International Journal of Machine Learning, AI & Data Science Evolution
ISSN: 3068-5073/ Open Access / IJMLAIDSE_Y2F2A001
Apex Academia Press / https://doi.org/10.63665/ijmlaidse-y2f2a001

*Original Article*

# A Self-Adaptive Zero-Trust Data Communication Framework Using Deep Reinforcement Learning in Software-Defined Networks

***Osita Miracle Nwakeze[1], Nwafor Anthony[2], Obaze Caleb[3]***

*[1 2]Department of Computer Science, Chukwuemeka Odumegwu Ojukwu University, Uli*
*[3]Department of Computer science, Dennis Osadebay University Asaba, Delta State, Nigeria.*

***Abstract***

*Software-Defined Networking (SDN) is quickly expanding, which provides flexibility in managing the network but opens systems to highly dynamic and advanced cyber threats. Conventional defence systems such as rule-based firewalls and threshold-based intrusion detection systems can be inadequate because they are not adaptable and are also static. In this work, a self-adaptive zero-trust data communication framework is suggested and combines Deep Reinforcement Learning (DRL) to enforce security in SDN environments in an intelligent and real-time manner. The framework includes a zero-trust engine and a DRL agent that is implemented in the SDN controller to assess the level of trust constantly and implement access control policies dynamically. The DRA agent treats network security as a Markov Decision Process (MDP) with states based on the traffic patterns, packet attributes, user behaviour, trust scores, and anomaly indicators, and actions based on allowing, denying, isolating, or re-authenticating requests. It is trained on simulated SDN traffic and the CICIDS 2017 dataset, and pre-processed by cleaning data, extracting features, normalizing and encoding data. Learning is driven by a reward function which rewards correct decisions and punishes false positives and false negatives to allow the agent to adjust to changing threats and dynamic traffic conditions. Experimental findings show that the framework has 96.8% detection accuracy, 95.7% precision, 97.3% recall and F 1-score of 96.5% which is much better than baseline rule-based and threshold-based systems. Network performance testing indicates that the latency and throughput are barely affected, which proves that there is efficient enforcement of the real time. Superior adaptability, scalability and robustness are brought out in comparative analysis. In general, the research confirms that combining DRL and a zero-trust architecture is an effective, intelligent, and scalable solution to ensure the SDN environments are secure and the network performance does not suffer.*

***Keywords***

*Software-Defined Networking (SDN), Zero Trust Architecture, Deep Reinforcement Learning (DRL), Network Security, Adaptive Access Control.*

Article History

Received: 05.04.2026

Accepted: 16.04.2026

Published: 22.04.2026

## 1. Introduction

The fast development of digital communication systems and the rise in the use of cloud-based services have greatly increased the attack surface of modern networks. Old fashioned security models which are largely based on the use of perimeter-based security are not adequate any more to deal with the advanced and long-lasting cyber threats. Hackers are now using internal weaknesses and it becomes imperative to implement more dynamic and intelligent security practices. In this regard, the idea of Zero Trust Architecture (ZTA) has become a powerful one, which supposes that even entities located within or outside the network should not be trusted by default and that identity verification and constant monitoring thus become strict (Rose et al., 2020; Khan et al., 2026).

Meanwhile, the development of network architectures like Software-Defined Networking (SDN) has brought programmability and centralized control to network management. SDN helps network administrators to configure, control, and optimize network resources dynamically by separating the control plane and the data plane. Such flexibility offers a good base on which higher security mechanisms can be implemented. However, the dynamic nature of SDN environments also introduces new vulnerabilities, such as controller attacks and flow rule manipulation, which necessitate intelligent and adaptive security solutions (Barach, 2025; Bijlwan and Kumar, 2026).

In order to solve these problems, Deep Reinforcement Learning (DRL) has been integrated into network security and has received a lot of attention. DRL allows systems to acquire the best decision-making strategies by interacting with the environment, based on reward-based feedback mechanisms. In contrast to the conventional rule-based systems, DRL is able to adjust to changing threats by continually updating its policies as a result of observed network behavior (Kanimozhi and Ramesh, 2025; Niknami and Wu, 2023). This contributes to its special applicability in handling dynamic and complex environments like SDN where real-time decision-making is essential to ensure both security and performance (Ke et al., 2025; Chenghao et al., 2022).

More recent research has shown that DRA-based systems can be successfully used to address distributed denial-of-service attacks and flow scheduling in zero-trust networks (Eke and Onyejegbu, 2024; Mahida, 2026). Also, uncertainty-aware reinforcement learning algorithms, including PPO and Bayesian neural networks, have been investigated to promote trust assessment in dynamic situations (Kalekar, 2026). These developments underscore the prospects of DRM as the foundation of responsive, clever security systems in SDN ambiance.

Thus, this paper suggests a zero-trust data communication scheme which is self-adaptive and which applies deep reinforcement learning in an SDN context. The framework will offer dynamic security policies and threat mitigation by assessing the level of trust and implementing it in real-time to offer intelligent and real-time access control. The proposed system should increase network resilience, decrease unauthorized access, and improve the overall data communication protection in contemporary distributed networks by integrating the principles of zero trust, SDN programmability, and DRL learning capabilities (Muhammad, 2026).

## 2. Methodology

The present work has taken a systematic and simulation-based method to come up with self-adaptive zero-trust data communication architecture based on deep reinforcement learning in a SDN setting. This approach combines the concept of ZTA and DRL to come up with a smart and dynamic security model. The framework is designed to have application, control and data plane layers, where the control layer is implemented with a centralized controller like an OpenDaylight that has a global network topology and implements dynamic policies. The control layer has a zero-trust engine to guarantee that all entities undergo continuous authentication, authorization and trust evaluation. The deep reinforcement learning agent works in the controller, monitoring network conditions like traffic flows, user behaviour, and anomaly signals and acting on them by permitting/denying/isolating/re-authenticating requests as a result of a learning process that rewards correct decisions and punishes mistakes. The simulations of networks are done with Mininet, and common datasets (CICIDS and UNSW-NB15) are used to train and validate. The proposed framework is measured based on the accuracy, precision, recall, F1-score, latency, and throughput, and compared with the conventional security measures to prove the enhancement of adaptability, detection efficiency, and the security of the network in general.

### *A. System Architecture*

The proposed framework has a system architecture based on the layered model of SDN, which incorporates smart security provided by ZTA and DRL. The architecture shown in Figure 1 consists of three main layers: the application layer, control layer, and data plane layer, which mutually interact in order to provide secure and adaptive data communication. Network services, user applications and security management interfaces are found in the application layer and they establish high-level requirements and policies. The control layer serves as the core of the system and is implemented using a centralized SDN controller such as Open Daylight, which maintains a global view of the network and dynamically manages traffic flows.

Two important modules are combined within the control layer, the zero-trust engine and the DRL agent. The zero-trust engine oversees the ongoing authentication, authorization and trust assessment of every user, device and application, so that no entity is trusted by default. It uses dynamic scores of trusts using contextual data including identity, behaviour and access history. The DRA agent is a kind of intelligent decision-making unit that constantly monitors the network conditions, the traffic features and the anomaly warning signs and decides the best security measures, such as permitting, rejecting, isolating or re-authenticating network requests. The flow rules are generated out of the decisions created by the DRL agent and implemented by the SDN controller.

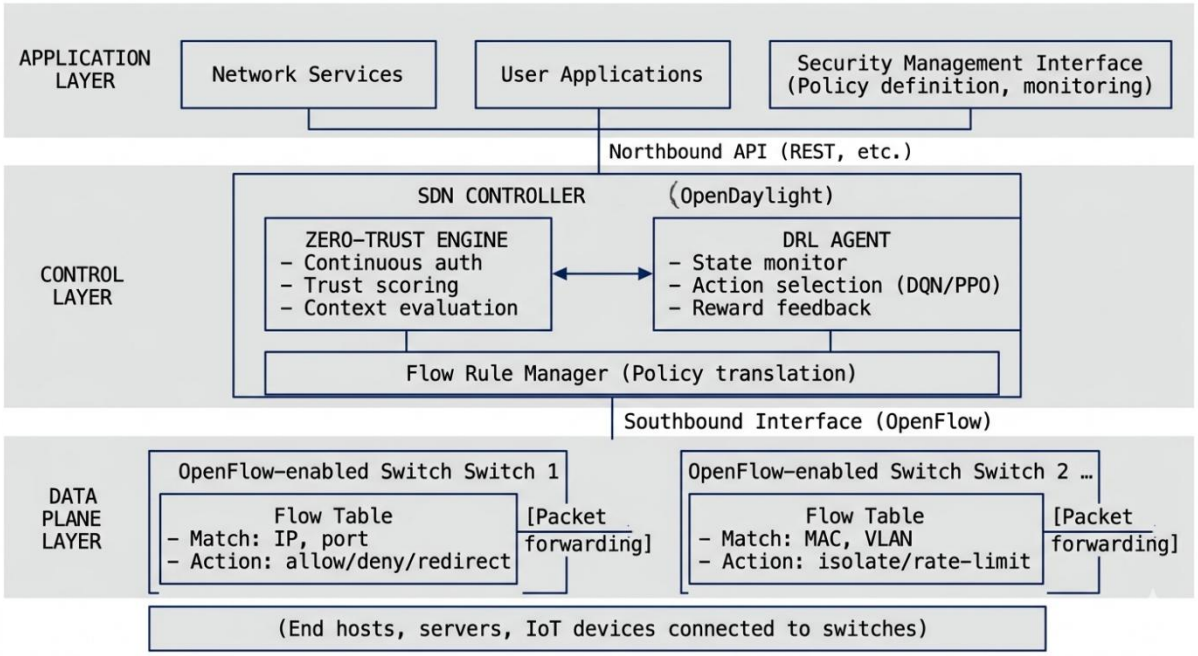


**Fig-1: Architecture of the Proposed Layered Structure of the SDN**

The data plane includes switches and forwarding devices functioning based on OpenFlow and which implement the policies that have been specified by the control layer. These devices process flow rules sent by the controller and packet forwarding is optimized according to them to achieve the enforcement of security decisions in real-time. The control layer and data plane communicate by using the standard southbound interfaces e.g. OpenFlow protocol and the northbound interfaces help the controller and application layer to interact. This combined architecture guarantees a centralized control, smart decision making and dynamic enforcement of a policy and thus increases the security, flexibility and efficiency of data communication in the network.

*(a) The Proposed Deep Reinforcement Learning Framework*

Figure 2 shows the suggested DRA structure that will be used as the intelligent decision-making part of the system allowing the SDN environment to adaptively and real-time control security. The DRL agent, which operates on the control layer and is coupled with the zero-trust engine, formulates the network security as a Markov Decision Process (MDP), comprising of states, actions, rewards, and a learning policy. The state space contains traffic flow statistics, packet level attributes, user behaviour patterns, trust scores and anomaly indicators whereas the action space contains the possible responses, which include allow access, deny request, isolate suspicious node or cause re-authentication. The reward function directs the learning process whereby positive rewards to right decisions (blocking malicious traffic) and negative rewards to wrong decisions (false positives and false negatives) are assigned. The DRL agent overcomes the constraints of the fixed rule-based systems by constantly communicating with the network environment and adjusting to changing threats and dynamic traffic situations.

To apply learning and policy implementation, the DRL architecture approximates optimal policy and value functions based on deep neural networks, and applies methods like Deep Q-Networks (DQN) method to scale state spaces with high dimensionality. The framework is combined with the SDN controller, like Open Daylight, where the decisions of the agent can be converted into flow rules and real-time deployed throughout the network. This integration makes adaptive security policies to be consistently enforced against immediate threats, provide proactive mitigation of risks, better detection rates, and increased overall network resilience. The DRA framework, thus, offers a scalable and smart system to self-adaptive zero-trust data communication within current SDN settings.

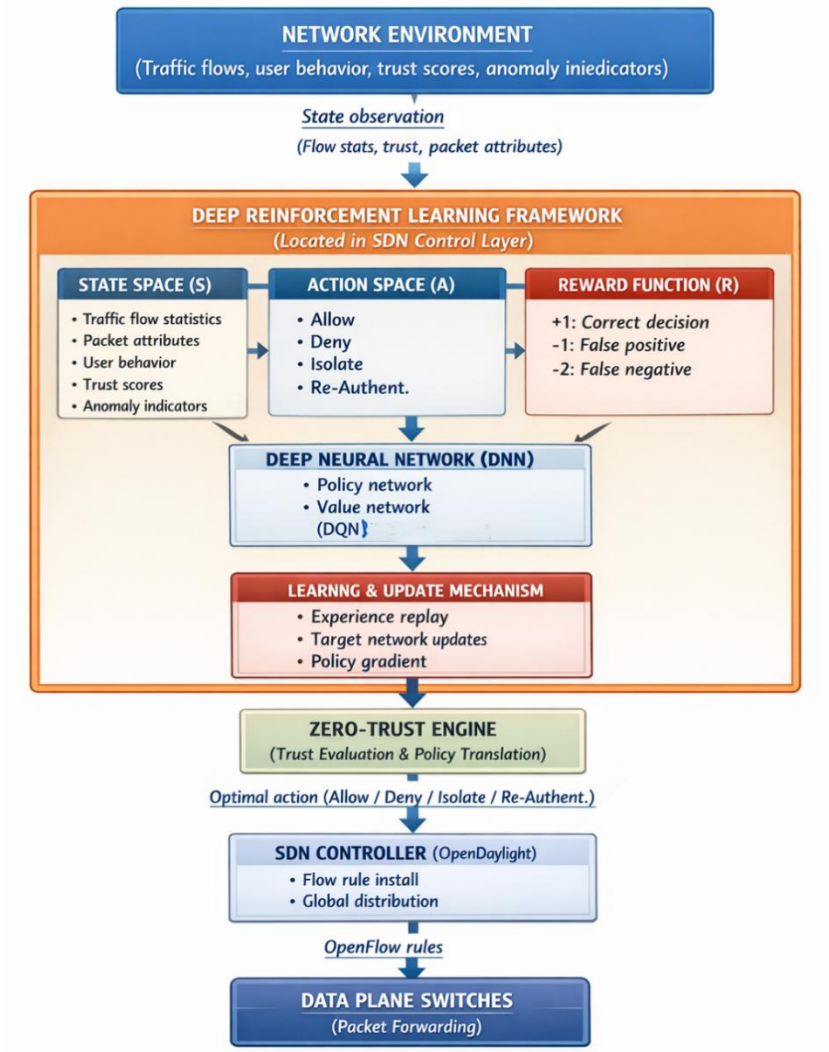


**Fig-2: Architecture of the Proposed DRL Framework**

*(b) The Proposed Zero-Trust Engine*

The zero-trust engine is a fundamental part of the system architecture, which derives its design concepts on the premises of ZTA to guarantee that none of the users, devices, or applications is trusted default, irrespective of whether or not they are located inside or outside of the network. The engine runs in the control layer of the SDN environment and is in charge of implementing continuous authentication, authorization and dynamic trust evaluation of all access requests as illustrated in Figure 3. It embraces a never trust, always verify concept and therefore all entities must be authenticated before network resources are allowed.

The below Figure 3 illustrated, the zero-trust engine has a number of functional modules, such as identity verification, access control, trust evaluation and policy enforcement. The identity verification module, authenticates the users and devices with the help of credentials like IP addresses, device fingerprints, or cryptographic tokens. Access control module is used to decide the level of access that is given based on the predefined security policies and the trust evaluation module is used to calculate the dynamic trust scores based on the contextual information like user behaviour, access history and results of anomaly detection. These trust scores are constantly updated to give an indication of the current network situation and the possible threat to security.

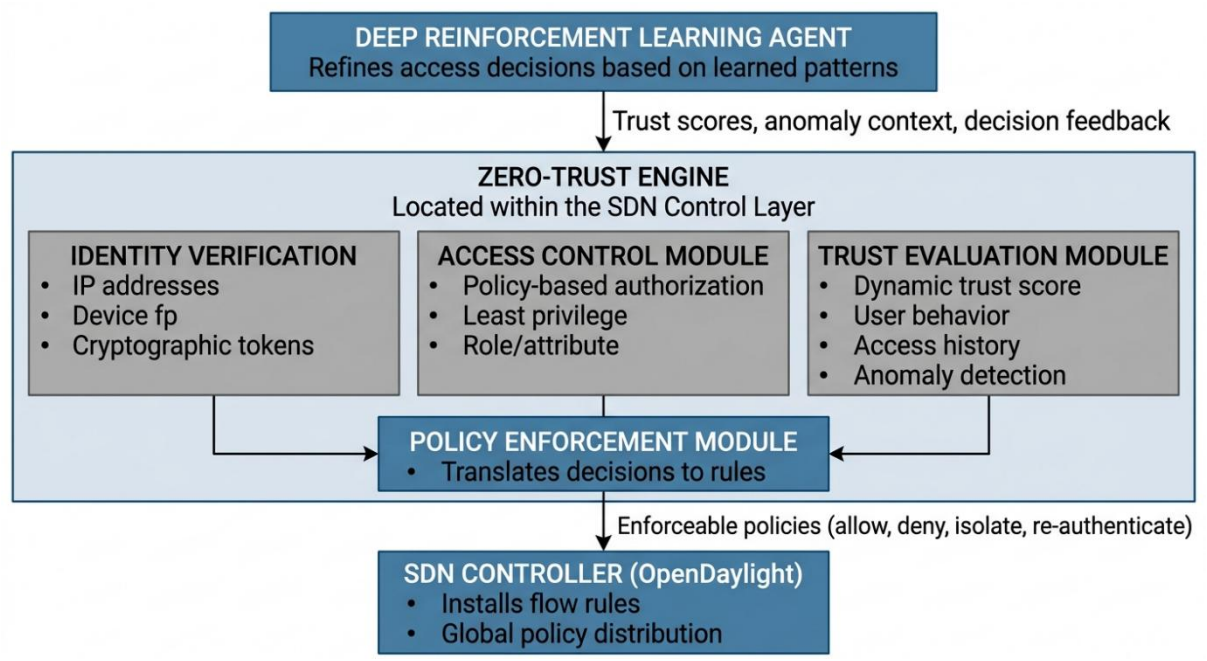


**Fig-3: Architecture of the Zero-Trust Engine**

The zero-trust engine is connected to the Deep Reinforcement Learning agent to maximise flexibility, training access control decisions on the observed patterns of normal and malicious behaviour. The DRL agent provides a decision to the engine which is then converted into enforceable security policies, e.g. permitting honest traffic, rejecting suspicious requests, isolating infected nodes or re-authentication. These policies are then sent to SDN controller to be implemented in the network.

In general, the suggested zero-trust engine guarantees fine-grained, real-time protection by dynamically verifying all entities and changing in response to changing threats. The combination with SDN and DRL allows proactive and smart defence mechanism which can considerably minimise the risk of unauthorised access, lateral movement and insider attacks within the network.

### *B. Data Collection*

To achieve this, data collection in this study is a combination of realistic network simulations and the CICIDS 2017 benchmark intrusion detection dataset to give a complete training and evaluation environment of the DRL agent. Mininet is used to perform the realistic network simulations, simulating a SDN setup with OpenFlow-capable switches and a centralized controller, like Open Daylight. Such simulations produce a variety of traffic patterns, such as regular user behaviour and artificially added malicious behaviour, enabling the DRL agent to train in changing network states and train real-time decision making. Traffic simulation gives us the ability to control network parameters on a fine scale with flow rate, packet size, and attack intensity being the parameters we can control to enable the model to adapt to different operational conditions.

Besides the simulated traffic, a second dataset, CICIDS 2017, is employed as a second, real-world dataset to improve the generalization and robustness of the model. CICIDS 2017 includes labelled flow-based network traffic of a complete week of normal and attack case scenarios such as Denial-of-Service (DoS), Distributed DoS, port scans, brute force attacks and infiltration attempts. The data goes through cleaning, normalization, feature extraction, and encoding to generate features including the size of packets, inter-arrival time, source/destination IP and ports, type of protocol and signs of anomaly. This data used together with the states of the zero-trust engine constitute the state space of the DRL agent. The combination of simulated SDN traffic and CICIDS 2017 within the framework will guarantee that the DRL agent will learn effective policies that can identify simulated and real-world threats and enhance the detection accuracy, adaptability, and overall network security performance.

### *C. Data Preprocessing*

A vital phase of the preprocessing of the simulated SDN traffic and the CICIDS 2017 dataset before training the DRL agent is data preprocessing. Raw network data is frequently noisy, inconsistent, has missing values and irrelevant features that may make models perform poorly. The preprocessing phase involves data cleaning, feature extraction, normalization, encoding and labelling. In the process of data cleaning, the missing or corrupted records are addressed by methods like the mean/mode imputation of numeric/categorical values and duplicate entries are eliminated. Z-score is used to identify and deal with outliers that could cause the learning process to be skewed by extreme values (Nwakeze and Mohammed, 2025). The feature extraction picks up the pertinent flow-level and packet-level features (addresses of the source/destination IP, ports, protocol type, flow duration, packet sizes, inter-arrival times, trust / anomaly scores) of the zero-trust engine. To normalize numerical features, Min-Max scaling is used to scale numerical features to have similar ranges to minimize bias when training neural networks (Umerah et al., 2025). One-hot encoding is used to encode categorical variables (protocol type, access decision labels) to convert them into numeric values which may be used by the DRL model. Lastly, the processed data is labelled and divided into training, validation and testing data, making sure that the DRL agent will be able to learn effectively and apply it to latent network conditions. Such preprocessing model ensures the agent receives high quality input that will guide it to come up with adaptive, zero-trust policies in the SDN environment.

### *D. System Implementation*

The proposed self-adaptive zero-trust system implementation will require the SDN environment that is simulated with Mininet offering a realistic environment to develop OpenFlow-enabled switches, hosts, and a centralized SDN controller like OpenDaylight. The controller is the control layer that has a universal perspective of the network and implements dynamic flow rules that are produced by the DRL agent and zero-trust engine. This system will enable real time monitoring, real time enforcement of policies and real time decision-making throughout the network. The implementation of the DRL agent is in Python with machine learning packages like TensorFlow and Keras, which enable you to create deep neural networks to learn a policy approximation and a value function. The zero-trust engine will be implemented at the control layer and will constantly calculate dynamic trust scores of all the entities in the network, which will be used to provide meaning to the actions of the DRL agent.

The DRL agent is coded in Python and based on TensorFlow and Keras, and it is trained on a mix of simulated SDN traffic and CICIDS 2017 dataset. Training: Feed in the pre-processed features of traffic to the agent, which then interacts with the network environment to discover what can be done, and then updates its policy via the reinforcement learning process. The reward mechanism uses positive rewards when selecting the right action (blocking malicious traffic) and punishment when selecting the wrong action (false positive or false negative) to have the agent learn the best security policies in a series of episodes. The zero-trust engine offers dynamic trust scores in order to enhance contextual decision-making. In the training phase, hyperparameters like learning rate, discount factor and exploration rate are tuned to converge. The decisions made by the DRA agent are implemented in real time with OpenFlow flow rules and the performance of the system is gauged by measuring factors like accuracy, precision, recall, F1-score, latency, and throughput, which illustrates a scalable, adaptive, and intelligent framework to secure SDN environments.

## 3. Results And Discussions

The analysis is oriented at the detection performance of the framework, adaptive decision making, network performance, and the general security improvements. The effectiveness of the proposed approach is measured with metrics like accuracy, precision, recall, F1-score, network latency, and throughput.

### *A. DRL Model Training Performance*

A mixture of the CICIDS 2017 dataset and simulated SDN traffic was used to train the DRA agent. The training was done on 500 episodes, the hyperparameters were: learning rate = 0.001, discount factor $\gamma = 0.95$, exploration rate e that decreases to 0.1. The cumulative reward and loss function were used to monitor the convergence of the agent. Table 1 shows the average cumulative reward per episode at various stages of training.

**Table 1: Average Cumulative Reward Results per Episode**

| Training Phase | Episodes | Average Cumulative Reward |
|---|---|---|
| Phase 1 | 1–100 | 45.6 |
| Phase 2 | 101–250 | 112.3 |
| Phase 3 | 251–400 | 178.9 |
| Phase 4 | 401–500 | 210.4 |

Table 1 shows that the cumulative reward is increasing, which means that the DRL agent is learning the best policies to use to secure the network. Initial episodes exhibit reduced rewards owing to discovery, whereas subsequent episodes exhibit increased aptitude of the agent to the malicious traffic and permit legitimate flows, owing to successful reinforcement learning.

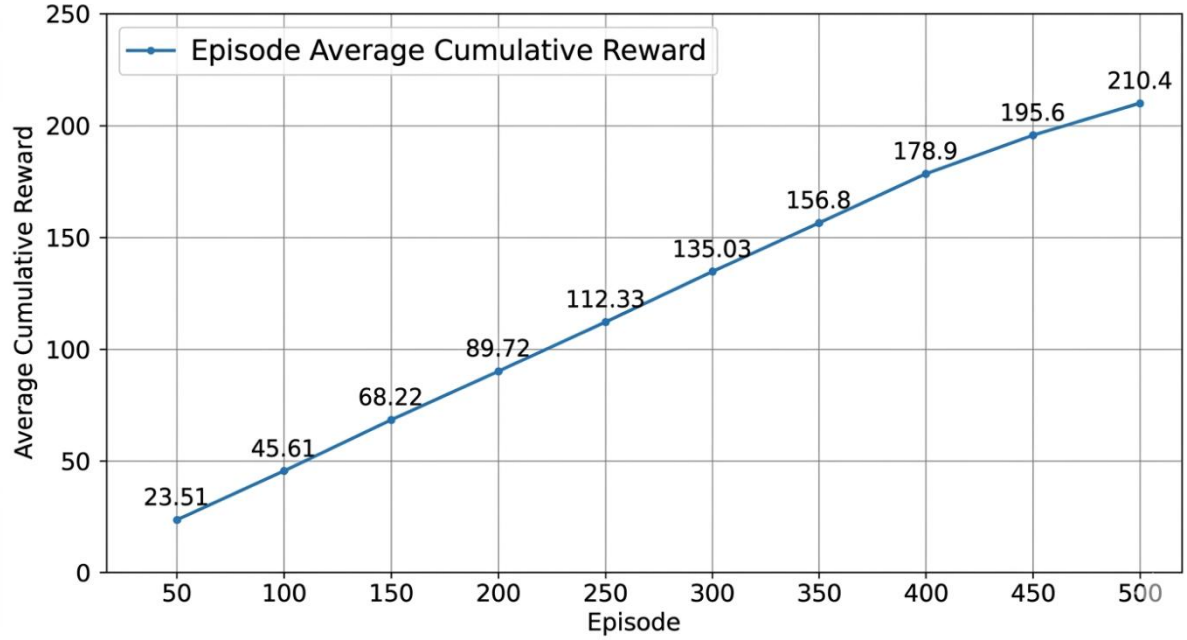


**Fig-4: Training Reward Curve Showing Cumulative Reward Vs. Episodes**

It is evident that the average cumulative reward per episode in Figure 4 increases steadily and steadily over the 500 episodes as the average cumulative reward is 23.5 in episode 50 and 210.4 in episode 500, which means that the DRL agent learns progressively the best policies in network security. The exploration phase occurs in the early episodes, and the agent tries various actions to learn about the network environment, which is reflected in lower rewards. With training, the agent learns more patterns of malicious and legitimate traffic, leading to more cumulative rewards. At episode 500 the agent is showing convergence of an optimal policy, indicating that it is able to reliably strike a trade-off between blocking threats and legitimate network flows. The trend proves the suitability of the reward function and the DRL learning process in the development of adaptive and self-learning security behaviour in the SDN environment.

### *B. Threat Detection Performance*

DRA, the DRA-based zero-trust framework performance was tested using accuracy, precision, recall and F1-score measures on simulated and CICIDS 2017 traffic. Table 2 summarizes the results compared with a baseline rule-based SDN security system.

**Table 2: Threat Detection Performance Results**

| Metric | Rule-Based SDN | DRL-Based Zero-Trust Framework |
|---|---|---|
| Accuracy (%) | 85.2 | 96.8 |
| Precision (%) | 82.5 | 95.7 |
| Recall (%) | 78.9 | 97.3 |
| F1-Score (%) | 80.7 | 96.5 |

The DRL-based framework is much more effective than the traditional rule-based SDN security, with an accuracy of 96.8% as opposed to 85.2% as in Table 2. An increase in precision and recall means that the system is effective in filtering malicious traffic as well as efficient in filtering legitimate traffic, thus minimizing false positives and false negatives. The F1-score of 96.5% indicates the general high strength and reliability of the proposed framework in threat detection.

### *C. Network Performance Evaluation*

Besides measuring security, the network efficiency was determined using latency and throughput measures to make sure that adaptive security enforcement would not compromise network performance. Table 3 shows the average network latency and throughput under normal and attack traffic conditions.

**Table 3: Network Performance Implementation Result**

| Attempt | Traffic Condition | Latency (ms) | Throughput (Mbps) |
|---|---|---|---|
| 1 | Normal Traffic | 12.1 | 942 |
| 2 | Normal Traffic | 12.4 | 938 |
| 3 | Normal Traffic | 12.3 | 940 |
| 4 | Normal Traffic | 12.5 | 941 |
| 5 | Normal Traffic | 12.2 | 939 |
| 1 | Attack Traffic | 14.6 | 912 |
| 2 | Attack Traffic | 15.0 | 908 |
| 3 | Attack Traffic | 14.8 | 910 |
| 4 | Attack Traffic | 14.9 | 911 |
| 5 | Attack Traffic | 14.7 | 909 |
| 1 | DRL Enforcement (Normal) | 12.7 | 934 |
| 2 | DRL Enforcement (Normal) | 12.9 | 932 |
| 3 | DRL Enforcement (Normal) | 12.8 | 935 |
| 4 | DRL Enforcement (Normal) | 12.6 | 933 |
| 5 | DRL Enforcement (Normal) | 12.7 | 936 |
| 1 | DRL Enforcement (Attack) | 13.4 | 927 |
| 2 | DRL Enforcement (Attack) | 13.6 | 929 |
| 3 | DRL Enforcement (Attack) | 13.5 | 928 |
| 4 | DRL Enforcement (Attack) | 13.7 | 926 |
| 5 | DRL Enforcement (Attack) | 13.5 | 929 |

Table 3 results show that the DRL-based framework has low latency and high throughput even in attack conditions. Despite slight delays in latency when mitigating against a threat with high intensity, the network performance is not greatly affected, proving that adaptive security decisions can be made without seriously affecting the service. The architecture allows balancing the network efficiency and security effectively, with high throughput and zero-trust policies.

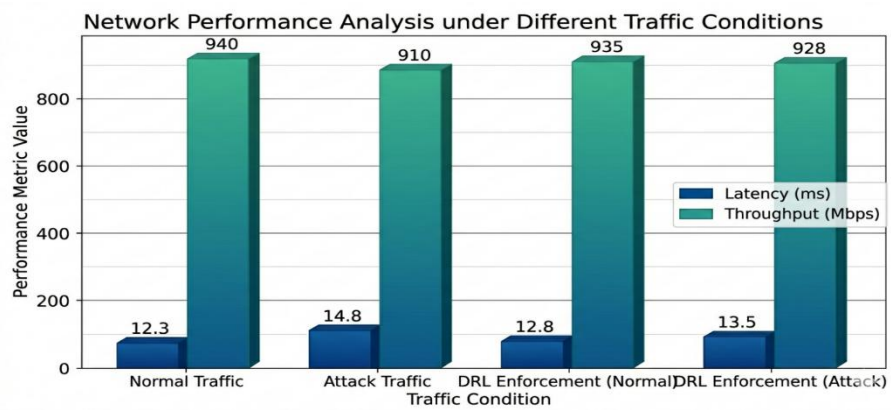


**Fig-5: Comparison Of Latency and Throughput Across Different Traffic Conditions**

According to the network performance results in Figure 5, the zero-trust framework that is based on DRL has shown low latency and high throughput in the conditions of both normal traffic and attack traffic. Although attack traffic raises the latency by 12.3ms to 14.8ms and the throughput by 940Mbps to 910Mbps, the addition of DRL enforcement reduces these effects, with latency decreasing to 12.8ms with normal flows and 13.5ms with attacks, and throughput recovering to 935Mbps and 928 These findings show that the DRL agent adaptive policies can be effective in managing the network resources and ensuring security and reducing degradation in performance even in the event of attacks. The insignificant differences between DRA-enforced traffic and natural traffic conditions suggest that the framework provides a satisfactory trade-off between the security and network efficiency, which means that the threat mitigation is robust without affecting the SDN performance in general.

### *D. Comparative Analysis with Baseline Systems*

A comparative analysis with the traditional SDN security mechanisms, such as fixed firewall rules and threshold-based intrusion detection was carried out to further justify the advantages of the suggested framework. Table 4 presents the main differences in detection performance and adaptability.

**Table 4: Comparative Performance Results with Baseline Techniques**

| Feature | Rule-Based SDN | Threshold IDS | DRL Zero-Trust Framework |
|---|---|---|---|
| Adaptivity to new threats | Low | Medium | High |
| Detection accuracy (%) | 85.2 | 89.1 | 96.8 |
| False positive rate (%) | 12.3 | 8.9 | 3.2 |
| Real-time enforcement | Partial | Partial | Full |
| Scalability | Limited | Moderate | High |

Table 4 has pointed out the benefits of the DRL-based zero-trust model as compared to the traditional security methods like rule-based SDN and threshold-based IDS. The framework is highly adaptive to new threats, which allows it to identify new attack patterns never seen before, but baseline systems are only low to medium adaptive. It has the highest accuracy in detection at 96.8% and the lowest false positive of 3.2% which greatly compares to the rule-based (85.2% accuracy, 12.3% false positives) and threshold-based (89.1% accuracy, 8.9% false positives) systems. The DRA framework also enables complete enforcement in real-time and is highly scalable, keeping security consistent within dynamic SDN environments without any manual policy changes. On the whole, this set of findings supports the idea that DRA + a zero-trust architecture is a strong, flexible, and effective solution to the contemporary network security issues.

The experimental findings indicate that the DRL-based zero-trust model is effective in learning adaptive security policies via reinforcement learning. The cumulative reward per episode, which increases with episode number, starting at 23.5 at episode 50 to 210.4 at episode 500, shows that the DRL agent has successfully converged to optimal policies. In the early episodes, exploration and policy adjustment can be observed, whereas in later episodes, the malicious traffic is consistently identified, and legitimate flows are permitted. The success of the reward role and the capacity of the agent to adapt to dynamic network conditions is confirmed by this learning process, which gives a basis to reliable, self-learning security behaviour in the SDN environment.

The framework performs greatly better in terms of performance compared to baseline rule-based SDN and threshold-based IDS strategies. Threat detection metrics are high (96.8%), precision (95.7%), recall (97.3%), and F1-score (96.5%), which indicates that the system is able to identify malicious flows with high accuracy and minimize false positives. Latency and throughput are also kept within normal traffic conditions even during attacks and this indicates that adaptive security decisions do not affect performance. The comparative analysis underscores the superior adaptability, real-time enforcement and scalability, which support the capability of the framework to respond to dynamic threats without human intervention. On the whole, the findings confirm that DRA can be effectively implemented to guarantee the security of the modern SDN by integrating it with a zero-trust architecture and offering a scalable, reliable, and efficient solution.

## 4. Conclusion

The paper designed and tested a self-adaptive zero-trust data communication model of the Software-Defined Networks, and used Deep Reinforcement Learning to implement intelligent, real-time security enforcement. The framework consists of a zero-trust engine and a DRL agent, which constantly assesses the level of trust and applies access control policies on the network. Using the simulated SDN traffic and the CICIDS 2017 benchmark data set to train the DRA agent enabled the system to train on the best security policies, which helps it to differentiate between legitimate and malicious traffic. These episode-by-episode cumulative reward patterns indicate that there is a steady learning and convergence to an optimal policy, which confirms that the reward function is working and the agent can adjust to changing network dynamics.

The results of the evaluation show that the suggested DRL-based framework is much more efficient in comparison to conventional rule-based SDN and threshold-based IDS frameworks. The system has a high detection accuracy of 96.8%, precision of 95.7%, recall of 97.3% and F1-score of 96.5%, meaning that it is able to recognize threats well, but with minimal false positives and false negatives. Measurements of network performance also confirm that adaptive security enforcement does not affect efficiency with latency and throughput being very similar to normal traffic conditions even under attack conditions. These results demonstrate the potential of the framework to ensure a balanced trade-off between security and network performance, in order to cause less disruption in the event of real-time threat mitigation.

The comparison to the baseline systems highlights the superiority of the framework in regard to its adaptability, scalability and real-time enforcement of policies. The DRA zero-trust framework does not require human intervention to learn new patterns of threat or implement security policies, unlike the more traditional systems that are based on static or a threshold mechanism. On the whole, this work shows that the interplay between zero-trust and DRL in SDN setting offers a powerful, intelligent and scalable answer to the current network security dilemma, which can withstand the changes in cyber threats and still achieve the best network performance.

## 5. References

[1] Barach, J. (2025). Towards zero trust security in SDN: A multi-layered defence strategy. In Proceedings of the 26th International Conference on Distributed Computing and Networking (pp. 331–339). Association for Computing Machinery. https://doi.org/10.1145/3700838.3703671

[2] Bijlwan, V. P., & Kumar, N. (2026). Deep reinforcement learning in software defined networking: A survey, research challenges, and future perspectives. IEEE Communications Surveys & Tutorials, 28, 4624–4653.

[3] Chenghao, S., Wang, Y., Su, Y., & Wan, Y. (2022). Adaptive threat mitigation in SDN using improved D3QN. In Second International Conference on Cloud Computing and Mechatronic Engineering (I3CME 2022) (Proc. SPIE 12339, 1233911). https://doi.org/10.1117/12.2652679

[4] Eke, R. N., & Onyejegbu, L. N. (2024). A deep reinforcement learning deep Q network-based approach for systematic detection of distributed denial of service attacks in software-defined networking. SSRN. https://ssrn.com/abstract=5023193 or http://dx.doi.org/10.2139/ssrn.5023193

[5] Kalekar, V. (2026). Uncertainty-aware reinforcement learning for zero trust: An empirical evaluation of PPO, MC dropout, and Bayesian neural networks. In Proceedings of the 12th International Conference on Information Systems Security and Privacy (ICISSP 2026) (Vol. 1, pp. 505–514). https://doi.org/10.5220/0014245800004061

[6] Kanimozhi, R., & Ramesh, P. S. (2025). Deep reinforcement learning-based intrusion detection scheme for software-defined networking. Scientific Reports, 15, 38827. https://doi.org/10.1038/s41598-025-24869-w

[7] Ke, W., Li, Z., Chen, P., Chen, B., & Lv, J. et al. (2025). DRL-AMIR: Intelligent flow scheduling for software-defined zero trust networks. Computers, Materials & Continua, 84(2), 3305–3319. https://doi.org/10.32604/cmc.2025.065665

[8] Khan, R. I., Habib, M. Q. B., Hasan, M. M., & Shad, K. W. U. (2026). Zero trust architecture in cloud-native environments: A scalable framework for cybersecurity. International Journal of Science and Research Archive, 18(1), 296–305. https://doi.org/10.30574/ijsra.2026.18.1.0063

[9] Mahida, A. (2026). An intellectual zero trust security framework using deep reinforcement learning for predictive threat mitigation in AI-based fraud detection systems. IEEE Access, 14, 24602–24617.

[10] Muhammad, Z. (2026). Zero trust architecture (ZTA) design and implementation: A comprehensive review. American Journal of Innovation in Science and Engineering, 5(1), 18–25. https://doi.org/10.54536/ajise.v5i1.5905

[11] Niknami, N., & Wu, J. (2023). DeepIDPS: An adaptive DRL-based intrusion detection and prevention system for SDN. Center for Networked Computing, Temple University, USA.

[12] Nwakeze, O. M., & Mohammed, N. U. (2025). Development of a deep learning-based framework for security of IoT network protocol. *International Journal of Innovative and Applied Research (IJIAR), 7*(10), 1–12. International Journal of Innovative and Applied Research. https://doi.org/10.58538/IJIAR/2136

[13] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207

[14] Umerah, A. T., Azaka, M., Nwakeze, O. M., Obaze, C. A., & Ibeh, S. C. (2025). AI-driven diagnostic imaging: Hybrid CNN-GNN models for early detection of cancer from pathological images. *International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), 14*(9), 579–588. International Journal of Latest Technology in Engineering, Management & Applied Science. https://EconPapers.repec.org/RePEc:bjb:journl:v:14:y:2025:i:9:p:579-588