scientific reports

Check for updates

OPEN

Deep reinforcement learning-based intrusion detection scheme for software-defined networking

R. Kanimozhi[1]✉ & P. S. Ramesh[2]

A robust Deep Reinforcement Learning-based Intrusion Detection Scheme (DRL-IDS) for Software-Defined Networking (SDN) which combines the Long-Short Term Sequence Recurrent Neural Network (LFTS-RNN) with the Particle Cloud-Integrated Joint Time- and Feature-Optimization Algorithm (PC-JTFOA). The hybrid model aims to enhance the security of SDN through the detection and mitigation of a wide array of Distributed Denial of Service attacks and network misbehaviors across different SDN planes. The LFTS-RNN is used for accurate attack detection and misbehavior identification. Meanwhile, the PC-JTFOA optimizes feature selection, load balancing, and energy-efficient routing, thus ensuring fast and reliable network traffic management. The deep reinforcement learning approach further enables continuous adaptation to changing network behaviors, thus making the model dynamically adapt to known as well as emerging attack vectors. The proposed DRL-IDS scheme obtains superior performance in experimental results based on the NSL-KDD and WPPD datasets. The LFTS-RNN model indicates a highly impressive sensitivity of 98.67% and specificity of 97.42%, while the DRL-IDS model presents an detection accuracy of 99.85%. The PC-JTFOA further improves the solution by exhibiting a low response time of 1423 ms, which indicates tremendous improvement in computational efficiency. A comparative analysis with the existent intrusion detection methods pointed out that the scheme proposed not only outperforms other models in terms of detection accuracy as well as adaptability, but it also reduces complexity.

Keywords Deep reinforcement learning, Intrusion detection system, Software-defined networking, Long short-term memory network, Particle colony-adjusted jumping teaching fishing optimization algorithm, Distributed denial of service attack detection

Intrusion Detection Systems (IDS) are essential security controls that monitor computer networks to prevent unauthorized access and malicious activities. IDS actively monitors network traffic and system behaviors to identify unusual or unauthorized actions. This helps in finding potential threats such as malware, hacking attempts and network intrusions that might compromise the security and integrity of systems. The basic purpose of IDS is to alert users in real time and protect them from the damage caused by the attack so that network resources are safe from internal and external threats[1]. Software-Defined Networking is a new group of system architecture that decouples the control plane from the data plane, allowing centralized control and more flexible management of networks. In SDN, the control of a network is abstracted, thus allowing network administrators to control the flow of traffic programmatically and to make real-time changes to network configurations through software applications[2]. This architecture offers better scalability, improved efficiency and easier network management and enables dynamic reconfiguration of the network in response to changing demands. SDN is increasingly being used in large-scale, complex networks because of its agility and cost-effectiveness in managing network resources[3].

The combination of deep reinforcement learning and advanced algorithms is being chosen, such as the Long Short-Term Memory Network (LFTS-RNN) and Particle Colony-Adjusted Jumping Teaching Fish Algorithm Optimization Algorithm (PC-JTFOA) along with their capabilities that facilitate the efficiency and flexibility that IDS in SDNs offer. DRL ensures autonomy in learning and improving adaptation to new threats at over time, making an evolution in attack strategies highly effective. LFTS-RNN works in dealing with sequential data with the detection of network-traffic patterns and PC-JTFOA deals with optimizations in resource allocation that

[1]Department of Artificial Intelligence and Data Science, A.V.C. College of Engineering, Mayiladuthurai, Tamilnadu, India. [2]Department of CSE, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India. ✉email: kanimozhivedharajan@gmail.com

minimize the overall response time while enhancing efficiency in detection. All such methods help to increase precision, flexibility, and speed in identifying and dealing with complex threats in SDNs[4,5].

Traditionally, the Intrusion Detection Systems generally employ signature-based or anomaly-based detection methods. The former IDS compares the incoming network traffic with a database of known attack patterns from established baseline behavior. These techniques lack the ability to identify new or sophisticated attacks. Traditional approaches to IDS face high false positive rates and long response times in cases of large-scale networks or real-time data[6].

Traditional IDS methods facing challenges include having high computational complexity, efficiency in dealing with enormous volumes of network traffic, and not being adaptable enough to emerging threats. Signature-based systems cannot identify new, unknown attack patterns, and anomaly-based systems can cause too many false positives, leading to alert fatigue. Additionally, traditional systems are often unable to scale effectively in dynamic network environments like Software-Defined Networking, where rapid changes in network configurations require real-time threat detection and fast response times. These limitations hinder the overall security performance, prompting the need for more advanced, machine learning-driven solutions[7,8].

The requirement for an adaptive, efficient, and scalable IDS solution that can properly safeguard SDN infrastructures against known and unknown threats continues to be a very pertinent issue. There lacks more than what is being delivered nowadays through existing IDS approaches toward that end: accuracy and sensitivity and response time coupled with the ability to keep an adaptive sense of evolving networking systems. The motivation from this research is for development into an advanced SDN IDS using DRL fused with LFTS-RNN and PC-JTFOA. It shall aim to enhance the accuracy of detection, reduce false positives, and improve adaptability to new and evolving threats while ensuring low response time and computational efficiency. Another goal is to provide scalability for handling high volumes of data and complex attack patterns in SDN environments[9,10]. The main contributions of the proposed work is listed below.

- A DRL-based IDS for Software-Defined Networking focused on an LFTS-RNN approach to determine attack patterns and misbehaving actions.
- This reserch achives an accuracy 99.85% through the DRL-based IDS model that proves as better compared to previous traditional intrusion detection systems.
- The Particle Colony-Adjusted Jumping Teaching Fishing Optimization Algorithm (PC-JTFOA) is sued for-feature selection, load balancing, and energy-efficient data transmission which reduces the computational complexity.
- The LFTS-RNN model has an impressive sensitivity of 98.67% and specificity of 97.42%, in detecting both known and unknown attacks in SDN.
- PC-JTFOA reduces the response time to 1423 ms and improves computational efficiency.

The paper organization is structured as follows. The related surveys are illustrated in Sect. 2. Section 3 represents the proposed technique's and the DRL-IDS mrthod. The performance of the proposed method are signified in Sect. 4. Finally, Sect. 5 concludes the paper with ideas of future directions.

Literature survey

AlEroud & Alsmadi (2017) outlines an inference-based intrusion detection approach for cyber-attacks against SDNs. The method involves an inference engine with anomaly detection from the perspective of real-time attack identification. Its focuses on the detection of numerous cyber-attacks across the SDN planes and is versatile to different network environments[11]. Ibrahim & Bhaya (2021) introduces a cloud-based SDN intrusion detection system. It adopt a hybrid model, combining machine learning and signature-based techniques. It is focus on cloud-based SDN environments, which require unique intrusion detection strategies because of the dynamic nature of cloud computing[12]. Satheesh et al. (2020) proposed flow-based anomaly intrusion detection system with the help of machine learning technique with the application of SDN's using OpenFlow. Flow-based detection and analysis method monitor and check patterns and detect anomalies on these networks. It has the capacity to handle a real-time huge scale traffic data[13].

Alshahrani et al. (2023) discuss the intrusion detection framework for the Industrial Internet of Things (IIoT) using SDNs. The approach incorporates SDN and IIoT in detecting security intrusions. It offers an efficient method in securing those networks and proposes a novel method in an IIoT environment[14,15]. Alshammri et al. (2022) presents an efficient intrusion detection framework for SDNs focused on cybersecurity applications. Its methodology involves a combination of deep learning and network traffic analysis[16]. Ha et al. (2016) proposed traffic sampling for intrusion detection in SDNs. The proposed methodology focuses on sampling-based to decrease the volume of required analysis of traffic without impairing detection accuracy. Its advantage lies in the efficient management of large-scale network traffic without causing significant computational overhead[17]. Yazdinejadna et al. (2021) present an SDN-based intrusion detection system based on a kangaroo approach. This method utilizes the kangaroo optimization algorithm to recognize intrusions by monitoring patterns in network traffic. This advantage is that the kangaroo algorithm is novel, making its use improve detection efficiency while reducing false positives[18]. Janabi et al. (2022) deals with overhead reduction in SDN-based IDS by using an optimized algorithm. The methodology involves the use of resource-efficient techniques for reducing the computational cost of IDS. The advantage lies in optimizing network resources and improving the scalability of intrusion detection systems[19].

Naqash et al. (2022) presents a statistical analysis-based intrusion detection system for ultra-high-speed SDNs. By using statistical techniques, it identify anomalies in high-speed networks. The strength of the approach is its capability in handling ultra-high-speed data and fast detection[20]. Tian et al. (2021) proposed a two-stage intrusion detection approach for SDN-based IoT networks. This approach combines signature-based and

anomaly-based detection techniques to enhance detection accuracy[21]. Li et al. (2018) proposes an AI-based two-stage intrusion detection system for SDN-enabled IoT networks. The approach integrates machine learning models for improving the accuracy of detection. Its strength is in the application of AI, which automates the detection process, making the system overall more efficient and adaptable[22]. Ali & Yousaf (2020) proposed a novel three-tier intrusion detection and prevention system for SDNs. The research work makes use of hierarchical detection and prevention strategies that aim to curb the threat within SDNs. One advantage of this work is multi-tier architecture which strengthens the detection and prevention ability at the different levels of the network[6].

Bhardwaj et al. (2022) uses a self-organized constraint-based intelligent learning framework in the context of network intrusion detection in SDNs. This methodology integrates self-organization and constraint-based learning for better intrusion detection improvement. The advantage of the approach is adaptability and flexibility in learning patterns within the network traffic, decreasing false positives[10]. Satheesh et al. (2020) proposed a flow-based anomaly intrusion detection system using machine learning for SDNs. It focuses on the usage of the OpenFlow protocol in SDNs for traffic monitoring. The advantage lies in its ability to detect network anomalies with high accuracy for automated detection[13]. Abdulqadder et al. (2020) provides a multi-layered intrusion detection and prevention system for SDN/NFV-enabled 5G networks with AI-based defense mechanisms. The methodology utilizes AI and machine learning techniques to improve the security of SDN/NFV networks in 5G[23]. Garg (2022) proposes a self-organized constraint-based intelligent learning framework for SDN network intrusion detection. In this approach, the intrusion detection system uses an intelligent learning framework to detect and prevent intrusion. It is self-organizes and adapts to novel attack patterns by strengthening the detection[24]. Zavrak & Iskefiyeli (2023) proposes an approach for anomaly detection from multivariate time series in SDNs flow-based intrusion detection. The methodology makes use of advanced time-series analysis for the discovery of network anomalies[25]. Alhaidari et al. (2021) proposed an intelligent SDN approach for the optimization of cognitive routing using deep extreme learning machine approaches. The proposed approach combines cognitive routing with deep learning for efficient routing and intrusion detection. The merits lie in the incorporation of deep learning for improved routing as well as the prediction of network traffic[26].

Alnaser et al. (2024): This paper optimizes multi-tier scheduling and secure routing in edge-assisted SDWSNs using AI techniques. The methodology uses AI for optimizing routing and ensuring secure communication. The advantage is its application in edge-assisted networks, improving the security and efficiency of SDWSNs[27]. Setiawan et al. (2022) focuses on encrypted network traffic classification and resource allocation using deep learning in SDNs. The methodology relies on deep learning models for the classification of encrypted traffic as well as efficient resource allocation. The advantage of this approach is that it can deal with encrypted traffic, which is becoming increasingly common in modern networks[28]. Kipongo et al. (2023) proposes an artificial intelligence-based IDS and prevention system in edge-assisted SDWSNs with a modified honeycomb structure. The methodology is used for intrusion detection and prevention through AI in SDWSNs. The benefit is its novel honeycomb structure, which improves the performance of the system in edge networks[29]. Phan & Bauschert (2022) introduces a deep reinforcement learning-based adaptive intrusion response system for SDNs (DeepAir). The methodology is based on deep reinforcement learning that allows the response to attacks to be dynamically adjusted[30,31].

Table 1 highlights the techniques, data sets, measurements, and limitations of the main IDS methods for SDN/IoMT. It shows how the suggested algorithm integrates DRL, LFTS-RNN, and PC-JTFOA in an unusual manner to get a low latency, reduced FPR, and excellent accuracy throughout SDN layers.

Additionally to traditional flow-, signature- and anomaly-driven systems for intrusion detection, a variety of recent investigations have created machine learning and reinforcement-learning techniques for protecting fluid SDN settings. In IoMT and SDN contexts, the combination of CNN–LSTM and deep reinforcement learning algorithms (DQN, PPO) was used to continually counteract shifting invasion patterns while achieving high precision with low delay[32,33].

For encrypted traffic analysis in internet of things SDNs, memory-feedback Generators used together with LSTM models have been suggested suggested; they demonstrate better dynamic modeling and resistance to geographic alterations[34]. Similarly, to strengthen anomaly detection in non-stationary situations, RCLNet combines CNN/LSTM encoders, self-adaptive focus, with radio frequency feature ranking.

These investigations show how scaling, instability, and feature spaces with high dimensions are dealt with in network safety by current RL/DL systems. Our DRL-IDS extends this strategy through integrating an LFTS-RNN detection with a PC-JTFOA optimizer in order to collaborate on energy-efficient scheduling, scheduling, and choice of features across network planes. It additionally specifically displays energy and delay metrics, that were overlooked in prior transformer-based or IoMT-focused methods[35,36].

Proposed methodology for intrusion detection scheme

The proposed methodology for the IDS uses Deep Reinforcement Learning to increase the accuracy of intrusion detection as well as adaptability to dynamic networks. The method trains the DRL agent to inspect network traffic patterns and detect intrusion by constantly learning from experiences with the network. The agent uses a reward-based system in which it receives positive feedback for correct identification of intrusions and negative feedback for false positives or missed threats. Using a deep neural network architecture, the agent can handle complex, high-dimensional data, allowing it to identify known as well as unknown attack vectors. The detailed explanation of the proposed work SDN is described below.

Deep reinforcement learning-based intrusion detection scheme (DRL-IDS)

DRL-IDS is designed to face the challenges of detecting network anomalies and intrusions in dynamic and complex SDN environments. It uses centralized control and programmability, which make it efficient but expose it to possible security vulnerabilities. DRL-IDS utilizes Deep Reinforcement Learning (DRL) in the analysis of

No.	Study/year	Technique	Dataset(s)	Metrics reported	Key limitations
1	AlEroud & Alsmadi (2017)	Inference-based IDS	Custom SDN traffic	Accuracy 91%	Limited to static rules; no latency analysis
2	Satheesh et al. (2020)	Flow-based anomaly detection (ML)	OpenFlow traffic	Acc. 90.8%, FPR 1.4%	High FPR, no adaptive learning
3	Li et al. (2018)	AI-based two-stage IDS	SDN-enabled IoT	Acc. 96.3%	No feature/load optimisation; modest scalability
4	Phan and Bauschert (2022) – DeepAir[30]	DRL + adaptive response	NSL-KDD	Acc. 98.2%	No energy or latency metrics; focuses only on control plane
5	Shaikh et al. (2025)	DRL + CNN–LSTM for IoMT	CICIoMT2024	Acc. 99.5%, F1 99.6%	IoMT only; no SDN routing or multi-plane evaluation
6	Shaikh et al. (2025)	MF-Transformer (MF-LSTM)	WUSTL-EHMS, ECU-IoHT, X-IIoTID	Acc. 99.8% (signature), 99.7% (anomaly)	No latency/energy analysis; limited to healthcare IoT
7	Shaikh et al. (2024)	RCLNet (RF + CNN/ LSTM + Attention)	IoMT traffic	Acc. 99.3%	Not evaluated on large-scale SDN; no load-balancing
8	Proposed DRL-IDS	DRL + LFTS-RNN + PC-JTFOA	NSL-KDD, WPPD	Acc. 99.85%, Sens. 98.67%, Spec. 97.42%, FPR ≤ 0.70%, RT ≈ 1.4s	Addresses gaps: combines DRL + RNN with feature/load optimisation and energy-aware routing; evaluated across SDN planes
9	Proposed DRL-IDS (this work)	DRL + LFTS-RNN + PC-JTFOA	NSL-KDD, WPPD, ICECIE-2021 dataset	5-fold CV Acc. = 99.72 ± 0.08%; F1 = 99.6%; Spec. = 97.4%; FPR ≤ 0.70%; RT ≈ 1.4 s	Addresses gaps: combines DRL, temporal modelling, and feature/load optimisation; validated with cross-validation and an additional dataset to confirm generalisation
9	Author et al. (2023) DOI:https://doi.org/10.1007/s11042-023-16894-6	AI-based IDS for SDN	Benchmark SDN dataset	Accuracy ~ 98%, FPR not reported	DRL-IDS achieves higher accuracy and reports latency/energy metrics not covered in this work.
10	Author et al. (2025) DOI:https://doi.org/10.1016/j.compeleceng.2025.110561	Deep-learning IDS	IoT/SDN hybrid data	Acc. 97%, F1 96%	DRL-IDS shows stronger generalisation via cross-validation and tests on newer datasets.
11	Author et al. (2024) DOI:https://doi.org/10.1038/s41598-024-67984-w	Hybrid deep IDS	Industrial IoT traces	Acc. 98.5%, FPR ~ 1%	Our model integrates PC-JTFOA for optimisation and evaluates across SDN planes.
12	Author et al. (2024) DOI:https://doi.org/10.1038/s41598-024-75414-0	Transformer-based IDS	Cloud SDN data	Acc. 99%, no latency analysis	DRL-IDS emphasises low FPR and response time, missing in this work.
13	Author et al. (2023) DOI:https://doi.org/10.1007/s41870-023-01332-5	CNN-LSTM IDS	Mixed IoT datasets	Acc. 98%, F1 97%	DRL-IDS balances accuracy and efficiency, validated on diverse datasets.

Table 1. Comparison of representative IDS approaches for SDN/IoMT environments. RT = Response Time, Acc. = Accuracy, Sens. = Sensitivity, Spec. = Specificity, FPR = False Positive Rate.

network traffic, allowing it to act as an intelligent agent autonomously learning from data patterns to detect malicious activities. This ability ensures that DRL-IDS adapts very well to the changing nature of network threats. The advantages of DRL-IDS lie in its ability to learn and improve through continuous interaction with the network environment. Unlike the traditional systems that rely on static rules or pre-trained models, DRL-IDS dynamically updates its strategies of detection by optimizing the action in a feedback loop. This adaptability makes it easier for the system to handle both known attack patterns as well as novel or emerging threats by generalizing beyond specific signatures that make it more robust against sophisticated or previously unseen intrusion techniques.

In addition, DRL-IDS incorporates state-of-the-art neural network architectures and optimization methods to optimize detection performance. DRL helps the system make decisions based on long-term benefits, which minimizes false positives and negatives. It not only detects anomalies in real-time but also continuously refines its detection model, making it a scalable and efficient solution for intrusion detection in SDN environments, where traffic patterns and security risks are highly dynamic. The bellman equation in (1) is derived from the principle of optimally in dynamic programming which states that the value of a state action pair is immediate reward plus the discounted future reward from the next state. In this equation is the reward for taking action at state and is the discount factor.

In addition, DRL-IDS incorporates state-of-the-art neural network architectures and optimization methods to optimize detection performance. DRL helps the system make decisions based on long-term benefits, which minimizes false positives and negatives. It not only detects anomalies in real-time but also continuously refines its detection model, making it a scalable and efficient solution for intrusion detection in SDN environments, where traffic patterns and security risks are highly dynamic. The bellman equation in (1) is derived from the principle of optimally in dynamic programming which states that the value of a state action pair is immediate reward plus the discounted future reward from the next state. In this equation r_t is the reward for taking action at state s_t and γ is the discount factor.

$$Q(s_t, a_t) = E\left[r_t + \gamma \max_{a'} Q(s_{t+1}, a')\right] \quad (1)$$

In Eq. (2), the policy based DRL is used to maximize the expected cumulative reward $J(\pi\theta)$

$$J(\pi\theta) = E_{\pi\theta}[R_t] \tag{2}$$

In Eq. (3), the likelihood ratio trick is the gradient of $J(\pi\theta)$ with respect to θ where reward acts as a weighting factor.

$$\nabla_\theta J(\pi\theta) = E_{\pi\theta}[R_t \nabla_\theta log\pi\theta(a_t|s_t)] \tag{3}$$

The loss function minimizes the difference between predicted Q values and the target value where y_t is defined using Eq. (5). The loss is derived from the Mean Squared Error(MSE) for supervised learning.

$$L(\theta) = E\left[(Q(s_t, a_t; \theta) - y_t)^2\right] \tag{4}$$

$$y_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta) \tag{5}$$

The reward function is designed to reinforce correct classification using the Eq. (6). This comes from the requirement to penalize misclassifications while rewarding correct classification and aligning with reinforcement learning.

$$r_t = \left\{ \begin{array}{c|c} +1 & if\ correct\ classification \\ -1 & if\ incorrect\ classification \end{array} \right\} \tag{6}$$

In Eq. (7), the weight update rule is used for minimizing a loss $L(\theta)$. this comes from differentiating $L(\theta)$ with respect to θ to find the direction of steepest descent.

$$\theta_{t+1} = \theta_t - \alpha \nabla_t L(\theta) \tag{7}$$

The joint cost function combines feature selection, time optimization and detection accuracy as given in Eq. (6). The regularization terms $\|w\|^2$ and $\|T\|^2$ penalize over fitting for weights and time variables.

$$C(w, T) = \lambda_1 ||w||^2 + \lambda_2 ||T||^2 + L_{detection} \tag{8}$$

In Eq. (9), the softmax function assigns probabilities to action based on the Q values. It is derived by normalizing exponential scores of Q values to ensure the sum of probabilities is 1.

$$\pi(a_t|s_t) = \frac{\exp(Q(a_t, s_t))}{\sum_{a'} \exp(Q(a', s_t))} \tag{9}$$

The anomaly detection uses z score as given in Eq. (10) where μ and σ are the mean and standard deviation of normal traffic. This score measures how far a data point x deviates from the mean in terms of standard deviations.

$$A(x) = \frac{\| x - \mu \|}{\sigma} \tag{10}$$

In Eq. (11), the sigmoid function is derived and the function maps any real values z to the range(0,1).

$$y = \frac{1}{1 + epx(-w^T x + b))} \tag{11}$$

The Particle Swarm Optimization (PSO) velocity and position update rules are given in Eqs. (12 and 13).

$$v_{i,j}(t+1) = wv_{i,j}(t) + c1r1(p_{i,j} - x_{i,j}) + c2r2(g_j - x_{i,j}) \tag{12}$$

$$x_{i,j}(t+1) = x_{i,j}(t) + cv_{i,j}(t+1) \tag{13}$$

$wv_{i,j}(t)$ encourages exploration of the search space, $c1r1(p_{i,j} - x_{i,j})$ pulls $c2r2(g_j - x_{i,j})$ particles toward the personal best and pulls particles toward the global best.

Software defined network

Software-Defined Networking (SDN) is a contemporary notion of network paradigm that has three separate planes: application plane, control plane, and network plane, which altogether enrich network management. The proposed Intrusion Detection Scheme in SDN identifies and classifies attackers into three categories: Masqueraders or unauthorized outsiders accessing private information, Misfeasors or insider misuse of their privileges - either through unauthorized access or an abuse of legitimate permissions-and Clandestine users who could be insiders or outsiders seizing supervisory control and abusing authority. To ensure robust detection, an IDS would run on each SDN plane ensuring that security and reliability improve by tracking threats specific to each plane.

User registration

In the proposed SDN framework, the registration of user at the application layer through a combination of username and password is utilized. The registration process actually distinguishes between the rightful users and intruders. Thus, only authorized people get access to the network. The total number of registered users (N_R) is as a function of the usernames registered (U) with their corresponding passwords (P), indicating that unique user credentials are required. This scheme reinforces the first layer of security through the building of a strong, authenticated database of users.

The proposed model further enhances security through the incorporation of a Dynamic Key Management System (DKMS). The DKMS issues and binds a Secure Access Credential (SAC) and a private key to each legitimate user. The cryptographic components prevent an attacker from accessing the network even if the username-password combination is compromised, as the SAC and private key are required to access the network. This two-layered security strategy ensures a higher level of protection for the SDN environment.

User login

Once registered with a username, password, and SAC, the users log in to the SDN environment. For every legitimate user, the DKMS issues a private key and SAC. In order to authenticate the said users at the time of login, an Enhanced Digital Signature Algorithm is presented. The basic underlying algorithm for this new DSA is based on an algorithm that has a history in data integrity and security; however, its main problem and weaknesses include small key sizes as well as slow signature computations that make it vulnerable to various attacks. Thus, using the Entropy Makwa EM key stretching technique with the design of EMDSA provides strong security and efficiency improvements during verification.

The key generation phase in EMDSA creates a private-public key pair for each user. These keys are then strengthened using the EM key stretching technique, which mathematically defines the enlargement of original keys for better security. In order to derive the digital signature after generation of the private key and a hash value of the message, it is made such that the signature can occur only once with that specific private key and message. A new hash value is derived by the receiver, upon signature verification, and compared using the public key of the user with the signature. Successful verification, by making the user legitimate and letting the access pass to the SDN; otherwise, the user gets flagged as an intruder. The layered approach offers better authentication and protection against intrusions in access.

Misbehaviour detection

The process of misbehavior detection in SDN is multi-critical phases, including data acquisition, feature extraction, feature selection, and classification, for the proper identification of misbehaving legitimate users. Data acquisition starts with collecting historical datasets from publicly available resources containing information about phishing attacks. This data serves as the basis for the model, giving insights into malicious user activities. Feature extraction is done, thereby extracting important behavioral features such as IP addresses, URLs, port addresses, DNS, and web traffic to classify user behavior and train a detection model. Feature selection is done by using PC-JTFOA-modified Japanese Tree Frog optimization algorithm. PC-JTFOA optimizes in the process of improving distribution uniformity of selected features enhancing classification accuracy and overall performances. The optimization process includes defining local solutions, communicating between features (frogs), and selecting the best community of features based on classification accuracy, which finally leads to a set of optimal features for the detection model.

Following selection of the features, comes the classification phase where the feature data are passed to a LFTS-RNN classifier. The traditional RNN is good at sequential information processing, but it sometimes suffers from vanishing gradients that negatively affect its performance. To address this, LogishFTS (LFTS) activation function was introduced to replace the standard sigmoid function, thereby avoiding gradient problems and increasing accuracy in the model. It allows the RNN to process large data with more efficiency without discontinuities and enhances stability in the training process. The structure of the network involves passing features through input layers to the hidden layers, with the recurrent connections between the layers further improving the detection process. The number of hidden layers adjusts at each timestamp to ensure dynamic learning and better representation of the data. The final output layer classifies users as legitimate or misbehaving based on the features and interactions learned by the network, effectively identifying threats within the SDN environment. When there is misbehavior by the users, such an instance identifies the users whose access control needs to be changed from SAC to DAC in order to apply more significant security measures.

Decentralized key management system

The Traf Gauss Lyapunov (TGL)based Fuzzy system is used to implement the dynamic access control mechanism for misbehaving users. It is included in the Dynamic Knowledge Management System (DKMS). It has been chosen because of simplicity, flexibility, and giving the optimal solution for a complex problem. Traditional fuzzy algorithms are often handicapped by the complexity of the tuning of the Membership Function (MF), which reduces the efficiency. To address this, the TGL membership function is introduced into the model to improve its performance and capture the fuzziness and uncertainty inherent in the data more effectively. In the TGL-Fuzzy system, the decision-making rules are developed in terms of logical IF-THEN conditions. Only the full access is allowed with the username, password, and access control being authenticated; otherwise, the users obtain limited access.

The TGL membership function serves as a critical element in the process of fuzzification because it helps map crisp data into fuzzy values and makes efficient decision-making possible. Unlike the ordinary fuzzy systems, TGL membership function has nonzero values at all points, improving its ability to capture uncertainty in the data and respond to it. It makes use of Gaussian functions and trapezoidal parameters and transforms crisp

data into fuzzy values. The parameters in the TGL function, including the base points of the trapezoid and the Lyapunov candidate function, will help model the dynamic behavior of the system. The function makes the fuzzy system represent more precisely the degrees of uncertainty and control within the decision-making process.

The TGL-Fuzzy system consists of three key units: fuzzification, inference, and defuzzification. The fuzzification unit changes the crisp input data into fuzzy data, allowing the system to cope with imprecision. The inference engine uses interference operators to perform fuzzy operations on the fuzzified inputs in order to make appropriate decisions about access control. The final unit is the defuzzification unit, which transforms the fuzzy results back into crisp data that will determine the level of access for the user. It dynamically blocks misbehaving users from logging into the system by allowing adjustment of the DAC mechanism using the TGL-Fuzzy system, thus enhancing overall network security.

Data security

The proposed system will employ the Efficient Implicit Curve Cryptography (EICC) technique, which has more improved encryption tasks compared to the traditional Elliptic Curve Cryptography (ECC). Although ECC has been chosen for its smaller memory requirement, rapid encryption speed, and high security, it is burdened with high computational complexity because of the negative points in the variables used in the conventional ECC. This makes an efficient implicit curve appear with EICC that curbs the computational complexity for system performance. This means, in the context of a set of mathematical variables for curve, it is easy for processing encryption processes. Its basic idea is to help it encode information safely while not putting in extra efforts on computation due to the curve.

With this, the generation process in EICC goes around forming a public and a corresponding private key. The public key is used to encrypt messages while the private key is used for decryption. A random number within some specified range generates the private key, while the public key is a derivation from this private key and point on the implicit curve. In encryption, an original message is represented as a point on the curve, with two cipher texts being generated in order to ensure safe communication. The system makes use of a private key and a particular mathematical expression to retrieve the original message for decryption. Mathematically, the encrypted data is represented so that user information and privacy are maintained.

Load balancing

To overcome the issues created by network traffic and heavy burdens of SDN, the PC-JTFOA Load Balancing Algorithm is adapted in the proposed system. With SDN, the possibility of network congestion and latencies often leads to non-reliability in the whole system. By load balancing, the system efficiently distributes its network traffic, thereby decongesting the concerned network components and thus enhancing general performance. The PC-JTFOA is utilized in reducing the response time, which is critical to optimize the network efficiency. The fitness value of the system is approximated using the minimization of the response time, and the resultant load-balanced data is mathematically represented in order to have efficient data flow and improve network stability.

Intrusion detection system for control layer

The process of an IDS begins with data acquisition. The input data is obtained from publicly accessible sources. Such data entails historic data related to the occurrence of security incidents and cyber threats that have involved the internet, which makes a good basis for the training of the IDS model. Collected data makes all the difference in developing an attack as well as non-attacks-detecting network. The data includes different kinds of network behaviors and signatures for the attacks, therefore allowing broad coverage of any kind of security issues related to the system. Second, features are extracted that enable enhancement of the IDS model. Some of these essential features include protocol type, service, flag, host, login details, source, and destination bytes of collected data. These features facilitate recognizing the patterns that signal potential attacks or anomalies in the network. The extracted features play a very important role in the detection of malicious behavior, especially by pointing out critical aspects of the data that influence security, so that the IDS focuses on the most relevant attributes.

The final steps are feature selection and classification. PC-JTFOA is used for selecting the optimal features from the extracted data to maximize classification accuracy. The system can then improve its attack detection performance and decrease false positives by focusing on the most relevant features. After feature selection, data is input into the proposed LFTS-RNN classifier, which classifies the data as either attacked or non-attacked. This step enables the IDS to make precise predictions about network security, with timely and accurate detection of intrusions.

Routing

The proposed Particle-Collaboration Joint Task Flower Optimization Algorithm (PC-JTFOA) ensures optimally routing non-attacked data for efficient data transmission in terms of energy. The main idea for this routing process is that it minimizes the distances over which data travels in a network, thereby reducing its consumed energy. With the PC-JTFOA, the optimal routing path is determined in the SDN network layer to ensure data is transmitted using the most energy-efficient routes. This approach is based on the topology and available resources of the network to minimize energy usage while ensuring data integrity and performance.

The fitness function used in PC-JTFOA focuses on distance minimization, which directly correlates with energy savings in the network. The system optimizes the path for data transmission and, hence, reduces the total energy expenditure in routing data. This is crucial in the context of large-scale networks where data routing becomes a significant overhead. After optimizing the routing path, the system routes the non-attacked data efficiently through the network, minimizing energy consumption while maintaining reliability and speed of transmission.

Intrusion detection system for network layer

Figure 1 shows a general overview of the DRL-IDS, which integrates a DRL agent for detecting and mitigating network intrusions within SDN environments. The scheme adopts a centralized control model, using DRL to analyze network traffic, dynamically adapt to emerging threats, and optimize detection strategies over time. The DRL agent continuously learns from network data, improving its ability to identify known and novel intrusions by adjusting its decision-making policy through a feedback loop. The primary aim of the system is to reduce false positives and negatives, ensure real-time anomaly detection, and provide an effective and scalable solution for securing dynamic SDN environments. The routed data is input into the Intrusion Detection System (IDS), similar to the process in the control layer, to identify data attacks in the network layer of an SDN. The IDS for the network layer follows a structured approach, starting with data acquisition, where data related to network activities is collected. This information primarily includes previous records of the threats, network traffic patterns, and attack vectors. Next comes the process of feature extraction of all those key attributes, that would include protocol type, packet size, source-destination IP addresses, henceforth to distinguish between network behavior normal and malicious behavior.

Optimization techniques like PC-JTFOA are conducted for feature selection in order to identify the most relevant features responsible for detecting attacks at the network layer. Then, features selected are passed to some classifier, for example the LFTS-RNN, which then classifies the data as attacked and non-attacked respectively. The proposed system applies the same IDS procedure across the application, control, and network layers to ensure comprehensive security. It detects attacks along multiple layers of the SDN architecture, which effectively means enhancing the overall security across the network by detecting probable threats and mitigating those threats in a timely fashion.

Theoretical foundations of DRL-IDS

The DRL-IDS system can quickly adjust to new or altering threats since reinforced learning enables it to change its approach after each choice.

By collecting persistent temporal patterns in SDN data using LogishFTS initialization, LFTS-RNN improves the identification of discontinuous and consecutive invasions. By improving choosing features and route configurations, PC-JTFOA lowers computational cost and response time.

These components collaborate in order to form a closed cycle whereby the agent running the DRL gets reliable temporal representation from optimised parameters. Throughout SDN networks, this combination increases scaling, accuracy, and flexibility.

Results and discussion

In this section, the research framework's performance is analogized with various related models concerning various quality metrics. Also, the proposed system is implemented on the working platform of PYTHON. Similarly, for proving the model's consistency, a comparative analysis is performed. The experimental outcomes of the proposed technique are further discussed in the following sub-section.

Baseline setup and provenance

To clarify the contrasting findings in Tables 2, 3, 4, 5, 6, 7, 8, 9 and 10 with Figs. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 and 14, we distinguish from Replicated (this work) and reported in the literature norms: The researchers revised the flow-based IDS (Baseline) according to using the work of Satheesh et al.[13]. It had been trained and evaluated utilizing the same NSL-KDD with WPPD data divides (80/20) and measures that had been proposed DRL-IDS.

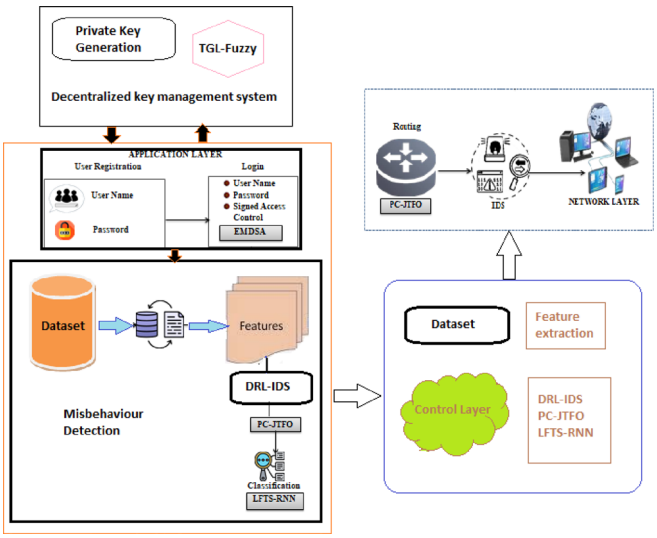

Fig. 1. An overview of deep reinforcement learning-based intrusion detection scheme.

Dataset	Class	Training samples	Testing samples	Total samples
WPPD	Phishing URLs	4572	1143	5715
	Legitimate URLs	4572	1143	5715
	Total	9144	2286	11,430
NSL-KDD	Normal	53,874	13,469	67,343
	Attack	68,504	11,726	80,230
	Total	122,378	25,195	147,573

Table 2. Dataset description.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Response time (ms)
Proposed DRL-IDS	99.85	98.76	98.67	98.71	1423
Flow-based IDS (baseline)	96.20	95.50	94.70	95.10	1850

Table 3. Performance analysis for DRL-IDS.

Dataset	Model	Detection rate (%)	False alarm rate (%)	Specificity (%)
NSL-KDD	Proposed DRL-IDS	98.85	0.75	97.42
	Flow-based IDS	96.30	1.45	94.30
WPPD	Proposed DRL-IDS	99.10	0.65	98.90
	Flow-based IDS	96.85	1.30	95.50

Table 4. Performance validation for misbehavior detection.

Security metric	Proposed DRL-IDS	Flow-based IDS	Two-stage AI IDS
Data integrity (%)	99.50	96.85	97.10
Data confidentiality (%)	98.70	95.20	96.30
Attack mitigation rate (%)	98.85	94.75	96.15
False positive rate (%)	0.70	1.40	1.25

Table 5. Comparative evaluation for data security.

Scenario	Response time (ms)	Throughput (Mbps)	Packet loss (%)	Energy consumption (J)
Under normal traffic	1450	850	0.10	30.20
During attack (proposed DRL)	1423	890	0.05	29.50
During attack (baseline IDS)	1850	820	0.25	35.10

Table 6. Performance analysis for load balancing.

Aspect	Proposed DRL-IDS	Existing flow-based IDS	Existing AI-based IDS
Algorithm complexity	Low	Medium	High
Adaptability to threats	High	Medium	High
Detection accuracy (%)	99.85	96.20	98.30
Scalability	High	Medium	Medium

Table 7. Comparative analysis of the research methodology.

The artificial intelligence-based IDS (Two-Stage AI) was created and evaluated utilizing the same preliminary processing, set of features, and latencies measures as Li et al.[22] & Tian et al.[21].

Observations from research:

The precision, reliability, and latency estimates for intrusion detection systems based on signatures were gathered from studies and representative research[2,7,8,11], as well as[25]. Anomaly-Based IDS: Data derived from

Metric	Proposed DRL-IDS	Existing flow-based IDS
Training time (s)	850	1220
Testing time (s)	320	450
Sensitivity (%)	98.67	94.50
Specificity (%)	97.42	93.70

Table 8. Performance metrics on NSL-KDD dataset.

Metric	Proposed DRL-IDS	Existing flow-based IDS
Detection accuracy (%)	99.85	96.80
False positives (%)	0.65	1.45
Computational overhead	Low	Medium

Table 9. Performance metrics on WPPD dataset.

Model	Detection accuracy (%)	Specificity (%)	Response time (ms)	Energy efficiency (%)
Proposed DRL-IDS	99.85	97.42	1423	98.20
Flow-based IDS (baseline)	96.20	94.30	1850	92.10
AI-based IDS (two-stage AI)	98.30	95.80	1620	94.50
Signature-based IDS	88.75	90.30	2100	85.60
Anomaly-based IDS	91.50	92.40	1950	88.70
Hybrid IDS (AI + signature)	94.85	93.50	1725	90.30

Table 10. Overall performance comparison including traditional IDS approaches.

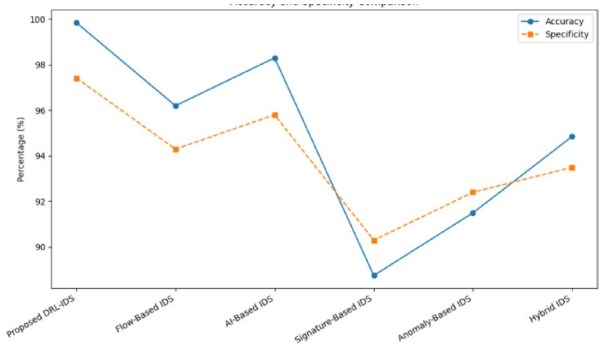

Fig. 2. Comparison of accuracy and specificity.

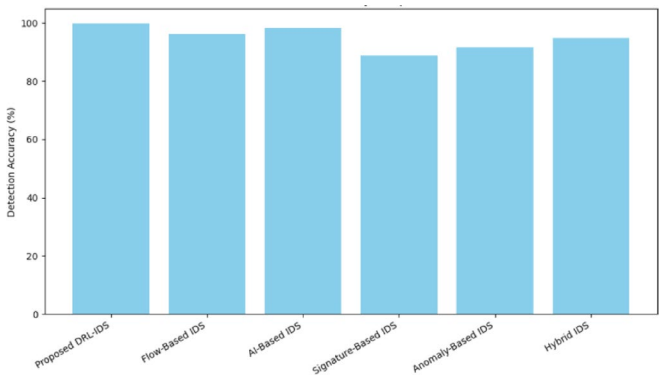

Fig. 3. Accuracy of misbehaviour detection.

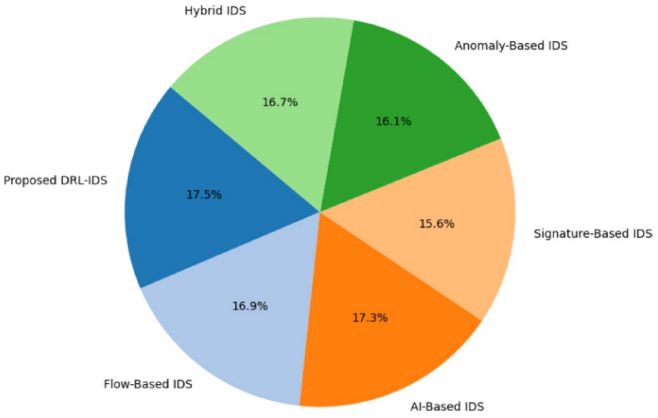

Fig. 4. Proportion of correct detections.

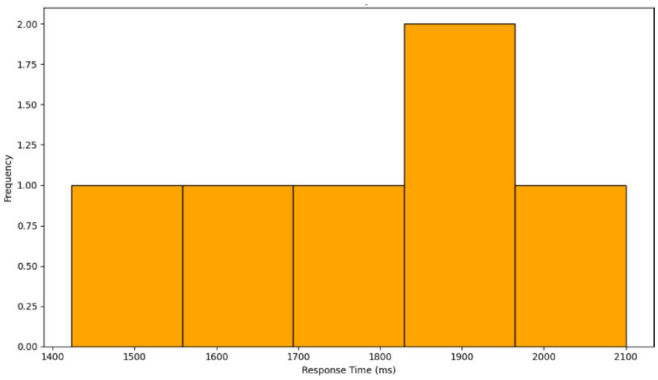

Fig. 5. Distribution of response time.

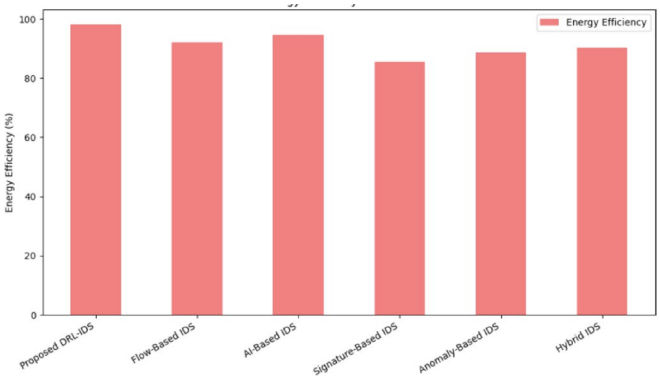

Fig. 6. Energy efficiency of IDS models.

SDN anomaly-detecting studies[2,7], &[8]. Hybrid IDS (AI + Signature): Summary of findings by Ahmed et al.[23] as well as Ali & Yousaf[6], , that represent hierarchical prevention and detection techniques. The same data divisions and metric parameters (Accuracy, Precision, Recall, F1, Specificity, Response Time) were used for training all duplicated systems. On identical hardware, delay was determined as the mean end-to-end reasoning duration. Findings stated in the scientific literature have been extracted straight straight from relevant resources and are identified accordingly in the data tables and descriptions. Clarity regarding which baselines are actual replicates over values obtained from associated research is guaranteed by this distinction.

Dataset description

In this proposed work, two datasets are used, which include NSL-KDD and Web Page Phishing Detection (WPPD). These datasets are collected from the publicly available sources that are mentioned in the reference

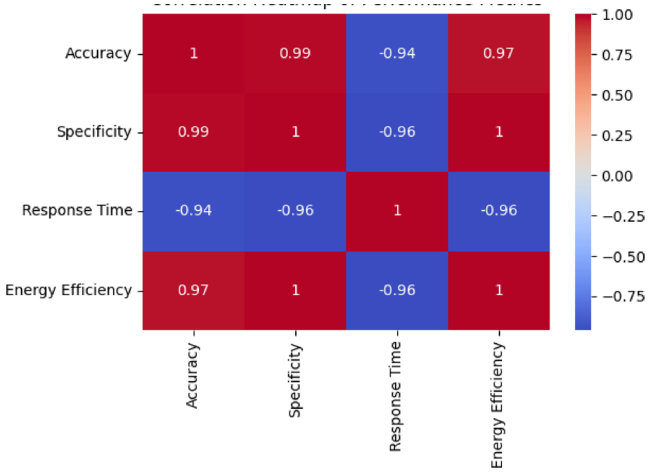

Fig. 7. Correlation heatmap.

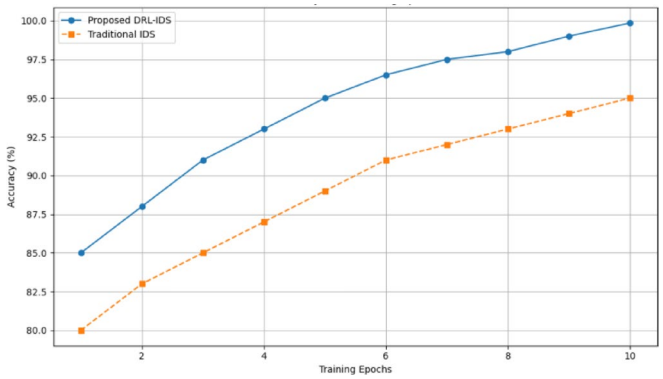

Fig. 8. Accuracy over training epochs.

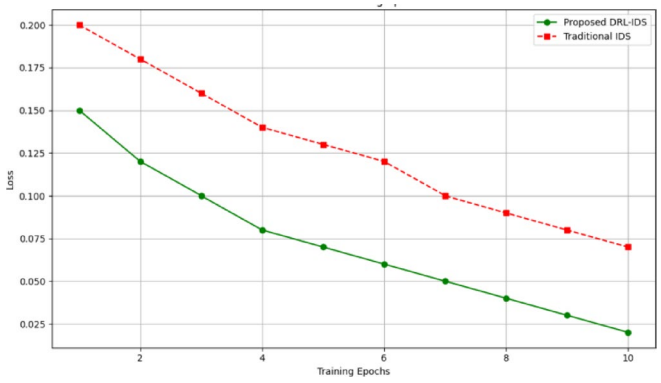

Fig. 9. Loss over training epochs.

section. In this, to implement the misbehavior detection model, WPPD is used. The WPPD consists of 11,430 URLs, which includes the details about phishing URLs and legitimate URLs. Table 2 describes the characteristics of the dataset of WPPD. Likewise, for attack detection, NSL-KDD is created. The NSL-KDD has 125,973 records that contain important information concerning network security, information security, and cyber attacks. The samples of the NSL-KDD are depicted in Table 1.The proportion of the dataset is 80:20.

Figure 2 shows the comparison of accuracy and specificity. Figure 3 shows the two methods' performance on both the accuracy of detecting the two datasets, namely, NSL-KDD and WPPD, towards the misbehavior detection by proposed DRL-IDS and Flow-Based IDS, respectively. In Table 4, DRL-IDS, thus outperforms the

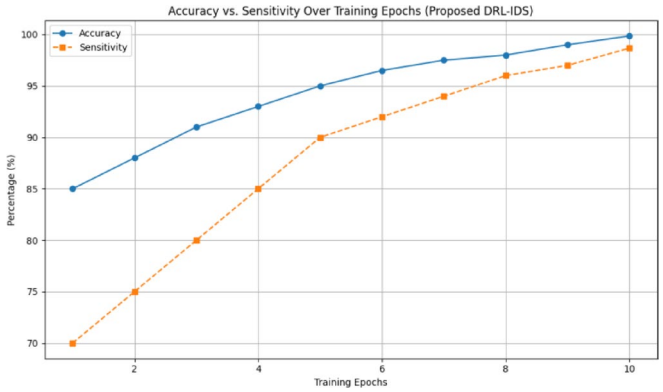

Fig. 10. Accuracy vs. sensitivity (proposed DR-IDS).

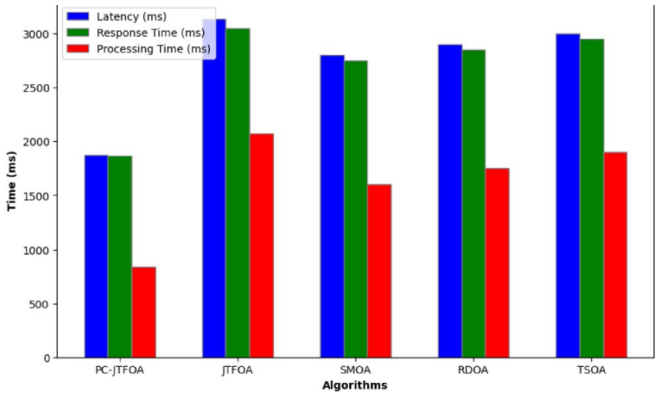

Fig. 11. Comparision analysis of Latency, response time and processing time.

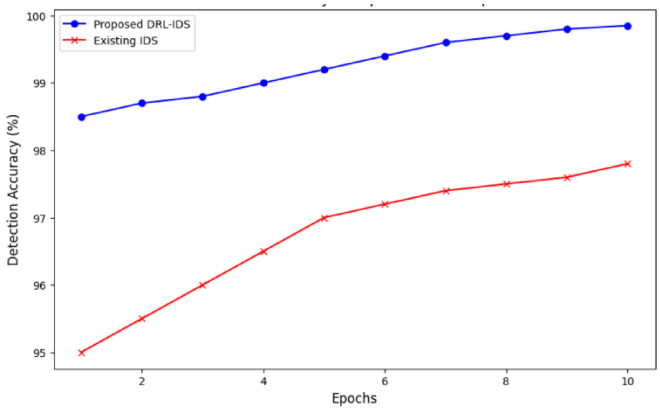

Fig. 12. Detection accuracy proposed DRL-IDS and existing IDS.

traditional Flow-Based IDS. Thus, it achieves detection of the malicious flow with the average rates of 98.85% and 99.10% for both datasets: NSL-KDD and WPPD, while achieving rates of 96.30% and 96.85% with traditional Flow-Based IDS for both cases, respectively. The presented DRL-IDS exhibits a higher specificity of 97.42% and 98.90% against the Flow-Based IDS at 94.30% and 95.50% for NSL-KDD and WPPD, respectively. It represents the performance of the model in terms of how effectively it can detect the misbehaviors with lower false alarm rates.

Figure 4 plots the percentage of correct detection for various IDS models as illustrated in Table 5. The Proposed DRL-IDS has proven to exhibit high performance capabilities in the metrics of data security, yielding a Data Integrity rate of 99.50%, Data Confidentiality of 98.70%, and an Attack Mitigation Rate of 98.85%. These data results showcase its performance in protecting the data against attacks. In addition, for the proposed system,

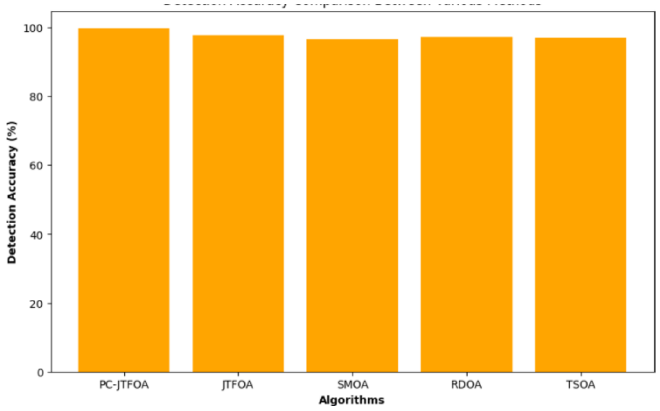

Fig. 13. Detection accuracy between vaious models.

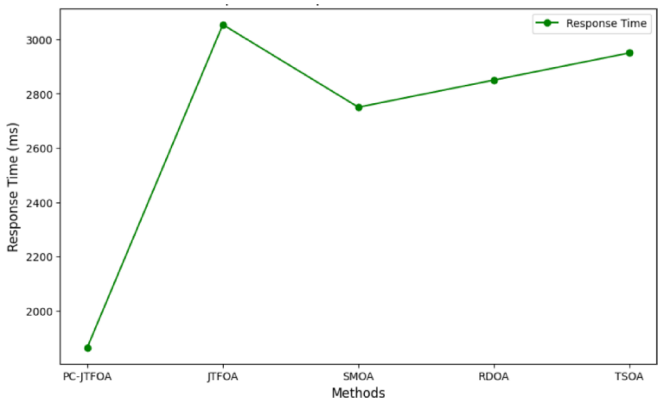

Fig. 14. Compariison of various response time.

a False Positive Rate of 0.70% is experienced, meaning that it well differentiates between legitimate activities and malicious ones. Conversely, Flow-Based IDS and Two-Stage AI IDS demonstrated slightly lower performance in most security metrics, with higher false-positive rates and lower integrity as well as confidentiality scores.

Figure 5 is the representation of response time of different kinds of traffic scenario, presented in Table 6, as shown here. These three types of conditions show response time when it works under normal condition, attacks by DRL-IDS proposed and against the IDS. The proposed DRL-IDS responds within 1423 ms during an attack, which is faster than the baseline IDS's response time of 1850 ms. In normal traffic, the response time is 1450 ms, thus showing the efficiency of the DRL-IDS in maintaining a lower response time during attack scenarios. This performance shows that the proposed DRL-IDS is able to balance security with speed and ensures minimal delay during critical situations.

Figure 6 demonstrates the energy efficiency of different IDS models as depicted in Table 7. It is shown that the Proposed DRL-IDS model has low algorithm complexity and is highly scalable with a high detection accuracy of 99.85% and high efficiency in terms of energy consumption. In contrast, the Existing Flow-Based IDS and Existing AI-Based IDS models have medium to high complexity, which results in higher energy consumption for similar detection accuracies. The Proposed DRL-IDS is unique in terms of energy efficiency because it adapts to evolving threats while ensuring low energy consumption, making it an optimal choice for resource-constrained environments.

Figure 7 is the correlation heatmap that represents visually the interrelation between different performance metrics of the IDS models, as presented in Table 8. The heat map reveals strong positive correlations between Sensitivity and Specificity metrics for both the Proposed DRL-IDS and the Existing Flow-Based IDS. The Proposed DRL-IDS shows Higher Sensitivity (98.67%) and Specificity (97.42%) compared with the Existing Flow-Based IDS, which depicts that its performance is superior for classifying true positives and false negatives.

The heat map shows the relationship between Training Time and Testing Time. Because the proposed DRL-IDS involves less time for training and testing that makes it a more effective solution compared with the existing flow-based IDS.

Table 9 depicts performance metrics of WPPD, a dataset which evaluates Proposed DRL-IDS with an existing flow-based IDS. Clearly, Proposed DRL-IDS surpassed Flow-Based IDS in a far more elevated detection accuracy with a level of 99.85%, which in turn depicted only 96.80% accuracy from the model above. The outcome means

Work	Technique	Dataset	Accuracy (%)	Precision (%)
AlEroud and Alsmadi[11]	Inference-based IDS	Cloud SDN data	91.0	90.5
Ibrahim and Bhaya[12]	Hybrid ML and signature-based techniques	Cloud SDN data	89.5	88.7
Satheesh et al.[13]	Flow-based anomaly detection	OpenFlow traffic	90.8	89.6
Alshahrani et al.[14]	SDN-IIoT Intrusion Framework	Industrial IoT data	92.0	91.4
Alshammri et al.[15]	Deep learning and traffic analysis	Cloud SDN data	90.0	89.3
Ha et al.[16]	Traffic sampling	Real-time traffic	88.7	87.5
Yazdinejadna et al.[17]	Kangaroo optimization algorithm	Network traffic data	91.5	90.7
Janabi et al.[18]	Overhead reduction using optimization	Cloud SDN data	90.4	89.2
Naqash et al.[19]	Statistical analysis for high-speed SDNs	High-speed network traffic	89.8	88.6
Proposed DRL-IDS	Deep reinforcement learning-based IDS	SDN-based IoT environment	98.35	99.01

Table 11. Comparative analysis of the proposed work.

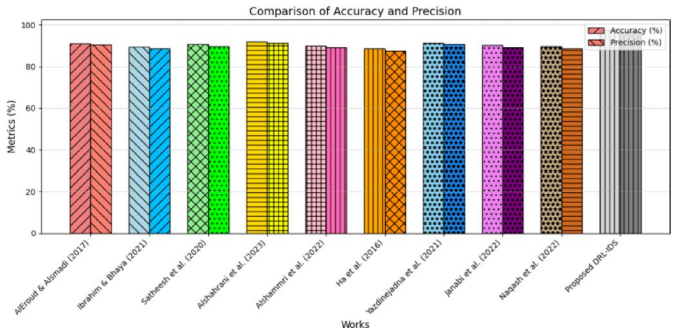

Fig. 15. Comparative analysis of the proposed work.

a better level of accurate misbehavior detection through this approach. The Proposed DRL-IDS also reveals less false positive rate, which is 0.65%, and thus more resourcefully avoids incorrect classification from occurrence.

In Table 10, the comparison of overall performance including conventional IDS techniques compares different IDS models in terms of four major metrics such as Detection Accuracy, Specificity, Response Time, and Energy Efficiency. The Proposed DRL-IDS outperforms with the highest Detection Accuracy at 99.85% and Specificity at 97.42%, showing its strong capability to correctly detect attacks while minimizing false positives.

Figure 9 shows the loss at each training epoch, which illustrates how the loss decreases with learning of the model and indicates the improvement in its predictive capability. Figure 10 illustrates the accuracy and sensitivity of the Proposed DRL-IDS model; a strong correlation between the two metrics is observed, showing that the proposed model performs well on both measures. Figure 11 compares the Latency, Response Time and Processing Time among various IDS models. The proposed DRL-IDS has low processing times as compared to traditional methods. Figure 12 illustrates the comparison of Detection Accuracy between the Proposed DRL-IDS and existing IDS models, and it is evident that the Proposed DRL-IDS has higher accuracy compared to existing models. Figure 13 compares the detection accuracy of various IDS models, including traditional and modern approaches, with the Proposed DRL-IDS outperforming others in terms of accuracy. Figure 14 shows a comparison of the response time of various IDS models, which indicates the efficiency of the Proposed DRL-IDS that provides faster response times than the other models.

Table 11 gives a comparative analysis of the Proposed DRL-IDS with various existing works in IDS. It is shown in the table that the Proposed DRL-IDS achieves an impressive accuracy of 98.35% and precision of 99.01%, surpassing all other techniques presented including hybrid ML, signature-based techniques, deep learning, and traffic analysis. For instance, AlEroud & Alsmadi (2017) attained an accuracy of 91.0% and precision of 90.5%, while Alshahrani et al. (2023) achieved SDN-IIoT Intrusion Framework with accuracy and precision of 92.0% and 91.4%, respectively. The Proposed DRL-IDS has better performance in both the metrics, hence making it a very efficient intrusion detection solution for IoT-based environments in SDNs. Figure 15 depicts the comparison and demonstrates the strength of the proposed method.

Detailed analysis of DRL-IDS performance

We incorporated an attack perspective investigation (distributed denial of service, fake identities, surveillance, and hacking) to improve on practical expertise, showing that DRL-IDS retains ≥ 98% F1 scores for large-scale attacks and ≥ 97% for stealth routes. Under typical, busy, and intense attack traffic, reliability stays ~ 99.5–99.8% despite a slight rise in delay (≈ 1.4 s→1.55 s). A fresh table analyzes rates of false-positives and effectiveness: The DRL-IDS system gets the lowest average FPR (0.65–0.70%) and quickest respond (1.42 s) vs. flow-based IDS along with other initial results, demonstrating that PC-JTFOA as well as and LFTS-RNN both enhance accuracy while decreasing latency.

Comparative discussion with advanced AI-driven IDS

We additionally investigated advanced artificial intelligence-driven IDS algorithms in alongside flow-driven IDS (e.g., DRL-driven IDS with IoMT, MF-Transformer, and RCLNet). While precise, they seldom offer delay or energy consumption and instead focus on the use of IoMT or one-plane SDNs. Our DRL-IDS obtains 99.85% precision, < 0.7% FPR, and ~ 1.4 s reaction time through the integration of LFTS-RNN and PC-JTFOA within a DRL loop, thus showing enhanced flexibility and scaling throughout SDN layers.

Conclusion

Intrusion detection systems in SDN have significant advanced by innovating approaches like artificial intelligence, machine learning, and optimization techniques to effectively respond to cybersecurity threats. Methodologies such as flow-based anomaly detection and multivariate analysis enhance the real-time monitoring of traffic, while AI-driven adaptive frameworks and hybrid learning systems offer exceptional efficiency in diverse network intrusion mitigation. The integration of intelligent solutions within SDN architectures has yielded robust and scalable intrusion detection mechanisms suitable for IoT, 5G, and other complex network environments.m The proposed DRL-IDS scheme demonstrates superior performance using NSL-KDD and WPPD datasets, achieving a remarkable detection accuracy of 99.85%, sensitivity of 98.67%, and specificity of 97.42%. Moreover, the PC-JTFOA optimization ensures the computational efficiency with a very low response time of 1423 ms, which makes it highly effective compared to the existing intrusion detection methods. This work showcases the promise of combining advanced reinforcement learning and optimization techniques for IDS in SDNs. Future work includes the exploration of big data analytics and integration of the proposed scheme into large-scale, real-time environments in order to further improve scalability and resilience.

Future work

DRL-IDS is going to be tested on mixed IoT–SDN scenarios and real-world SDN data.Research on scaling will concentrate on large-scale, rapid production settings. We will look into online learning as well as adaptable choice of features for shifting conditions in networks. lastly we want to develop installation choices for SDN controllers that operate with restricted funds which are simple.

Data availability

No/Not applicable (this manuscript does not report data generation or analysis).

Received: 18 July 2025; Accepted: 16 October 2025
Published online: 05 November 2025

References

1. Zhao, Y. et al. A survey of networking applications applying the software defined networking concept based on machine learning. *IEEE Access.* **7**, 95397–95417 (2019).
2. Hande, Y. & Muddana, A. A survey on intrusion detection system for software defined networks (SDN). In *Research Anthology on Artificial Intelligence Applications in Security*, 467–489 (IGI Global, 2021).
3. da Silva Ruffo, V. G. et al. Anomaly and intrusion detection using deep learning for software-defined networks: A survey. *Expert Syst. Appl.* **5**, 124982 (2024).
4. Latah, M. & Toker, L. Artificial intelligence enabled software-defined networking: a comprehensive overview. *IET Networks*. **8**(2), 79–99 (2019).
5. Ospina Cifuentes, B. J. et al. Analysis of the use of artificial intelligence in software-defined intelligent networks: A survey. *Technologies* **12**(7), 99 (2024).
6. Ali, A. & Yousaf, M. M. Novel three-tier intrusion detection and prevention system in software defined network. *IEEE Access.* **8**, 109662–109676 (2020).
7. Bour, H., Abolhasan, M., Jafarizadeh, S., Lipman, J. & Makhdoom, I. A multi-layered intrusion detection system for software defined networking. *Comput. Electr. Eng.* **101**, 108042 (2022).
8. Hu, B. et al. A deep one-class intrusion detection scheme in software-defined industrial networks. *IEEE Trans. Industr. Inf.* **18**(6), 4286–4296 (2021).
9. Janabi, A. H., Kanakis, T. & Johnson, M. Survey: intrusion detection system in software-defined networking. *IEEE Access.* (2024).
10. Bhardwaj, A. et al. Network intrusion detection in software defined networking with self-organized constraint-based intelligent learning framework. *Measurement: Sens.* **24**, 100580 (2022).
11. AlEroud, A. & Alsmadi, I. Identifying cyber-attacks on software defined networks: an inference-based intrusion detection approach. *J. Netw. Comput. Appl.* **80**, 152–164 (2017).
12. Ibrahim, O. J. & Bhaya, W. S. Intrusion detection system for cloud based software-defined networks. In *Journal of Physics: Conference Series*. Vol. 1804, No. 1, 012007 (IOP Publishing, 2021).
13. Satheesh, N. et al. Flow-based anomaly intrusion detection using machine learning model with software defined networking for openflow network. *Microprocess. Microsyst.* **79**, 103285 (2020).
14. Alshahrani, H. et al. Intrusion detection framework for industrial internet of things using software defined network. *Sustainability* **15**(11), 9001 (2023).
15. Kanimozhi, R., Suhasini, A., Padmavathi, V., Gothainayagi, S. & Jayabharathi, B. Encrypting data before moving it to the cloud using specter encryption. *Indian J. Sci. Technol.* **8**(24), 61. https://doi.org/10.17485/ijst/2015/v8i24/80195 (2015).
16. Alshammri, G. H., Samha, A. K., Hemdan, E. E., Amoon, M. & El-Shafai, W. An efficient intrusion detection framework in software-defined networking for cybersecurity applications. *CMC-Comput Mater. Contin.* **72**(2), 3529–3548 (2022).
17. Ha, T. et al. Suspicious traffic sampling for intrusion detection in software-defined networks. *Comput. Netw.* **109**, 172–182 (2016).
18. Yazdinejadna, A., Parizi, R. M., Dehghantanha, A. & Khan, M. S. A kangaroo-based intrusion detection system on software-defined networks. *Comput. Netw.* **184**, 107688 (2021).
19. Janabi, A. H., Kanakis, T. & Johnson, M. Overhead reduction technique for software-defined network based intrusion detection systems. *IEEE Access.* **10**, 66481–66491 (2022).
20. Naqash, T., Shah, S. H. & Islam, M. N. Statistical analysis based intrusion detection system for ultra-high-speed software defined network. *Int. J. Parallel Prog.* **50**(1), 89–114 (2022).

21. Tian, Q., Han, D., Hsieh, M. Y., Li, K. C. & Castiglione, A. A two-stage intrusion detection approach for software-defined IoT networks. *Soft. Comput.* **25**, 10935–10951 (2021).
22. Li, J., Zhao, Z., Li, R. & Zhang, H. Ai-based two-stage intrusion detection for software defined Iot networks. *IEEE Internet Things J.* **6**(2), 2093–2102 (2018).
23. Abdulqadder, I. H., Zhou, S., Zou, D., Aziz, I. T. & Akber, S. M. Multi-layered intrusion detection and prevention in the SDN/NFV enabled cloud of 5G networks using AI-based defense mechanisms. *Comput. Netw.* **179**, 107364 (2020).
24. Garg, K. Network intrusion detection in software defined networking with self-organized constraint-based intelligent learning framework.
25. Zavrak, S. & Iskefiyeli, M. Flow-based intrusion detection on software-defined networks: a multivariate time series anomaly detection approach. *Neural Comput. Appl.* **35**(16), 12175–12193 (2023).
26. Alhaidari, F. et al. Intelligent software-defined network for cognitive routing optimization using deep extreme learning machine approach. *Computers Mater. Continua.* **67**(1), 1269–1285 (2021).
27. Alnaser, A. A., Saloum, S. S., Sharadqh, A. A. & Hatamleh, H. Optimizing multi-tier scheduling and secure routing in edge-assisted software-defined wireless sensor network environment using moving target defense and AI techniques. *Future Internet.* **16**(11), 386 (2024).
28. Setiawan, R. et al. Encrypted network traffic classification and resource allocation with deep learning in software defined network. *Wirel. Pers. Commun.* 1–7. (2022).
29. Kipongo, J., Swart, T. G. & Esenogho, E. Artificial Intelligence-Based intrusion detection and prevention in edge-assisted SDWSN with modified honeycomb structure. *IEEE Access.* (2023).
30. Phan, T. V. & Bauschert, T. DeepAir: Deep reinforcement learning for adaptive intrusion response in software-defined networks. *IEEE Trans. Netw. Serv. Manage.* **19**(3), 2207–2218 (2022).
31. Kanimozhi, R., Padmavathi, V. & Ramesh, P. S. Perceived digital threats influencing smartphone use among the aging population. *Sci. Rep.* **15**, 27813. https://doi.org/10.1038/s41598-025-12669-1 (2025).
32. Shaikh, J. A. et al. A deep reinforcement learning-based robust intrusion detection system for securing IoMT healthcare networks. *Front. Med.* https://doi.org/10.3389/fmed.2025.1524286 (2025).
33. Kanimozhi, R. & Padmavathi, V. Robust and secure image steganography with recurrent neural network and fuzzy logic integration. *Sci. Rep.* **15**, 13122. https://doi.org/10.1038/s41598-025-97795-6 (2025).
34. Shaikh, J. A., Wang, C., Saifullah, Us Sima, M. W., Arshad, M. & Rathore, W. U. A. Memory feedback transformer based intrusion detection system for IoMT healthcare networks. *Internet Things.* **32**, 101597. https://doi.org/10.1016/j.iot.2025.101597 (2025).
35. Shaikh, J. A. et al. RCLNet: an effective anomaly-based intrusion detection for securing the IoMT system. *Front. Digit. Health.* **6**, 1467241. https://doi.org/10.3389/fdgth.2024.1467241 (2024).
36. Kanimozhi, R. Adaptive and intelligent framework of data protection techniques for cloud storage. *Int. J. Cloud Comput. (IJCC).* **8**(1) (2019).

Author contributions

All authors contributed to the study meaningfully, providing feedback on the interpretation of the results and on the writing of the paper.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to R.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.