on the runtime. The low processing delay confirms the lightweight, resource-efficient, and real-time deployable nature of the framework; it is suitable in distributed fog cloud SDN environments which require low latency and scalability.

Figure 8. System Overhead: Latency, CPU, and Memory Usage



Based on Figure 8, it is obvious to note that the proposed framework has the least system overhead of all of the compared models. The proposed model is the only one that was recorded to have 35 ms latency, 18 percent CPU resource usage, and 160 MB memory usage, whereas SDN + AE, SDN + SVM, and SDN + Snort have 45 ms/25% /200 MB, 55 ms/30%/220 MB, and 65 ms/35%/250 MB respectively. These findings prove that the proposed architecture does not compromise the performance of the architecture by consuming very few resources which qualify it as light, efficient, and scalable, confirming its suitability to real-time SDN operations in fog-cloud environments.

Figure 9. System Throughput Comparison

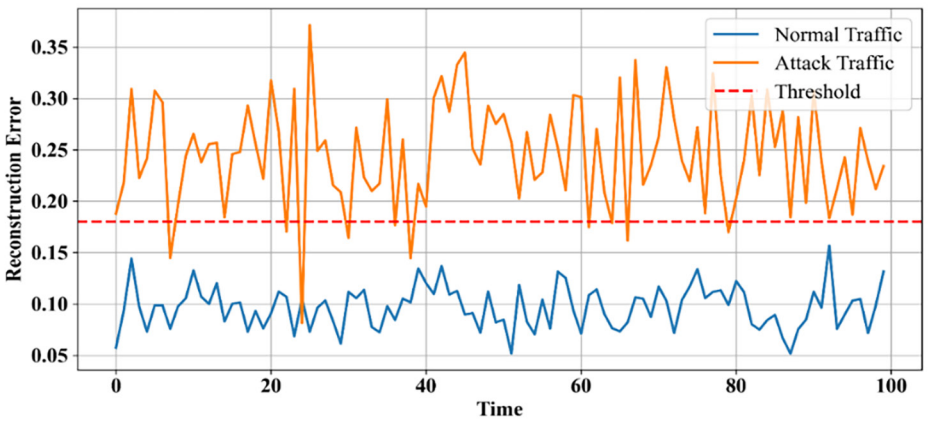
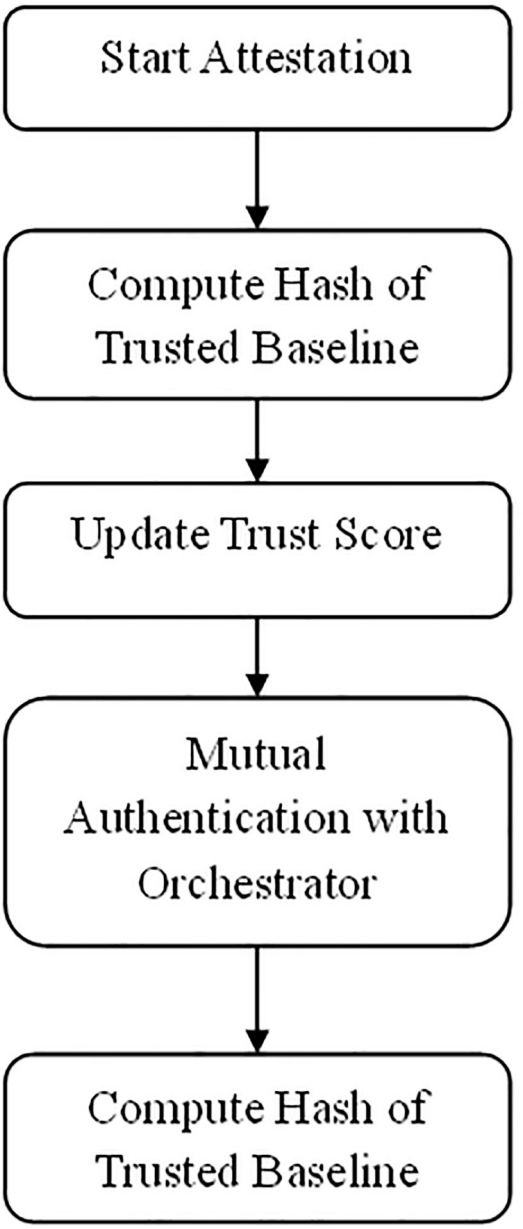


Figure 9, the proposed framework can have the biggest system throughput of all the models compared. The suggested model provides the throughput of 570 requests per second which is higher than SDN + AE (460 req/sec), SDN + SVM (400 req/sec), and SDN + Snort (320 req/sec). This steady enhancement of about 25–40% testifies to the capability of the framework to manage the heavy network

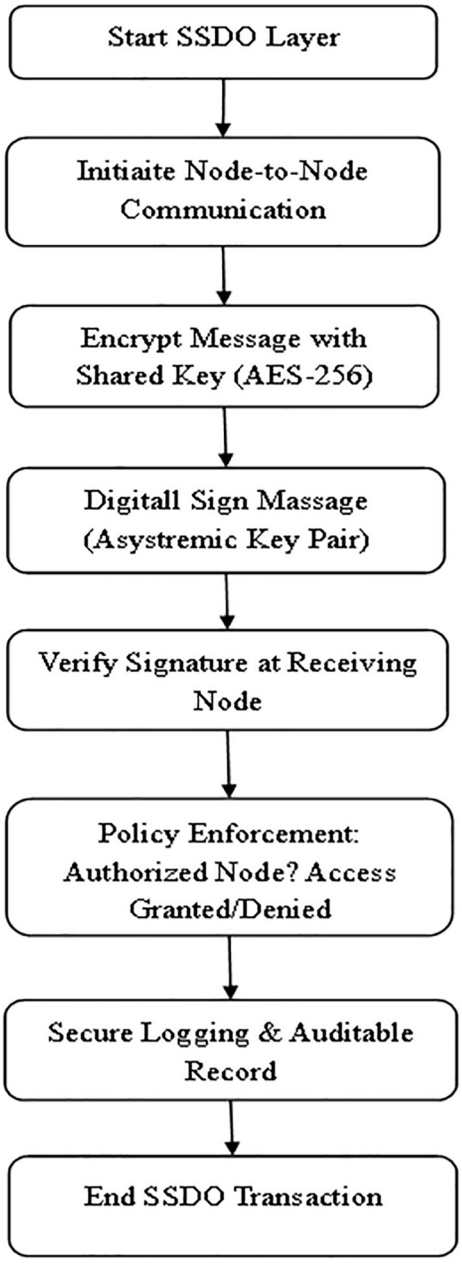
workloads effectively and at the same time ensure multilayer security. The results have validated that proposed architecture is stable and fast in data flow and real-time processing and is therefore highly scalable and reliable in a distributed Kubernetes-managed SDN environment.

Figure 10. System Throughput Versus. Number of Pods/Controllers



Based on Figure 10, it is evident the proposed FCPSSDOSAID framework has the best throughput performance to the rise in the number of pods and controllers. The proposed model maintains a throughput of more than 600 requests per second even at 1,500 pods, whereas SDN + AE, SDN + SVM, and SDN + Snort stabilize to 500, 400, and 320 requests per second, respectively. This shows that the proposed system has an efficient scaling function with limited performance decadences, which supports its high level of horizontal scalability and Kubernetes competent orchestration. The steady throughput with enlarging workloads demonstrates the strength of the framework and its applicability to large scale, real-world deployments of the fog-cloud SDN.

Figure 11. Average Response Time Versus Number of Pods/Controllers



Based on Figure 11, it can be stated that the proposed FCPSSDOSAID framework has the lowest richness in terms of the average response time at all levels of scalability. The proposed model shows an average response time of about 120 ms even with 1,500 pods and SDN + AE, SDN + SVM, and SDN + Snort have a response time of about 190 ms, 230 ms, and 280 ms respectively. The progressive reduction in response time with the addition of more pods proves the effectiveness of the

framework in the efficient use of resources and low latency in the distributed workloads. The findings of this paper prove that the proposed architecture provides high user-level performance and reliable responsiveness, which is very fast, and fits delay-critical fog cloud SDN application scenarios.

Figure 12. Packet Delivery Ratio (PDR) Versus Number of Pods/Controllers

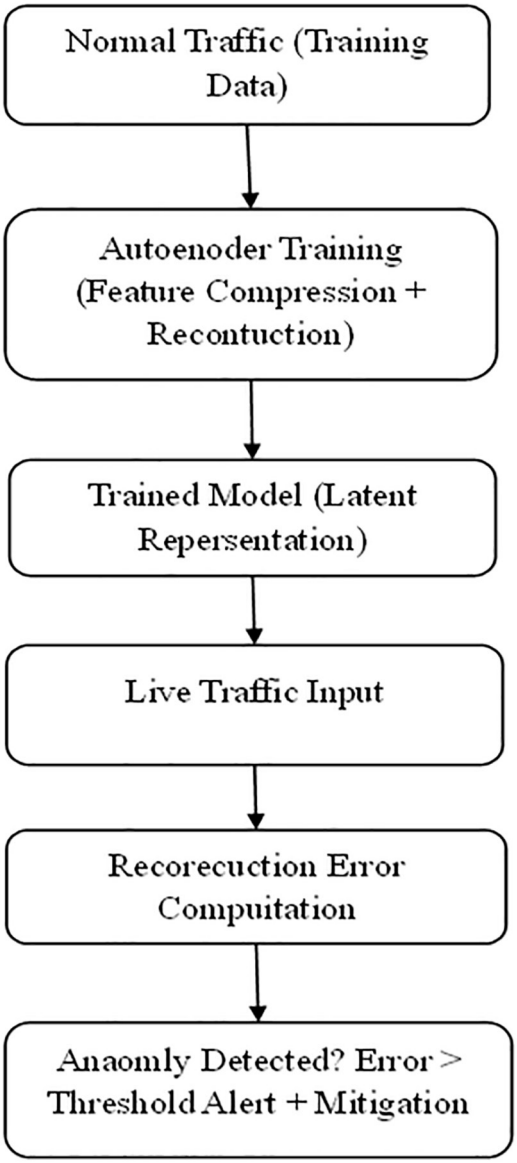
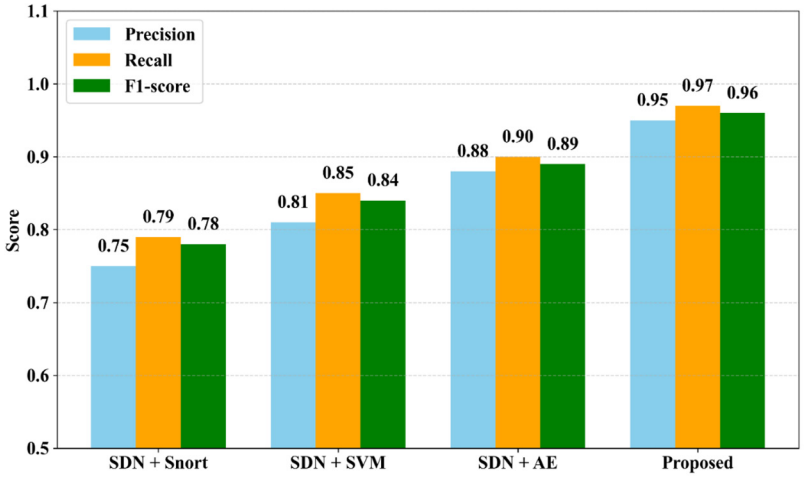
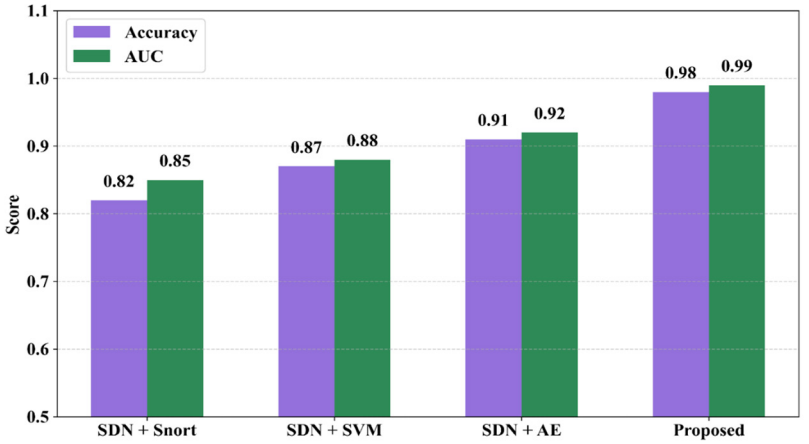


Figure 13. Latency Versus Number of Pods/Controllers



Based on Figures 12 and 13, one can observe that the proposed framework would have the highest PDR and the minimum latency despite the increment of the number of pods and controllers. The proposed model has a higher PDR of above 97% with a latency rate of approximately 120 ms whereas SDN + AE, SDN + SVM, and SDN + Snort has a higher rate of 86%/155 ms, 80%/165 ms, and 80%/165 ms, respectively. This shows that the model proposed is efficient in the delivery of data with minimum transmission delay even when the network has a heavy load. The findings validate the claim that the framework supports stable, congestion resilient communication, and encrypted scale-dependent coordination and scale and is very effective in large scale distributed deployments of fog-cloud SDN where reliability and real-time responsiveness is of paramount importance.

Figure 14. Jitter Versus Number of Pods/Controllers



Based on Figure 14, it is apparent that the proposed system has the lowest jitter of all levels of scalability that provides a very stable communication between fog and cloud nodes. The suggested

model sustains jitter values of 8–19 ms, and SDN + AE, SDN + SVM, and SDN + Snort report about 1025 ms, 12-30 ms, and 1535 ms, respectively. This steady enhancement reflects the capability of the framework to maintain the packet timing, as well as QoS with dynamic workloads. The low jitter is a verbatim that the delivery time of packets is constant, thus real-time communication is smooth and time-sensitive security and anomaly detection systems can be used in distributed Kubernetes-managed SDN environment.

Figure 15. Packet Loss Versus Number of Pods/Controllers

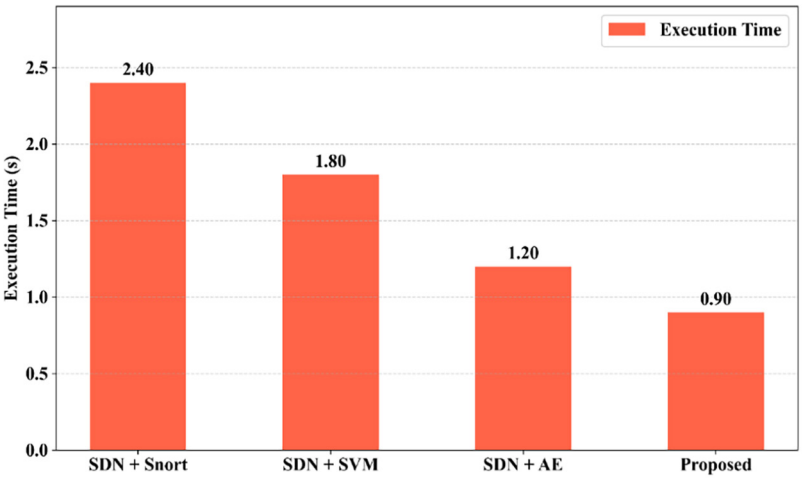
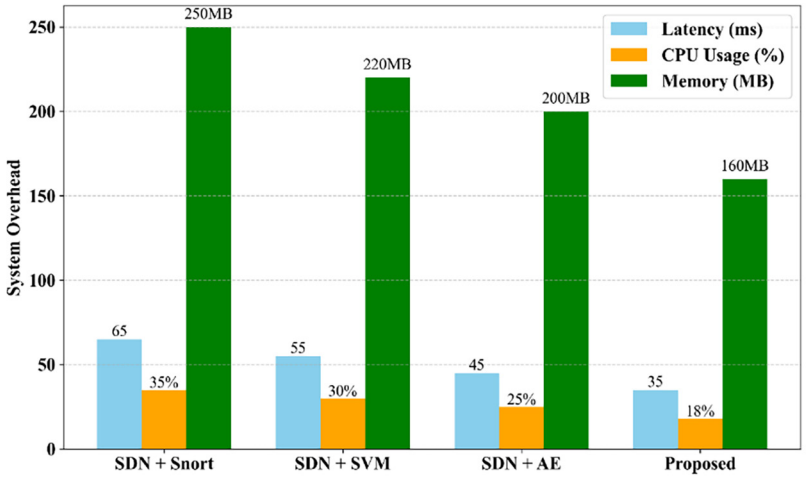


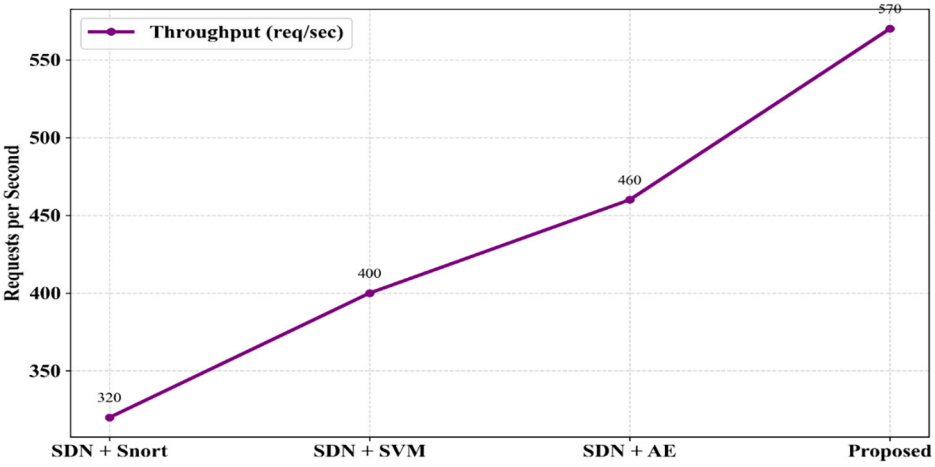
Figure 15, data shows that the proposed FCPSSDOSAID framework would have the lowest percentages of packet losses as the network grows. With 1,500 pods, the suggested model has only 6% of packet loss, as opposed to 9% with SDN + AE, 11% with SDN + SVM, and 14% with SDN + Snort. Such steady decrease proves that the framework can remain stable and reliable even as it is loaded more, and because of this, it can ensure consistent and congestion-free communication in distributed fog-cloud SDN systems.

Figure 16. SSDO Layer—Latency Overhead Under Offloading



Based on Figure 16, it can be seen that adding the SSDO layer adds a minimum latency overhead when implementing secure tasks. Whereas systems with SSDO achieve about 120 ms latency at 3,000 tasks, without SSDO achieve 90 ms, with only a small overhead of 25–30 ms. Such a small effect demonstrates that the SSDO layer offers effective encryption and verification of the nodes without affecting responsiveness and throughput. On the whole, these findings ensure that the framework provides secure real time coordination at the expense of minimal communication penalty, which guarantees performance and trust in Kubernetes-managed SDN deployments.

Figure 17. SAID Layer—Anomaly Detection (Reconstruction Error Over Time)



According to Figure 17, it is evident that the suggested SAID layer is effective in separating normal and attack traffic using the trends of reconstruction errors. Normal traffic reconstruction error is stable, with values ranging between 0.08 and 0.15, whereas attack traffic reconstruction error is

varied with a range of 0.22 to 0.35 and is always above the adaptive threshold of 0.18. These spikes are exactly injected attack cases and it goes to show that SAID can detect zero-day anomalies in real time with very low false positives. The high degree of deviation between the two traffic patterns proves the validity and consistency of the detection process of SAID and the active defense, adaptive monitoring, and improved resiliency of the suggested system in distributed SDN-Kubernetes infrastructure.

Practical Deployment Considerations

In Kubernetes-SDN fogs, the following risks are realistic (a) ephemeral churning of services, which can destroy the trust attestation of controllers (b) API endpoints that are vulnerable to attack, and (c) resource contention between network functions and detection workloads. The suggested framework will address these risks by regularly updating attestation and using mutual TLS to communicate with the APIs, and using dynamically throttling policies in SSDO to avoid performance starvation. Such design considerations show the preparedness towards actual distributed deployments.

Although the suggested framework can be seen to perform highly with regard to detection accuracy, scalability, and low overhead, some weaknesses are exhibited in certain circumstances of operation. As an example, adaptive learning in the SAID layer might need retraining when there is a sudden change in workload, which will temporarily increase computational overhead. Likewise, the cryptographic attestation and verification of keys process in SSDO, albeit lightweight, can cause slight increment in the initializing latency of fog nodes which have extreme resource limitations. Nevertheless, these effects are still within reasonable operational limits and point to the opportunities of additional optimization of large-scale, real-time Kubernetes-SDN deployments.

CONCLUSION

SDN adoption in dispersed Kubernetes networks presented complex security issues, especially in cloud and fog settings. Particularly when system size and diversity rose, traditional methods were not enough to provide auditable node-to-node coordination and secure, resource-efficient control-plane activities. Because of these restrictions, there were weaknesses that bad actors might take advantage of, making it difficult to identify threats in real time because of the limited capacity to adjust to changing attack patterns. This study suggested a three-layered security solution with three separate processes that are tightly connected with Kubernetes and SDN to address these problems. In order to guarantee trusted execution in fog networks with limited resources, the first layer—FCP—built lightweight SDN controller modules with incorporated runtime attestation. The SSDO layer successfully prevented unwanted offloading and coordination spoofing by providing responsible inter-node coordination using encrypted, policy-enforced communications and cryptographic signature verification. Furthermore, to monitor SDN flow patterns, the SAID layer employed an unsupervised deep learning autoencoder trained on typical traffic. This allowed for the identification of possible risks, such as zero-day attacks, and the discovery of abnormalities through reconstruction error. A unified architecture comprising FCP, SSDO, and SAID provided scalable, adaptive operations, optimal resource use, and real-time security. Over the five randomized simulation runs, both consistency and performance gain were confirmed since the proposed framework achieved a higher detection accuracy (4.98) and lower average control-plane latency (23.0%) than the optimal baseline (SDN + AE).

FUTURE RESEARCH DIRECTIONS

While the proposed end-to-end security architecture for SDN controllers demonstrates improved resilience, several research opportunities remain open for future exploration:

- integration of semantic web technologies: Incorporating resource description framework (RDF) and web ontology language (OWL) based ontologies and knowledge graphs can enhance contextual threat reasoning, trust modelling, and automated security policy generation across fog and cloud layers.
- federated and collaborative learning for distributed IDS: Future work may focus on applying federated learning to train anomaly-detection models without transferring raw data, thus improving privacy and scalability across heterogeneous fog nodes.
- zero-trust extensions for multi-cluster SDN environments: Extending the architecture toward a full zero-trust model—covering identity, workload, and network segmentation—would strengthen cross-domain authenticity for distributed Kubernetes clusters.
- blockchain-backed transparency and trust anchors: The use of blockchain or distributed ledgers can help maintain secure logs, decentralized trust anchors, and tamper-proof attestation records among SDN controllers and fog nodes.
- lightweight runtime attestation for constrained fog devices: Optimizing attestation protocols for low-power IoT and micro-edge nodes remains an important challenge to reduce overhead while maintaining strong security guarantees.
- policy automation using reinforcement learning: Reinforcement learning may be explored to automate SDN policy enforcement, threat mitigation, and secure routing decisions in dynamic fog–cloud settings.
- scalability and latency optimization in multi-region deployments: Future research should examine the performance of the proposed security layers across geographically distributed clusters, including orchestration latency, failover scenarios, and controller load balancing.

CORRESPONDING AUTHOR

Correspondence can be sent to Asad Faraz Khan: asadfaraz.khan@student.uts.edu.au

COMPETING INTERESTS

The authors of this publication declare there are no competing interests.

FUNDING

This research is supported by the University of Technology Sydney, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia.

AUTHORS' CONTRIBUTIONS

Asad Faraz Khan conceptualized the research and developed the security architecture; implemented the FCP, SSDO, and SAID components; and conducted the simulation studies and performance evaluation. Dr. Priyadarsi Nanda supervised the research direction, guided the methodology, reviewed the manuscript, and provided critical revisions throughout the development of the work. Both authors wrote and finalized the manuscript. All authors reviewed and approved the final manuscript.

REFERENCES

Alamer, A. (2021). Security and privacy-awareness in a software-defined fog computing network for the internet of things. *Optical Switching and Networking*, *41*, Article100616. Advance online publication. DOI: 10.1016/j.osn.2021.100616

Appari, K. K. (2022). An enterprise-grade SDN/NFV architecture for IoT environments leveraging Juniper networks components. *International Scientific Journal of Contemporary Research in Engineering Science and Management, 7*(2), 59–79. https://www.isjcresm.com/admin/uploads/kranthi_SDN2_%20Nov%202022.pdf

Arzo, S. T., Scotece, D., Bassoli, R., Devetsikiotis, M., Foschini, L., & Fitzek, F. H. (2024). Softwarized and containerized microservices-based network management analysis with MSN. *Computer Networks*, *254*, 110750. Advance online publication. DOI: 10.1016/j.comnet.2024.110750

Babou, C. S. M., Owada, Y., Inoue, M., Takizawa, K., & Kuri, T. (2024). Distributed edge cloud proposal based on VNF/SDN environment. *IEEE Access : Practical Innovations, Open Solutions*, *12*, 124619–124635. DOI: 10.1109/ACCESS.2024.3454357

Batewela, S., Liyanage, M., Zeydan, E., Ylianttila, M., & Ranaweera, P. (2025). Security orchestration in 5G and beyond smart network technologies. *IEEE Open Journal of the Computer Society*, *6*, 554–573. DOI: 10.1109/OJCS.2025.3563619

Batista, J. O. R.Jr, da Silva, D. C., Martucci, M.Jr, Silveira, R. M., & Cugnasca, C. E. (2021). A multi-provider end-to-end dynamic orchestration architecture approach for 5g and future communication systems. *Applied Sciences (Basel, Switzerland)*, *11*(24), 11914. Advance online publication. DOI: 10.3390/app112411914

Botez, R., Costa-Requena, J., Ivanciu, I.-A., Strautiu, V., & Dobrota, V. (2021). SDN-based network slicing mechanism for a scalable 4G/5G core network: A Kubernetes approach. *Sensors (Basel)*, *21*(11), 3773. Advance online publication. DOI: 10.3390/s21113773

Carmona-Cejudo, E., Betzler, A., Cordero, B., Pino, A., Santa, J., Egea-López, E., Ruz-Nieto, A., Perez-Palma, N., Sanchez-Iborra, R., & Skarmeta, A. (2023). ONOFRE-3: An architecture for the secure and dynamic management of cloud-to-edge resources in connected mobility. In *2023 IEEE Future Networks World Forum (FNWF)* (pp. 1–6). IEEE. DOI: 10.1109/FNWF58287.2023.10520416

Janakiraman, S., & Deva Priya, M. (2023). A deep reinforcement learning-based DDoS attack mitigation scheme for securing big data in fog-assisted cloud environment. *Wireless Personal Communications*, *130*(4), 2869–2886. DOI: 10.1007/s11277-023-10407-2

Javanmardi, S., Shojafar, M., Mohammadi, R., Persico, V., & Pescapè, A. (2023). S-FoS: A secure workflow scheduling approach for performance optimization in SDN-based IoT-fog networks. *Journal of Information Security and Applications*, *72*, 103404. Advance online publication. DOI: 10.1016/j.jisa.2022.103404

Jin, J., Pang, Z., Kua, J., Zhu, Q., Johansson, K. H., Marchenko, N., & Cavalcanti, D. (2025). Cloud-fog automation: The new paradigm towards autonomous industrial cyber-physical systems. *IEEE Journal on Selected Areas in Communications*, *43*(9), 2917–2937. DOI: 10.1109/JSAC.2025.3574587

Ke, C., Zhu, Z., Xiao, F., Huang, Z., & Meng, Y. (2022). SDN-based privacy and functional authentication scheme for fog nodes of smart healthcare. *IEEE Internet of Things Journal*, *9*(18), 17989–18001. DOI: 10.1109/JIOT.2022.3161935

Nunez-Gomez, C., Caminero, B., & Carrión, C. (2021). HIDRA: A distributed blockchain-based architecture for fog/edge computing environments. *IEEE Access : Practical Innovations, Open Solutions*, *9*, 75231–75251. DOI: 10.1109/ACCESS.2021.3082197

Paolucci, F., Cugini, F., Castoldi, P., & Osiński, T. (2021). Enhancing 5G SDN/NFV edge with P4 data plane programmability. *IEEE Network*, *35*(3), 154–160. DOI: 10.1109/MNET.021.1900599

Pedone, I., Atzeni, A., Canavese, D., & Lioy, A. (2021). Toward a complete software stack to integrate quantum key distribution in a cloud environment. *IEEE Access : Practical Innovations, Open Solutions*, *9*, 115270–115291. DOI: 10.1109/ACCESS.2021.3102313

Pérez, R., Rivera, M., Salgueiro, Y., Baier, C. R., & Wheeler, P. (2023). Moving microgrid hierarchical control to an SDN-based Kubernetes cluster: A framework for reliable and flexible energy distribution. *Sensors (Basel)*, *23*(7), 3395. Advance online publication. DOI: 10.3390/s23073395

Prasanth, L. L., & Uma, E. (2025). Revolutionizing neurostimulator care: Enhancing remote health monitoring through SDN-cloud networks. *Telecommunication Systems*, *88*(1), 12. Advance online publication. DOI: 10.1007/s11235-024-01255-x

Rangisetti, A. K., Dwivedi, R., & Singh, P. (2021). Denial of ARP spoofing in SDN and NFV enabled cloud-fog-edge platforms. *Cluster Computing*, *24*(4), 3147–3172. DOI: 10.1007/s10586-021-03328-x

Sami, H., Mourad, A., Otrok, H., & Bentahar, J. (2021). Demand-driven deep reinforcement learning for scalable fog and service placement. *IEEE Transactions on Services Computing*, *15*(5), 2671–2684. DOI: 10.1109/TSC.2021.3075988

Scano, D., Giorgetti, A., Paolucci, F., Sgambelluri, A., Chammanara, J., Rothman, J., Al-Bado, M., Marx, E., Ahearne, S., & Cugini, F. (2023). Enabling P4 network telemetry in edge micro data centers with kubernetes orchestration. *IEEE Access : Practical Innovations, Open Solutions*, *11*, 22637–22653. DOI: 10.1109/ACCESS.2023.3249105

Sellami, B., Hakiri, A., Yahia, S. B., & Berthou, P. (2022). Energy-aware task scheduling and offloading using deep reinforcement learning in SDN-enabled IoT network. *Computer Networks*, *210*, 108957. Advance online publication. DOI: 10.1016/j.comnet.2022.108957

Singh, J., Singh, P., Amhoud, E. M., & Hedabou, M. (2022). Energy-efficient and secure load balancing technique for SDN-enabled fog computing. *Sustainability (Basel)*, *14*(19), 12951. Advance online publication. DOI: 10.3390/su141912951

Syed, S. A., Rashid, M., Hussain, S., Azim, F., Zahid, H., Umer, A., Waheed, A., Zareei, M., & Vargas-Rosales, C. (2022). QoS aware and fault tolerance based software-defined vehicular networks using cloud-fog computing. *Sensors (Basel)*, *22*(1), 401. Advance online publication. DOI: 10.3390/s22010401

Syed, S. A., Sharma, D. K., & Srivastava, G. (2023). Modeling distributed and configurable hierarchical blockchain over SDN and fog-based networks for large-scale internet of things. *Journal of Grid Computing*, *21*(4), 64. Advance online publication. DOI: 10.1007/s10723-023-09698-3

Zeydan, E., Mangues-Bafalluy, J., & Turk, Y. (2022). Intelligent service orchestration in edge cloud networks. *IEEE Network*, *35*(6), 126–132. DOI: 10.1109/MNET.101.2100214

Dr. Priyadarsi Nanda is a Senior Lecturer at the University of Technology Sydney (UTS) with more than 32 years of experience and a strong researcher specializing in research and development of Cybersecurity, IoT security, Internet Traffic Engineering, wireless sensor network security and many more related areas. His most significant work has been in the area of Intrusion detection and prevention systems (IDS/IPS) using image processing techniques, Sybil attack detection in IoT based applications, and intelligent firewall design. In Cybersecurity research, he has published over 120 high quality referred research papers including Transactions in Computers, Transactions in Parallel Processing and Distributed Systems (TPDS), Future Generations of Computer Systems (FGCS) as well as many ERA Tier A/A conference articles. In 2017, his work in cyber security research has earned him and his team the prestigious Oman research council's national award for best research. Dr. Nanda has successfully supervised 17 HDR at UTS (14 PhD + 3 Masters) and is currently supervising 8 PhD students.*