A Secure Deep Q-Reinforcement Learning Framework for Network Intrusion Detection in IoT-Fog Systems

1st Sujit Bebortta
Department of Computer Sceince
Ravenshaw University
Cuittack, 753003, India
sujitbebortta1@gmail.com

2nd Subhranshu Sekhar Tripathy
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar, 751024, India
subhranshu.008@gmail.com

3rd Vandana Sharma
Department of Computational Sciences
CHRIST (Deemed to be University)
Delhi NCR Campus, India
vandana.juyal@gmail.com

4th Jnana Ranjan Behera
School of Electronics
KIIT Deemed to be University
Bhubaneswar, 751024, India
jnanaranjan07@gmail.com

5th Aishwarya Nayak
Department of Computer Science & Engineering
DRIEMS University
Cuttack, 754022, India
aishwaryanayak22@gmail.com

***Abstract*—IoT-Fog system security depends on intrusion detection system (IDS) since the growing number of Internet-of-Things (IoT) devices has increased the attack surface for cyber threats. The dynamic nature of cyberattacks often makes it difficult for traditional IDS techniques to stay up to date. Because it can adapt to changing threat landscapes, deep Q-reinforcement learning (DQRL) has become a potential technique for ID in IoT-Fog situations. In this paper, an IDS system for IoT-Fog networks based on DQRL is proposed. The suggested solution makes use of fog nodes' distributed computing power to provide real-time IDS with excellent accuracy and minimal latency. With feedback from the network environment, the DQRL agent learns to recognize and categorize network traffic patterns as either normal or intrusive. Adaptive exploration techniques, effective reward functions, and deep neural networks for feature extraction are adopted by the system to improve predictive performance. The evaluation findings show that, in terms of detection accuracy, precision, recall and f-measure, the proposed DQRL provides flexibility to changing threat patterns as compared to conventional IDS techniques. A vast array of cyberattacks, such as malware infections, denial-of-service (DoS) attacks, and command-and-control communications, are successfully recognized and categorized by the system. It is possible that the suggested solution will be crucial in safeguarding IoT-Fog networks and preventing cyberattacks**

Keywords— Internet of Things, Fog Computing, Intrusion detection system, Deep Q-Reinforcement Learning, Performance Evaluation.

I. Introduction

The Internet of Things (IoT) and Fog computing have fused, transforming technology. Data processing and network organization have changed, creating new opportunities and difficulties. IoT devices—intelligent sensors, actuators, and networked devices—are driving this shift. Industry includes smart cities, healthcare, industrial automation, and environmental monitoring [1]. IoT devices have become essential to current data ecosystems, enabling massive data collecting, processing, and transmission at extreme rates. The rise ofIoT devices has made networks more vulnerable to various threats, notwithstanding this innovation.

Many industries use IoT devices, a significant driver of this paradigm change. These devices improve infrastructure in "smart cities," monitor vital signs in healthcare, and optimize manufacturing processes in "industrial automation." processing, Data sensing, and transmission efficiency make IoT devices appealing [2]. They can be used in an intelligent factory's automated assembly line, a hospital's patient monitoring system, or A city's traffic management system. These devices' ubiquitous availability and versatility signal their transformative potential.But as IoT devices spread, they've also shown the ugly side of the technological revolution. Due to their low processing storage space, power, and battery life, bad guys target these devices. Unauthorized access, data breaches, and malware infections are common cybersecurity concerns for IoT devices, but DDoS attacks, botnet infiltrations, and data exfiltration are more insidious. Due to their limited security resources, IoT devices are vulnerable to several assaults [3]. This issue emphasizes the need for innovative security methods to secure IoT systems and their sensitive data without overburdening these devices' limited capabilities.

Fog computing represents a crucial architectural approach to address IoT device limitations. FOG computing brings cloud capabilities to the network edge, unlike traditional cloud computing. Move processing power closer to the action to increase responsiveness, real-time decision-making, and bandwidth utilization. Nodes and gateways in the Fog layer process data near IoT devices. Fog computing's distributed nature solves latency-sensitive applications and IoT data mountains.However, Fog computing has security requirements. As data processing shifts to the periphery, a network's attack surface expands [4]. IoT-Fog systems must be protected. Defending against today's growing number of network breaches, vulnerabilities, and new attack vectors requires robust Fog layer intrusion detection and response. Secure Deep Q-Reinforcement Learning Framework, a cutting-edge solution for IoT-Fog system security, addresses this demand.For urgent IoT-Fog system technical and security challenges, a Secure Deep Q-Reinforcement Learning Framework was created. IoT devices have limited processing speed, storage space, and battery life. These restrictions expose them to several attacks. Resource-limited IoT devices can face DDoS attacks, virus breakouts, and unauthorized data access. They need imaginative security techniques to protect these gadgets without using too much of their limited resources [5].

Fog computing addresses IoT computational and latency issues. Bringing data processing and analytics closer to the network edge improves performance using fog computing. This distributed computing model raises security concerns, particularly in Fog layer network intrusion detection. Fog computing connects edge devices to cloud servers by processing, data storage, and communications. Nodes in fog computing are heterogeneous and deployed in many scenarios. Fog software should manage resources effectively. Figure 1 shows the Fog software architecture's main components.

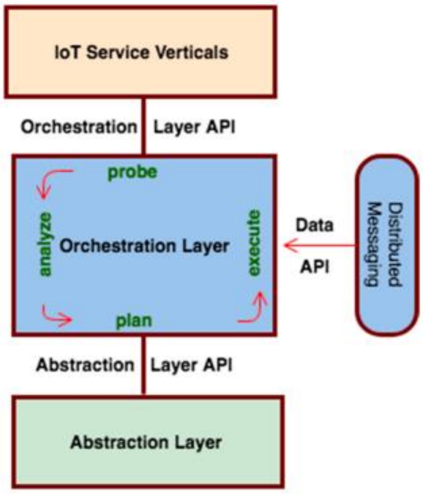

Figure 1 Fog software architecture's main components [6]

A significant challenge is the ever-changing threat landscape. The ways hackers use to break into networks and systems change and get more complex. Cyberattacks are becoming increasingly difficult, so traditional security methods typically fall behind. Automated security solutions that identify, evaluate, and stop emerging threats are popular [7]. By merging machine learning with decision-making, Reinforcement Learning can address these shifting security challenges. This research helps solve IoT-Fog system technical and security challenges. In this study, we describe an IoT-Fog-specific security architecture. This technique uses Fog computing to protect IoT devices' data-generating and processing sites from network vulnerabilities. This case helps the framework overcome difficulties and reap the benefits of IoT-Fog settings. Based on deep neural networks and reinforcement learning, Deep Q-reinforcement learning is a robust machine learning paradigm. Deep Q-reinforcement understanding lets the framework swiftly discover complex patterns, learn from past experiences, and make decisions. This capability considerably improves real-time network intrusion detection and proactive response. Introduce a detailed system model to operationalize the concept.

This paradigm governs the interactions between IoT devices, Fog nodes, and cloud computing resources. Designing and testing the suggested security framework is guided by the system model.An empirical study evaluated the framework in realistic IoT-Fog situations. The following sections detail the framework's pros and cons, revealing its efficiency and improvement possibilities. Finally, this paper offers a Secure Deep Q-Reinforcement Learning Framework to address IoT-Fog system technical and security challenges, and it also examines the architecture, system model, empirical results, comments, and future research directions for IoT-Fog network intrusion detection.

II. Realted Work

Sudqi Khater et al. introduced lightweight MLP-based vector space representation [8]. Next, they tested the suggested approach on the Australian Defense Force Academy Windows Dataset (ADFA-WD) and ADFA with Linux Dataset (ADFA-LD), a new generation system dataset featuring exploits and attacks on numerous programs. An et al. [9] reported apriori-inspired hypergraph clustering. We found the susceptible FCs link through our research. DDoS analysis ensured the model's resource consumption rate could be pushed.

Mourad et al. [10] developed a fog-assisted vehicular edge computing (VEC) system to offload IDS functions to federated automobile nodes in the low-latency ad hoc vehicular fog. Abdel-Basset et al. [11] developed a forensics-based DL (Deep-IFS) to detect IIoT traffic breaches. In addition to LocalGRU-learned local representations, the proposed MHA learns and captures global graphics with longer-range dependencies. Layers have a residual connection to prevent data loss. Pacheco et al. [12] presented an adaptive IDS based on ANN to identify fog node tampering and ensure transmission accessibility.

Prabavathy et al. [13] offer an online sequential extreme learning machine (OS-ELM) fog computing detection method. The proposed technique involves two steps: attack detection at local fog nodes and a cloud server summary of the IoT system. A 97.36% success percentage was achieved while testing this model on the NSL-KDD dataset. The authors claim fog nodes detect 25% faster than cloud-based implementation. Diro and Chilamkurti[14] used fog computing to create an IoT IDS. The authors suggested a distributed deep-learning intrusion detection system. Experimental results show that distributed parallel architecture is more accurate than centralized model. We train and test the IDS using the NSL[1]KDD dataset. With a 4.97% false alarm rate, the model detected 93.66 per cent of events.

The authors of [15] suggest creating a framework to discover all attack channels and counteract IoT system threats. The five-stage, interconnected graphical security model begins with data processing, where system data and security metrics are provided and processed. Security and gap models are constructed in the second stage. All IoT system attack vectors are represented in this model. The attack path shows how attackers can penetrate nodes to reach their target. Security visualization and analysis of the IoT network, including attack pathways, occur in the third and fourth stages. Malware detection and prevention in IoT networks were suggested in [16]. Fog computing is critical to improving virus detection and data security. Cloud and fog computing powered the malware detection system and circumvented smart device IDS deployment limits]. A framework was created to demonstrate the potential of IoT networks to limit malware proliferation. Using the ADFA-LD and ADFA-WD datasets, Borisaniya et al. [17] developed a modified vector space representation technique with multiple classifiers to obtain above 95% detection accuracy. Frequency-based models helped Xie et al. [18] detect ADFA-LD attack behaviour with a false positive rate of less than 20%.

Xie et al. [19] applied a one-class SVM model to short sequences like their predecessors. Overall, they improved, although false positives were close to 20%. A semantic model for anomaly identification using ADFA-LD Dataset short sequences is presented by G. Creech et al. [20]. They constructed a lexicon of words and phrases from the dataset and tested it with HMM, ELM, and one-class SVM algorithms. They achieved 15% FPR with ELM and 80% with SVM [29,30]. They assessed ADFA-WD using HMM, ELM, and SVM. HMM had 100% accuracy with 25.1% FPR, ELM 91.7% with 0.23% FPR, and SVM 99.58 % with 1.78% FPR.Illy et al. [21] presented a lightweight Fog-to-Things IDS. Using numerous base learners trained with well-established methods, the suggested method created multiple ensemble classifiers to recognize and categorize attacks. The NSL-KDD dataset shows that the IDS model outperforms alternative fog computing intrusion detection methods [22]. The recommended way has high binary and multiclass classification accuracy. Pacheco et al. [23] suggested artificial neural network intrusion detection for IoT fog nodes. The recommended solution restores connectivity after identifying a compromised fog node. Whether from hacking or hardware failures, the proposed solution worked in experiments.

Our Secure Deep Q-Reinforcement Learning Framework bridges Fog computing and deep reinforcement learning to help IoT-Fog systems detect network intrusions. This connection creates a proactive, flexible security solution to handle new threats.

III. Proposed Framework

In this section, proposed IoT fog based task offloading framework is introduced. Performance indexes including latency and the corresponding energy consumption are formulated for specific offloading scheme.

3.1 System Model

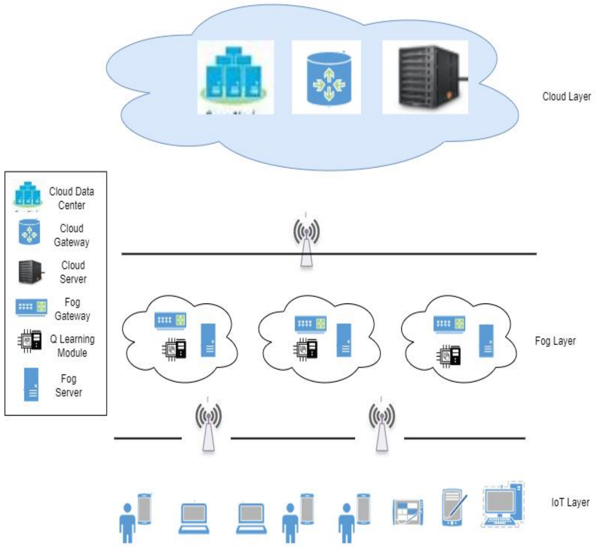

Figure 2 Three layer IoT fog framework

Figure 2 illustrate the proposed three layer task offloading framework for IoT fog computing.

The bottom layer is the IoT layer which contains several devices which is generating multiple tasks in unit time. The middle layer is the fog layer which contains distributed nodes. Each node in the fog layer received the information generated from each task in IoT layer. Then the nodes aggregate the sensor data, filter them and process them for further processing. The data are distributed in fog layer so the easy processing is possible in this layer. The next layer is the cloud layer which stores the processing result for further reuse.

A. Computation Model

For our proposed framework let there are n (where n=1, 2,…s) number of sensors are there in IoT layer. The Sensors are generating t (where t=1, 2…p) number of tasks which are sending to the fog nodes. In next layer i.e fog layer is having m (where m=1, 2….f) number of distributed fog nodes for computation and processing. The task is having properties like $t_{size.}$

The latency in the proposed framework is the combination of transmission latency and computation latency.

The transmission latency of task t_i when uploaded to fog node m_j is

$$l^t_{i,j} = \frac{t_{size}}{m_{tr}} \qquad (1)$$

Where t_{size} is the task size and m_{tr} is the transmission data rate of the fog node.

The computation latency of t_i when uploaded and executed on fog node m_j is

$$l^c_{i,j} = b_{i,j}\frac{t_{size}}{m_{cap}} \qquad (2)$$

Where $b_{i,j}$ is the binary variable $b_{i,j} \in (0,1)$ used to identify whether the task t_i is uploaded to the fog node m_j or not. m_{cap} is the capacity of fog node m_j and calculated as

$$m_{cap} = m_{pro} + m_{bw} \qquad (3)$$

Where m_{pro} is the processing power of fog node m_j calculated as

$$m_{pro} = m_{mips} + m_{cpu} \qquad (4)$$

Where m_{mips} is the number of instructions (MIPS) in fog node m_j and m_{cpu} is the CPU utilization of fog node $m_{j.}$

The total latency will be

$$L_{i,j} = l^t_{i,j} + l^c_{i,j} \qquad (5)$$

The energy consumption of the fog node is the combination of transmission energy and computation energy.

The transmission energy is the energy consumed for transmitting task t_i to fog node m_j and calculated as

$$e^t_{i,j} = t_{size} + energy\ consumed\ per\ unit\ data\ transmission(6)$$

The computation energy is the energy consumed for execution of task t_i on fog node m_j and calculated as

$$e^c_{i,j} = t_{size} + energy\ consumed\ per\ unit\ data\ execution(7)$$

The total energy consumption will be

$$E_{i,j} = e^t_{i,j} + e^c_{i,j}(8)$$

B. Deep Q-Reinforcement Learning Model

In Internet of Things (IoT) scenarios, deep Q-reinforcement learning (DQRL) has become a promising method for network intrusion detection. Combining Q-learning, a reinforcement learning algorithm that learns to choose the best course of action in a given environment to maximize a reward signal, with deep learning is known as DQRL. DQRL agents can be trained to recognize and categorize network traffic patterns as either normal or invasive in the context of network intrusion detection systems by using the input they receive from the network environment.

DQRL can be applied to intrusion detection systems in the IoT due to a number of important benefits. Firstly, DQRL agents have the ability to learn from big and intricate datasets of network traffic, which is crucial for efficient intrusion detection in IoT where data diversity and volume are always increasing. Second, DQRL agents are highly adapted to handle changing cyberthreats and the dynamic nature of IoT networks because they are adaptive and have the capacity to continually enhance their performance over time. Third, real-time threat mitigation and network edge intrusion are made possible by the distributed deployment of DQRL agents among IoT devices. In order to improve intrusion detection accuracy, current research in DQRL for NID is concentrated on creating more scalable and effective algorithms, enhancing the generalization skills of DQRL agents, and integrating other data sources including device logs and sensor readings. DQRL is expected to be a major contributor to IoT network security and cyberattack prevention as it develops further.

In a Markov Decision Process (MDP), its states are illustrative of the system's sub-problems. Thus, for some state $s \in S$, we can define the value function with policy μ as $\vartheta_\mu(s)$, and the expected return corresponding to the value function which acts with policy μ can be mathyematically defined as follows:

$$\vartheta_\mu(s) = E_\mu[R(t) + \delta\vartheta_\mu(S_{t+1})|S_t = s], \quad (9)$$

where δ refers to the discount factor such that $\delta \in [0,1]$.

TD learning decomposes the estimation above with bootstrapping. Given a value function V : S → R, the simplest version, TD(0), is the following one-step bootstrapping:

The temporal difference in the learning process decomposes the above expression in Eq.(9) by leveraging bootstrapping. Hence, for a given value function represented by the mapping $V: S \rightarrow R(t)$, and the simplest order temporal difference can be represented by the below single-step bootstrapping as,

$$V_\mu(S_t) \leftarrow V_\mu(S_t) + \theta[R(t) + \delta V_\mu(S_{t+1}) - V_\mu(S_t)], \quad (10)$$

where $R(t) + \delta V_\mu(S_{t+1})$ represents the temporal difference target, and $R(t) + \delta V_\mu(S_{t+1}) - V_\mu(S_t)$ provides the temporal difference error for the above Eq.(10).

The value of a policy, as represented by the Q-function, offers a means to gauge the efficacy of a certain intrusion detection activity in the context of intrusion detection in an IoT-Fog system. We compute the Q-value, which indicates the predicted long-term benefit associated with doing a particular action in a given condition, to find the optimal course of action. The intrusion detection system is able to learn and adjust to changing threats since the Q-function is updated on a regular basis depending on feedback and experiences from the system. Thus, we can define the Q-function for our proposed system as below:

$$q_{\mu}(s,a) = E_{\mu}\left[R(t+1) + \delta\vartheta_{\mu}(S_{t+1})|S_t = s, A_t = a\right]. \quad (11)$$

The simplest method for implementing an efficient intrusion detection policy in an IoT-Fog system is to take a greedy approach, in which the action with the highest Q-value is chosen at each stage. This can be accomplished by acting greedily for the policy function $\mu'(s,a) = arg\max_{a'} q_{\mu}(s,a')$, such that the improvement in performance can be ensured by, $\mu'(s,a) = \max_{a'} q_{\mu}(s,a') \geq q_{\mu}(s,a)$. However, if the system doesn't consider other options, this greedy strategy could result in less than ideal performance. We can solve this by adding a small probability that an action, regardless of its Q-value, will be chosen at random. The system can balance exploitation (picking the most well-known action) with exploration (trying new actions to perhaps find better ones) with this technique, which is known as ε-greedy exploration. By averaging the Q-values of the greedy and random acts, one may determine the Q-value of an ε-greedy policy. This can be stated as,

$$q_{\mu}\left(s,\mu'(s)\right) = (1-\epsilon)\max_{a\in A} q_{\mu}(s,a) + \frac{\epsilon}{|A|}\sum_{a\in A} q_{\mu}(s,a). \quad (12)$$

From Eq.(12), it may be noteworthy to mention that sum of $\frac{\mu(s,a)-(\epsilon/|A|)}{1-\epsilon}$ with action $a \in A$ will equal to 1. Considering the fact that the maximization of the Q-function will not be lesser than its weighted average, we obtain the below expression for our assumption,

$$\begin{aligned} q_{\mu}\left(s,\mu'(s)\right) &= (1-\epsilon)\max_{a\in A} q_{\mu}(s,a)\sum_{a\in A}\frac{\mu(s,a)-(\epsilon/|A|)}{1-\epsilon} + \frac{\epsilon}{|A|}\sum_{a\in A} q_{\mu}(s,a) \\ &\geq (1-\epsilon)\sum_{a\in A}\frac{\mu(s,a)-(\epsilon/|A|)}{1-\epsilon} q_{\mu}(s,a) \\ &+ \frac{\epsilon}{|A|}\sum_{a\in A} q_{\mu}(s,a) = q_{\mu}\left(s,\mu(s)\right) \end{aligned} \quad , \quad (13)$$

From the above Eq.(13), it is clear that the Q-value to act upon the $\varepsilon -$greedy policy μ' is more than the original policy $\mu(s,a)$, and hence ensures that the improvement in performance of the learning agent.

In the DQRL, the off-policy approach allows learning from its past experience, and hence the simplest Q-learning update rule considering the set of states and actions over a given problem space can be expressed as,

$$Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \theta\left[R(t) + \delta\max_{A_{t+1}} Q(S_{t+1},A_{t+1}) - Q(S_t,A_t)\right] \quad , \quad (14)$$

where A_t represents set of time-dependent action and can be sampled over ε-greedy strategies w.r.t the Q-value, further it is worthwhile to mention that the actions A_{t+1} are selected greedily in the proposed DQRL framework.

IV. Results and Discussions

The simulation framework for fog computing, iFogSim, is used to establish the topology of the proposed paradigm. Three layers make up the topology: IoT devices, fog nodes, and cloud nodes. Sensor nodes gather information from their surroundings and forward it to fog nodes for further processing. Sensor node data is collected, processed, and sent to the cloud by fog nodes for additional study. Fog nodes can access centralized processing and storage resources thanks to the cloud. A gateway node is connected to four fog nodes in the experimental configuration. The computing power and available network bandwidth of the fog nodes vary widely. Numerous sensor nodes can be connected to a single fog node since the mapping between the sensor nodes and fog nodes is many-to-one. Numerous applications, such as network intrusion detection in the Internet of Things, can make use of the suggested paradigm. Fog nodes are able to recognize and categorize network traffic as either normal or intrusive by examining patterns in the traffic. Afterwards, suitable action can be taken to lessen network attacks using this knowledge.

IoT devices, fog nodes, and cloud nodes are the three levels that make up the system in this figure. Sensor nodes gather environmental data from their location at the network's edge. Data from sensor nodes is gathered and processed by fog nodes, which are situated closer to the IoT devices. Fog nodes can access centralized processing and storage resources from cloud nodes, which are located in centralized data centers. The data flow between the levels is depicted by the different layers in Figure 2. Data is sent from IoT devices to fog nodes, which in turn send data to cloud nodes, which in turn send data back to fog nodes. To schedule tasks and allocate resources, fog and cloud nodes communicate with one another. A possible method for network intrusion detection in the IoT-Fog system is the suggested paradigm. Through the utilization of fog nodes' distributed processing capabilities, the paradigm can offer real-time intrusion detection with minimal latency and optimal accuracy. In this study, we used the well-known benchmark NSL-KDD dataset, that comprises of a set of labeled network traffic data and is useful for building the model for testing and training the performance of the IDS. There are many different kinds of traffic in the collection, including malicious and legitimate traffic. The type of attack, such as a user-to-root, spoofing, root-to-local, or denial-of-service (DoS) attack, is indicated on the malicious traffic. When creating IDS for IoT-Fog systems, intelligent learning systems can benefit greatly from the NSL-KDD dataset.

Figure 3 depicts the comparison of training accuracy obtained for the proposed DQRL algorithm over convention algorithms viz., Deep Neural Network (DNN) and conventional reinforcement learning (RL). It was observed that with the increase in number of fog nodes, the training accuracy increases as the model is trained over its past experience from preceding fog nodes. Figure 4 provides the comparison of test accuracy over varying number of fog nodes. It was observed that as the number of fog nodes increase, the test accuracy observes a decrease. This is due to the reason that the testing sample instances becomes decrease with increase in fog nodes. It was also observed from our experimental findings that the proposed DQRL approach outperformed all others due to the $\varepsilon -$greedy approach incorporated in the learning phase of the model. In figure 5, the comparison for precision, recall, and f-measure is illustrated for the three models considered in this study. It was inferred from the figure that the DQRL algorithm depicted superior performance for all the three performance metric considered in this study. In Figure 6, the energy consumption of the DQRL model is compared with the DNN and RL algorithms with varying the number of tasks processed. It was observed that the proposed DQRL was more robust towards these dynamic behaviour of the system and outperformed the other two by showing relatively lower energy consumption. Finally, Figure 7 provides the delay incurred for processing the tasks by various models considered in the paper such as DNN, RL, and DQRL. It was noted from the simulation finding that the delay for the DQRL model was the lowest as compared to the DNN and RL models, depicting its adaptiveness in heterogeneous and dynamically changing IoT-Fog system.

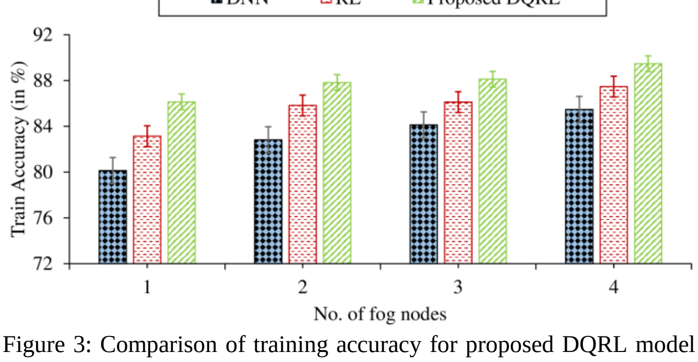

Figure 3: Comparison of training accuracy for proposed DQRL model with DNN and RL over different number of fog nodes.

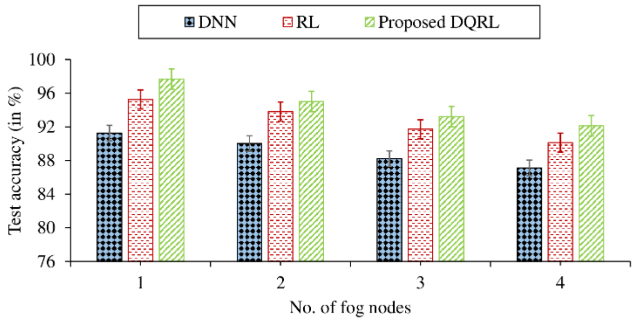

Figure 4: Comparison of test accuracy for proposed DQRL model with DNN nad RL over varying number of fog nodes.

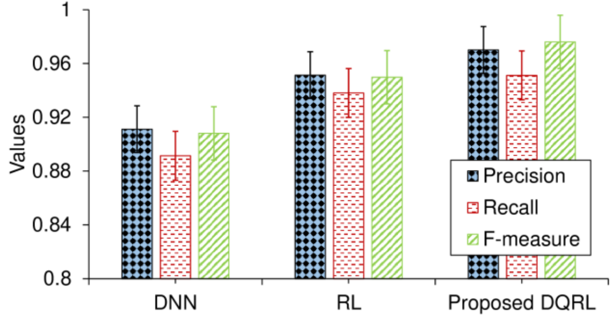

Figure 5: Comparison of precision , recall, and F-measure for proposed DQRL along with DNN and RL algorithm.

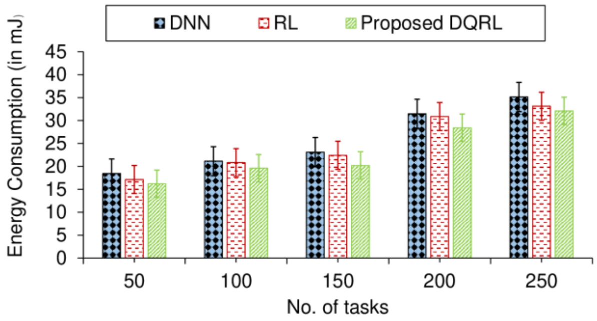

Figure 6: Comparison of energy consumption in millijoules for DNN, RL, and Proposed DQRL.

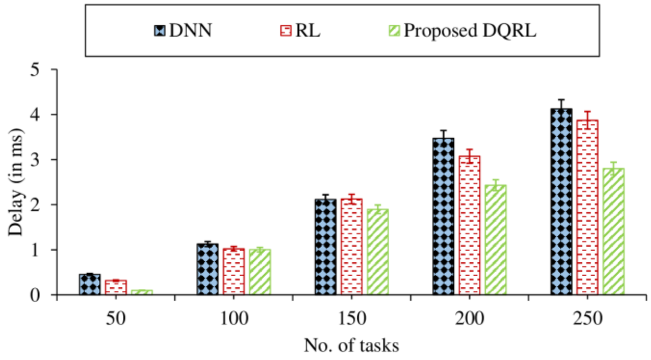

Figure 7: Comparison of delay in milliseconds for DNN, RL, and Proposed DQRL.

V. Conclusions and Future Work

While the attack surface for cyber threats has grown due to the increasing popularity of IoT devices, intrusion detection is essential for guaranteeing the security of IoT-Fog systems. The dynamic nature of cyberattacks often makes it difficult for traditional ID techniques to stay up to date. The suggested DQRL model's capacity to learn from and adjust to changing threat landscapes has made it a viable method for intrusion detection in IoT-Fog situations. An IDS for IoT-Fog networks based on DQRL was proposed in this paper. The suggested method made use of the fog nodes' dispersed computing power to provide real-time ID with low latency, high precision, and energy efficiency. Based on input from the network environment, the DQRL agent was able to recognize and categorize network traffic patterns as either regular or intrusive. The system used a number of methods to improve performance, such as effective incentive systems, adaptive exploration tactics, and deep neural networks for feature extraction. The suggested DQRL-based system outperforms conventional IDS methods in terms of detection accuracy and flexibility to changing threat patterns, as evidenced by evaluation results over the well-known NSL-KDD IDS dataset. A broad variety of cyberattacks, including as malware infections, denial-of-service attacks, and command-and-control communications, were successfully recognized and categorized by the system. It is possible that the suggested solution will be crucial in safeguarding IoT-Fog networks and thwarting cyberattacks. To sum up, the suggested DQRL-based ID system presents a viable method for accurate, adaptable, and real-time ID in IoT-Fog situations. The system is well-suited to protect IoT-Fog networks from assaults because of its capacity to learn from and adjust to changing threat patterns. Prospective avenues for investigation encompass researching the utilization of diverse deep learning architectures for feature extraction and investigating the application of DQRL to ID in more fog computing scenarios.

References

[1] P. Kumar, G. P. Gupta, and R. Tripathi, “A distributed ensemble design-based Intrusion Detection System using fog computing to protect the internet of things networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 10, pp. 9555–9572, 2020. doi:10.1007/s12652-020-02696-3

[2] B. Sudqi Khater, A. W. Abdul Wahab, M. Y. Idris, M. Abdulla Hussain, and A. Ahmed Ibrahim, “A lightweight perceptron-based intrusion detection system for Fog Computing,” *Applied Sciences*, vol. 9, no. 1, p. 178, 2019. doi:10.3390/app9010178

[3] B. S. Khater *et al.*, “Classifier performance evaluation for lightweight IDS using fog computing in IOT security,” *Electronics*, vol. 10, no. 14, p. 1633, 2021. doi:10.3390/electronics10141633

[4] Y. Labiod, A. Amara Korba, and N. Ghoualmi, “Fog computing-based intrusion detection architecture to protect IOT Networks,” *Wireless Personal Communications*, vol. 125, no. 1, pp. 231–259, 2022. doi:10.1007/s11277-022-09548-7

[5] S. Tu *et al.*, “Mobile fog computing security: A user-oriented smart attack defense strategy based on DQL,” *Computer Communications*, vol. 160, pp. 790–798, 2020. doi:10.1016/j.comcom.2020.06.019

[6] Bonomi, F.; Milito, R.; Natarajan, P.; Zhu, J. Fog computing: A platform for internet of things andanalytics. In Big Data and Internet of Things: A Roadmap for Smart Environments; Springer: Cham, UAE, 2014; pp. 169–186.

[7] O. A. Alzubi *et al.*, “Optimized machine learning-based intrusion detection system for fog and edge computing environment,” *Electronics*, vol. 11, no. 19, p. 3007, 2022. doi:10.3390/electronics11193007

[8] B. Sudqi Khater, A. W. Abdul Wahab, M. Y. Idris, M. Abdulla Hussain, and A. Ahmed Ibrahim, “A lightweight perceptron-based intrusion detection system for Fog Computing,” *Applied Sciences*, vol. 9, no. 1, p. 178, 2019. doi:10.3390/app9010178

[9] X. An, J. Su, X. Lü, and F. Lin, “Hypergraph Clustering Model-based association analysis of DDOS attacks in fog computing intrusion detection system,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, 2018. doi:10.1186/s13638-018-1267-2

[10] A. Mourad, H. Tout, O. A. Wahab, H. Otrok, and T. Dbouk, “*ad hoc* vehicular fog enabling cooperative low-latency intrusion detection,” *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 829–843, 2021. doi:10.1109/jiot.2020.3008488

[11] M. Abdel-Basset, V. Chang, H. Hawash, R. K. Chakrabortty, and M. Ryan, “Deep-IFS: Intrusion detection approach for industrial internet of things traffic in fog environment,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7704–7715, 2021. doi:10.1109/tii.2020.3025755

[12] J. Pacheco, V. H. Benitez, L. C. Felix-Herran, and P. Satam, “Artificial Neural Networks-based intrusion detection system for internet of things fog nodes,” *IEEE Access*, vol. 8, pp. 73907–73918, 2020. doi:10.1109/access.2020.2988055

[13] S. Prabavathy, K. Sundarakantham, and S. M. Shalinie, “Design of cognitive fog computing for intrusion detection in internet of things,” *Journal of Communications and Networks*, vol. 20, no. 3, pp. 291–298, 2018. doi:10.1109/jcn.2018.000041

[14] A. A. Diro and N. Chilamkurti, “Distributed attack detection scheme using Deep Learning Approach for Internet of Things,” *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2018. doi:10.1016/j.future.2017.08.043

[15] M. Ge, J. B. Hong, W. Guttmann, and D. S. Kim, “A framework for automating security analysis of the internet of things,” *Journal of Network and Computer Applications*, vol. 83, pp. 12–27, 2017. doi:10.1016/j.jnca.2017.01.033

[16] S. Shen *et al.*, “Multistage signaling game-based optimal detection strategies for suppressing malware diffusion in fog-cloud-based IoT Networks,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1043–1054, 2018. doi:10.1109/jiot.2018.2795549

[17] B. Borisaniya and D. Patel, “Evaluation of modified vector space representation using ADFA-LD and ADFA-WD Datasets,” *Journal of Information Security*, vol. 06, no. 03, pp. 250–264, 2015. doi:10.4236/jis.2015.63025

[18] M. Xie, J. Hu, X. Yu, and E. Chang, “Evaluating host-based anomaly detection systems: Application of the frequency-based algorithms to ADFA-LD,” *Network and System Security*, pp. 542–549, 2014. doi:10.1007/978-3-319-11698-3_44

[19] M. Xie, J. Hu, and J. Slay, “Evaluating host-based anomaly detection systems: Application of the one-class SVM algorithm to ADFA-LD,” *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2014. doi:10.1109/fskd.2014.6980972

[20] W. Haider, G. Creech, Y. Xie, and J. Hu, “Windows based data sets for evaluation of robustness of host based intrusion detection systems (IDS) to zero-day and stealth attacks,” *Future Internet*, vol. 8, no. 4, p. 29, 2016. doi:10.3390/fi8030029

[21] P. Illy, G. Kaddoum, C. Miranda Moreira, K. Kaur, and S. Garg, “Securing fog-to-things environment using intrusion detection system based on Ensemble Learning,” *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019. doi:10.1109/wcnc.2019.8885534

[22] A. R. STMicroelectronics et al., “From CIC-IDS2017 to lycos-IDS2017: A corrected dataset for Better Performance: IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology,” ACM Conferences, https://dl.acm.org/doi/10.1145/3486622.3493973 (accessed Nov. 3, 2023).

[23] J. Pacheco, V. H. Benitez, L. C. Felix-Herran, and P. Satam, “Artificial Neural Networks-based intrusion detection system for internet of things fog nodes,” *IEEE Access*, vol. 8, pp. 73907–73918, 2020. doi:10.1109/access.2020.2988055