



Intrusion Detection in Software-Defined Networks: Leveraging Deep Reinforcement Learning with Graph Convolutional Networks for Resilient Infrastructure

Fuqdan A. Al-Ibraheemi¹, Firas Hazzaa^{2,3}, Mohanad Sameer Jabbar^{4,*}, Jamal Fadhil Tawfeq⁵, Ravi Sekhar⁶, Pritesh Shah⁷, Sushma Parihar⁸

¹College of Dentistry, University of Al-Ameed, Iraq

²Ministry of Higher Education and Scientific Research, Baghdad, Iraq

³Visiting Fellow, School of Engineering and Build Environment, Anglia Ruskin University, Chelmsford, UK

⁴Medical Instruments techniques Engineering Department, Technical College of Engineering, Al-Bayan University, Baghdad, Iraq

⁵Department of Medical Instrumentation Technical Engineering, Medical Technical College, Al-Farahidi University, Baghdad, Iraq

^{6,7,8}Symbiosis Institute of Technology (SIT) Pune Campus, Symbiosis International (Deemed University) (SIU), Pune, 412115, Maharashtra, India

Emails: fuqdanal_ibrahimi@alameed.edu.iq; fi7600@gmail.com; mohanad.s@albayan.edu.iq; jamaltawfeq55@gmail.com; ravi.sekhar@sitpune.edu.in; pritesh.shah@sitpune.edu.in; sushmap@sitpune.edu.in

Abstract

Protecting Software-Defined Networking (SDN) environments from intrusions and unauthorized access requires a high level of security. Security issues have arisen because of the widespread use of Software-Defined Networking (SDN), especially regarding intrusions that may cause disruptions to network operations by gaining unauthorized access. Intrusion is a danger to an SDN architecture's security, efficacy, and dependability because it involves manipulation or disruption. To improve SDN security through Intrusion Detection Systems (IDS), this study suggests a novel approach that makes use of Graph Convolutional Networks (GCN) and Deep Reinforcement Learning (DRL). The approach, which makes use of the NSL-KDD dataset, shows enhanced performance measures for intrusion detection, such as accuracy (93.8%), recall (93%), F1-score (92%), and precision (94.2%). This work establishes the groundwork for resilient infrastructure against threats and advances the security posture of SDN environments.

Keywords: Computer Science; Network; Virtual Learning Environment (VLE); Fuzzy-based Convolutional Neural Network (FCNN); Software-Defined Networking (SDN).

1. Introduction

With the rise of Software-Defined Networks (SDN), modern networking has evolved into something much more sophisticated. SDN makes network architectures more flexible and faster than ever before. The management layer can be based on data to achieve flexible, combined network management techniques. One major issue associated with the rise of inventions is the penetration of SDN, which poses new problems. Through manipulation or interruption, the term intrusion is used to describe unauthorized and destructive access to a network. An architecture that behaves in such a way can have substantial effects on security, effectiveness, and reliability. SDN was forced to undergo a fundamental update when intrusion concerns emerged. Networks normally have their control mechanisms dispersed among several components to prevent attackers from completely destroying the entire network. SDN, however, may be vulnerable to cyber attacks because it centralized control in software-based controllers. Having access to the central controller from an unauthorized source gives the individual considerable power over the network, potentially leading to severe and far-reaching consequences. To prevent loss of data privacy, security, and accessibility as a result of these incursions, we must resolve this issue in a complete manner

[4]. Intrusions into SDN are complex and involve many factors. Among the driving factors is the intent to gain from vulnerabilities. Cybercriminals may hack into an SDN environment to steal confidential information, carry out fraudulent operations, or use ransomware to attack. It may be a goal of state-sponsored organizations to spy on or disrupt essential infrastructure. Hackers and script monkeys may enter SDN with the objective of disrupting or promoting their ideologies. Due to the complex root cause of these attacks, SDNs need effective safety measures. The SDN environment has many intruders, each of which can pose its challenges. Control rules must be approved for SDN controller access because an attacker may alter network protocols, traffic routing techniques, or control rules without it [6]. Hacking and service interruptions can result from manipulating data. Furthermore, secret pathways can be extracted without permission. Additionally, SDN's design can be exploited to launch DDoS attacks. The attacks may lock network resources, making them inaccessible. In addition to introducing new vulnerabilities, SDN implementation increases attack surfaces. In light of these dangers, we must continue to take steps to reduce them. Multifaceted approaches are needed to secure SDNs. The security of SDN controllers and network components is dependent upon authentication and authorization systems. Keeping patches up-to-date, along with two-factor authentication, prevents vulnerabilities. Additionally, separating the networks prevents unauthorized individuals from harming the networks horizontally through the separation of the networks [8].

SDNs can track and identify intrusions, which is important for identifying and resolving intrusions. To detect atypical network behavior or variations in established network operations, controllers and networks can provide data related to network operations. SDN architectures must be protected against intrusions via real-time threat notifications and automated responses. Data exchange and collaboration are essential for the development of a successful defense system. The threat intelligence that organizations and industries share must evolve as threats do. The dissemination of standard procedures is another key component of SDN security. The combination of an Intrusion Detection System (IDS) with Deep Reinforcement Learning algorithms can enhance the security of Software-Defined Networking (SDN). Section 2- Related works, Section 3- Methodology, Section 4- Results, and Section 5- Conclusions are organized in the paper.

2. Related Works

Study findings indicate DNNs are deep learning models able to identify and categorize unplanned and unexpected internet attacks based on an adaptable identity. Since network behavior and attacks continue to evolve, it became necessary to evaluate datasets generated by static and evolving approaches. Based on a comprehensive analysis of the results, deep neural networks (DNN) outperformed traditional machine learning (ML) classifiers. Study results suggest that deep learning (DL) can be used to develop SDN intrusion detection systems. As a result of an experimental study, DL proved suitable for detecting abnormalities in an SDN environment. Based on tests, Deep IDS does not interfere with Open Flow controllers. Thus, Deep IDS became suitable. Examines [12] implementing a priority-based model using SDN for controlling packet transmissions. Through the use of virtual networks, the proposed model ensures the effective allocation of bandwidth and facilitates reallocation. Using comparative analysis, they assessed existing routing methods and analyzed network resilience to determine which were most effective. In their study, the authors developed methods for identifying and detecting security issues in the structure of SDN.

Several ML algorithms were tested in the study [13], including the Nave Bayes algorithm and the K-means clustering algorithm. By combining two algorithms, they identified and recognized malicious attacks. As a result, the goal was achieved in the end. In addition to the fact that machine learning may improve detection rates and reduce false alarm rates as well as reduce computing and communication expenses, ML approaches may also improve efficiency and reduce costs. In the study [14], it can be found that data can be transmitted, processed, and exchanged through a separate network using a decentralized approach. System overload and inefficient performance were addressed using this approach. With the help of a feature selection technique, the researchers reduce the amount of data sent across networks and the number of features extracted; this reduces the amount of data sent across networks. Since Naive Bayes classifiers are efficient in classifying flows, they were used for classifying the data collected. In their study [15], researchers found two ways to detect anomalies in networks based on flows. The gated recurrent unit-long short-term memory (GRU-LSTM) and random forest (RF) models use machine learning techniques to detect network interruptions using SDN. In this process, flow-based anomaly detection is performed using the GRU-LSTM model since it offers a higher level of accuracy and efficiency than machine learning.

According to the study [16], a statistical analysis-based intrusion detection system (SABIDS) was developed using machine learning. The RYU controller is integrated with the SABIDS system to block IP addresses that cause traffic. Several network scenarios can be addressed by their solution. An anomaly IDS based on SDN detects cyber-attacks as well as unusual activity in IoT networks. ToN-IoT data was analyzed and evaluated using machine learning techniques. The use of machine learning-based IDS could provide reliable security protection, according to research. An article [18] proposes a new method of improving SDN intrusion detection through hybridization

at several levels. Using K-nearest neighbors (KNNs) and extreme learning machines (ELMs), we built the system based on a combination of extreme learning machines and hierarchical extreme learning machines. A measurement accuracy of 84.29 percent was found to be the best in the experiments. In the study, instantaneous identification was achieved by using machine learning [19] and intrusion prevention systems. An evaluation of the system in relation to 5G networks was conducted using convolutional neural networks (CNNs). Software-defined systems were included in the automated IDS. In a study [20], researchers investigated challenge-based collaborative intrusion detection networks (CIDNs). CIDNs based on challenges demonstrated efficient performance in SDN environments, even when they were fighting against new attacks and betrayals. As a result of CIDN's approach, it was able to detect and recognize betrayal attacks [21-23].

3. Methodology

Random forest-recursive feature elimination algorithms are used after preprocessing the data with min-max normalization to select the features. Deep Reinforcement Learning with Graph Convolutional Networks (DRL+GCN) contributes to enhanced security for SDN. Diagram 1 shows how the DRL+GCN system is structured.

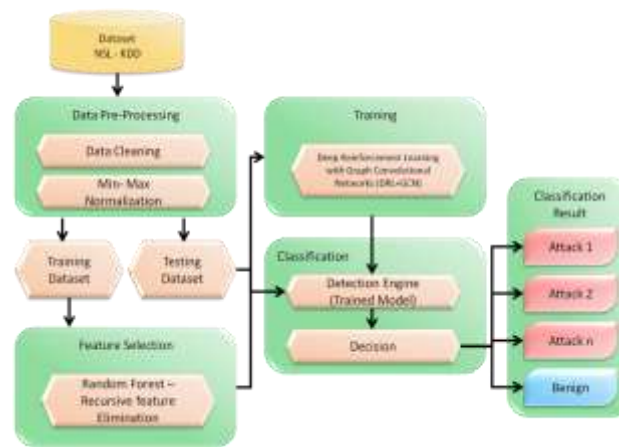


Figure 1: DRL+GCN system's architecture diagram

3.1 Dataset

A dataset called NSL-KDD [18] was used in this study. When compared to the NSL-KDD dataset, the KDD Cup 99 dataset has a significant number of unnecessary records. During the collection, 39 distinct assaults were identified, and each one was categorized into four different categories: R2L, Probe, DoS, and U2R. Several of these assaults have been added to the testing set. A table showing the composition of previously identified and newly discovered assaults can be found below.

Table 1: Distribution of KDD - Test set

Dataset	U2R	DoS	Probe	R2L
Known dataset	37	5741	1106	2199
	18.50	76.98	45.68	79.85
New dataset	163	1717	1315	555
	81.50	23.02	54.32	20.15

3.2 Pre-processing

3.2.1 Data cleaning

As part of data cleaning, pre-processing and cleaning information about network traffic can help detect anomalies more accurately. Organizing data forms, correcting missing values, and removing noise is done with this process. It is crucial for the performance of the intrusion detection model to be cleaned in an SDN environment as it

changes. The enhancement of network security and dependability is achieved by identifying and responding to network threats.

3.2.2 Min-max normalization

As a result of linear modifications applied to the original data, min-max normalization ensures equivalent value evaluations of the original and modified data. A solution to this problem can be found by using equation (1).

$$K_{new} = \frac{k - \min(k)}{\max(k) - \min(k)} \quad (1)$$

K_{new} = Normalization produces a new value

k = This is the old value

$\max(k)$ = A dataset's maximum value

$\min(k)$ = Values in the dataset with the lowest value

3.3 Selecting Features

3.3.1 Random Forest Recursive Feature Elimination (RF-RFE)

The Recursive Feature Elimination method is a strategy that utilizes a greedy algorithm to rank attributes. For achieving the most prominent characteristics, the Recursive Feature Elimination (RFE) technique is employed to remove the least important attributes from the complete attribute set, one at a time, during iteration. Recursion is an essential requirement due to the chance for major changes in the characteristic attributes of certain processes when assessing across a different selection of attributes through step-wise removal. This applies to attributes that are connected together. The last attribute set is created based on the sequence through which the attributes are eliminated. The attribute selection technique involves the extraction of the initial n characteristics from this ranking. The Random Forest algorithm is a predictive methodology that employs a collection of models. A composite predictor is created through the integration of numerous individual prediction trees. When making a final prediction for a dataset, an absolute rule is applied to the various forecasting options available. To encourage the generation of dissimilar and varied insights, each tree is constructed by utilizing a portion of the dataset from the preparation set. The algorithm integrates random contingency into the process of searching for optimal splitting with the objective of maximizing the differences across the trees. The relationship between the proportion of significance of distinctive characteristics and the recursive attribute removal procedure can be observed in the RF-RFE approach. The RF-RFE methodology relies on the concept of constructing a "random forest model" and selecting the most suitable or least effective operational attribute. Remove the particular attribute and proceed to reiterate the procedure, continuing with the other attributes. The procedure is executed until the characteristics present in the dataset are employed. The attributes are prioritized based on the sequence of their removal.

3.4 Graph Convolutional Networks with Deep Reinforcement Learning (DRL+GCN)

DRL is used with this hybrid model to enhance intrusion detection rules and increase tolerance and prevention against emerging attacks. An anomaly identification engine utilizing reinforcement learning is central to ANIDS's anomaly detection capability. With this engine, the detection model is automatically updated, allowing it to adapt and learn new network traffic patterns and those associated with novel attack patterns. Compared to existing IDS research that relies on simulations, our proposed model is designed to operate continuously in network conditions. The speed and accuracy of our technology are therefore both considered. In our methodology, we consider network traffic information as the current factors of the environment in the framework of reinforcement learning (RL). The RL agent is represented by the intrusion detection engine, where the action executed by the agent corresponds to the results of the intrusion detection process. The reward signal is calculated based on the reliability of the identified results. The architecture of this system is depicted in Figure 2. To help with the implementation of the self-update functionality, reinforcement learning employs two distinct modes: learning mode and detection mode. The organization of the processes in these two modes is listed below:

Learning Mode

- i) The reinforcement learning agent processes the state factors, which are derived from raw network traffic data and generates a response.
- ii) The reward function module determines the reward using the action and label that provide feedback to the RL agent.
- iii) The reinforcement learning agent modifies its policy, namely the intrusion detection model, by considering the reward and states.
- iv) Return to stage i.

Detection mode

- i) The RL agent processes the state factors, which are derived from raw network traffic data and generates a response.
- ii) The reward function modules provide a simulated reward to the reinforcement learning agent to maintain the functionality of the process.
- iii) Return to stage i.

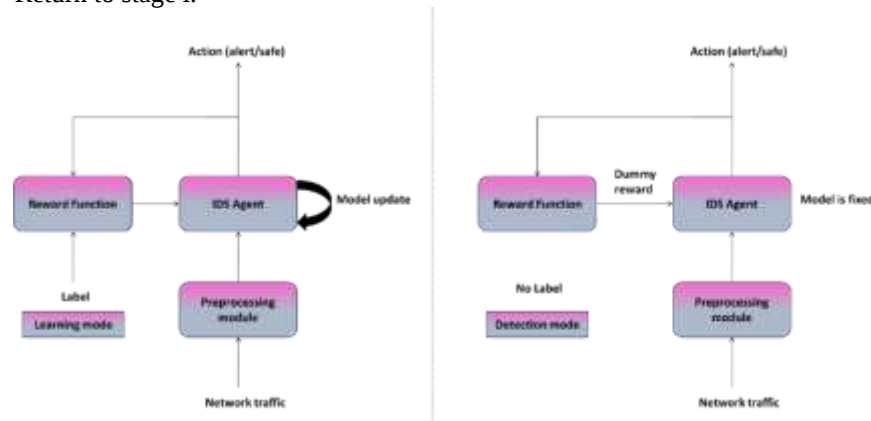


Figure 2: The operation modes of the reinforcement learning

During the learning phase, the DRL agent assesses the quality of the detection results by monitoring the reward signal. In case of a decrease in the reward, the detection model will be updated using the most recent data to improve the efficacy of IDS. During the detection mode, the RL agent uses an established detection model to analyze network data and the compensation provided is a temporary reward that serves to maintain the operational process. The distinction between these two modalities depends on the way in which the reward function computes the actual reward when combined with the label. A switch flag is used to enable flexible switching between the two modes in the system. The current design permits the system to assess and update the detection model at any given moment. It should be observed that the DRL agent provides a specific action as the outcome of intrusion detection. Our system has the capability to be enhanced for the purpose of intrusion protection. The structure of our DRL model depends upon a Deep Q Network (DQN) agent. This agent leverages a deep neural network to estimate the anticipated value of rewards. GCN is a type of neural network that processes and modifies graph structures. Considering an undirected graph description $H_{bh} = (U, F)$, let $Y \in \mathbb{R}^{n \times N}$ be a matrix that contains m nodes together with their related characteristics. The attribute vector for node u , denoted as $w_u \in \mathbb{R}^N$, is defined to have a size of N . The non-linear function representation of each GCN layer can be derived from the adjacency matrix $B \in \mathbb{R}^{n \times n}$ is shown in Equation (2).

$$G^{(1)} = \delta (\bar{B} G^{(0)} X^{(0)} + a^{(0)}) \quad (2)$$

The parameters $G^{(0)} = W, X^{(0)} \in \mathbb{R}^{N \times E}$ and $a^{(0)} \in \mathbb{R}^F$ represent the weight and bias, respectively. In particular, E represents the number of output characteristics per node. Where $\delta(\bullet)$ represent a non-linear activation function that includes ReLU. The normalized symmetric adjacency matrix, denoted as $\bar{B}$, can be generated by using Equation (3) that follows:

$$\bar{B} = \hat{C}^{-\frac{1}{2}} \hat{B} \hat{C}^{-\frac{1}{2}} \quad (3)$$

The degree matrix of the graph H , denoted as C , is considered in the presence of self-loops, represented by matrix $\hat{B} = B + J$. J denotes the identity matrix shown in Equation (4). Through stacking many convolutional layers, it is possible to collect information from distant neighbours.

$$G^{(k+1)} = \delta(\bar{B} G^{(k)} X^{(k)} + c^{(k)}) \quad (4)$$

k - k^{th} layer,

$X^{(k)}$ - Weight matrix associated with k^{th} GCN layer,

$c^{(k)}$ - bias matrix,

Following the creation of the graph displaying alerts, the graph representation is fed into a two-layer Graph Convolutional Network (GCN) that facilitates the transmission of information among the nodes. The activation function utilized is ReLU and the result of the second layer is transmitted through a soft max classifier, as shown in Equation (5).

$$Y = softmax(\bar{B}ReLU(\bar{B}WX^{(0)} + a^{(0)})X^{(1)} + b^{(1)}) \quad (5)$$

GCN facilitates the collection of data from neighbouring nodes through the use of a series of convolutional layers. This process helps in the identification of connections between alerts, hence facilitating the identification of potential threats. Graph Convolutional Network system architecture is displayed in Figure 3.

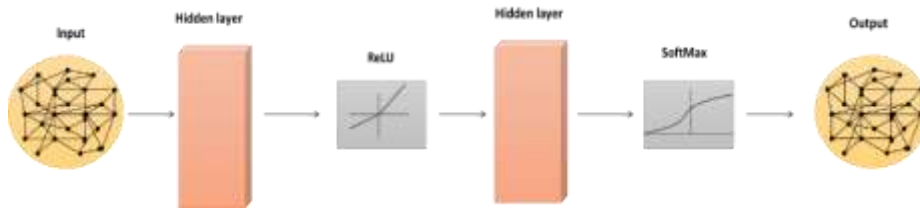


Figure 3: Graph Convolutional Network system architecture

Intrusion detection systems can be improved using a hybrid strategy that combines Deep Reinforcement Learning (DRL) with Graph Convolutional Networks (GCNs). System and network protection requires detection of intrusions efficiently. Despite constantly evolving threat environments, deep reinforcement learning algorithms (DRLs) are well suited for learning optimal policies with iterative experiments. As an alternative, graph convolutional networks can detect complex relationships between complex data structures, such as graphs of network traffic. Combining these two approaches allows for multi-layered intrusion detection.

A system with previous interactions can identify and adapt to changing patterns of attack based on past interactions. When GCNs are added, property extraction is improved since node relationships and networks are replicated. In this way, it is capable of detecting network traffic problems that conventional tools cannot. An environment that is rapidly changing can be made more accurate and resilient by using technologies such as DRL and GCN. By using academics and cyber security experts, the model can be improved, making it possible for it to adapt as new attacks are discovered. This algorithm depicts a pseudocode for Deep Reinforcement Learning coupled with Graph Convolutional Networks (DRL+GCN).

Algorithm 1: Graph Convolutional Networks (DRL+GCN) for Deep Reinforcement Learning

Preparing the models f DRL and GCN for initializing process

Preparing of intrusion detection environments

A range of episodes (num_episodes) was selected

A reset is necessary for the environment

Initial_state equal to state

total_reward equal to Zero

While not yet completed:

action = DRL.select_action(state)

next_state, reward, done = environment.step(action)

DRL.update(state, action, next_state, reward)

```

GCN.update(graph_data)
total_reward += reward
state = next_state
if episode % target_update_interval == 0:
if episode % gc_n_update_interval == 0:
if episode % intrusion_check_interval == 0:
    detected_intrusion = perform_intrusion_detection(DRL, GCN, environment)
    if detected_intrusion:
        take_action_to_mitigate()

```

1. Result

This study utilizes "VMware's Mini net Virtual Machine." An emulator implemented in Python enables the creation of virtual networking infrastructures. This infrastructure enables the interconnection of virtual hosts by means of diverse components, including switches, links and controllers. The device is equipped with Linux network software and has the capability to facilitate Open Flow for customized routing and SDN support. Due to the requirement of a Linux server for the installation of Mini net, we choose the utilisation of "Oracle VM Virtual Box" for our simulation purposes. The experiment is executed on a desktop system operating on a 32-bit Ubuntu 18.04 LTS. The computer was equipped with a Core-i7 processor and possessed a total of 16 GB of RAM.

Accuracy evaluates the system's capacity to detect and categorize network intrusions. This serves to minimize the appearance of false positives and promote the entire safety of the network. Achieving a considerable amount of accuracy is crucial, as it helps to ensure that the intrusion detection system is capable of differentiating between regular and threatening network activities. The identification of risks and weaknesses is of greatest significance in securing software-defined networks. The comparative analysis of various approaches, including DNN [11] and GRU-RNN [11], revealed that their respective accuracies were 80.7% and 89%. Based on our evaluation of our proposed Deep Reinforcement Learning with Graph Convolutional Networks (DRL+GCN) method, we found that it exhibits a greater accuracy of 93.8%, as shown in Figure 4.

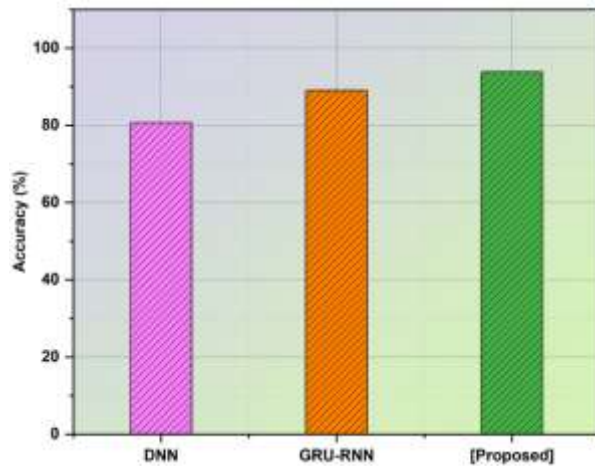


Figure 4: Accuracy

Precision and accuracy are presented in Table 2. This metric measures how well the system is able to detect and categorize security risks in the network. Therefore, false positive alerts will be reduced because it estimates the ratio between identified intrusions and a total number of detected intrusions. High precision ensures that a smaller percentage of legal network activity is incorrectly classified as an intrusion, thereby enhancing the intrusion detection system's efficacy and reliability. In comparison with existing methods such as DNN (85%) and GRU-RNN (89%), the proposed DRL+GCN model provides 94.2% precision. The data presented in Figure 5 demonstrate that our proposed approach provides the best precision value and reliability..

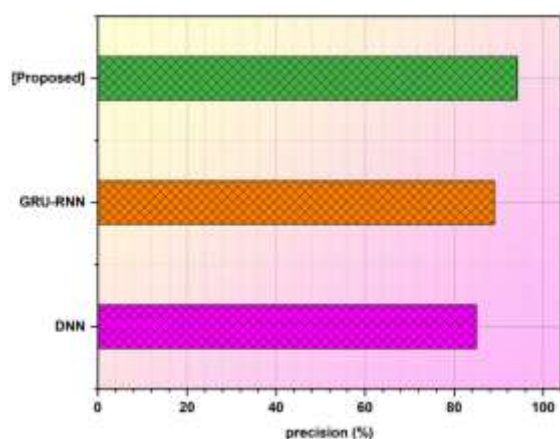


Figure 5: Precision

Table 2: Results of accuracy and precision

Methods	Accuracy (%)	Precision (%)
DNN	80.7	85
GRU-RNN	89	89
[Proposed]	93.8	94.2

The recall test measures the system's ability to detect and classify intrusions and reduce the likelihood of false negatives. In this way, security measures are enhanced by minimizing the probability of unknown attacks by predicting the ratio between detected intrusions and actual incursions. To minimize cyber threats associated with network vulnerabilities and information thefts, maximizing recall rates is of paramount importance. DNN and GRU-RNN, which are existing methods, have a recall ratio of 81% and 89%, respectively. Alternatively, our proposed method, DRL+GCN, had a recall ratio of 93%, which is shown in Figure 6. As a result, our suggested approach performs better than existing approaches.

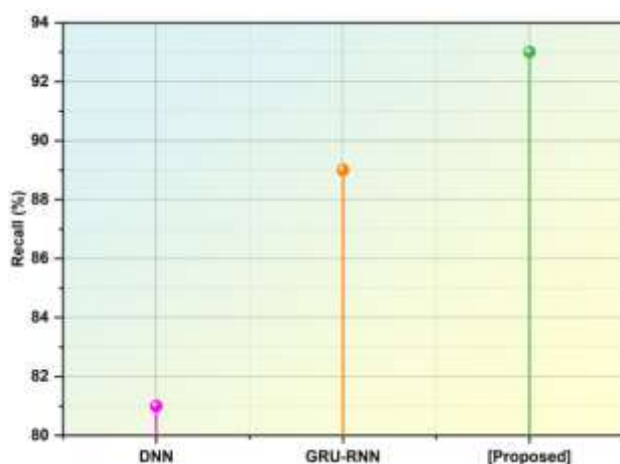


Figure 6: Recall

According to Table 3, precision and accuracy were both high. When detecting network attacks, the F1 score evaluates precision versus recall. In this example, we demonstrate that the model is competent at classifying safe and malicious network behaviors. It is an indication that the intrusion detection structure has high F1 scores, and precision and recall work together to prevent false positives and false negatives and ensure network security. The F1 scores for the existing DNNs and GRUs-RNNs are 81% and 89%, respectively. The F1-score of our proposed

DRL+GCN method is higher than that of the CTR-free method, showing that it is more efficient than the CTR-free method. A graph showing the F1-score's performance is shown in Figure 7.

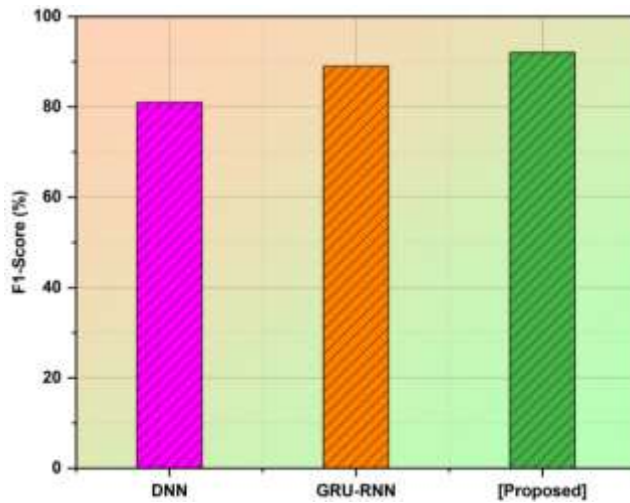


Figure 7: F1-score

Table 3: Results of F1-score and recall

Methods	F1-Score (%)	Recall (%)
DNN	81	81
GRU-RNN	89	89
[Proposed]	92	93

2. Conclusion

Detecting intrusions in SDN involves monitoring and assessing network traffic and behavior to identify and address possible security risks and unauthorized access. A flexible and manipulable approach that strengthens network security. A combination of Deep Reinforcement Learning and Graph Convolutional Networks (DRL + GCN) is presented in this paper to enhance the security of SDN using IDS. The NSL-KDD dataset is utilized in this research. The experimental results demonstrate that the proposed methodology exhibits enhanced performance in terms of accuracy (93.8%), recall (93%), F1-score (92%) and precision (94.2%) for the purpose of intrusion detection. The restriction observed in intrusion detection relates to its elevated computing complexity and resource demands, which could hinder the real-time implementation and scalability of extensive network systems. The upcoming scope includes the evaluation of hardware acceleration and the development of innovative algorithms to enable the real-time implementation and scalability of substantial intrusion detection techniques.

References

- [1] Bhardwaj, A., Tyagi, R., Sharma, N., Khare, A., Punia, M.S. and Garg, V.K., "Network intrusion detection in software-defined networking with self-organized constraint-based intelligent learning framework." *Measurement: Sensors*, 24, p.100580. 2022.
- [2] Abou El Houda, Z., Senhaji Hafid, A. and Khoukhi, L., "A novel unsupervised learning method for intrusion detection in software-defined networks." In *Computational Intelligence in Recent Communication Networks* (pp. 103-117). Cham: Springer International Publishing, 2021.
- [3] Girdler, T. and Vassilakis, V.G., "Implementing an intrusion detection and prevention system using Software-Defined Networking: Defending against ARP spoofing attacks and Blacklisted MAC Addresses." *Computers & Electrical Engineering*, 90, p.106990. 2021.
- [4] Kumar, C., Biswas, S., Ansari, M.S.A. and Govil, M.C., "Nature-inspired intrusion detection system for protecting software-defined networks controller." *Computers & Security*, 134, p.103438. 2023.
- [5] Shaji, N.S., Muthalagu, R. and Pawar, P.M. "SD-IIDS: intelligent intrusion detection system for software-defined networks." *Multimedia Tools and Applications*, pp.1-33. 2023.

- [6] Ahmad, A.A., "Solution Model for Intrusion Detection in Software Defined Networking (SDN) using Machine Learning." 2021.
- [7] Alshammri, G.H., Samha, A.K., Hemdan, E.E.D., Amoon, M. and El-Shafai, W., "An efficient intrusion detection framework in software-defined networking for cybersecurity applications." *CMC-Comput. Mater. Contin.*, 72, pp.3529-3548. 2022.
- [8] Bour, H., Abolhasan, M., Jafarizadeh, S., Lipman, J. and Makhdoom, I., "A multi-layered intrusion detection system for software defined networking." *Computers and Electrical Engineering*, 101, p.108042. 2022.
- [9] Scaranti, G.F., Carvalho, L.F., Junior, S.B., Lloret, J. and Proença Jr, M.L., "Unsupervised online anomaly detection in Software Defined Network environments." *Expert Systems with Applications*, 191, p.116225. 2022.
- [10] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A. and Venkatraman, S., "Deep learning approach for intelligent intrusion detection system." *Ieee Access*, 7, pp.41525-41550. 2019.
- [11] Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M. and El Moussa, F., "DeepIDS: Deep learning approach for intrusion detection in software defined networking." *Electronics*, 9(9), p.1533. 2020.
- [12] Satheesh, N., Rathnamma, M.V., Rajeshkumar, G., Sagar, P.V., Dadheech, P., Dogiwal, S.R., Velayutham, P. and Sengan, S., "Flow-based anomaly intrusion detection using machine learning model with software defined networking for OpenFlow network." *Microprocessors and Microsystems*, 79, p.103285, 2020.
- [13] Jayasri, P., Atchaya, A., Parveen, M.S. and Ramprasath, J., "Intrusion detection system in software defined networks using machine learning approach." *International Journal of Advanced Engineering Research and Science*, 8(8), 2021.
- [14] Janabi, A.H., Kanakis, T. and Johnson, M., "Overhead reduction technique for software-defined network based intrusion detection systems." *IEEE Access*, 10, pp.66481-66491, 2022.
- [15] Dey, S.K. and Rahman, M.M., "Effects of machine learning approach in flow-based anomaly detection on software-defined networking." *Symmetry*, 12(1), p.7, 2019.
- [16] Naqash, T., Shah, S.H. and Islam, M.N.U., "Statistical Analysis Based Intrusion Detection System for Ultra-High-Speed Software Defined Network." *International Journal of Parallel Programming*, 50(1), pp.89-114, 2022.
- [17] Alshammari, T.M. and Alserhani, F.M., "Scalable and Robust Intrusion Detection System to Secure the IoT Environments using Software Defined Networks (SDN) Enabled Architecture." *Int. J. Comput. Networks Appl*, 9(6), pp.678-688, 2022.
- [18] Latah, M. and Toker, L., "An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks." *CCF Transactions on Networking*, 3(3-4), pp.261-271, 2020.
- [19] Bocu, R. and Iavich, M., "Real-Time Intrusion Detection and Prevention System for 5G and beyond Software-Defined Networks." *Symmetry*, 15(1), p.110, 2022.
- [20] Li, W., Wang, Y., Jin, Z., Yu, K., Li, J. and Xiang, Y., "Challenge-based collaborative intrusion detection in software-defined networking: an evaluation." *Digital Communications and Networks*, 7(2), pp.257-263, 2021.
- [21] S. A. Shawkat, B. A. Tuama, and I. Al Barazanchi, "Proposed system for data security in distributed computing in using triple data encryption standard and Rivest Shamir Adlemen," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 6, pp. 6496–6505, 2022, doi: 10.11591/ijece.v12i6.pp6496-6505.
- [22] F. Hazzaa, S. Yousef, E. Sanchez and M. Cirstea, "Lightweight and Low-Energy Encryption Scheme for Voice over Wireless Devices," *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Washington, DC, USA, 2018, pp. 2992-2997, doi: 10.1109/IECON.2018.8591451.
- [23] F. Hazzaa, S. Yousef, N. H. Ali and E. Sanchez, "The Effect of Nodes Density on Real Time Traffic in Mobile Ad Hoc Network," *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*, London, UK, 2019, pp. 209-212, doi: 10.1109/ICGS3.2019.8688314