IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 March 2024, accepted 14 April 2024, date of publication 22 April 2024, date of current version 6 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3392279

RESEARCH ARTICLE

DQQS: Deep Reinforcement Learning-Based Technique for Enhancing Security and Performance in SDN-IoT Environments

ZABEEHULLAH[1], FAHIM ARIF[1], (Senior Member, IEEE), NAUMAN ALI KHAN[1], JAVED IQBAL[1], (Senior Member, IEEE), FATEN KHALID KARIM[2], NISREEN INNAB[3], AND SAMIH M. MOSTAFA[4,5]

[1]National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[3]Department of Computer Science and Information Systems, College of Applied Sciences, Almaarefa University, Diriyah, Riyadh 13713, Saudi Arabia
[4]Computer Science Department, Faculty of Computers and Information, South Valley University, Qena 83523, Egypt
[5]New Assiut Technological University (NATU), New Assiut City 71684, Egypt

Corresponding authors: Javed Iqbal (javed.iqbal@mcs.nust.edu.pk) and Samih M. Mostafa (samih_montser@sci.svu.edu.eg)

This work was supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R300), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT The Internet of Things (IoT) is an emerging technology that allow smart devices to communicate through various heterogeneous channels (wired or wireless). However, for conventional networks, it has become a challenging task to efficiently control and manage the data flows of a huge number of devices. Software-defined networking (SDN) is a new way of thinking about networking. Because it is programmable, flexible, agile, and gives you a big picture of the network, it has tried to solve some IoT problems, like scalability, heterogeneity, and complexity. In large-scale SDN-IoT networks, there is a requirement for routing protocols that are both efficient and secure in order to ensure a superior level of quality of service (QoS) and quality of experience (QoE). To address the above stated challenges, a novel deep reinforcement learning (DRL) known as DQQS model is proposed. The aim is to achieve QoS and QoE while also ensuring the security of the SDN-IoT network. The proposed DQQS model dynamically extracts patterns from the past network history by interacting with the underlying network and generating optimized routing policies. This article employs three network metrics—throughput, latency, and the probability of avoiding malicious nodes—to measure the performance of DQQS. Simulations reveal that the proposed framework outperforms four state-of-the-art routing algorithms: OSPF, L-L Routing, Sailfish Routing, and RL-Routing in terms of both throughput and latency. For instance, in an attacked environment, the proposed DQQS model achieved the highest throughput value of 14.5 Mbps, surpassing OSPF at 8 Mbps, L-L at 8.2 Mbps, Sailfish at 9 Mbps, and RL at 9.5 Mbps. Similarly, this model exhibited superior performance in latency, recording the lowest latency value of 52 ms, compared to OSPF 88 ms, L-L 85 ms, Sailfish 72 ms, and RL 75 ms routing algorithms. The experimental results demonstrate that this new DQQS model is a pioneering deep reinforcement learning-based technique that optimally addresses secure routing in the SDN-IoT environment, ensuring enhanced quality of service and experience, and outperforming state-of-the art DL methodologies in both security and network performance metrics.

INDEX TERMS Deep reinforcement learning, Internet of Things, malicious node detection, optimal network management, routing optimization, software defined network, security.

I. INTRODUCTION

The Internet of Things (IoT) encompasses a vast network of diverse devices connected across multiple communication

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Abdur Razzaque.

interfaces [1]. Notably, IoT devices are characterized by high mobility and extensive coverage capabilities. However, the intricate nature of these networks, coupled with their heterogeneous composition and multiple communication platforms, results in several challenges. These challenges range from network load balancing, ensuring the quality of service (QoS), optimizing routing processes, and safeguarding security measures. Software-defined networking (SDN) emerges as a pivotal solution for the effective management and control of large, intricate, and diverse IoT networks [2]. The amalgamation of SDN with IoT, aptly termed SDN-IoT, signifies a transformative shift in networking paradigms [3]. The architecture of SDN-IoT can be dissected into three main planes: the sensing plane, the data plane, and the control plane, as illustrated in Fig. 1. The sensing plane predominantly consists of varied IoT devices, including but not limited to sensors and actuators. Through SDN-enabled switches that are present in the data plane, these devices relay the data they collect to the SDN-IoT gateway. Meanwhile, the control plane serves as a distinction between the SDN-IoT central controller and the fundamental network components like routers and switches. This distinction allows the controller to acquire an all-encompassing perspective of the network, facilitating key network functions such as channel resource administration and packet forwarding. It is worth noting that the SDN-empowered switches within the data plane function in strict accordance with the directives from the SDN-IoT controller, ensuring precise forwarding of IoT data clusters.

OpenFlow [4] is a traditional SDN routing protocol commonly used for facilitating communication between the data plane and the control plane through the Northbound Interface (NBI) within the SDN-IoT environment. The OpenFlow protocol enables the SDN centralized controller to comprehensively manage the network, and dynamically seek out and select routing paths. SDN's key characteristics, such as programmability, global network perspective, and centralized control, empower it to effectively govern network flows and behaviors in real-time [5]. However, as the number of IoT devices grows and network traffic escalates within the SDN-IoT landscape, the demand for Quality of Service (QoS), Quality of Experience (QoE), and secure routing have intensified [6]. Upon delving into the literature on SDN routing, two primary limitations have been identified in SDN and OpenFlow-based routing: 1) Security concerns, and 2) QoS and QoE issues. The SDN OpenFlow routing protocol is susceptible to various security breaches, including Man-in-the-Middle attacks, Distributed Denial of Service (DDoS) attacks, and Bitrate Oscillation Attacks [7]. Consequently, SDN-IoT routing also remains exposed to severe detrimental attacks [8]. In the SDN-IoT architecture, the default routing protocol employed within the SDN controller is the Open Shortest Path First (OSPF), which operates based on the shortest routing path strategy. In the SDN-IoT context, the performance of the routing protocol degrades when security attacks occur or the volume of packet forwarding requests increases [9]. This degradation results in network congestion, packet loss, latency, jitter, and other issues. The acronyms used in this article are shown in Table 1.

In this article, we tried our best to address the above-mentioned challenges. The proposed DRL-based novel strategy addresses the secure routing optimization in SDN-IoT and improves QoS and QoE. The proposed technique was tested through extensive experimentation and simulation over three important network metrics (throughput, latency, and probability of avoiding malicious nodes).

TABLE 1. Acronyms used in paper.

S.No	Acronyms	Explanation
1	AE	Autoencoder
2	AI	Artificial Intelligence
3	CNN	Convolutional Neural Network
4	DBN	Deep Belief Network
5	DDPG	Deep Deterministic Policy Gradient
6	DDoS	Distributed Denial of Service
7	DL	Deep Learning
8	DNN	Deep Neural Network
9	DRL	Deep Reinforcement Learning
10	EDA	Eavesdropping Attack
11	GAN	Generative Adversarial Network
12	IoT	Internet of Things
13	LSTM	Long Short Term Memory
14	MIoT	Medical Internet of Things
15	MIMA	Man-in-the-Middle Attack
16	ML	Machine Learning
17	NBI	North Bound Interface
18	OSPF	Open Shortest Path First
19	PTA	Physical Tampering Attack
20	QoE	Quality of Experience
21	QoS	Quality of Service
22	SDN	Software Defined Network
23	SDN-IoT	Software Defined Internet of Things
24	SBI	South Bound Interface
25	SVM	Support Vector Machine

The main contributions of this article are mentioned below.

- A novel deep reinforcement learning (DRL) technique known as DQQS model is designed to optimize secure routing while enhancing Quality of Service (QoS) and Quality of Experience (QoE) within SDN-IoT environments.
- The DQQS model specifically addresses security concerns in the sensing and data layers of SDN-IoT networks. It aims to establish secure routing by avoiding malicious or compromised nodes, thereby maintaining high standards of QoS and QoE. This approach is identified as the first of its kind in using DRL for tackling security challenges in SDN-IoT routing while upholding QoS and QoE.
- The model demonstrates superior performance in providing rapid and secure routing in SDN-IoT settings, effectively countering security threats at both the sensing and data layers. Comparative tests show that DQQS outperform the state-of-the art routing protocols (OSPF, L-L, Sailfish, RL-Routing) across all QoS and QoE metrics. Additionally, it exhibits high accuracy in attack detection and classification on NSL-KDD and IoT

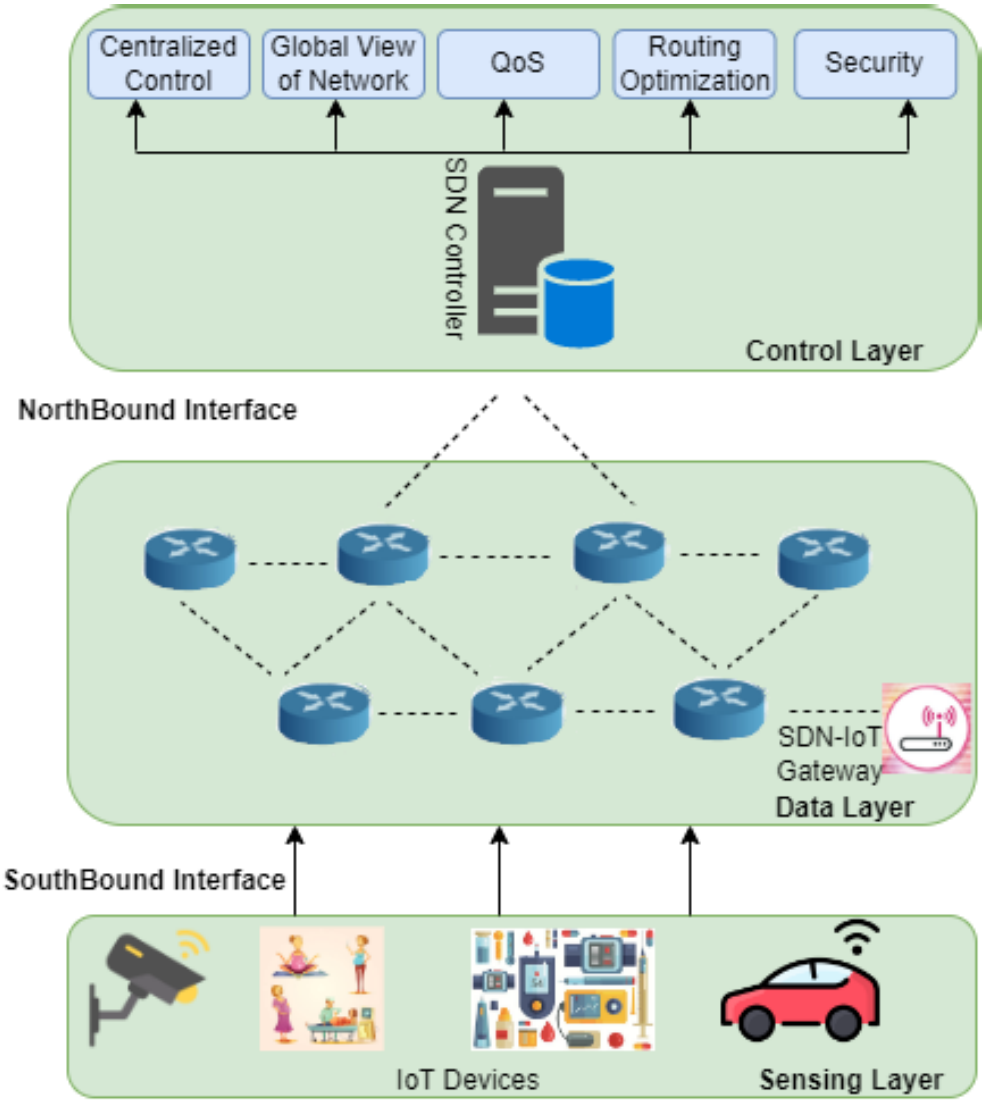

FIGURE 1. Illustration of SDN-IoT framework depicting the control plane, data plane, and sensing plane.

datasets, surpassing naive deep learning models like LSTM and CNN.

The rest of the paper is organized as follows:

Section II outlines the related work of the proposed model. Section III presents the problem statement, while Section IV elucidates the workings and architecture of our proposed model. Section V, assesses the performance of the proposed model, and Section VI describes the results discussions, limitations, and future research directions. Finally, Section VII concludes the article.

II. RELATED WORK

This section discusses the relevant literature concerning Deep Learning (DL) and Machine Learning (ML)-based routing optimization for SDN-IoT, aimed at enhancing SDN-IoT's Quality of Service (QoS) and Quality of Experience (QoE). In SDN-IoT, network traffic routing plays a crucial role in ensuring efficient data transmission, minimizing latency, enhancing security, and optimizing resource utilization. In [10], the authors introduced a DRL-based routing optimization technique in SDN. Their DRL agent learns the interdependencies between network traffic load and network performance, selecting optimal sets of link weights to minimize latency and packet loss rates. Experimental results of the proposed technique demonstrate the superiority of their proposed approach over traditional hop count routing across various network topologies. In [11], the authors proposed both centralized and distributed DRL-based routing techniques. Through extensive experimentation, they determined that the centralized approach excels in managing dynamic network traffic due to its remarkable reconvergence capabilities. In [12], the authors put forward a routing optimization algorithm founded on Quality of Service (QoS) parameters, utilizing Deep Reinforcement Learning for

SDN-based Data Center Networks. Their proposed technique achieved 21% higher average throughput and 17% lower average delay compared to Dijkstra's algorithm. In [13], the authors presented DeepMonitor, a framework designed for SDN-based IoT networks, facilitating fine-grained traffic analysis for distinct IoT traffic types at network edges. Simulation outcomes highlight the efficacy of this technique in addressing overflow issues within flow tables at edge nodes. The average number of match-fields in a flow rule,as achieved by this proposed technique, witnessed an approximate increase of 37% and 41.9%.

In [14], the authors presented a routing optimization technique in the context of SDN-IoT, leveraging traffic-aware Quality of Service (QoS). Simulations reveal that this approach effectively reduces latency and mitigates flows that impact QoS. In [15], the authors introduced an SDN-IoT routing technique grounded in application-aware QoS principles. This method addresses QoS prerequisites of high-priority applications while adapting to the current network status to optimize routing paths. Simulation results demonstrate the superiority of this technique over MINA in terms of metrics such as jitter, packet loss rate, and latency. As described in [16], a Deep Learning (DL)–based model for predicting traffic load and managing channel assignments is outlined. The utilization of a Convolutional Neural Network (CNN) model has yielded promising results. The work detailed in [3] outlines an SDN technique tailored for IoT environments, dynamically tailoring distinct quality levels to diverse IoT tasks within highly heterogeneous wireless networking scenarios. In [17], a technique aimed at controlling congestion in SDN is proposed. This involves dynamic traffic splitting through the analysis of statistics gathered by each network switch. Simulation results showcase the success of this approach in reducing packet loss rates and alleviating overutilized links. Elaborated upon in [18], the authors devised a deep reinforcement learning-based smart routing algorithm to render distributed computing and communication infrastructure thoroughly feasible. This is achieved while simultaneously adhering to latency constraints imposed by service requests from a diverse audience. Highlighted in [19], the authors introduced a pioneering deep learning-based algorithm for predicting Traffic Load (TL), forecasting future TL, and anticipating network congestion. Simulation results robustly demonstrate the superiority of our proposal over conventional channel assignment algorithms.

In their work [20], introduced a novel deep learning strategy, DLICA, designed for efficient channel assignment in SDN-IoT environments to alleviate network congestion. Reference [21] developed a comprehensive framework for joint multi-channel reassignment and traffic control in the core backbone network of SDN-IoT. Their primary goal was to maximize throughput while minimizing packet loss and time delays. Reference [22] asserted that SDN has effectively addressed challenges in IoT, such as complexity and heterogeneity. Reference [23] presented a machine-learning model with SDN-enabled security for predicting network resource consumption and enhancing sensor data delivery. Furthermore, they introduced a cost-effective centralized SDN architecture to mitigate network threats among deployed sensors. Reference [24] applied a combination of the Balancing Module (BM), Spider Monkey Optimization (SMO), and Crow Search Algorithm (CSA) to improve multi-path selection efficiency in SDN. The Balancing Module incorporates Gaussian distribution principles to achieve equilibrium between exploration and exploitation. Such a balance aids in evading the pitfalls of local optima and improving convergence speeds. In the research presented by [25], deep reinforcement learning techniques were applied to resource scheduling within the control plane of SDN. The advanced algorithm was developed with the aim of enhancing resource distribution, and it successfully exhibited superior network performance. Specifically, it showcased improvements in Quality of Service (QoS) metrics, notably in delay and throughput, when compared to both random and round-robin strategies.

Furthermore, [26] proposed a multi-agent reinforcement learning framework in SDN-IoT to detect and mitigate DDoS attacks and effectively manage route flash crowd events without impacting benign network traffic. Another noteworthy contribution was made by [27], where they presented an SDN-centric efficient clustering mechanism for IoT, leveraging the Improved Sailfish Optimization (ISFO) algorithm. This innovative design promotes efficient clustering of IoT devices and is adeptly incorporated into the SDN controller to streamline the management of Cluster Head (CH) nodes. The summary of the literature work is presented in Table 2.

III. PROBLEM STATEMENT

This article focuses on addressing the challenge of efficient and secure routing optimization in SDN-IoT environment. The sensing and data layers of the SDN-IoT setup are vulnerable to security attacks. Attackers can gain control of the sensing layer (sensors) and data layer (switches) devices and manipulate these devices by generating fake traffic, disabling SDN-enabled switches for a specific period, stealing sensitive information, and deleting flow entries of switches to degrade network performance. This section describes the problem definition and an overview of the SDN-IoT network model. The symbols employed in the network model are shown in Table 3.

A. PROBLEM DEFINITION AND SDN-IoT NETWORKING MODEL

First and foremost, we present some fundamental definitions to facilitate comprehension of the architecture and packet flow processes within the SDN-IoT network. Here, $\mathbb{N}(\mathbb{S}, \mathbb{L})$ signifies the network configuration consisting of $\mathbb{S}$ SDN-enabled switches and $\mathbb{L}$ undirected links

TABLE 2. A table describes the state-of-the-art deep learning (DL) and machine learning (ML)-based routing techniques.

Reference	Purpose	Methodology	Advantages	Limitations
[10]	Routing optimization in SDN	DRL-based routing optimization	Superiority over traditional hop count routing	Not secure due to sensing layer (sensors)
[11]	Centralized and distributed DRL-based routing techniques	Centralized and Distributed DRL-based routing	Centralized approach excels in dynamic network traffic	Not secure due to data layer (switches) devices
[12]	Routing optimization based on QoS parameters for SDN-based Data Center Networks	DRL for SDN	21% higher throughput and 17% lower delay than Dijkstra's	Complex DL architecture
[13]	Fine-grained traffic analysis for IoT networks	DeepMonitor framework	Addresses overflow issues at edge nodes	Not works in different situations
[14]	Routing technique based on application-aware QoS principles	Application-aware QoS routing	Superiority over MINA in jitter, packet loss rate, and latency	impacting quality of experience
[15]	Predicting traffic load and managing channel assignments	CNN model	Promising results	Security threats
[16]	Controlling congestion in SDN	Dynamic traffic splitting	Reduction in packet loss rates and alleviation of overused links	impacting quality of experience
[3]	Smart routing algorithm for distributed computing and communication infrastructure	Deep reinforcement learning-based algorithm	Adherence to latency constraints of diverse service requests	Computational expensive
[17]	Predicting Traffic Load (TL) and anticipating network congestion	Deep learning-based algorithm	Superiority over conventional channel assignment algorithms	Insecured
[18]	Efficient channel assignment in SDN-IoT environments	DLICA strategy	Alleviate network congestion	impacting quality of experience
[19]	Predicting network resource consumption and enhancing sensor data delivery	Machine-learning model with SDN-enabled security	Mitigate network threats among sensors	Computational expensive
[20]	Improve multi-path selection efficiency in SDN	BM, SMO, and CSA	Escape from local optima and enhanced convergence rates	Not generalized solution
[21]	Resource scheduling in SDN's control plane	Deep reinforcement learning	Improved network performance compared to Random and Round Robin	impacting quality of experience
[22]	Detect and mitigate DDoS attacks in SDN-IoT	Multi-agent reinforcement learning framework	Manage DDoS without impacting benign traffic	Not Optimize solution
[23]	Efficient clustering scheme for IoT using ISFO algorithm	SDN-based clustering using ISFO	Effective management of CH nodes	Not secure due to sensing layer (sensors)

interconnecting these switches. Within the SDN-IoT network environment, each SDN-enabled switch, denoted as s_i, is equipped with a flow table, and overall network control and management are executed by the SDN controller $\mathbb{C}$. The SDN controller furnishes packet forwarding rules to the switches, enabling the transmission and reception of data among neighboring switches. The operation of the conventional routing protocol is depicted in Fig. 2.

In a conventional routing protocol, the sensing layer dispatches a message to an SDN-enabled switch situated in the data layer. Upon receiving the message, the switch consults its flow table for a relevant flow entry. If a flow entry is present, the switch acts in accordance with the specified rule. However, in the absence of a corresponding flow entry, the message is forwarded to the SDN controller to solicit future instructions concerning this flow entry, as illustrated in Figure 2. Notably, a significant concern arises due to the lack of security and the vulnerability inherent in the conventional routing protocol when faced with sophisticated and contemporary security threats. In the presence of malicious entities within the network, the performance of the SDN-IoT network can degrade in terms of Quality of Service

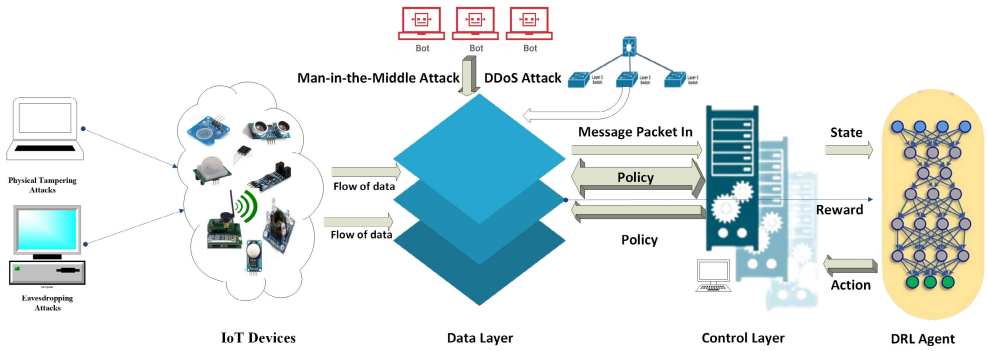

FIGURE 2. Workflow of conventional routing protocol and proposed DQQS technique.

TABLE 3. Symbols used in network model.

Symbols	Explanation
$\mathbb{S}$	Represents the number of SDN-enabled switches
$\mathbb{L}$	Represents the links between SDN-enabled switches
$\mathbb{F}_{s_i}$	Represents maximum flows in any switch $s_i, i = 1, 2, 3...$
$\mathbb{F}_{s_i}(t)$	Represent how many flow entries can be accommodated by switch s_i at any time t
$\mathbb{R}_{p(s_s,s_d)}$	Routing policy request from source switch s_s to the destination switch s_d
s_s	Represent the source switch
s_d	Represent the destination switch
$\mathbb{RS}$	Set of relay switch
$\mathbb{N}(\mathbb{S}, \mathbb{L})$	Network with $\mathbb{S}$ number of SDN-enabled switches and $\mathbb{L}$ links between switches

(QoS), leading to issues such as increased congestion and reduced throughput, as well as adversely impacting Quality of Experience (QoE) by introducing elevated levels of jitter and latency.

It holds true that the advent of Software-Defined Networking (SDN), driven by its remarkable attributes including programmability, flexibility, and global view of the network, has led researchers to tackle the aforementioned challenges by leveraging SDN's capabilities. However, SDN-IoT is not intelligent enough to efficiently deal with the heterogeneous, dynamic, and unpredictable traffic flows and state-of-the-art security challenges to optimize the routing optimization. Fortunately, DRL can be implemented in SDN-IoT to make the system more intelligent, efficient, and capable enough to optimize routing to enhance QoS and QoE while dealing with the latest security threats. In this article, the proposed technique based on DRL is called DQQS. In the proposed technique, DRL agent learns while interacting with the environment. The agent engages in an iterative process of learning through interaction with the environment, receiving rewards for its actions. Over time, the DRL agent progressively refines its performance by assimilating knowledge gained from its environment.

B. CONSIDERING SECURITY THREATS

This article addresses two categories of threats: sensing layer threats and data layer threats. We assert that this article is distinctive in its nature, being one of the first to consider both types of threats while simultaneously upholding Quality of Service (QoS) and Quality of Experience (QoE) in the SDN-IoT environment.

In SDN-IoT research literature, multiple efforts have been made to confront this challenge. In [28], authors have introduced a model termed DQSP, designed to achieve secure routing optimization while preserving QoS within the SDN-IoT environment. However, a notable drawback of this proposed technique pertains to the absence of consideration for sensing layer threats. Indeed, if accurate and reliable data is not obtained from the sensing layer, the feasibility of making informed and intelligent decisions based on erroneous information becomes questionable. Therefore, it is of utmost importance to address threats posed by both the sensing and data layers to effectively realize efficient and secure routing.

1) SENSING PLANE ATTACKS

In the SDN-IoT environment, IoT devices located in the sensing plane gather ambient information and relay it to the data plane. Within the sensing plane, numerous IoT devices are present, rendering them susceptible to various forms of attacks. Adversaries could exploit vulnerabilities in IoT devices, thereby initiating diverse network attacks aimed at compromising network performance in terms of Quality of Service (QoS) and Quality of Experience (QoE). Examples of potential attacks that could be executed within the sensing plane encompass eavesdropping attacks, physical tampering attacks, privilege escalation attacks, malicious node injection attacks, and sybil attacks, among others.

- Eavesdropping Attack (EDA): EDA takes place when intruders or hackers intercept, delete, or modify vital information from the sensing plane while it is being transmitted to the data plane. For instance, in the context of Medical Internet of Things (MIoT), devices gather sensitive patient information that must remain confidential. Attackers could breach these security measures, manipulate the sensitive data, and then transmit it to the data plane. Consequently, such tampered data has the

potential to undermine both the Quality of Service (QoS) aspects of SDN-IoT, specifically trust and accuracy, as well as the Quality of Experience (QoE), manifesting as jitter.

- Physical Tampering Attack (PTA): PTA disrupts network performance in terms of routing, Quality of Service (QoS), and Quality of Experience (QoE). Within PTA, one or more sensor nodes fall under the control of attackers, which can then be manipulated to generate fabricated, contradictory, and excessive information. The objective of such manipulation is to degrade network performance.

2) DATA PLANE ATTACKS

After the data plane has received genuine and accurate information from the sensing plane, frequent communication commences between the data plane and the control plane. In the SDN-IoT domain, the control plane formulates rules and routing policies that are subsequently implemented in the data plane. Consequently, this recurrent communication between the data and control planes renders the system susceptible to a variety of attacks. Examples of attacks targeting the data plane include the Man-in-the-Middle (MITM) attack, Distributed Denial of Service (DDoS) attack, and Bitrate Oscillation attack, among others.

- Man-in-the-Middle Attack (MIMA): In MIMA, attackers gain control over one or more participating nodes (such as switches or sensors) within the network and exploit these nodes to degrade network performance. For instance, a compromised node within the network may intentionally discard critical packets at specific times. Additionally, attacked nodes may intermittently remain inactive, leading to network congestion. In the context of the SDN-IoT environment, the MIMA attack deliberately deletes flow entries of the attacked SDN-enabled switch, thereby increasing the likelihood of packet loss.
- Distributed Denial of Service Attack (DDoS): Through the utilization of a DDoS attack, intruders disrupt network performance, resulting in issues such as network congestion, packet loss, and latency. In this attack, assailants harness compromised nodes to inundate the network with a barrage of meaningless packets, effectively obstructing regular network communication. Within the SDN-IoT environment, a malicious node forwards flow requests to an SDN-enabled switch. However, if the switch's flow table lacks the corresponding entry for the received flow request, this situation can precipitate a deterioration in network performance.

IV. THE PROPOSED MODEL ARCHITECTURE

In this section, we will discuss the architecture and algorithmic detail of our proposed model. Generally, the proposed model is divided into four layers. Each layer has its own responsibilities and functions. The working of our proposed scheme is shown in a Fig. 3.

A. FOUR LAYERED ARCHITECTURE

1) SENSING LAYER

The primary and foundational layer of the proposed model is the sensing layer. This layer comprises a multitude of IoT devices in the form of sensors and actuators. Its primary duty is to collect data and subsequently transfer it to the succeeding layer. Given the significant diversity and heterogeneity inherent in the sensors composing the sensing layer, it becomes susceptible to a range of security attacks. Ensuring the precise and reliable collection of data, followed by its seamless transfer to the subsequent layer, stands as a primary objective of our proposed model.

2) DATA LAYER

The second layer of the proposed model is known as the data layer. This stratum encompasses various networking devices, including SDN-enabled switches, routers, and more. The primary function of this layer is to facilitate data movement among switches based on the guidance and directives emanating from the control layer. Consequently, the data layer is often referred to as the 'dumb layer.' The communication between the data layer and the control layer is facilitated through the South Bound Interface (SBI).

3) CONTROL LAYER

The third and perhaps the most pivotal layer within our proposed model is recognized as the control or controller layer. This layer functions as the system's central hub and brain. Endowed with a comprehensive network-wide perspective, the control layer oversees all state alterations transpiring within the network. Its principal role lies in orchestrating network functionality, encompassing the adaptation of routing strategies in response to instances of network congestion, node failures, and link failures.

4) DRL LAYER

To imbue the network with heightened intelligence, efficiency, security, adaptability, and responsiveness to the dynamic and unpredictable flow of network traffic, we have integrated a DRL layer into our proposed model. Interaction between the DRL layer and the controller layer takes place through the North Bound Interface (NBI). Owing to the global network view inherent in the controller layer, the DRL layer acquires comprehensive awareness of the underlying network's status. This enables the DRL layer to formulate optimized routing policies by analyzing the rewards associated with multiple policies and making adjustments to diverse performance parameters. Furthermore, the extensive training undergone by the DRL agent using historical network data empowers it to fashion an optimal routing policy for real-time network scenarios.

B. DETAILED DESCRIPTION OF THE PROPOSED MODEL

In this section, we elucidate the intricate workings and algorithms underpinning the proposed technique. As previously

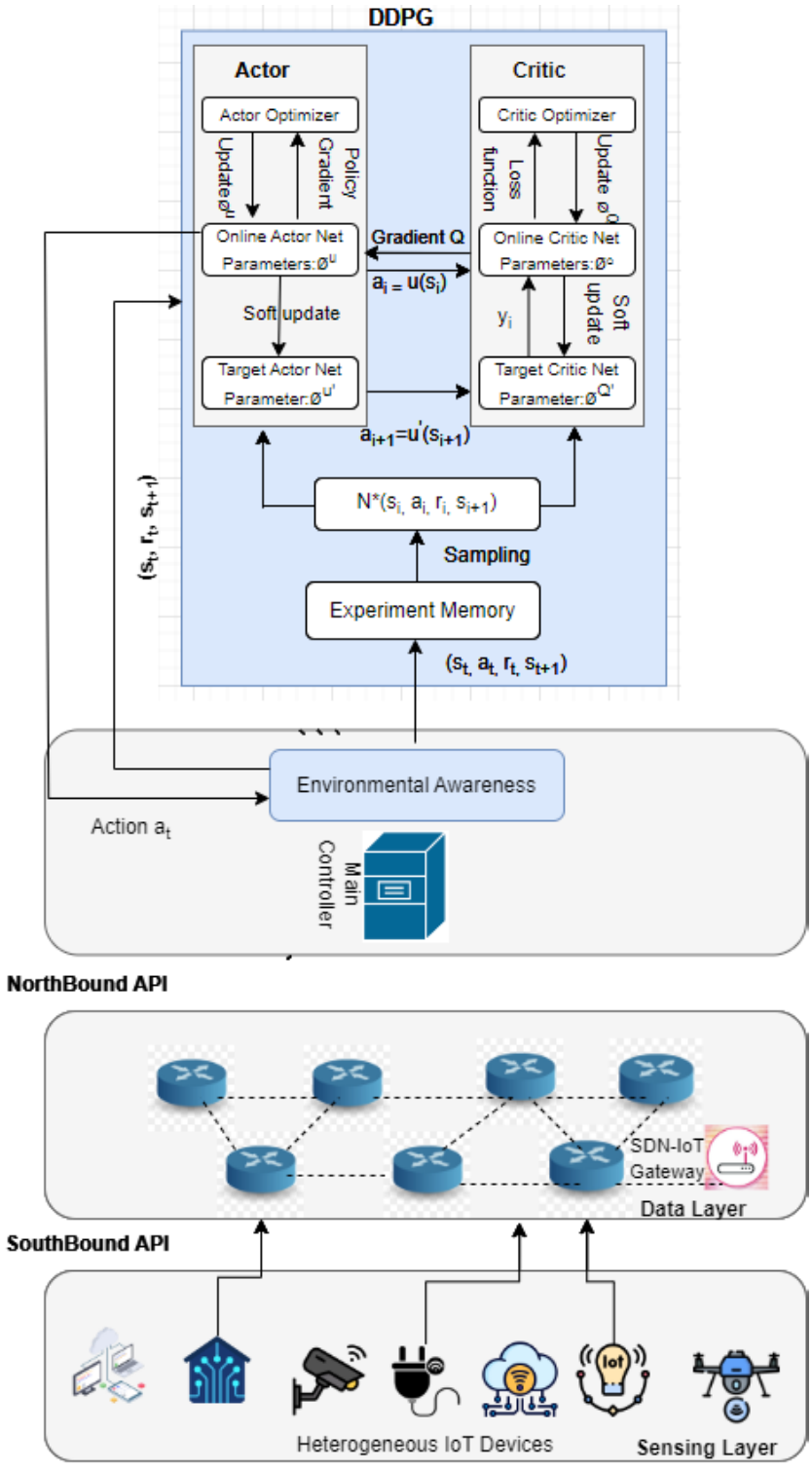

FIGURE 3. The Proposed DQQS Model's Structure: Sensing Layer, Data Link Layer, Controller Layer, and DDPG Layer.

mentioned, conventional routing methods lack the aptitude to effectively handle the ever-changing and unpredictable nature of real-time SDN-IoT networks. Compounding this, network attacks exacerbate the predictability of network traffic and degrade both Quality of Service (QoS) and Quality of Experience (QoE). The impact of network attacks extends to the capabilities of SDN-enabled switches and the quality of network links. To address these challenges, we have

incorporated a DRL agent into our proposed technique to optimize routing while ensuring security. The subsequent section elaborates on the three components of the DRL agent: state, action, and reward.

1) STATE

We consider routing as taking place within a unit of time, with each unit equivalent to one time step. Consequently, the total routing time between the source switch s_s and the destination switch s_d is denoted as $\mathbb{T}$. The DRL agent evaluates the reward for a transmission task within a single unit time slot, which encompasses determining the time required to select the subsequent SDN-enabled hop switch and transmit data to it. The DRL agent leverages three factors in the reward calculation: 1) message input frequency $\lambda_s(t)$, 2) message holding rate of the flow table $\rho_s(t)$, and 3) channel holding rate between the SDN controller and the switch $\sigma_s(t)$. For a given switch node $s_i, i = (1, 2, 3, \ldots)$ during a unit time $t, t = (1, 2, 3, \ldots)$, these three factors collectively define the actual state of the system. Upon incorporating these three factors:

$$\begin{aligned} s(t) = [&\lambda_{s_1}(t), \lambda_{s_2}(t), \lambda_{s_3}(t), \ldots .\lambda_{s_N}(t), \\ &\rho_{s_1}(t), \rho_{s_2}(t), \rho_{s_3}(t), \ldots .\rho_{s_N}(t), \\ &\sigma_{s_1}(t), \sigma_{s_2}(t), \sigma_{s_3}(t), \ldots .\sigma_{s_N}(t)] \end{aligned} \tag{1}$$

Equation (1), $\rho_{s_i} = \frac{\mathbb{F}s_i(t)}{\mathbb{F}s_i}$ signifies the capacity of the flow table in the SDN-enabled switch s_i. In this context, $\mathbb{F}s_i(t)$ represents the number of flow entries currently accommodated within switch s_i at any given time (t), while $\mathbb{F}s_i$ indicates the maximum number of flows that can be supported by switch s_i.

2) ACTION

Smart and intelligent routing relies heavily on the selection of the next hop (switch). In the action stage, the primary responsibility of the DRL agent is to identify and choose the next available switch for data transfer. This action stage is depicted by Equation (2).

$$\mathbb{P}(t) = \mathbb{P}^{prest}_{s_1}(t), \mathbb{P}^{prest}_{s_2}(t), \mathbb{P}^{prest}_{s_3}(t), \ldots .\mathbb{P}^{prest}_{s_N}(t) \tag{2}$$

As shown in a Equation (2), $\mathbb{P}^{prest}_{s_i}(t)$ can be defined in the vector form $\mathbb{P}^{prest}_{s_i}(t) = \{\mathbb{P}^{prest}_{s_i,s_j}(t) | J \in \{1, 2, 3 \ldots N\}, J \neq I\}$. $\mathbb{P}^{prest}_{s_i,s_j}(t)$ shows the relation between the switch s_i and the switch s_j. Every element of $\mathbb{P}^{prest}_{s_i,s_j}(t) \in [0, 1]$, where $\mathbb{P}^{prest}_{s_i,s_j}(t) = 0$ means there is no connection between switch s_i and switch s_j at any unit time t and $\mathbb{P}^{prest}_{s_i,s_j}(t) \in [0, 1]$ shows the switch s_j weight that which is selected as next hop of switch s_i.

3) REWARD

In Deep Reinforcement Learning (DRL), the efficiency and effectiveness of the agent's actions are assessed through the reward function. Consequently, the reward associated with each action varies. Within our proposed technique, key parameters defining the reward function include switch processing delay, switch forwarding delay, switch queuing delay, switch packet loss rate, and flow table status. QoS and QoE-related parameters impacting the reward function encompass propagation delay, jitter, link packet loss rate, and latency, respectively. The reward function is articulated in Equation (4).

$$Attack(t) = \alpha\{RW^{attack(s\dashrightarrow d)}_{IoT_i}(t) + RW^{attack(d\dashrightarrow c)}_{s_i}(t)\} \tag{3}$$

$$\begin{aligned} RW(t) = \frac{1}{|Trans|} \sum_{i \in Trans} &[Attack(t) + \beta RW^{QoS}_{s_i}(t) \\ &+ \sigma RW^{QoE}_{s_i}(t) \end{aligned} \tag{4}$$

Equation (3) outlines the attacks on IoT device IoT_i and switch s_i during any given unit time t. The term $RW^{attack(s\dashrightarrow d)}IoT_i(t) + RW^{attack(d\dashrightarrow c)}s_i(t)$ accounts for the potential occurrence of two types of attacks, aiming to undermine QoS and QoE. One attack could arise from the sensing layer to the data layer, denoted as $s \dashrightarrow d$. The other attack could stem from the data layer to the control layer, denoted as $d \dashrightarrow c$. We incorporate the value from Equation (3) into Equation (4).

Before delving into Equation (4), it is important to acknowledge that the transmission task within any unit time t encompasses two distinct phases. In the first phase, SDN-enabled switch s_i transfers the data, and in the subsequent phase, the data is directed to the SDN-enabled destination switch s_j via a communication link. In Equation (4), $|Trans|$ signifies the maximum number of data transmission jobs during any given unit time t. Parameters α, β, and σ are function tuning parameters, the values of which are adjusted to optimize the function, either by maximizing or minimizing it. Notably, the sum of these tuning parameters is constrained to $\alpha + \beta + \sigma = 1$. The term $Attack(t)$ denotes the attack reward affecting switch s_i and IoT_i, indicating the severity and intensity of security attacks on these entities. In essence, severe security attacks diminish the reward, while less severe attacks yield the opposite effect.

We break the Equation (3),

$$RW^{attack(s\dashrightarrow d)}_{IoT_i}(t) = -FalsInfo - StealSenInfo \tag{5}$$

$$\begin{aligned} RW^{attack(d\dashrightarrow c)}_{s_i}(t) = -DEL^{process}_{s_i} - DEL^{queue}_{s_i} \\ - DEL^{forward}_{s_i} - PLR_{s_i} + FTS_{s_i} \end{aligned} \tag{6}$$

After dissecting Equation (3), we derive Equation (5) and Equation (6). Equation (5) characterizes the impact of sensing layer attacks on IoT devices. Attacks on these devices can introduce false information into the system or compromise sensitive data. Severe attacks on the sensing layer can significantly disrupt system performance in terms of QoS and QoE, thus leading to a reduction in the reward value. Similarly, Equation (6) encapsulates the effect of an attack on the data layer, targeting an SDN-enabled switch s_i. Potential consequences of such a data layer attack on switch s_i include switch processing delay, queuing delay, forwarding delay, and packet loss rate.

Now, QoS and QoE rewards are defined in Equation (7) and Equation (8) respectively.

$$RW_{s_i}^{QoS}(t) = \sum_{j\in\{1,2,3,4...N\},j\neq i} [\mathbb{P}_{s_i,s_j}^{prest}(t) - DEL_{s_i,s_j}^{propagate} - LPLR_{s_i,s_j} - Jitter_{s_i,s_j}] \quad (7)$$

$$RW_{s_i}^{QoE}(t) = \sum_{j\in\{1,2,3,4...N\},j\neq i} [\mathbb{P}_{s_i,s_j}^{prest}(t) + HighLatency_{s_i,s_j}] \quad (8)$$

Equation (7), $DEL_{s_i,s_j}^{propagate} - LPLR_{s_i,s_j} - Jitter_{s_i,s_j}$ shows that propagation delay $DEL_{s_i,s_j}^{propagate}$, Link packet loss rate $LPLR_{s_i,s_j}$, and jitter $Jitter_{s_i,s_j}$ effect the QoS reward. Similarly, Equation (8) $HighLatency_{s_i,s_j}$ shows high latency effects QoE reward.

C. INTEGRATION OF THE PROPOSED TECHNIQUE WITH DDPG

In this section, we discuss our proposed technique and its application for efficient and secure routing. The integration of the proposed technique with DDPG [29] and the global view of the SDN-IoT environment enables the establishment of secure routing mechanisms. This is achieved by defining new states, actions, and rewards.

1) DDPG

DDPG is one of the most commonly used DRL techniques. The fourth layer of our proposed model (DRL agent) utilizes DDPG, as illustrated in Fig. 3. It employs the actor-critic model of DRL, wherein the actor comprises the actor network and the target actor network denoted as $\tau(s|\theta^{\tau})$ and τ', respectively. Similarly, the critic includes the main critic network and the target critic network $\eta(s,a)$ and η', respectively. The structure of both the main network and the target network is the same. The current policy is determined by the actor-network $\tau(s|\theta^{\tau})$, which maps states to actions. The critic network $\eta(s,a)$ employs the Bellman equation for learning, and typically, the output of the actor serves as the input for the critic.

2) SAMPLE COLLECTION

The exploration policy is employed to generate samples from the environment, and sample records $(s(t), a(t), r(t), s(t+1))$ are stored in a replay buffer B following the DDPG mechanism. Here, $s(t)$ and $a(t)$ denote the initial state and policy network output, respectively. Additionally, the action $a(t)$ is executed on the state $s(t)$, resulting in the corresponding rewards $r(t)$ and the subsequent state $s(t+1)$. The procedure for the sample collection process in SDN-IoT is illustrated in Fig. 3 and also outlined in Algorithm 1.

3) TRAINING

The training process is represented in an Equation (9)

$$Train(\theta) = \frac{1}{M}\sum_{t}(y(t) - \eta(s(t), a(t)|\theta^{\eta}))^2 \quad (9)$$

Deep Q-learning is used to train the critic net. As shown in equation (9), the actor network takes the state $s(t)$ as input and provides the action $a(t)$ as an output. Then, the critic network takes the action $a(t)$ as input and provides $\eta(s(t), a(t)|\theta^{\eta})$ as an output.

$$\eta(s(t), a(t)) \longleftarrow \eta(s(t), a(t)) + \zeta(r(s(t), a(t))) + \omega_{a(t+1)}\eta(s(t+1), a(t+1)) - \eta(s(t), a(t)) \quad (10)$$

In Equation (9), target Q-value $y(t)$ is defined as follow

$$y(t) = r(t) + \omega\eta'(s(t+1), \tau'(s(t+1)|\theta^{\tau'})|\theta^{\eta'}) \quad (11)$$

As shown in Equation (11), the summation of reward and Q-value $r(t)+\omega\eta'(s(t+1))$ gives the target value. The input state $s(t+1)$ gives the output action $\tau'(s(t+1)|\theta^{\tau'})$. By using the policy gradient technique, the gradient of the actor-network is given as:

$$\frac{\delta J(\theta^{\tau})}{\delta\theta^{\tau}} = Z_s[\frac{\delta\eta(s,a|\theta^{\eta})}{\delta a}\frac{\delta\tau'^{(s|\theta^{\tau})}}{\delta\theta^{\tau}}] \quad (12)$$

Parameters updation process is explained in Equation (13)

$$\nabla_{\theta^{\tau}} J \approx \frac{1}{M}\sum_t \nabla_a \eta(s,a|\theta^{\eta})|_{s=s(t),a=\tau(s(t))} \nabla_{\theta^{\tau}} \tau(s|\theta^{\tau})|_{s(t)} \quad (13)$$

Against the same state $s(t)$, main actor provides multiple actions. Hence, different actions can be used as an input for main critic to achieve different Q values. Equation (14) and Equation (15) update the target network

$$\theta^{\eta'} \longleftarrow \phi\theta^{\eta} + (1-\phi)\theta^{\eta'} \quad (14)$$

$$\theta^{\tau'} \longleftarrow \phi\theta^{\tau} + (1-\phi)\theta^{\tau'} \quad (15)$$

The detailed training mechanism of the proposed technique is elaborated in Algorithm 2.

Algorithm 1 Data Sample Collection Process From the Underlying Environment

```
Initialization of buffer B
Initialization of both main critic and main actor networks η(s, a|θ^η), τ(s|θ^τ) along with their weights θ^η, θ^τ respectively.
Initialization of target critic network η' and τ' along with their weights θ^η' ⟵ θ^η and θ^τ' ⟵ θ^τ
for epic = 1 to Trans do
    Initial state s(t)
    for t=1 to T do
        action selection a(t) = τ(s(t)|θ^τ)
        action execution a(t) and observe reward r(t)
        new state observation s(t+1)
        Transition in buffer B (s(t), a(t), r(t), s(t+1))
    end for
end for
```

Algorithm 2 Proposed DQQS Model Training

```
for epic = 1 to Trans do
    for t=1 to T do
        Transition from buffer B (s(t), a(t), r(t), s(t+1))
        The training process is represented in an Equation (9)
        Target Q-value y(t) is defined in Equation (11)
        Parameters updation process is explained in Equation (13)
        Updating the target network by using Equation (14) and Equation (15)
    end for
end for
```

V. EXPERIMENTATION AND THE PROPOSED TECHNIQUE EFFICACY EVALUATION

A. EXPERIMENTAL PROTOCOL

We established the SDN-IoT environment using the Mininet 2.3.0 simulation tool [30]. Subsequently, we deployed a Deep Learning model within the ONOS SDN controller, utilizing the Python-based TensorFlow framework. Our environment is equipped with the latest version of TensorFlow, v2.12.0. The simulations were carried out on a laptop with an 8th-generation Intel Core i9 processor, 16 GB of RAM, and a 1TB hard disk. The proposed model serves a dual purpose: attack classification and routing optimization. For attack classification, we employed two datasets, namely the NSL-KDD dataset and the IoT dataset. We compared the performance of our model with that of naive Deep Learning models, using metrics such as Accuracy, Precision, Recall, and F1-score. In the routing optimization task, we evaluated the proposed technique using three key metrics: throughput, latency, and the probability of avoiding malicious nodes. We also conducted a comparative analysis against the four state-of-the art routing algorithms named Open Shortest Path First (OSPF) [31], Least Loaded (LL) routing algorithm [32], RL-Routing protocol [33], and sailfish optimization algorithm [27]. To achieve this, we created a simulated environment comprising 50 sensor nodes and 20 SDN-enabled switches. Within these 20 switches, we designated two nodes as the source node (s_i) and the destination node (s_j), respectively. It is important to note that certain sensor nodes and SDN-enabled switches are susceptible to attacks, potentially jeopardizing network performance and effectiveness. As discussed in previous sections, our proposed four-layer model is vulnerable to security attacks and threats, particularly in two of its layers: the sensing layer and the data layer. The parameters used in the proposed model are given in Table 4.

TABLE 4. List of parameters used in the proposed DRL model.

Parameters	Value	Description
ω	0.9	The discount factor
ζ	0.1	Learning rate
ϕ	0.01	The soft target updating parameter
(α, β, σ)	(.4,.4,.2),(.4,.3,.3)	Different configurations of reward parameters

B. COMPUTATIONAL COMPLEXITY

We have employed a DRL model within our proposed framework to achieve both security and optimized routing in the SDN-IoT environment. Measuring the computational complexity of our model presents a considerable challenge, as it relies on multiple factors, including network architecture complexity, state and action spaces, and training iterations. To estimate the computational complexity, we employed two methods: Theoretical Analysis and Scalability Testing. These techniques collectively demonstrate that the computational complexity of our proposed model, from the data pre-processing stage to the optimized routing policy, is notably superior to OSPF, L-L Routing, and Sailfish Routing in the context of a heterogeneous and dynamic environment. The proposed technique takes 230 seconds to perform routing optimization in an SDN-IoT environment, whereas OSPF, L-L Routing, and Sailfish Routing require 350 seconds, 430 seconds, and 310 seconds respectively.

C. DESCRIPTION OF DATA SETS

The two datasets that we are going to describe are considered as a benchmark in the domain of network security and SDN-IoT security.

1) IoT DATA SET

IoT-23 [34] is a benchmark dataset of IoT traffic collected from heterogeneous IoT devices. The proposed technique will be tested and evaluated on this IoT dataset. Within the IoT dataset, the division of sub-data is such that 20 sub-datasets are gathered from malicious IoT devices, and 3 sub-datasets are collected from benign IoT devices. In this dataset, there are 23,145 traffic flows and four classes, where each flow belongs to one of the four classes. These four classes are: 1) Benign; 2) C and C; 3) DDoS; and 4) PortScan. Among these four classes, only 'benign' belongs to the normal class, and the other three are considered as security attacks. Each record contains 21 columns, representing various characteristics of traffic flow. The IoT dataset is configured in such a way that the DDoS class contains 14,294, benign contains 100,000, PortScan contains 122, and C and C contains 6,706.

2) NSL-KDD DATA SET

The NSL-KDD dataset is an updated version of the KDD-cup99 dataset [35]. It contains 125,973 training records and 22,544 testing records, with a total of 41 attributes. The dataset comprises two main classes: 1) Normal class; 2) Attacked Class. The Attacked class is further divided into four attack classes: 1) DoS; 2) Probing; 3) Remote to Local (R2L); and 4) User to Root (U2R). A DoS attack targets a

node by inundating it with a massive amount of traffic flows, rendering the target node dysfunctional and unable to provide an appropriate response. In a probing attack, the primary objective of the attacker is to extract important information from the target node. In an R2L attack, as the name suggests, the attacker's main objective is to gain local access through a remote device. An example of an R2L attack is retrieving a password. In a U2R attack, the attacker's primary purpose is to gain access to the root privileges of the target system. Table 5 illustrates the class distribution of the NSL-KDD dataset.

D. STATISTICAL ANALYSIS

In this subsection, we have used four metrics (Accuracy, Precision, Recall, and F1-score) to compare the threat identification and prediction capabilities of the proposed model with other DL models (LSTM, CNN, DT, SVM). For comparison and evaluation purposes, each DL model was trained 150 times and then tested on a separate test dataset. The results of each model with the best detection rate are displayed. Now, let's define each metric with their respective formulas:

- **Accuracy:** It measures the proportion of correctly classified instances (or data points) out of the total number of instances in the dataset. In other words, it quantifies how often the model's predictions or classifications match the actual labels or ground truth.
  Mathematically, accuracy is calculated as:
  $$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{16}$$
  In Equation 16, TP denotes True Positive, TN denotes true negative, FP denotes false positive, and FN denotes false negative.
- **Precision:** Precision is concerned with the proportion of true positive predictions (correctly identified positive instances) out of all instances predicted as positive, including both true positives and false positives.
  Mathematically, precision is calculated as:
  $$Precision = \frac{TP}{TP + FP} \tag{17}$$
- **Recall:** It measures the ability of the model to correctly identify all the relevant instances from the total actual positive instances in the dataset.
  Mathematically, recall is calculated as:
  $$Precision = \frac{TP}{TP + FN} \tag{18}$$
- **F1-score:** is a metric used in machine learning to provide a balance between precision and recall, especially in situations where there is an uneven class distribution (class imbalance). It is the harmonic mean of precision and recall.
  Mathematically, the F1-score is calculated as:
  $$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{19}$$

E. EXPERIMENT ANALYSIS

In this subsection, we have used four metrics (Accuracy, Precision, Recall, and F1-score) to compare the threat identification and prediction capabilities of the proposed model with other DL models (LSTM, CNN, DT, SVM). For comparison and evaluation purposes, each DL model was trained 150 times and then tested on a separate test dataset. The results of each model with the best detection rate are displayed.

1) EXPERIMENTATION ON IoT DATA SET

We conducted experiments on the IoT dataset to assess the performance of the proposed systems in IoT environments. For assessment purposes, we partitioned the dataset into training and test sets using a random 7:3 ratio within each class, resulting in 84,855 training samples and 36,367 test samples. The performance of all models was then evaluated using the previously separated test data. Figure 4 evaluates and describes the confusion matrix for the proposed framework using the IoT-23 dataset. The accuracy of attack detection stands as a vital metric, measuring the performance and effectiveness of the proposed framework. Utilizing the confusion matrix as a technique to ascertain accuracy levels proves to be highly effective. The IoT-23 dataset's confusion matrix, depicted in Fig. 4, clearly underscores the excellent performance of the proposed framework in accurately predicting the actual attack classes. Moreover, the Receiver Operating Characteristic (ROC) illustrates the degree of separation and assesses the model's efficiency in accurately classifying normal and abnormal classes. We conducted a validation of our proposed framework on the IoT-23 dataset, employing ROC analysis with and without the feature selection approach. Fig. 5 and Fig. 6 showcase the outcomes derived from the ROC analysis and accuracy of the proposed framework respectively. Likewise, Fig. 7, Fig. 8, and Fig. 9 depict the multiclassification comparison between the DL models and the proposed framework. Table 6 provides a comprehensive display of the results obtained from the proposed framework on the IoT-23 dataset.

2) EXPERIMENTATION ON NSL-KDD DATA SET

We trained and tested our proposed technique along with other DL models on the NSL-KDD dataset. As mentioned earlier, this dataset comprises distinct training and test sets. We utilized 125,973 records for training and 22,544 records for testing. All models were evaluated on the original test dataset for an unbiased analysis. Fig. 10 and Fig. 11 display the information regarding the confusion matrix and ROC analysis of the NSL-KDD dataset, respectively. Additionally, Fig. 12 illustrates the accuracy comparison between the proposed framework and naive DL models on the NSL-KDD dataset. For binary classification results on the NSL-KDD dataset, Fig. 13 and Fig. 14 present detailed outcomes. Table 7 provides the detailed experimental results on the NSL-KDD dataset.

TABLE 5. NSL-KDD dataset distribution.

Class	Training	Weight%	Testing	Weight%
Normal	67342	53.46%	9710	43.07%
DoS	45927	36.46%	7460	33.09%
Probing	11656	9.25 %	2421	10.74%
R2L	995	0.79%	2885	12.79%
U2R	52	0.041%	67	0.29%
Total	125973	100%	22543	100%

TABLE 6. Performance Comparison of LSTM, SVM, Decision Trees (DT), CNN Models with the Proposed Model on IoT-23 Dataset, DDoS, CandC, and PortScan Tasks.

		←——DDoS——→			←——CandC——→			←——PortScan——→		
Model	A	R	P	F1	R	P	F1	R	P	F1
LSTM	93.5%	100%	100%	100%	47%	100%	63.9%	100%	85.4%	92.1%
SVM	93.1 %	100%	100%	100%	47%	100%	63.9%	100%	83.7%	91.1%
DT	93.7 %	100%	100%	100%	48.4%	100%	65.2%	100%	86.4%	92.7%
CNN	93.7 %	100%	100%	100%	48.4%	100%	65.2%	100%	86.4%	92.7%
DQQS	95.9 %	100%	100%	100%	80%	100%	88.9%	100%	90.4%	95.0%

A = Accuracy, R = Recall, P= Precision, F1= F1- score, DQQS =Proposed Model

TABLE 7. Results of NSL-KDD test dataset.

		←——Normal——→			←——Abnormal——→		
Model	A	R	P	F1	R	P	F1
LSTM	82.0%	97.5%	71.0%	82.1%	70.0%	97.2%	81.3%
SVM	72.1 %	97.8%	61.2%	75.2%	53.1%	96.9%	68.6%
DT	81.5 %	97.3%	70.8%	81.9%	69.6%	97.1%	81.0%
CNN	80.5 %	96.5%	68.7%	80.3%	69.5%	96.6%	80.8%
DQQS	85.5 %	98.8%	78.0%	87.2%	72.5%	98.5%	83.5%

A = Accuracy, R = Recall, P= Precision, F1= F1- score, DQQS =Proposed Model

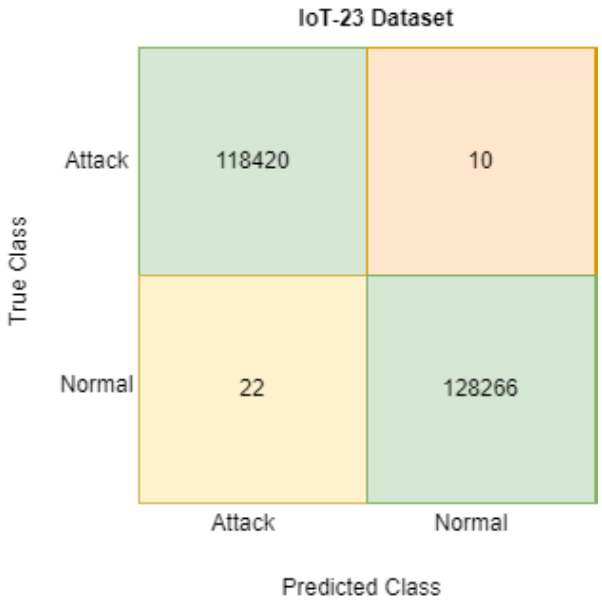

FIGURE 4. Confusion matrix for the proposed framework on the IoT-23 dataset.

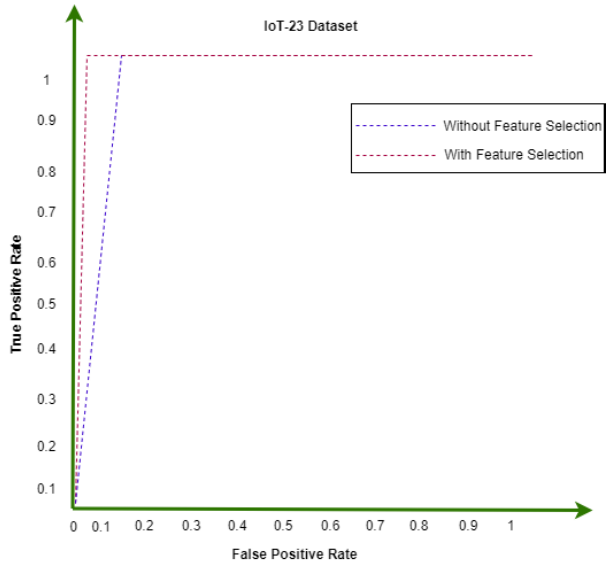

FIGURE 5. Analysis of IoT-23 dataset with respect to ROC.

Table 8 showcases the processing time, learning time, and detection time of various DL models and the proposed technique across two benchmark datasets: IoT-23 and NSL-KDD, specifically focusing on attack detection. Five methods, including the proposed framework, underwent evaluation on the IoT-23 dataset. For instance, SVM recorded 510 seconds of processing time, 25 seconds for learning, and 22 seconds for detection, respectively. The DT model

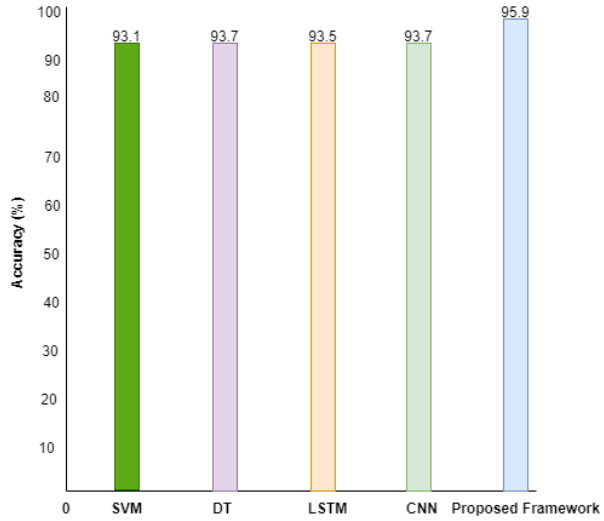

FIGURE 6. Performance comparison of the proposed technique with naive deep learning models in terms of attack detection and prediction Accuracy (%) on the IoT-23 dataset.

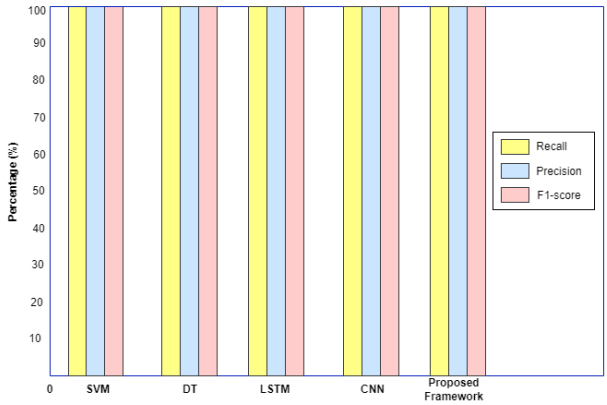

FIGURE 7. Multiclassification performance comparison of the proposed technique with naive deep learning models in term of Recall, Precision, and F1-score by classifying the DDoS attack on the IoT-23 dataset.

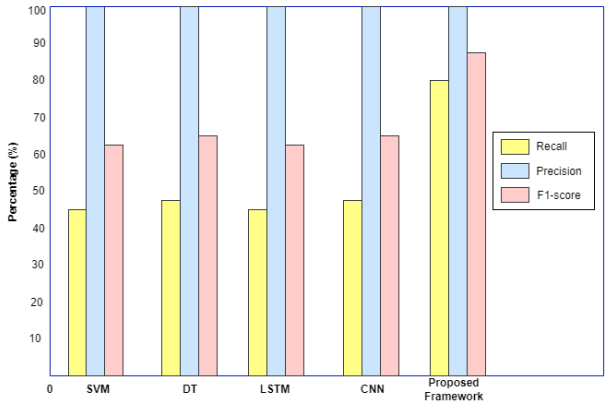

FIGURE 8. Multiclassification performance comparison of the proposed technique with naive deep learning models in term of Recall, Precision, and F1-score by classifying the C and C attack on the IoT-23 dataset.

demands 540 seconds for processing time, with 26 seconds devoted to learning and 24 seconds for detection. LSTM,

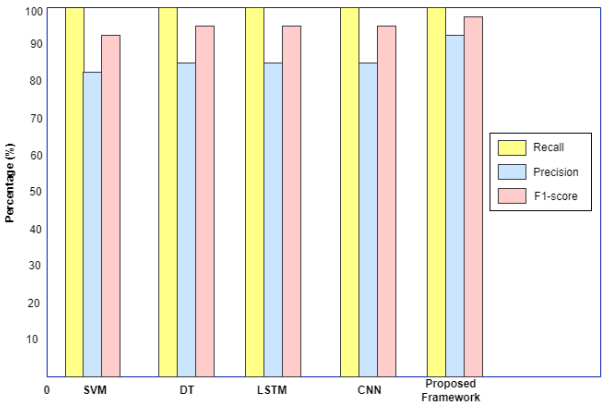

FIGURE 9. Multiclassification performance comparison of the proposed technique with naive deep learning models in term of Recall, Precision, and F1-score by classifying the PortScan attack on the IoT-23 dataset.

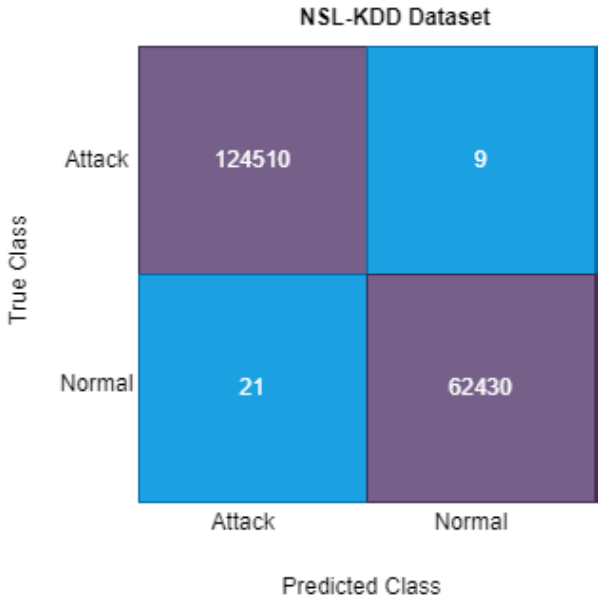

FIGURE 10. Confusion matrix for the proposed framework on the NSL-KDD dataset.

on the other hand, requires 370 seconds for processing, accompanied by 17 seconds for learning and 16 seconds for detection. In a similar vein, CNN utilizes 387 seconds for processing, 23 seconds for learning, and 19 seconds for detection. Notably, the proposed framework demonstrates the shortest processing, learning, and detection times among all DL models, clocking in at 290 seconds, 13 seconds, and 6 seconds, respectively. On the NSL-KDD dataset, the proposed framework also surpassed the DL models, achieving optimal processing, learning, and detection times of 300 seconds, 11 seconds, and 7 seconds, respectively.

F. EFFICACY AND EFFICIENCY EVALUATION

To assess the effectiveness and efficiency of the proposed technique, it is evaluated and tested against four network

TABLE 8. Comparing the computational processing time of the proposed framework with DL models across different datasets. Note: Processing Time, Learning Time and Detection time are measured in seconds.

Datasets	Methods	Processing Time	Learning Time	Detection Time
IoT-23	SVM	510	25	22
	DT	540	26	24
	LSTM	370	17	16
	CNN	387	23	19
	Proposed Framework	290	13	6
NSL-KDD	SVM	490	26	23
	DT	520	24	22
	LSTM	410	22	19
	CNN	400	25	21
	Proposed Framework	300	11	7

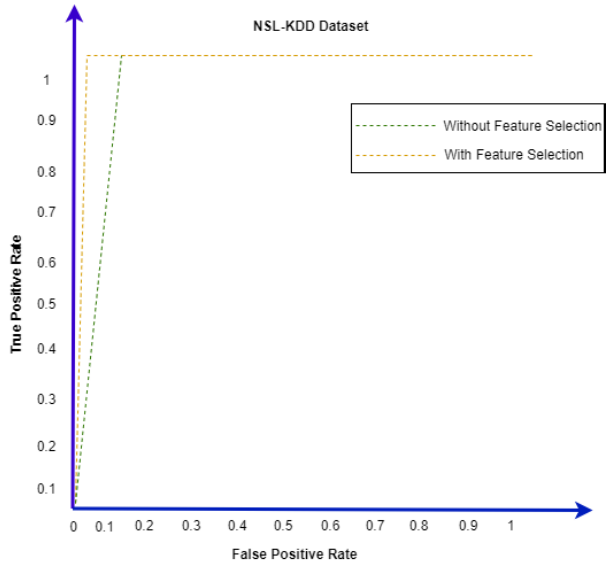

FIGURE 11. Analysis of NSL-KDD dataset with respect to ROC.

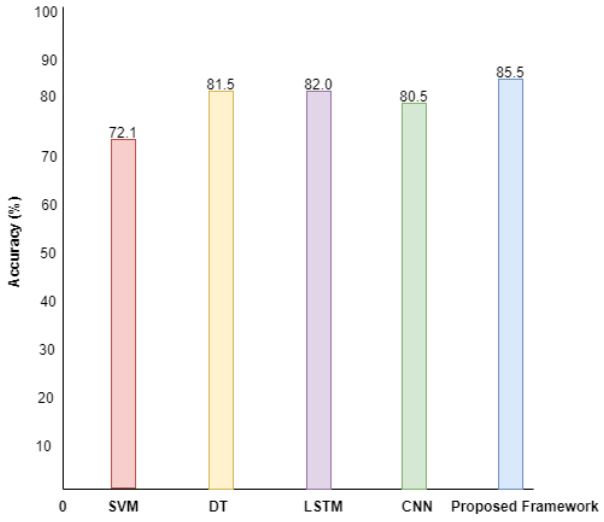

FIGURE 12. Performance comparison of the proposed technique with naive deep learning models in terms of attack detection and prediction Accuracy (%) on the NSL-KDD dataset.

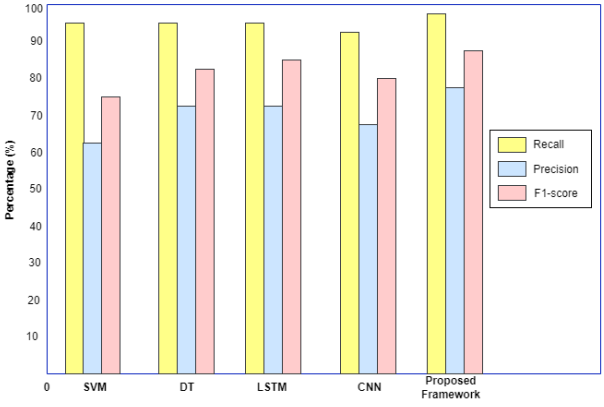

FIGURE 13. Binary classification performance comparison of the proposed technique with naive deep learning models in term of Recall, Precision, and F1-score by classifying the normal traffic on the NSL-KDD dataset.

metrics: throughput, packet loss rate, and the probability of routing through attacked nodes. These metrics are crucial for network evaluation, especially when considering the influence of sensing plane attacks (Eavesdropping Attack (EDA), Physical Tampering Attack (PTA)) and data plane attacks (Man-in-the-Middle Attack (MIMA), Distributed Denial of Service Attack (DDoS)). In the following subsections, the proposed technique is comprehensively evaluated using these metrics, and comparisons are made with the state-of-the-art routing protocols.

1) THROUGHPUT

Here, the performance assessment of the proposed technique in terms of throughput is conducted, and it is compared with the state-of-the-art routing techniques (OSPF, LL, RL, sailfish). Rigorous evaluations have been carried out in both attacked and non-attacked environments. In the non-attacked environment, all sensor nodes and switch nodes function normally. On the other hand, the attacked environment involves random attacks on sensors and switches, leading to performance degradation in terms of QoS and QoE.

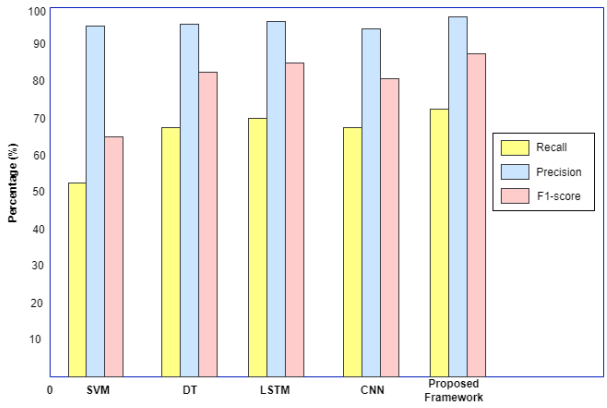

FIGURE 14. Binary classification performance comparison of the proposed technique with naive deep learning models in term of Recall, Precision, and F1-score by classifying the abnormal traffic on the NSL-KDD dataset.

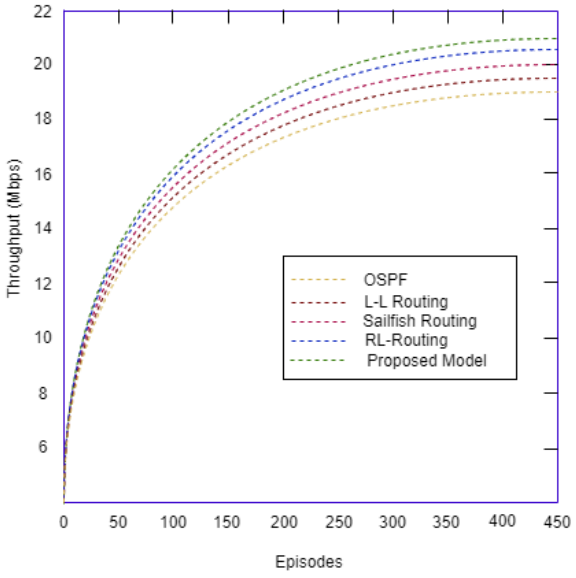

FIGURE 15. Comparison between the proposed technique and state-of-the art routing algorithms on the network metric Throughput in a non-attacked environment.

Fig. 15 depicts the performance of both the proposed model and related routing algorithms in a non-attacked environment. The results demonstrate that all techniques exhibit an excellent packet delivery rate. Fig. 16 display the performance of the proposed framework and state-of-the art routing algorithms in an attacked environment. Initially, we set the reward parameters α, β, and σ to 0.3, 0.3, and 0.4, respectively. As depicted in Fig. 16, the packet delivery rate of the proposed model is higher than the other routing algorithms under the Man-in-the-Middle (MIM) Attack. This improved performance of the proposed model can be attributed to its intelligent decision-making, which enables it to select secure routing paths.

2) LATENCY

The time taken by a packet to travel from the source node s_i to the destination node s_j is commonly referred to as

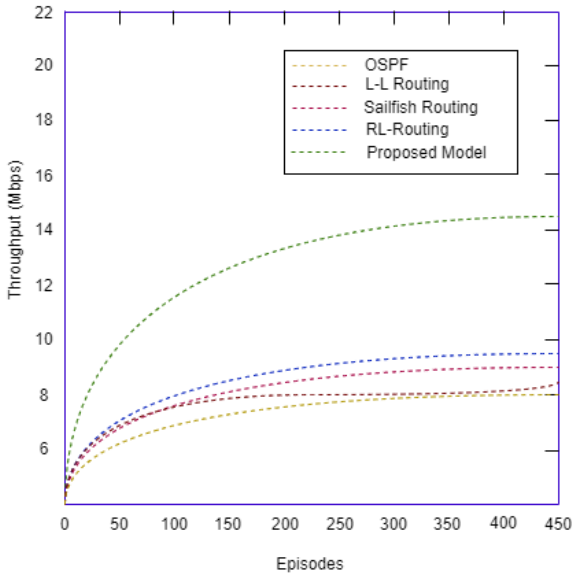

FIGURE 16. Comparison between the proposed technique and state-of-the art routing algorithms on the network metric Throughput in an attacked environment.

network latency. Higher latency indicates greater delay, while lower latency implies minimal delay. Delay is influenced by various factors including propagation delay, transmission delay, queuing delay, and processing delay. In this subsection, we will compare the latency of the proposed technique with the four state-of-the art routing algorithms. Experimental results indicate that in a non-attacked environment, both protocols exhibit similar behavior and almost the same latency value. However, when the environment transitions from non-attacked to attacked, the number of attacked nodes (sensors and switches) and malicious nodes gradually increases. In an attacked environment, the latency value of all routing algorithms increases due to the constant obstruction and disruption caused by malicious nodes. Thanks to the intelligent behavior and learning capability of the DRL agent in the proposed model, its performance is notably better, stable, and boasts lower latency compared to related routing algorithms. The trained agent avoids malicious nodes by learning from the underlying environment, thereby reducing latency.

Furthermore, through extensive experimentation in an attacked environment, we have observed that different network attacks exert varying impacts on latency. The results demonstrate that data plane attacks (MIMA and DDoS) have a more pronounced impact on latency when compared to sensing plane attacks (PTA and EDA). Among the data plane attacks, DDoS has a greater impact on latency compared to MIMA. Additionally, we have identified that altering the values of the reward parameters $[\alpha, \beta, \sigma]$ influences latency. The outcomes of our experimentation in non-attacked and attacked environments are depicted in Fig. 17 and Fig. 18.

TABLE 9. Performance comparison between the proposed framework and state-of-the art routing algorithms.

Environment	Ref.	Methods	T(Mbps)	L(ms)	P(%)	Security	Routing
Non-Attacked	[31]	OSPF	18.5	42	N/A	N/A	✓
	[32]	L-L Routing	19	40	N/A	N/A	✓
	[27]	Sailfish Routing	20	33	N/A	N/A	✓
	[33]	RL-Routing	20.5	31	N/A	N/A	✓
	T.A	Proposed Model	21	30	N/A	N/A	✓
Attacked	[31]	OSPF	8	88	0.19		✓
	[32]	L-L Routing	8.2	85	0.20		✓
	[27]	Sailfish Routing	9	72	0.43		✓
	[33]	RL-Routing	9.5	75	0.40		✓
	T.A	Proposed Model	14.5	52	0.81	✓	✓

T.A = This Article, T = Throughput, L= Latency, P= Probability, N/A = Not Applicable

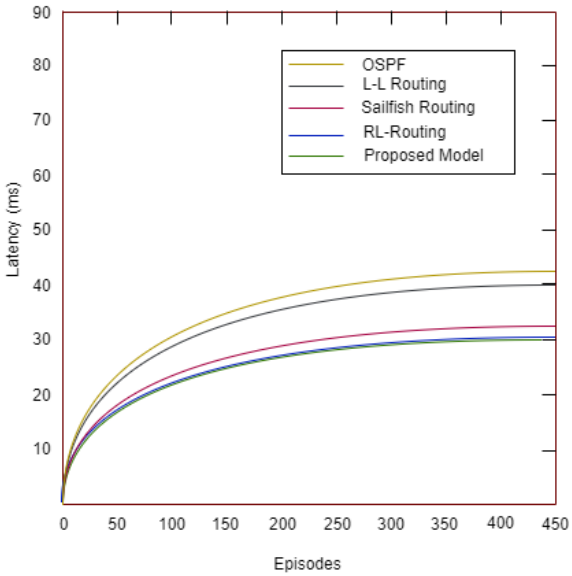

FIGURE 17. Comparison between the proposed technique and state-of-the art routing algorithms on the network metric Latency in a non-attacked environment.

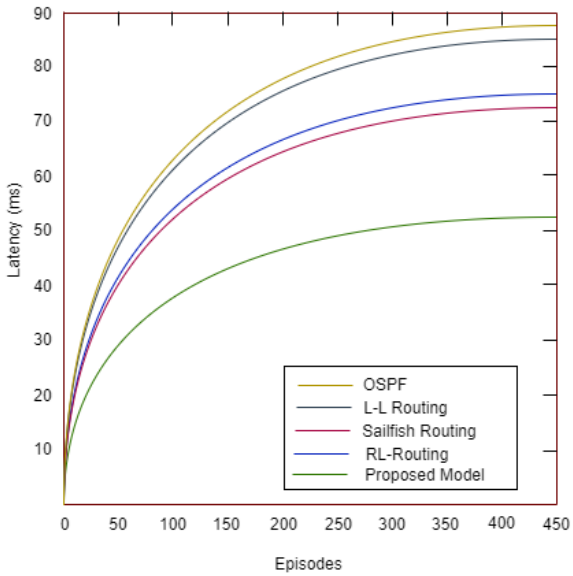

FIGURE 18. Comparison between the proposed technique and state-of-the art routing algorithms on the network metric Latency in an attacked environment.

3) THE PROBABILITY OF AVOIDING THE ATTACKED SWITCHES

In this subsection, we assess the probability of successfully avoiding malicious and attacked nodes when the SDN-IoT network is under attack. Our experimentation results clearly indicate that the proposed technique exhibits a significantly higher probability of successfully avoiding malicious and attacked nodes compared to the other state-of-the art routing protocols. This is evidenced by Fig. 19, which portrays our experimental findings.

VI. RESULTS AND DISCUSSION

The proposed technique has demonstrated excellent performance compared to the four state-of-the art routing algorithms (OSPF, L-L, Sailfish, RL). In an attacked SDN-IoT environment, the performance comparison of the proposed technique with these four algorithms across three network metrics is shown in a Table 9. The results indicate that the four routing protocols have failed to tackle security attacks, and dynamic, and unpredictable traffic flows and hence unable to provide secure routing with high-quality QoS and QoE. On the other hand, the proposed technique steadily improves its performance through continuous interaction with the underlying environment and the generation of optimized policies. This is evident in the third performance metric (Probability of avoiding malicious nodes) in Table 9, where the proposed technique outperformed all four routing algorithms comprehensively by the value 0.81%.

Despite the excellent performance of the proposed technique, there are some limitations in the proposed model. The first limitation concerns the hidden layers of the proposed model. This model contains three hidden layers to address security attacks as well as to optimize routing. However, it is a fact that complex problems are solved more efficiently with an increasing number of hidden layers. Thus, increasing the number of hidden layers can improve the efficiency of the proposed model in terms of throughput and by avoiding the present malicious nodes. The second limitation of the proposed model pertains to security attacks.

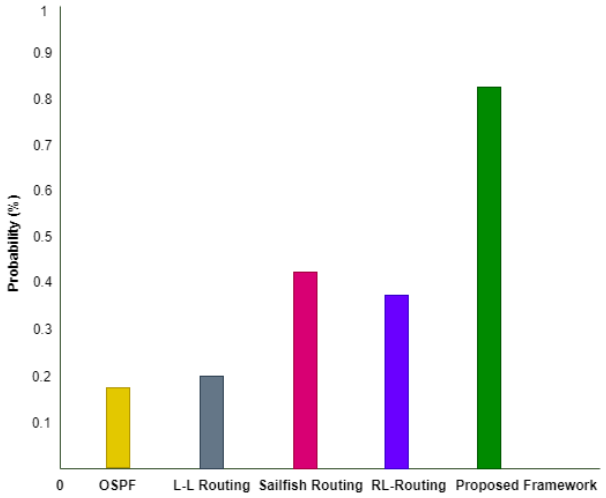

FIGURE 19. Comparison between the proposed technique and state-of-the art routing algorithms on the metric of the probability of successfully avoiding the attacked nodes.

All experimentation and testing were conducted using well-known and older security attacks. However, it is important to acknowledge that the performance of the proposed model may be adversely affected when exposed to state-of-the-art and the latest security attacks. The third limitation of our proposed technique pertains to the utilization of datasets for attack detection and classification. We employed two datasets, namely NSL-KDD and IoT-23, for this purpose. While these datasets are not particularly large, they do contain a fair number of records. In the future, it would be valuable to assess the effectiveness of the proposed technique on larger datasets. The fourth limitation concerns dataset imbalance. Our evaluation involved testing the proposed technique on both balanced and imbalanced datasets. Notably, the results on balanced datasets were significantly superior to those on imbalanced datasets. Therefore, future work should focus on enhancing the efficiency of the proposed technique when dealing with imbalanced datasets.

VII. CONCLUSION AND FUTURE WORK

In this article, we proposed a Deep Reinforcement Learning (DRL)-based efficient and secure routing technique named DQQS designed for the SDN-IoT environment. The core functionality of this proposed model is to ensure secure routing while maintaining both Quality of Service (QoS) and Quality of Experience (QoE) within the network. To validate its accuracy in identifying and classifying attacks, we conducted experiments using four AI metrics (Accuracy, Precision, Recall, and F1-score) and compared it with naive DL models (LSTM, CNN, DT, SVM). Subsequently, we evaluated the technique against three QoS and QoE-related metrics: Throughput, Latency, and Probability of avoiding attacked nodes. The simulations demonstrate that the proposed model exhibits excellent accuracy in identifying attacks and surpasses the state-of-the art routing models by a significant margin, particularly when nodes are under attack. This exceptional performance in the SDN-IoT attacked environment can be attributed to the intelligent DRL agent, which formulates secure, optimized routing policies by analyzing rewards and interacting with the underlying environment. While discussing future research directions, the first future research direction is the usage of activation functions in the proposed algorithm. The proposed model uses three activation functions: ReLU, Tanh, and Softmax. In future research work, new activation functions or combinations of these three activation functions can be employed to enhance the security and efficiency of the proposed model. The second future research direction is the exploitation of other DL models, such as Autoencoder (AE), Deep Belief Network (DBN), and Generative Adversarial Network (GAN).

ACKNOWLEDGMENT

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R300), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

REFERENCES

[1] C. Brewster, I. Roussaki, N. Kalatzis, K. Doolin, and K. Ellis, "IoT in agriculture: Designing a Europe-wide large-scale pilot," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 26–33, Sep. 2017.

[2] A. Rostami, P. Ohlen, K. Wang, Z. Ghebretensae, B. Skubic, M. Santos, and A. Vidal, "Orchestration of RAN and transport networks for 5G: An SDN approach," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 64–70, Apr. 2017.

[3] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A software defined networking architecture for the Internet-of-Things," in *Proc. IEEE Netw. Operations Manag. Symp. (NOMS)*, May 2014, pp. 1–9.

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[5] T. A. Q. Pham, Y. Hadjadj-Aoul, and A. Outtagarts, "Deep reinforcement learning based QoS-aware routing in knowledge-defined networking," in *Quality, Reliability, Security and Robustness in Heterogeneous Systems* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 272, T. Duong, N. S. Vo, and V. Phan, Eds. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-030-14413-5_2.

[6] C. Kharkongor, T. Chithralekha, and R. Varghese, "A SDN controller with energy efficient routing in the Internet of Things (IoT)," *Proc. Comput. Sci.*, vol. 89, pp. 218–227, Jan. 2016.

[7] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 623–654, 1st Quart., 2016.

[8] S. B. A. Khaliq, M. F. Amjad, H. Abbas, N. Shafqat, and H. Afzal, "Defence against PUE attacks in ad hoc cognitive radio networks: A mean field game approach," *Telecommun. Syst.*, vol. 70, no. 1, pp. 123–140, Jan. 2019.

[9] W. B. Shahid, B. Aslam, H. Abbas, S. B. Khalid, and H. Afzal, "An enhanced deep learning based framework for web attacks detection, mitigation and attacker profiling," *J. Netw. Comput. Appl.*, vol. 198, Feb. 2022, Art. no. 103270.

[10] G. Kim, Y. Kim, and H. Lim, "Deep reinforcement learning-based routing on software-defined networks," *IEEE Access*, vol. 10, pp. 18121–18133, 2022.

[11] P. Cong, Y. Zhang, Z. Liu, T. Baker, H. Tawfik, W. Wang, K. Xu, R. Li, and F. Li, "A deep reinforcement learning-based multi-optimality routing scheme for dynamic IoT networks," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 108057.

[12] A. A. Magadum, A. Ranjan, and D. G. Narayan, "DeepQoSR: A deep reinforcement learning based QoS-aware routing for software defined data center networks," in *Proc. 12th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, Jul. 2021, pp. 1–7.

[13] T. G. Nguyen, T. V. Phan, D. T. Hoang, T. N. Nguyen, and C. So-In, "Federated deep reinforcement learning for traffic monitoring in SDN-based IoT networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 4, pp. 1048–1065, Dec. 2021.
[14] N. Saha, S. Bera, and S. Misra, "Sway: Traffic-aware QoS routing in software-defined IoT," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 1, pp. 390–401, Jan. 2021.
[15] G.-C. Deng and K. Wang, "An application-aware QoS routing algorithm for SDN-based IoT networking," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 00186–00191.
[16] S. P. K. Et. al., "A deep learning method for effective channel allotment for SDN based IoT," *Turkish J. Comput. Math. Educ. (TURCOMAT)*, vol. 12, no. 2, pp. 1721–1728, Apr. 2021.
[17] A. Kannan, S. Vijayan, M. Narayanan, and M. Reddiar, "Adaptive routing mechanism in SDN to limit congestion," in *Information Systems Design and Intelligent Applications* (Advances in Intelligent Systems and Computing), vol. 862, S. Satapathy, V. Bhateja, R. Somanah, X. S. Yang, and R. Senkerik, Eds. Singapore: Springer, 2019, doi: 10.1007/978-981-13-3329-3_23.
[18] L. Zhao, J. Wang, J. Liu, and N. Kato, "Routing for crowd management in smart cities: A deep reinforcement learning perspective," *IEEE Commun. Mag.*, vol. 57, no. 4, pp. 88–93, Apr. 2019.
[19] F. Tang, Z. Md. Fadlullah, B. Mao, and N. Kato, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5141–5154, Dec. 2018.
[20] F. Arif, Y. Abbas, S. Ahmad, and M. Waseem, "DLICA: Deep learning based novel strategy for intelligent channel adaption in wireless SDN-IoT environment," in *Proc. Int. Conf. Commun., Comput. Digit. Syst. (C-CODE)*, May 2023, pp. 1–6.
[21] T. Wu, P. Zhou, B. Wang, A. Li, X. Tang, Z. Xu, K. Chen, and X. Ding, "Joint traffic control and multi-channel reassignment for core backbone network in SDN-IoT: A multi-agent deep reinforcement learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 231–245, Jan. 2021.
[22] R. Samadi and J. Seitz, "Machine learning routing protocol in mobile IoT based on software-defined networking," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2022, pp. 108–111.
[23] K. Haseeb, I. Ahmad, I. I. Awan, J. Lloret, and I. Bosch, "A machine learning SDN-enabled big data model for IoMT systems," *Electronics*, vol. 10, no. 18, p. 2228, Sep. 2021.
[24] H. D. Praveena, V. Srilakshmi, S. Rajini, R. Kolluri, and M. Manohar, "Balancing module in evolutionary optimization and deep reinforcement learning for multi-path selection in software defined networks," *Phys. Commun.*, vol. 56, Feb. 2023, Art. no. 101956.
[25] M. Moslehi, H. Ebrahimpor-Komleh, S. Goli, and R. Taji, "A QoS optimization technique with deep reinforcement learning in SDN-based IoT," *Majlesi J. Electr. Eng.*, vol. 15, no. 3, pp. 105–113, Sep. 2021.
[26] D. K. Dake, J. D. Gadze, and G. S. Klogo, "DDoS and flash event detection in higher bandwidth SDN-IoT using multiagent reinforcement learning," in *Proc. Int. Conf. Comput., Comput. Model. Appl. (ICCMA)*, Jul. 2021, pp. 16–20.
[27] R. Mohammadi, S. Akleylek, and A. Ghaffari, "SDN-IoT: SDN-based efficient clustering scheme for IoT using improved sailfish optimization algorithm," *PeerJ Comput. Sci.*, vol. 9, p. e1424, Jul. 2023.
[28] X. Guo, H. Lin, Z. Li, and M. Peng, "Deep-reinforcement-learning-based QoS-aware secure routing for SDN-IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6242–6251, Jul. 2020.
[29] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[30] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw.*, Oct. 2010, pp. 1–6.
[31] J. Moy, "RFC2328: OSPF version 2," Tech. Rep., 1998. Accessed: Dec. 15, 2023. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2328.html
[32] L. Li and A. K. Somani, "Dynamic wavelength routing using congestion and neighborhood information," *IEEE/ACM Trans. Netw.*, vol. 7, no. 5, pp. 779–786, Oct. 1999.
[33] Y.-R. Chen, A. Rezapour, W.-G. Tzeng, and S.-C. Tsai, "RL-routing: An SDN routing algorithm based on deep reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 3185–3199, Oct. 2020.
[34] A. Parmisano, S. Garcia, and M. Erquiaga, "A labeled dataset with malicious and benign IoT network traffic," Stratosphere Lab., Praha, Czech Republic, Tech. Rep., 2020. Accessed: Jan. 10, 2024. [Online]. Available: https://www.stratosphereips.org/datasets-iot23
[35] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.

ZABEEHULLAH received the B.Sc. degree in software engineering from the Department of Software Engineering, University of Engineering and Technology, Taxila, Pakistan, in 2012, and the M.S. degree in computer science from the Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Software Engineering, National University of Science and Technology, Islamabad. His research interests include the Internet of Things (IoT), the Internet of Medical Things (IoMT), software-defined networks, SDN-IoT, AI, deep learning applications in SDN-IoT for quality of service, routing communication, and security.

FAHIM ARIF (Senior Member, IEEE) received the degree in telecommunication engineering from UET Lahore, in 1995, and the Ph.D. degree in software engineering from the National University of Sciences and Technology, in 2009. However, he completed his research work at Carleton University, Ottawa, Canada, in 2007. He attended the Professional Development Program, George Mason University, Fairfax, USA, in 2013. He is currently with the National University of Sciences and Technology (NUST), Islamabad, Pakistan. He has published more than 70 research publications in international journals and conferences. He has more than 35 years of experience in administration, research, and academics. His research interests include software engineering, SQA, remote sensing, and machine learning applications in the software domain.

NAUMAN ALI KHAN received the B.S. degree in computer science from the University of Engineering and Technology, Peshawar, Pakistan, in 2008, the M.Phil. degree in computer science from Quaid-i-Azam University, Islamabad, Pakistan, in 2012, and the Ph.D. degree in communication and information systems from the University of Science and Technology of China, Hefei, China. He is currently an Assistant Professor with the Department of Computer Software Engineering National University of Science and Technology, Islamabad. During his professional career, he remained one of the active organizers of Frontier Information Technology (FIT), an international conference in collaboration with COMSATS University, Abbottabad Campus. His research interests include machine learning, social network analysis, wireless big data mining, social-tie inference, and prediction. He received the prestigious Gold Medal during the bachelor's degree.

JAVED IQBAL (Senior Member, IEEE) received the M.S. degree from BTH, Sweden, in 2009, and the Ph.D. degree in electronics and telecommunication engineering from the Polytechnic University of Turin, Italy, in 2015. He was awarded the scholarship for Ph.D. studies from the Higher Education Commission of Pakistan (HEC), Pakistan, under the UESTP Italy Project. He is currently an Associate Professor with the Department of Computer Software Engineering,

National University of Science and Technology, Pakistan. Prior to joining NUST, he was an Associate Professor and the Head of the Department of Computer Systems Engineering (DCSE), University of Engineering and Applied Sciences (UEAS), Pakistan. As a PI/Co-PI, he has secured more than ten million PKR research funding and awards. He was a part of the Italian Government funded project of five million euros for the topic "Socially Aware Networking for Human Cooperation in Mobility" a PRIN Project, from February 2012 to December 2014. He has published in prestigious journals and conferences (more than 100 impact scores). He has tendered his technical committee services in various conferences and editorial services in reputed high IF journals. His current research interests include vehicular networking, software-defined networking, cloud computing, the IoT, and machine learning.

FATEN KHALID KARIM received the Ph.D. degree in computing and information technology from Flinders University, Adelaide, SA, Australia. She is currently an Assistant Professor with the Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia. Her research interests include cloud computing and information technology. She has published several research articles in her field.

NISREEN INNAB received the Ph.D. degree in computer information systems, in 2008. She is currently an Associate Professor with the Department of Computer Science and Information Systems, College of Applied Sciences, Almaarefa University, Riyadh, Saudi Arabia. She is also the Program Director of the Information Systems Program. She has published many scientific research in international high-prestigious journals indexed in Web of Science and Scopus. She has supervised many theses for the Master of Information Security. Furthermore, she also supervised many graduation projects for computer science, information systems, and health information systems. Her research interests include information security, cyber security, machine learning, artificial intelligence, and the IoT.

SAMIH M. MOSTAFA received the bachelor's and M.Sc. degrees in computer science from the Computer Science and Mathematics Department, Faculty of Science, South Valley University, in 2004 and 2010, respectively, and the Ph.D. degree in computer science from the Advanced Information Technology Department, Graduate School of Information Technology, Kyushu University, Japan, in 2017. His research interests include machine learning and CPU scheduling. He is currently a fellow of the Academy of Scientific Research and Technology (ASRT), Egypt.