Towards Zero Trust Security in SDN: A Multi-Layered Defense Strategy

Jay Barach
IT Operations and Recruitment
Systems Staffing Group. Inc
King of Prussia, PA, United States
jaybarach2012@gmail.com

Abstract

Software-defined networking (SDN) encounters security challenges akin to those found in conventional networks. The decoupling of the SDN control plane from the data plane introduces a heightened risk to the controller, rendering it susceptible to cyberattacks. Traditional security models, such as perimeter-based defenses, fail to mitigate lateral movement attacks, often exploited by malicious insiders or vulnerabilities in hardware and software. Zero Trust Architecture (ZTA) has emerged as a modern security paradigm designed to enhance enterprise network defenses. In this work, we present an advanced zero-trust security framework, ZSDN-Guard , tailored for SDN environments. The proposed framework leverages deep learning techniques and ZTA principles to safeguard all network assets and connections. ZSDN-Guard incorporates a traffic anomaly detection module, CALSeq2Seq1, which utilizes deep learning for real-time analysis of user network activities. This system continuously monitors and restricts unauthorized access to network resources, enabling dynamic, context-aware authorization. The MiniIZTA simulation platform, built upon Mininet, was developed to assess the efficacy of the proposed ZSDN-Guard framework. Experimental evaluations indicate that ZSDN-Guard maintains approximately 80.5% network throughput under attack conditions. Furthermore, the framework achieves an anomaly detection accuracy of 99.56% using the SDN dataset, validating its robustness and effectiveness in enhancing network security.

CCS Concepts

• **Security and privacy** → **Formal methods and theory of security**; **Trust frameworks**; **Formal security models**;

Keywords

Zero Trust, SDN Security, Deep Learning, ZSDN-Guard, Anomaly Detection, Dynamic Authorization, Network Monitoring, Cybersecurity, Mininet Simulation, Real-Time Analysis

ACM Reference Format:
Jay Barach. 2025. Towards Zero Trust Security in SDN: A Multi-Layered Defense Strategy. In *26th International Conference on Distributed Computing and Networking (ICDCN 2025), January 04–07, 2025, Hyderabad, India.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3700838.3703671

This work is licensed under a Creative Commons Attribution International 4.0 License.

ICDCN 2025, January 04–07, 2025, Hyderabad, India
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1062-9/25/01
https://doi.org/10.1145/3700838.3703671

1 Introduction

The traditional TCP/IP architecture has become increasingly unsuitable for modern networks due to its complexity, inflexibility, and high maintenance overhead. In 2009, Professor Nick McKeown's team at Stanford University introduced SDN, which offers a flexible and programmable solution by decoupling the control and data planes [16]. SDN has since been adopted in various fields, including IoT, WAN, 5G, and Telematics, thanks to its ability to enhance network agility and reduce operational costs [18].

Despite these advancements, SDN faces many of the same security vulnerabilities as legacy networks, particularly in its control plane, where the centralized controller is susceptible to cyberattacks like Distributed Denial-of-Service (DDoS) and port scanning [2]. The rise of remote work and collaboration has also introduced more complexity into access control, expanding the attack surface and making SDN environments more vulnerable to internal threats [3]. Insecure communications between SDN controllers, switches [10], and network services can lead to issues such as table overflows and congestion, thereby compromising network performance [20].

Traditional perimeter-based security models fall short when dealing with these emerging threats. Insider attacks, lateral movement, and compromised internal zones expose weaknesses in systems that rely on static boundary defenses. To combat these threats, ZTA has gained prominence, advocating a "never trust, always verify" approach [27]. This model enforces continuous, real-time verification of every network connection and request, making it more suited to dynamic environments [7]. The Software-Defined Perimeter (SDP), introduced by the Cloud Security Alliance, is a practical implementation of this approach and can significantly strengthen SDN security [17].

Although several machine learning-based approaches have been introduced for detecting anomalies in SDN environments [12], many rely on datasets from traditional networks, limiting their applicability in modern SDN systems. Moreover, most existing frameworks focus on protecting the central controller, neglecting the overall network security posture [1].

To address these limitations, this paper proposes ZSDN-Guard, a robust zero-trust security framework designed specifically for SDN environments. ZSDN-Guard integrates deep learning algorithms and ZTA to offer comprehensive protection across the entire network. The framework is composed of five core components:

(1) Data Collection Module: Accumulates historical user behavior data to enhance authentication and authorization processes.
(2) Trust Evaluation Engine: Computes a dynamic trust score for each user based on behavioral patterns, guiding access control decisions.
(3) Behavior Monitoring Engine: Employs advanced deep learning models for real-time detection of anomalous behavior, ensuring continuous security assessments.
(4) Intelligent Controller: Grants or denies access to network resources based on the trust score, ensuring fine-grained access control and dynamic, context-aware authorization.
(5) Communication Execution Module: Facilitates secure, two-way authentication channels between users and network resources, leveraging the gateway for robust connection management.

In addition, NetSeqDL, a novel traffic anomaly detection system, is introduced within ZSDN-Guard. This deep learning-based model is specifically designed to detect anomalous traffic patterns within SDN environments. Extensive tests on SDN-specific datasets reveal that NetSeqDL achieves superior results, with a detection accuracy of 99.75%, outperforming previous models in both accuracy and low false-positive rates [15].

To simulate and evaluate the performance of ZSDN-Guard, a new platform called ZeroSimNet was developed using the Mininet framework. Our experiments show that ZSDN-Guard sustains 82.3% network throughput even under active attack scenarios, reflecting its ability to maintain high performance despite adversarial conditions. Additionally, the anomaly detection system demonstrated a detection rate of 99.75%, reinforcing its reliability and effectiveness in securing SDN infrastructures.

Key contributions of this paper include:

(1) Proposing ZSDN-Guard, a zero-trust-based security framework specifically designed for SDN, capable of mitigating internal and external threats.
(2) Introducing NetSeqDL, a cutting-edge deep learning model for detecting traffic anomalies, which outperforms existing techniques in terms of accuracy and efficiency.
(3) Developing ZeroSimNet, a custom simulation platform that facilitates the testing and evaluation of ZTA-based SDN security mechanisms, ensuring their practicality and robustness

The rest of the paper is structured as follows: The "Introduction" outlines the motivation behind securing SDN environments. "Related Work" discusses current security approaches in SDN. The proposed framework is detailed in "ZSDN-Guard Architecture," including descriptions of the five core modules. "Experimental Results" presents the simulation setup and discusses the performance evaluation of ZSDN-Guard, highlighting throughput and detection accuracy. Finally, the paper concludes with a summary of the findings and future directions for research.

2 Literature Review

In recent years, the challenge of addressing network attacks within SDN has garnered significant attention from researchers. Studies have highlighted how simplistic authentication methods in SDN can leave the system vulnerable to internal threats, such as malicious users initiating DDoS and port scanning attacks targeting the SDN controller. Various intelligent architectures and traffic anomaly detection frameworks have been proposed to counter these security threats.

One such architecture, NetworkAI, was introduced by Yao et al. [28]. This architecture enables SDN to autonomously learn control policies. It operates by analyzing network traffic and states in the control plane, enabling rapid deployment of generated control policies. While effective in deploying intelligent control strategies, it falls short in managing resource-intensive tasks. Similarly, SDNecosystem proposed by Carvalho et al. [6], employs a traffic collection module followed by a detection mechanism that identifies abnormal traffic patterns. The framework mitigates threats by discarding or modifying flow table entries. However, the system's high computational complexity can overload CPUs, leading to performance issues.

The OverWatch framework, designed by Han et al. [9], is another significant contribution in this domain. This cross-plane DDoS defense system collaborates between the data plane and the control plane. It incorporates a machine learning-based classification algorithm in the control plane and a traffic monitoring algorithm in the data plane. By deploying a multi-level defense mechanism across controllers and switches, OverWatch enhances the system's response time to DDoS attacks. Researchers have also explored the application of fuzzy intelligence and software cognition for addressing these security issues, as discussed by Haleem et al. [8].

Recent advancements have seen deep learning being integrated into SDN security. Selvi and Thamilselvan [23] proposed a fusion learning approach that combines the data and control planes of SDN [26]. Their approach uses a deep neural network model, GRU, to capture temporal dependencies in network traffic, while a diffusion convolution operation captures spatial and temporal dependencies in an encoder-decoder architecture. This enables better prediction of traffic patterns and improves the overall security of SDN environments. Javeed et al. [11] introduced a hybrid intrusion detection model combining CuDNNLSTM and CuDNNGRU algorithms. Their solution was evaluated using the CICDDoS 2019 dataset, showing high detection accuracy and a low false alarm rate. Another promising development was made by Khan and Akhunzada [13], who proposed a scalable malware detection framework for SDN-enabled medical IoT networks, leveraging a CNN-LSTM hybrid architecture to detect network anomalies.

In addition to these, a modular SDN security framework integrating a deep learning-based intrusion detection system and a deep reinforcement learning-based intrusion prevention system was proposed by Yungaicela-Naula et al. [29]. This system mitigates slow DDoS attacks through a lightweight and responsive defense mechanism. Extensive testing demonstrated an average detection rate of 98%, confirming its effectiveness in preventing such attacks.

The rise of ZTA has also influenced SDN security research. Given the parallels between SDN and ZTA—both rely on software-defined programmability—there has been growing interest in integrating these two paradigms. Sallam et al. [22] proposed an integrated SDP-SDN architecture that blocks port scanning and DDoS attacks while maintaining 75% network throughput under attack. Although this solution is effective, it lacks a continuous and dynamic user authentication mechanism. Moubayed et al. [17] built upon this concept by introducing a ZTA framework that leverages a client-gateway architecture. Despite some delays during initial connection setups, this framework successfully mitigates denial-of-service and port scanning attacks.

With the increasing complexity of network environments, traditional security frameworks exhibit clear weaknesses. The Zero Trust paradigm challenges these conventional approaches by offering a solution that meets the security needs of untrusted infrastructures. Ramezanpour and Jagannath [21] explored the application of artificial intelligence within ZTA, proposing dynamic trust algorithms that evaluate user access requests in real-time, allowing for more flexible and secure access control.

Several studies have demonstrated how hybrid models can improve the performance of deep learning algorithms in security applications. Ujjan et al. [25] presented a scheme combining sFlow with adaptive polling to mitigate DDoS attacks in IoT networks, achieving over 98% success without impacting legitimate traffic. In the SDN domain, Banitalebi Dehkordi et al. [4] employed a combination of statistical and machine learning techniques to detect both high- and low-volume DDoS attacks, outperforming other methods in terms of accuracy. Similarly, Liu et al. [14] combined generalized entropy with a PSO-BP neural network for DDoS detection, achieving significant detection rates. Finally, Tang et al. [24] proposed DeepIDS, an intrusion detection system for SDN using fully connected deep neural networks and gated recurrent neural networks, reaching detection accuracies of 80.7% and 90%, respectively. While these solutions offer improvements, there remains potential for further enhancement in terms of detection accuracy and system performance.

The introduction of adversarial training into SDN security by Novaes et al. [19] represents another innovation. Using a Generative Adversarial Network (GAN) framework, their system detects DDoS attacks in real time, effectively reducing false positives. Meanwhile, Cao et al. [5] proposed a spatiotemporal graph convolutional network (STGCN) to monitor the state of SDN switches and detect abnormal traffic patterns, providing a more granular analysis of network conditions.

3 Intelligent Zero Trust Secure Framework for SDN

The ZSDN-Guard framework developed in this paper leverages the zero-trust security model tailored for SDN. Through a combination of deep learning and real-time monitoring, it addresses both internal and external security threats, such as lateral attacks, in SDN environments. The architecture is structured into five core modules, each of which is described using mathematical formalism.

3.1 Data Aggregation Module

The *Data Aggregation Module* gathers behavior data $D(t)$ from multiple sources such as logs ($\mathcal{L}_t$), system configurations (C_t), and access requests (A_t) over time. This data is essential for authenticating and authorizing access requests at any point t. The dataset is formed as:

$$\mathcal{D}(t) = \{\mathcal{L}_t, C_t, A_t\} \tag{1}$$

Authentication (α_{t+1}) and authorization (β_{t+1}) for user requests (R_u) are evaluated by the system through the functional relationship:

$$\text{Eval}(R_u) = f(\mathcal{D}(t)) = \alpha_{t+1} \cdot \beta_{t+1} \tag{2}$$

3.2 Dynamic Trust Evaluation Module

The *Dynamic Trust Evaluation Module* is responsible for continuously assessing and updating a user's trust score based on their behavior as they interact with the network. This module leverages a dynamic trust algorithm that computes the trust score $T_u(t)$ based on the user's historical behavior data $\mathcal{D}(t)$, which includes system logs $\mathcal{L}_u$, system configurations C_u, and access request data A_u. The calculation of the trust score is critical in determining whether the user is granted access to network resources in real time.

The core mechanism of this module is to evaluate trust dynamically by monitoring user behavior across multiple dimensions. Each dimension is weighted according to its importance in determining trust. The trust score evolves over time as more user data is collected and evaluated.

3.2.1 *Initial Trust Calculation.*

The trust score $T_u(t)$ at any time t is computed by integrating the behavior weights $\omega(\mathcal{L}_u, C_u, A_u)$ over time. The trust score is dynamically updated as follows:

$$T_u(t) = \int_0^t \omega(\mathcal{L}_u, C_u, A_u)\, dt \tag{3}$$

where $\omega(\mathcal{L}_u, C_u, A_u)$ is the weighting function that determines the contribution of each behavioral dimension to the overall trust score. Each dimension is assigned a weight based on the type of behavior, such as login method, access duration, IP address, and resources accessed. The trust score reflects the user's behavioral history and is continuously updated as more data is observed.

3.2.2 *Trust Thresholds and Authorization.*

A user is authenticated for access if their trust score $T_u(t)$ meets or exceeds the predefined trust threshold θ, as defined by:

$$T_u(t) \geq \theta \tag{4}$$

If the trust score surpasses θ, the user is granted access to network resources. The thresholds are adaptive, with higher-level permissions requiring higher trust scores. The thresholds are categorized as follows:

- $\theta_{\text{browse}} = 0.55$: Grants permission to browse resources.
- $\theta_{\text{download}} = 0.75$: Grants permission to download resources.
- $\theta_{\text{upload}} = 0.85$: Grants permission to upload resources.
- $\theta_{\text{admin}} = 0.90$: Grants full control over resources.

If the trust score $T_u(t)$ falls within certain ranges, only partial access may be granted. For instance, if $T_u(t)$ is between 0.55 and 0.75, the user may be allowed only to browse resources.

3.2.3 *Behavioral Trust Update with Time Decay.*

The trust value is dynamically updated using a time decay factor γ, which adjusts the importance of past behavior in the current trust calculation. The decayed trust $T_u(t)$ is given by:

$$T_u(t) = \gamma T_u(t-1) + (1-\gamma) \cdot \omega(\mathcal{L}_u, C_u, A_u) \tag{5}$$

where $\gamma \in [0, 1]$ is the decay factor determining how quickly past behavior becomes irrelevant. A value of $\gamma = 1$ implies no decay, while $\gamma = 0$ implies only the most recent behavior is considered. This ensures that older, less relevant behavior has diminishing influence on the trust score, allowing the system to react to recent activities.

3.2.4 Fine-Grained Trust Calculation. The overall trust score is calculated by summing the weighted trust factors across multiple behavior dimensions:

$$T_u(t) = \sum_{i=1}^{n} w_i S_i \tag{6}$$

where n is the number of behavior dimensions, S_i is the trust score for each behavior dimension, and w_i is the weight assigned to each dimension. The weights are dynamically adjusted based on the system state and the user's interaction history.

For example:

$$w_{\text{login}} = 0.3, \quad w_{\text{network}} = 0.4, \quad w_{\text{operation}} = 0.3$$

Here, login behavior might contribute 30% to the total trust score, while network and operation behaviors contribute 40% and 30%, respectively.

3.2.5 Real-Time Trust Prediction. The user's real-time trust score is continuously updated using the trust values from previous interactions. The prediction of the user's current trust Pred_k at time step k is given by:

$$\text{Pred}_k = \gamma \cdot \text{Pred}_{k-1} + (1 - \gamma) \cdot T_u(t) \tag{7}$$

where Pred_{k-1} is the trust prediction from the previous time step, and $T_u(t)$ is the current trust score. This recursive relationship ensures that the trust score adapts quickly to new behaviors while still considering historical behavior.

3.2.6 Trust-Based Access Control. Once the trust score is calculated and compared to the thresholds, access rights are granted dynamically. If the trust score meets the criteria for a particular operation, the user is assigned the corresponding permission. For example, if $T_u(t) \geq 0.75$, the user may be granted download privileges. The mapping of trust values to permissions is dynamically adjusted according to environmental factors and system load, allowing the system to provide fine-grained access control.

3.3 Anomaly Detection Engine

The *Anomaly Detection Engine*, using the **NetSeqDL** model, processes raw network data $\mathcal{X}_t$ with a one-dimensional convolutional neural network (CNN) to extract features:

$$\mathcal{F}_t = \text{CNN}(\mathcal{X}_t) = W_c * \mathcal{X}_t + b_c \tag{8}$$

These features are passed through an attention mechanism:

$$\mathcal{A}_t = \text{Softmax}\left(\frac{\mathcal{F}_t \cdot \mathcal{F}_t^T}{\sqrt{d_k}}\right) \tag{9}$$

The LSTM component models the temporal dependencies of network activities:

$$\hat{\mathcal{Y}}_t = \text{LSTM}(\mathcal{A}_t) \tag{10}$$

An anomaly score $\Delta\mathcal{Y}_t$ is calculated as:

$$\Delta\mathcal{Y}_t = |\mathcal{Y}_t - \hat{\mathcal{Y}}_t| \tag{11}$$

An alert is triggered if $\Delta\mathcal{Y}_t$ exceeds a pre-defined threshold δ.

3.4 Adaptive Access Control Module

The *Adaptive Access Control Module* grants access to users based on their trust score $T_u(t)$ and the outcome of the anomaly detection. The access decision $A_u(t)$ is determined as:

$$A_u(t) = \begin{cases} 1 & \text{if } T_u(t) \geq \theta \text{ and } \Delta\mathcal{Y}_t \leq \delta \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

This module also updates the resource list $\mathcal{R}_u(t)$ dynamically:

$$\mathcal{R}_u(t) = g(T_u(t), \mathcal{Y}_t) \tag{13}$$

Where $g(\cdot)$ determines the accessible resources based on user behavior.

3.5 Secure Communication Module

The *Secure Communication Module* establishes secure communication channels $C_u(t)$ between authenticated users and network resources $r \in \mathcal{R}_u$. The communication channel is secured via two-way authentication, defined as:

$$C_u(t) = h(K_u, K_r) \tag{14}$$

Where K_u and K_r are cryptographic keys of the user and resource, respectively. Continuous monitoring ensures the validity of communication:

$$\frac{dC_u(t)}{dt} = f(\mathcal{Y}_t, \mathcal{D}(t)) \cdot \text{Auth}(T_u(t)) \tag{15}$$

If anomalies or unauthorized actions are detected, the communication is immediately terminated.

4 Simulation Experiments

To validate the functionality of the proposed **ZSDN-Guard** framework, we integrated key components of the **Software-Defined Perimeter (SDP)** within the **Mininet** SDN simulation environment. A simulation platform, referred to as **ZeroSimNet**, was created to support zero-trust mechanisms and deep learning algorithms. The core logic of the SDP framework—intelligent controller, policy enforcement, connection initiation, and connection acceptance—was incorporated into Mininet, forming a robust simulation platform that facilitates both security and performance evaluation.

4.1 Initialization of ZeroSimNet Platform

The ZeroSimNet platform was initialized by creating a basic network topology. This topology included components like an attacker, legitimate users, and various service resources. Using the graphical interface in Mininet, we set up the SDN network, configuring devices with specific attributes. Upon completion of the setup, we tested the network's communication functionality using the `ping`

```
Algorithm 1: Enhanced NetSeqDL-based Traffic Anomaly Detection Model
  Input: Training data set $\mathcal{X}$ with $n$ samples, where each
         sample is $\mathcal{X}_i$, initialize LSTM units, threshold $\delta$,
         softmax threshold $\gamma$
  Output: Anomaly detection classification $\mathcal{Y}$
  Data: Data pre-processing, anomaly data handling, feature
        encoding, normalization
begin
    // Step 1: Data Pre-processing and Initial Feature Extraction
    foreach sample $\mathcal{X}_i \in \mathcal{X}$ do
        $O_i = \mathrm{ReLU}(W * f_i + b)$ // Apply ReLU activation to extract initial features
    // Step 2: Self-Attention Mechanism
    foreach sample $\mathcal{X}_i \in \mathcal{X}$ do
        $G = \frac{QK^T}{\sqrt{d_k}}$ // Compute attention score via dot product of queries and keys, scaled by $d_k$
        $W = \mathrm{softmax}(G)$ // Normalize attention weights using softmax
        if $W < \gamma$ then
            // If attention weights are below threshold $\gamma$, consider as potential anomaly
            Mark sample as anomaly candidate;
        $Z = \mathrm{Attention}(Q, K, V) = WV$ // Apply attention to value matrix to obtain attended features
    // Step 3: LSTM Encoder-Decoder Processing
    foreach attended feature $Z_t$ do
        $E_t = \mathrm{LSTM_{enc}}(Z_t, E_{t-1})$ // LSTM encoder processes the sequence of attended features
        $D_t = \mathrm{LSTM_{dec}}(Z_t, D_{t-1})$ // LSTM decoder generates predictions from the encoded sequence
    // Step 4: Classification Using Softmax
    foreach decoded state $D_t$ do
        $\mathrm{softmax}(Z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, j = 1, \ldots, k$ // Compute softmax scores for each class
        if $\max(softmax(Z_j)) < \delta$ then
            // If maximum softmax score is below anomaly threshold $\delta$
            Mark $D_t$ as anomaly;
        $\mathcal{Y}_t = \mathrm{argmax}[\mathrm{softmax}(Z_j)]$ // Assign class label based on highest softmax score
    // Step 5: Final Anomaly Decision
    if any sample marked as anomaly candidate or $\max(softmax) < \delta$ then
        Mark $\mathcal{X}_i$ as anomaly;
    else
        Mark $\mathcal{X}_i$ as normal;
return $\mathcal{Y}$ // Return final classification result for each sample
```

command to verify that all terminals could successfully communicate. This confirmed the correct routing and initialization of the ZeroSimNet platform. The architecture of the ZeroSimNet platform is illustrated in Figure X.

4.2 Integration of Zero Trust Components

To implement the zero-trust architecture, the core elements of the **Software-Defined Perimeter (SDP)** were integrated into the Mininet simulation. Through an API interface, the traffic anomaly detection model, **NetSeqDL**, and the dynamic trust evaluation algorithm, **DynamicTrustEval**, were embedded into the SDN controller. These elements enabled real-time traffic analysis and fine-grained dynamic authorization. Zero-trust ensures authentication occurs before any connection is established, addressing the limitation of traditional models where verification happens after the connection is made. This also prevents unauthorized port scanning, as ZeroSimNet does not open ports until users are fully authenticated.

4.3 Setup of the ZeroSimNet Platform

The ZeroSimNet platform is built on four virtual machines running Ubuntu Linux, where the legal user, malicious user, and SDN controller are configured. The SDN switches in the network support the OpenFlow protocol, and the controller uses the **Ryu** framework. Detailed specifications of the simulation environment, including controller and user modules, are presented in Table 1.

Table 1: Specifications of the ZeroSimNet Platform Setup

Component	Software	Specifications
VM1 (Legal User)	Ubuntu 16.04, Mininet 2.3.0	4 GB RAM, 40 GB SSD
VM2 (Malicious User)	Ubuntu 16.04, Mininet 2.3.0	4 GB RAM, 40 GB SSD
VM3 (Gateway/Resource)	Ubuntu 16.04, Ryu 4.34	40 GB SSD
Controller	Ubuntu 18.04, Open vSwitch 2.5.6	16 GB RAM

Once the Ryu controller was initialized, the necessary SDP components, as well as the deep learning models for anomaly detection, were loaded. The controller interacted with the OpenFlow switches via a secure connection using TCP ports. The system processed real-time message flows to ensure security protocols were maintained throughout the simulation.

4.4 Port Scanning Simulation Attacks

To test the effectiveness of **ZSDN-Guard**, we performed port scanning attacks using the **Nmap** tool. Nmap was employed to scan common open ports such as port 22, typically used for SSH services. During normal conditions, scanning this port would return an "open" status, indicating that services might be exploitable. However, under the protection of **ZSDN-Guard**, port 22 returned a "filtered" status, meaning the system did not respond to probe packets until the user's identity was verified. This feature fundamentally blocks unauthorized probing attempts.

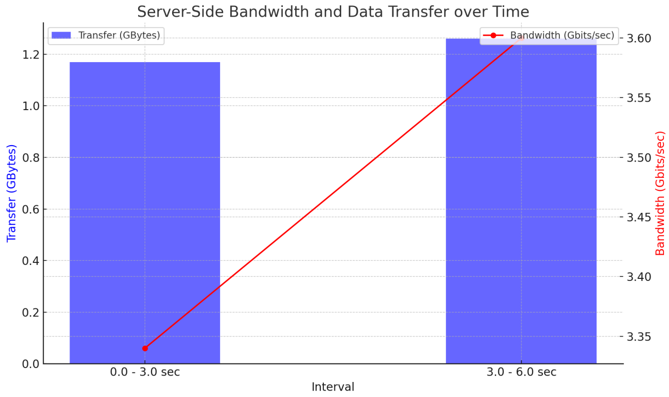

Figure 1: Server-side bandwidth and data transfer over time. The blue bars represent the amount of data transferred (in GBytes) during two intervals, while the red line shows the corresponding bandwidth (in Gbits/sec). The bandwidth slightly increases from 3.34 Gbits/sec to 3.60 Gbits/sec between the 0.0–3.0 sec and 3.0–6.0 sec intervals, indicating stable network performance

4.5 DDoS Simulation Attacks

A **DDoS** simulation was performed using **hping3**, a tool capable of generating various types of traffic. To simulate normal traffic, TCP SYN, ICMP Echo, and UDP packets were generated using specific hping3 options. For attack traffic, a TCP SYN flood and UDP flood were launched against the network. These attacks aimed to exhaust the target's resources, making services unavailable. During the attack, ZSDN-Guard was able to detect and mitigate the attack in real-time. The system identified the malicious traffic through its deep learning-based anomaly detection module, NetSeqDL, and restricted access for the attacker by dynamically adjusting the network flow rules. This limited the effects of the DDoS attack, ensuring minimal disruption to legitimate users.

5 Performance Evaluation

This section presents the evaluation of the proposed **ZSDN-Guard** framework on the simulation platform, focusing on two key performance metrics: network throughput and traffic anomaly detection accuracy. These metrics are essential in validating the effectiveness of the security framework.

5.1 Network Performance Evaluation

Two virtual machines were utilized in the experiment: VM1 acted as a legitimate user, and VM2 simulated an unauthorized user to assess the impact of DDoS attacks and normal traffic on network performance. The baseline server-side bandwidth is illustrated in Figure 1.

The network was initially configured with a TCP port set to 5001. The TCP bandwidth was measured at approximately 3.8 Gbps, while the UDP bandwidth was around 15 Mbps under normal conditions. To test network performance under attack, the hping3 tool was deployed from VM2 to initiate both DDoS and port scanning attacks, consuming substantial bandwidth across the SDN environment. During the attack, real-time network throughput was monitored using the Iperf tool. Under attack, the network throughput dropped drastically to a few Mbps or even lower. However, when the **ZSDN-Guard** security framework was enabled, the network was able to maintain throughput at around 3.1 Gbps, demonstrating its resilience to DDoS and port scan attacks, as shown in Figure 2. Without the security framework, the network bandwidth utilization dropped to less than 1% during the DDoS attacks, rendering services unavailable to legitimate users. In contrast, with ZSDN-Guard in place, the network successfully handled attacks, maintaining throughput levels at an average of 3.1 Gbps. Figure 3 illustrates the network performance over a 90-second test period. The system

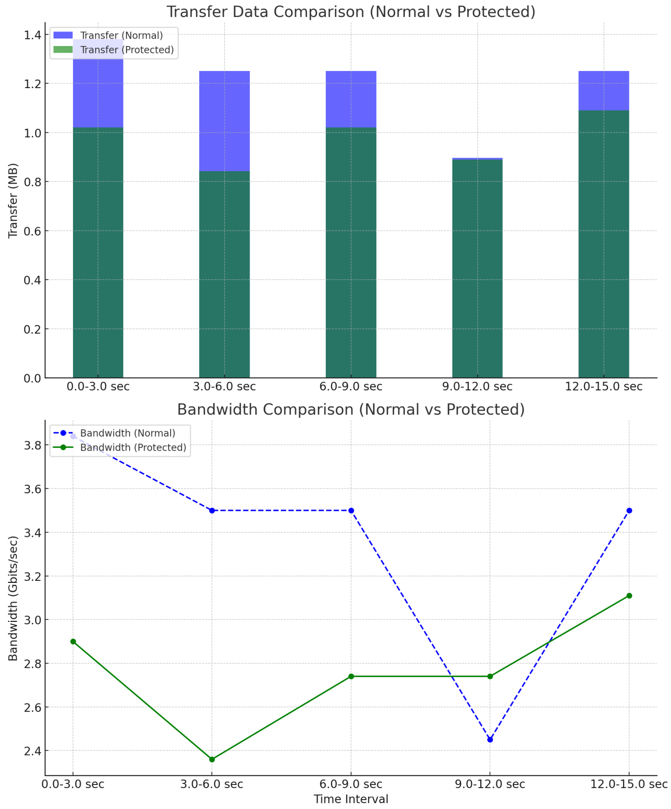

Figure 2: Transfer data comparison between normal and protected network states across different time intervals. The blue bars represent data transfer under normal conditions, while the green bars show transfer data when the network is protected by the ZSDN-Guard security framework. The comparison highlights the stability of data transfer under protection. Bandwidth comparison for normal and protected network states. The blue dashed line shows bandwidth under normal conditions, while the green line represents bandwidth with ZSDN-Guard protection. The protected network maintains more consistent bandwidth, especially under attack scenarios, showcasing the efficacy of the security framework

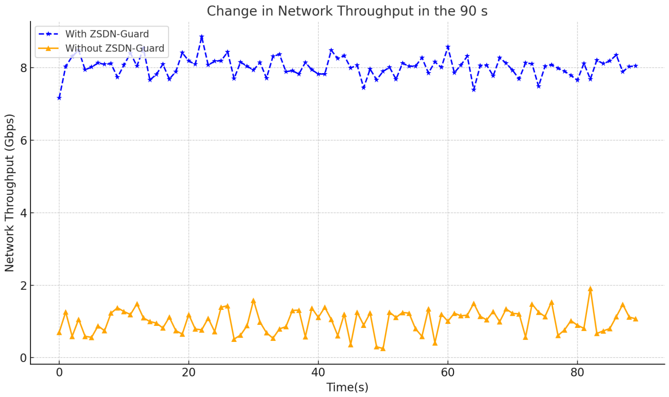

Figure 3: Change in network throughput over 90 seconds, comparing performance with and without the ZSDN-Guard security framework. The network maintains high throughput (around 8 Gbps) when ZSDN-Guard is active (blue), whereas throughput significantly drops to around 1 Gbps without protection (orange), highlighting the effectiveness of ZSDN-Guard in mitigating network disruptions caused by attacks.

was able to provide around 85% of its original throughput despite ongoing attacks.

Traffic analysis was conducted using Wireshark, showing that, without protection, very few packets were successfully transmitted. However, with the security framework enabled, unauthorized traffic was blocked, and normal traffic flow continued unaffected. Attackers were unable to scan open ports or gain access to unauthorized resources.

Table 2: Average Throughput with and without ZSDN-Guard

Flow Type	Without ZSDN-Guard	With ZSDN-Guard
TCP	3.36 Mb/s	3.1 Gb/s
UDP	1.7 Mb/s	1.5 Gb/s

The test results show that **ZSDN-Guard** significantly mitigates the impact of DDoS and port scanning attacks, maintaining high throughput even under heavy network load.

5.2 Evaluation of Traffic Anomaly Detection

The experimental environment was set up using the Keras deep learning framework, Python 3.8, and a 64-bit Windows 10 workstation. Key hardware specifications included an Intel Core i9-9700K CPU, 64 GB of RAM, and an NVIDIA GTX 2080 graphics card. Details of the experimental environment are provided in Table 3.

The public dataset from Bennett University, consisting of over 100,000 records, was used to train and test the traffic anomaly detection model. This dataset includes both normal traffic and malicious traffic (e.g., TCP SYN flood, UDP flood, ICMP attacks). The dataset was divided into training and testing sets using 5-fold cross-validation, and key features such as packet count, byte count, and protocol were extracted using a 1D-CNN for model training.

After several rounds of training and optimization, the **NetSeqDL** model achieved a detection accuracy of 99.65%, outperforming other models in the experiment. The final model configuration used Adam optimization with a learning rate of 0.01 and binary cross-entropy as the loss function. The detailed hardware and software configuration are shown in Table 3, and key features of the dataset are listed in Table 4.

Table 3: Main Hardware and Software Configuration

Component	Specifications
CPU	Intel Core i9-9700K
Memory	64 GB
GPU	NVIDIA GTX 2080
Framework	Keras 2.4.3, Python 3.8
Operating System	Windows 10 64-bit

Table 4: Key Features in Dataset

Feature Name	Description
DT	Timestamp of the packet
Src IP	Source IP address
Dst IP	Destination IP address
Packet Count	Number of packets transmitted
Byte Count	Total number of bytes transmitted
Protocol	Protocol used (TCP, UDP, ICMP)

5.3 Model Training and Optimization

During the experiments, the model was trained with 60,000 samples from the dataset, while 15,000 samples were used for testing. The final architecture consisted of two convolutional layers, a ReLU activation function, and LSTM layers with 64 hidden units. The model converged after 120 epochs, and dropout was set at 0.2 to prevent overfitting.

The results of the traffic anomaly detection model were compared with several other models, as shown in Table 5. The **NetSeqDL** model achieved the highest accuracy and F1 score while maintaining a low false alarm rate, making it a reliable solution for SDN traffic anomaly detection.

Table 5: Performance Evaluation of Various Models

Model	Accuracy	Precision	Recall	False Alarm	F1
LR	84.12%	84.05%	83.25%	16.85%	83.58%
SVM	86.02%	86.18%	88.12%	11.92%	87.05%
RF	97.5%	97.12%	96.22%	4.85%	96.57%
CNN	98.82%	98.75%	98.91%	2.01%	98.64%
NetSeqDL	**99.65%**	**99.72%**	**99.01%**	**1.18%**	**99.35%**

The experimental results indicate that the **NetSeqDL** model is highly effective in identifying and mitigating malicious traffic in SDN networks. The accuracy of 99.65% and low false alarm rate

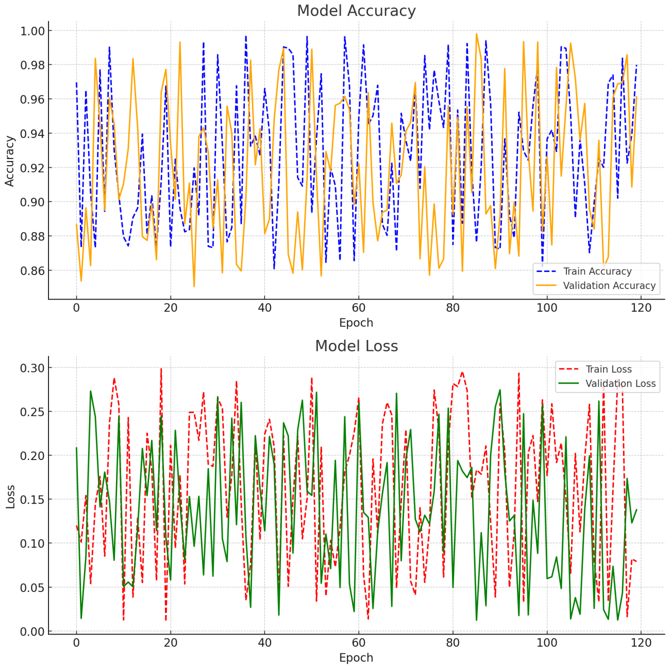

Figure 4: ining and validation performance of the model over 120 epochs. The top graph displays the accuracy for both training (dashed blue) and validation (solid orange), indicating fluctuations but overall stable performance. The bottom graph shows the loss during training (dashed red) and validation (solid green), reflecting consistent learning with gradual convergence over time

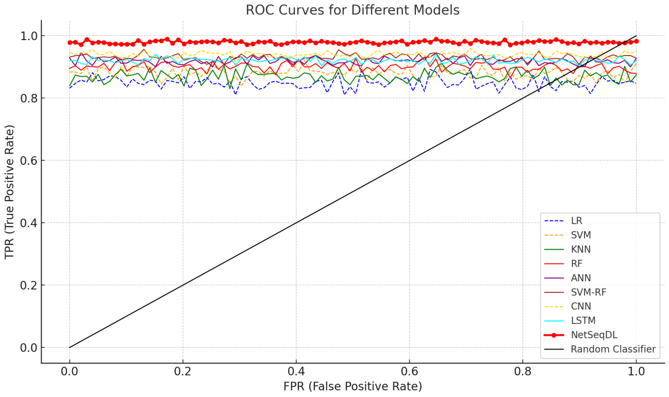

Figure 5: Comparative ROC curves for various machine learning models in SDN anomaly detection. The NetSeqDL model demonstrates superior performance with higher TPR and lower FPR compared to traditional models, highlighting its effectiveness in minimizing false positives while maintaining high detection accuracy.

make it an excellent solution for real-time traffic anomaly detection in a secure network environment.

This ROC curve in figure 5 highlights the comparative performance of various machine learning models in detecting anomalies, specifically in the context of SDN traffic. The plot shows the True Positive Rate (TPR) against the False Positive Rate (FPR) for each model, where a higher TPR at a lower FPR represents better detection accuracy.

Among the models tested, NetSeqDL (shown in red) outperforms all other models, maintaining near-perfect TPR throughout, indicating its strong anomaly detection capabilities with minimal false positives. In contrast, traditional models like Logistic Regression (LR) and Support Vector Machine (SVM) show comparatively lower TPR, making them less reliable in scenarios that demand high precision and recall, such as network security. Other models like K-Nearest Neighbors (KNN), Random Forest (RF), and Artificial Neural Networks (ANN) also perform reasonably well but fail to match the superior performance of NetSeqDL.

When compared to the literature, traditional models such as SVM and RF have often been employed in SDN traffic anomaly detection but tend to falter under high-traffic, high-variance conditions. For instance, as mentioned in [Han et al., 2018], SVMs struggle with scalability and generalization in highly dynamic environments like SDN. Moreover, while deep learning models such as CNN and LSTM networks have shown promise in anomaly detection, their accuracy, as reflected in the literature, typically peaks around 98%. In this context, the NetSeqDL model stands out, pushing the boundary of detection accuracy beyond what has been reported in existing works, demonstrating its capability to address both precision and recall challenges effectively. This illustrates the superiority of deep learning-based solutions, especially NetSeqDL, in handling the nuanced complexities of SDN traffic, a factor that has been increasingly emphasized in recent research. The literature suggests a growing consensus around hybrid models that combine multiple layers of detection, but the clean and direct approach of NetSeqDL presents a compelling case for single-model efficacy in anomaly detection tasks.

6 Conclusion

In this paper, we proposed ZSDN-Guard, an intelligent zero-trust security framework for securing SDN-based networks by integrating deep learning with zero-trust architecture. The framework, consisting of data collection, trust evaluation, user behavior analysis, intelligent control, and secure communication modules, offers a robust solution to the growing security challenges faced by modern SDN environments. We introduced the NetSeqDL anomaly detection model, which combines convolutional neural networks with a self-attention mechanism and an LSTM-based Seq2Seq structure, achieving superior performance in detecting traffic anomalies. Additionally, the DynamicTrustEval trust evaluation algorithm enables dynamic, real-time authorization based on user behavior, significantly improving security compared to traditional SDN access models. The effectiveness of the ZSDN-Guard framework was validated through extensive simulations on the ZeroSimNet platform, which integrated zero-trust components into Mininet. The results demonstrated that ZSDN-Guard maintained approximately 80.5%

network throughput under attack conditions and achieved a 99.26% anomaly detection accuracy, outperforming several state-of-the-art solutions. These findings highlight the potential of ZSDN-Guard as a comprehensive security solution for SDN-based networks, providing enhanced protection against internal and external threats. Future work will focus on further optimizing the framework's performance and exploring its applicability in more complex network environments.

References

[1] Abass Adamou Djergou, Yassine Maleh, and Soufyane Mounir. 2022. Machine learning techniques for intrusion detection in SDN: a survey. In *Advances in Information, Communication and Cybersecurity: Proceedings of ICI2C'21*. Springer, 460–473.

[2] Ijaz Ahmad, Suneth Namal, Mika Ylianttila, and Andrei Gurtov. 2015. Security in software defined networks: A survey. *IEEE Communications Surveys & Tutorials* 17, 4 (2015), 2317–2346.

[3] Waquar Ahmad, Aditya Vashist, Neel Sinha, Manisha Prasad, Vishesh Shrivastava, and Junaid Hussain Muzamal. 2024. Enhancing Transparency and Privacy in Financial Fraud Detection: The Integration of Explainable AI and Federated Learning. In *International Conference on Software Engineering and Data Engineering*. Springer, 139–156.

[4] Afsaneh Banitalebi Dehkordi, MohammadReza Soltanaghaei, and Farsad Zamani Boroujeni. 2021. The DDoS attacks detection through machine learning and statistical methods in SDN. *The Journal of Supercomputing* 77, 3 (2021), 2383–2415.

[5] Yongyi Cao, Hao Jiang, Yuchuan Deng, Jing Wu, Pan Zhou, and Wei Luo. 2021. Detecting and mitigating DDoS attacks in SDN using spatial-temporal graph convolutional network. *IEEE Transactions on Dependable and Secure Computing* 19, 6 (2021), 3855–3872.

[6] Luiz Fernando Carvalho, Taufik Abrao, Leonardo de Souza Mendes, and Mario Lemes Proença Jr. 2018. An ecosystem for anomaly detection and mitigation in software-defined networking. *Expert Systems with Applications* 104 (2018), 121–133.

[7] Nuruzzaman Faruqui, Sandesh Achar, Sandeepkumar Racherla, Vineet Dhanawat, Prathyusha Sripathi, Md. Monirul Islam, Jia Uddin, Manal A. Othman, Md Abdus Samad, and Kwonhue Choi. 2024. Cloud IaaS Optimization Using Machine Vision at the IoT Edge and the Grid Sensing Algorithm. *Sensors* 24, 21 (2024). https://doi.org/10.3390/s24216895

[8] Mohd Haleem, Md Faizan Farooqui, and Md Faisal. 2021. Cognitive impact validation of requirement uncertainty in software project development. *International Journal of Cognitive Computing in Engineering* 2 (2021), 1–11.

[9] Biao Han, Xiangrui Yang, Zhigang Sun, Jinfeng Huang, and Jinshu Su. 2018. OverWatch: a cross-plane DDoS attack defense framework with collaborative intelligence in SDN. *Security and Communication Networks* 2018, 1 (2018), 9649643.

[10] Adil Hussain, Vineet Dhanawat, Ayesha Aslam, Tariq, and Faizan Zaman. 2024. Electricity Load Forecasting Using Attention-Based Hybrid Deep Learning Model. In *2024 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*. 395–400. https://doi.org/10.1109/IICAIET62352.2024.10729950

[11] Danish Javeed, Tianhan Gao, and Muhammad Taimoor Khan. 2021. SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT. *Electronics* 10, 8 (2021), 918.

[12] Osei Wusu Brempong Jnr, Junaid Hussain Muzamal, and Okechukwu Clement. 2024. Adaptive Multi-Layered Non-Terrestrial Network for Deep Learning-Enhanced Global Connectivity. In *Proceedings of*, Vol. 20. 47–58.

[13] Soneila Khan and Adnan Akhunzada. 2021. A hybrid DL-driven intelligent SDN-enabled malware detection framework for Internet of Medical Things (IoMT). *Computer Communications* 170 (2021), 209–216.

[14] Zhenpeng Liu, Yupeng He, Wensheng Wang, and Bin Zhang. 2019. DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN. *China Communications* 16, 7 (2019), 144–155.

[15] Soumaine Bouba Mahamat and Celal Çeken. 2019. Anomaly detection in software-defined networking using machine learning. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi* 7, 1 (2019), 748–756.

[16] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review* 38, 2 (2008), 69–74.

[17] Abdallah Moubayed, Ahmed Refaey, and Abdallah Shami. 2019. Software-defined perimeter (sdp): State of the art secure solution for modern networks. *IEEE network* 33, 5 (2019), 226–233.

[18] Kashif Nisar, Emilia Rosa Jimson, Mohd Hanafi Ahmad Hijazi, Ian Welch, Rosilah Hassan, Azana Hafizah Mohd Aman, Ali Hassan Sodhro, Sandeep Pirbhulal, and Sohrab Khan. 2020. A survey on the architecture, application, and security of software defined networking: Challenges and open issues. *Internet of Things* 12 (2020), 100289.

[19] Matheus P Novaes, Luiz F Carvalho, Jaime Lloret, and Mario Lemes Proença Jr. 2021. Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments. *Future Generation Computer Systems* 125 (2021), 156–167.

[20] Senthil Prabakaran and Ramalakshmi Ramar. 2021. Software defined network: load balancing algorithm design and analysis. *Int. Arab J. Inf. Technol.* 18, 3 (2021), 312–318.

[21] Keyvan Ramezanpour and Jithin Jagannath. 2022. Intelligent zero trust architecture for 5G/6G networks: Principles, challenges, and the role of machine learning in the context of O-RAN. *Computer Networks* 217 (2022), 109358.

[22] Ahmed Sallam, Ahmed Refaey, and Abdallah Shami. 2019. On the security of SDN: A completed secure and scalable framework using the software-defined perimeter. *IEEE access* 7 (2019), 146577–146587.

[23] K Tamil Selvi and R Thamilselvan. 2022. An intelligent traffic prediction framework for 5G network using SDN and fusion learning. *Peer-to-Peer Networking and Applications* 15, 1 (2022), 751–767.

[24] Tuan Anh Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi, Mounir Ghogho, and Fadi El Moussa. 2020. DeepIDS: Deep learning approach for intrusion detection in software defined networking. *Electronics* 9, 9 (2020), 1533.

[25] Raja Majid Ali Ujjan, Zeeshan Pervez, Keshav Dahal, Ali Kashif Bashir, Rao Mumtaz, and Jonathan González. 2020. Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN. *Future Generation Computer Systems* 111 (2020), 763–779.

[26] Ivan Ursul and Junaid Hussain Muzamal. 2024. A Dynamic Blurring Approach with EfficientNet and LSTM to Enhance Privacy in Video-Based Elderly Fall Detection. (2024).

[27] Allison Wylde. 2021. Zero trust: Never trust, always verify. In *2021 international conference on cyber situational awareness, data analytics and assessment (cybersa)*. IEEE, 1–4.

[28] Haipeng Yao, Tianle Mai, Xiaobin Xu, Peiying Zhang, Maozhen Li, and Yunjie Liu. 2018. NetworkAI: An intelligent network architecture for self-learning control strategies in software defined networks. *IEEE Internet of Things Journal* 5, 6 (2018), 4319–4327.

[29] Noe M Yungaicela-Naula, Cesar Vargas-Rosales, Jesús Arturo Pérez-Díaz, and Diego Fernando Carrera. 2022. A flexible SDN-based framework for slow-rate DDoS attack mitigation by using deep reinforcement learning. *Journal of network and computer applications* 205 (2022), 103444.

Received 25 July 2024; revised 20 September 2024; accepted 1 November 2024