Egyptian Informatics Journal 27 (2024) 100515

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full Length Article

BSDN-HMTD: A blockchain supported SDN framework for detecting DDoS attacks using deep learning method

Parthasarathy Ramadass [a], Raja shree Sekar [b], Saravanan Srinivasan [c], Sandeep Kumar Mathivanan [d], Basu Dev Shivahare [d], Saurav Mallik [e,f,*], Naim Ahmad [g,*], Wade Ghribi [g,*]

[a] Department of Computer Science & Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai 600062, India
[b] Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India
[c] Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai 600062, India
[d] School of Computer Science and Engineering, Galgotias University, Greater Noida 203201, India
[e] Department of Environmental Health, Harvard T H Chan School of Public Health, Boston, MA 02115, USA
[f] Department of Pharmacology & Toxicology, The University of Arizona, Tucson, AZ 85721, USA
[g] College of Computer Science, King Khalid University, Saudi Arabia

ARTICLE INFO

Keywords:
Software Defined Networking (SDN)
Moving Target Defense (MTD)
Authentication
Proactive MTD
Reactive MTD
Honeypot
Blockchain

ABSTRACT

The surge in Distributed Denial of Service (DDoS) attacks within SDN environments demands more potent defense strategies. While Moving Target Defense (MTD) holds promise, current MTD approaches against DDoS suffer from security gaps due to overwhelming malicious traffic and static detection areas. In order to tackle these difficulties, we have implemented BSDN-HMTD, a combination of deep learning and blockchain technologies within SDN environments, as a framework. Our strategy starts by employing blockchain technology to authenticate users. We use the NTRU-based Nyberg Rueppel Digital Signature Algorithm for this purpose. This ensures that only authenticated user flows are allowed for validation and forwarding. Within the forwarding layer, Quantum Convolutional Neural Networks (QCNN) evaluate authentic flows by analyzing many characteristics, effectively differentiating between regular, malicious, and dubious flows. Utilizing an Enhanced Spotted Hyena Optimization (EHSO) method to activate switches in real-time modifies the vulnerable points of attack, so impeding attackers and simultaneously decreasing energy usage. The Forwarding Layer Organizer (FLO) oversees the detection of possible attacker surveillance activities and transmits the collected information to local controllers in the control layer. The controllers, functioning in a structured controller network, carry out proactive Moving Target Defense (MTD) techniques, such as host virtual IP hopping, which make attacker plans more complex and raise their operational expenses. Reactive MTD actions are implemented based on the results of flow validation. These actions utilize techniques such as secure honeypots and host virtual IP hopping to effectively prevent attacks. The blockchain securely logs all processed data related to packet validation, authentication, and honeypot activities to ensure the protection of data privacy. Our studies, conducted using Network Simulator-3.26 (NS-3.26), show that our proposed framework outperforms existing techniques in terms of several validation criteria.

1. Introduction

In recent days, the trends in healthcare, cloud services, and other technologies as well as different networks are dramatically increased over time. On the other hand, malicious users and hackers who are very smart are trying their utmost best to exploit the resources on every network, which also affects the Quality of Service (QoS) [1]. The software-defined network is one of the emerging technologies as it provides resilient scalability, flexibility, and network programmability to the environment. So, in recent years it is highly targeted by cyber attackers for its centralized controller nature [2]. As the intelligence from the switches is decoupled, attackers are showing their huge interest to deplete both data and the control plane by implementing DDoS attacks. An attacker tries to flood the system with multiple bots distributed

* Corresponding authors.
E-mail addresses: rajashree.cse@sathyabama.ac.in (R. shree Sekar), sauravmtech2@gmail.com, smallik@arizona.edu, smallik@hsph.harvard.edu (S. Mallik), nagqadir@kku.edu.sa (N. Ahmad), ygraby@kku.edu.sa (W. Ghribi).

https://doi.org/10.1016/j.eij.2024.100515
Received 6 January 2024; Received in revised form 4 July 2024; Accepted 30 July 2024
Available online 9 August 2024
1110-8665/© 2024 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Computers and Artificial Intelligence, Cairo University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

[3]. DDoS attacks in SDN can be implemented in three ways such as data plane DDoS, application plane DDoS, and control plane DDoS [4]. As a result of this, various researchers have mitigated this problem using distributed controllers. Further machine learning based methods also adopted to detect DDoS [5–8]. However, the static environments give a hint to the attacker to learn the targeting network. Thus, this static behaviour paves the way to face reconnaissance attacks where attackers scan the open ports and available IP addresses [9]. Hence, attackers have a greater opportunity to weaponize according to the observed vulnerabilities as well as they can also surpass the detection systems very easily [10]. Reconnaissance attacks are usually established in the form of releasing various kinds of probe data packets to the device/node/server and according to the response that the attacker gets from the network host is plotted as a target by the attacker [11].

To overcome these issues, Moving Target defence (MTD) has been recommended to thwart reconnaissance attacks and to change the static nature of surfaces (attack, detection, and prevention) as a countermeasure [12,13]. In general, MTD can be performed in different manners like port hopping, Ip hopping, migration, diversification, surface shifting, etc. These strategies are subsumed into two categories like proactive MTD implementation and reactive MTD implementation [14,15]. In MTD, IP hopping is one of the important strategies in which virtual IP addresses of the network entities are dynamically changed [16]. The main reason behind changing the IP addresses is to reduce the survival rate at the attackers hit list and to diminish the successful scanning to some extent. On the other hand, frequent Ip hopping reduces the attack success rate and frustrates the attackers for establishing their weaponized worms, viruses, and other malware. In SDN, IP hopping is performed by the controller as the network structure is very flexible to adaptation [17]. The controller keeps the real and hopped virtual IP addresses of network devices [18]. Besides, the majority of the research works don't pay adequate attention to managing the data plane, and control plane as well as learning the attacker behaviours [19]. For successful MTD implementation, deception technologies need to be incorporated for trapping the attackers without their knowledge to pretend that they are still successful in their established attacks without disrupting the legitimate users' services. In most of the research works, the logs related to the attacker behavior in terms of attack patterns are not preserved with greater security thus this results in security issues [20].

1.1. Motivation & objectives

In general, defender strategies are static hence attackers have the potential opportunities to learn the environment. The SDN technology that faces DDoS attacks is mitigated through moving target defenses (MTD) where defender strategies are dynamic. Various existing researches have investigated and contributed their knowledge to overcome DDoS attacks in SDN using MTD. Yet there are several problems that exist as mentioned below,

- Inadequate Security: In existing works, users are assumed as legitimate ones. By assuming so, unauthorized users will intrude into the network and this will result in huge security threats, huge traffic, overloading attacks, switch failure rates, etc. On the other hand, preservation of attacker behaviour is given with less security which can also be manipulated.
- Static Detection Surface: In some existing works, the attack surface is dynamically changed but the changes at the detection surface are not that focused. Hence this will offer an opportunity to the attackers for learning the detection surface.
- Ineffective Data Layer Management: The resources management at the data layer is performed in an ineffective manner which causes high energy consumption, improper resource utilization as well as increased computational overhead. With the aforementioned problems, the SDN network is lacking in terms of several QoS aspects such as high packet loss ratio, less throughput, less scalability, and high congestion rate.

Motivated from the above research issues, we have frame a novel research with aim of ensuring security to the SDN environment from cutting edge blockchain technology with MTD technique. The objective of this research by resolving the above problems are provided as follows,

- To develop and evaluate the BSDN-HMTD framework that integrates blockchain technology and Quantum Convolutional Neural Networks (QCNN) within Software-Defined Networking (SDN) environments.
- To enhance security by implementing robust user authentication mechanisms using blockchain technology, ensuring only legitimate users are allowed into the network and mitigating unauthorized intrusions, security threats, traffic overloads, overloading attacks, and switch failure rates.
- To dynamically configure both the attack and detection surfaces, preventing attackers from learning and exploiting the network's defensive mechanisms.
- To enhance Quality of Service (QoS) metrics in SDN networks by reducing packet loss ratio, increasing throughput, improving scalability, and decreasing congestion rates.

1.2. Research contribution

This research mainly focuses on improving the security SDN environment through Moving Target Defence (MTD), and Blockchain Technologies. The contribution of this research to SDN based MTD is provided below,

- Firstly, the malicious traffic in the networks is reduced by proposing an authentication method in the user layer. The users are authenticated to blockchain based on their credentials using NTRU based Nyberg Rueppel Digital Signature Algorithm (NTRU NR-DSA). The adoption of the authentication method also reduces the congestion and computation overhead.
- Secondly, the user flows are validated to reduce the DDoS rate at the preliminary level. The flow rule validation is done by switches in the environment in which the switches are periodically activated and idled to reduce the energy consumption rate using Enhanced Spotted Hyena Optimization (EHSO) algorithm. The selected switch performs flow validation using Quantum Convolutional Neural Networks (QCNN) which classifies flows as normal, malicious, and suspicious by considering several metrics.
- Thirdly, in the control layer, the proactive and reactive MTD is done to prevent, detect, recover, frustrate, and mitigate the attack strategies. For that, the proactive MTD is performed using virtual IP hopping based on Forwarding Layer Organizer (FLO) information. Once the event has occurred, the suspicious flows are validated in terms of packet validation using the QCNN algorithm. Based on the results, the reactive MTD is taken place in terms of virtual IP hopping and honeypot based secure flow migration.

Further, the proposed work performs simulation using Network Simulator- 3.26 to realize the proposed environment and compared with existing works in terms of validation metrics such as Defender success rate, survival rate, attack success rate, energy consumption, computation overhead, and malicious traffic. The simulation and comparative results show that the proposed work outperforms better than the existing works.

1.3. Novelty highlights

The novelties of the proposed work than the existing works are highlighted in this sub-section in form of Table 1 as follows,

Table 1
Novelties of proposed work.

Existing Work	Proposed Novelty
Performs both proactive and reactive MTD. However, the surfaces for performing MTD and flow forwarding by the existing works are kept static which made the attackers easily capture the network at a low cost.	The proposed work also performs both proactive and reactive MTD. But, both the attack and detection surfaces for the proposed work are changed dynamically (i.e. forwarding switch activation for switches, and DAG reconstruction for controller respectively) to confuse and frustrate the attacker at a high cost.
High malicious traffic in the existing works is a major concern which makes the network to be filled with attackers leading to a high rate of DDoS attacks.	The proposed work significantly reduces the malicious traffic in the network by performing user authentication in which only legitimate users take part in the networks.
The existing works performed MTD utilizing migration optimally. However, migrating without security and also a lack of logging the attacker behavior leads to high vulnerability to security attacks.	The proposed work performs the migration of flows in the switches securely and optimally. Honeypots are used by the proposed work to provide security as well attracting the attackers to capture their patterns to store in blockchain.

1.4. Paper organization

The remaining of the paper is organized as follows, section II provides the literature survey and gaps in the existing works in DDoS detection. Section III gives the problem statement in which the specific problems faced by the existing works with corresponding solutions are discussed. Section IV explains the proposed BSDN-HMTD model in which the proposed work is clearly explained with corresponding equations, pseudocodes, and diagrams. Section V illustrates the experimental results which provide a clear explanation of the simulation setup, comparative results, attacker cost analysis, and research summary. Finally, section VI explains the conclusion of the research work. Table 2 gives the notations used in the proposed work.

2. Literature survey

This section illustrates the survey of existing research works in DDoS detection of SDN by adopting MTD strategies. This section is further subdivided into two sub-sections. In addition, the research gaps focused by the existing works are focused in this section in form of Table 3.

2.1. MTD based DDoS detection

Authors in this work [21], have proposed an effective MTD implementation by incorporating various security parameters such as routes, operating systems, ports, IP addresses, and topology configurations in an SDN environment. These security metrics are classified into 3 classes named such as path, address, and attack and by considering these metrics, network changes are tracked over the time when the event takes place. To be more specific, host IP addresses, port numbers, and attack routes are based to ensure dynamicity and to diminish the attack success rates. The authors in [22] have mitigated a type of distributed denial of service attack in which attackers flood the neighborhood of the target (crossfire) in an SDN environment. For thwarting this type of attack in SDN, they have integrated intent-based networking (capable of offering flexible network maintenance) with the moving target defense mechanism (effective in confusing attackers) at the controller layer where policies (intents) are transfigured as rules. Here, MTD is performed in the form of port shuffling/redirecting to other nodes for reducing the defense cost as well as computational overhead.

Table 2
Notation table.

Notation	Description
$\mathbb{U}_{\mathbb{C}}^{i}$	User i with credentials
$\mathrm{pri}(x), \mathrm{pub}(x)$	Private and public key
SW_i	Switches
$fL_i(SW_i)$	Flow of the switches
$E, (j, k)$	Digital signature parameters
$L(\mathbb{U}_{\mathbb{C}}^{i})$	Legitimate user
$\forall$	Active switch selection feature set
ϑ	CNN parameter
UN	Unitary operator
$\beth$	Maliciousness score
$\aleph$	Anomaly Score

In this work [23], the authors have proposed a framework using reinforcement learning where an agent (single-player) has the privilege to make a secured dynamic system and software configuration to deceive the attacker's efforts from the reconnaissance stage itself by autonomously changing the exploitable configurations. Here, system and software configurations are made in the form of a moving target defense mechanism. Besides, to evaluate the agent performance, here Monto Carlo prediction algorithm is used. MTD implementation in terms of spatial and temporal aspects is taken into account by incorporating attack and defense surface transitions was proposed by authors in [24]. Here, a game concept named flip-it is utilized for uninterruptedly observing the attack and defense activities to make temporal and spatial decision framework. In addition to that, the best strategies are chosen based on this game approach. When compared to other game theories, this type of game concept based in stealth info has offered realistic results as per the authors statements.

Authors in [25], proposed a security enhancement for the DNS server is performed by using intent-aware moving target defense in the field of SDN by dynamically hopping the port numbers. Here, intent apps and MTD apps are running at the control plane where 3 frameworks in each application such as intents (policies), composing, and installation as well as monitoring, scanning detection, and migration decision making are comprised within those respectively. Further, the MTD application is completely monitoring the network clients when their traffics are directed toward the DNS server. On the other hand, intent apps are dynamically changing the port numbers of DNS services to deceive attacker strategies. The unknown cyber-attacks that take place in a network are mitigated and thwarted through a game theory and a decision-making module called Markov-game by incorporating a moving target defense mechanism by [26]. This framework is efficient in even the former information is incomplete. For handpicking best strategies to enrich the defense countermeasures, revenue-based technique (Markov decision process) is incorporated here. The vulnerable resources are adaptively changed with the help of the game theory concept. The authors in [27] proposed a moving target defense was implemented in SDN by incorporating an adaptive IP shuffling technique to deceive attackers from the reconnaissance stage itself, and for detecting attacks, a lightweight CNN is used. Initially, data packet sampling (the packets related to both normal and forged packets) and preprocessing (where packets are transfigured into a matrix) are taken into account as inputs for the CNN detector. Here, IP shuffling is performed in two cases. To be more specific, shuffling of IP addresses takes place whenever an event is identified by a CNN detector as well as at a predetermined time interval to avoid being missed from an alert.

2.2. Non-MTD based DDoS detection

Port scanning attacks and DDoS attacks are mitigated using long short-term memory and fuzzy systems was proposed by authors in [28] which deals with classification, anomalies identification, and attack mitigation. For this, they have used a dataset named CICDDoS 2019. Initially, network traffics are analyzed by considering Source IP/Port, Destination IP/port, Bits per Sec, etc. using LSTM as well as by incorporating Shannon entropy. On the other hand, intrusions are identified using fuzzy logic and eventually, anomalies are pinpointed using attack mitigation schemes. The authors in [29] proposed a route hopping is

Table 3
Limitations of existing works.

Approaches	References	Objective	Methods/Algorithms	Limitations
MTD based DDoS Detection	[21]	MTD based on many security oriented parameters	Three type security parameters such as path, address, and attack	• Lack of ensuring service availability • Poor QoS
	[22]	Cross fire DDoS detection using MTD	Intent based MTD	• Lack of logging the behaviour of the attacker
	[23]	RL based MTD for attack detection in SDN	Single agent RL algorithm	• Less detection accuracy • Less realistic than expected
	[24]	Game theory based MTD for thwarting security threats	Flipit differential game model	• High malicious • Easily vulnerable to more security threats
MTD based DDoS Detection (continue)	[25]	Preserving DNS privacy using MTD	Reinforcement algorithm	• Time consuming • Lack of reinforcement learning parameters • Inefficient container placement
	[26]	MTD for cyber-attack detection based on markov game	Markov Game Model	• High complexity • Highly vulnerable to security threats
	[27]	Deep learning based MTD for cyber-attack detection	Convolutional Neural Network	• Increased computation overhead • High complexity
Non-MTD based DDoS Detection	[28]	Deep learning-based anomaly detection in SDN	Long Short-Term Memory and Fuzzy Logic	• Increased time consumption • Less network scalability
	[29]	RL based secure route mutation	Deep Q-Learning	• Need high data • High latency
	[30]	Entropy based DDoS detection in SDN	Infuse both entropy and threshold techniques	• Not mitigating the DDoS consequences • High computation overhead
	[31]	Slow distributed denial of service attack detection for SDN	Considering several features of slow DDoS detection	• Static attack surface leads to high attack vulnerability • High detection time
	[32]	Machine learning based DDoS detection in SDN	K-Nearest Neighbours	• Single point of failure • Lack of feature of DDoS detection
	[33]	To detect and defence against DDoS by time series analysis	Time series analysis based on IP	• High computation overhead • High security issues
	[34]	To defence against DDoS using deep learning	General Adversarial Networks	• Lack of security in detection surface
	[35]	Blockchain based packet parsing in SDN	Blockchain technology	• Lack of control plane security • High illegitimate traffic

performed using a context-based Q learning approach. This is capable of fine-tuning the route hopping and learning adaptively by improving the convergence rate of learning and by reducing processing overheads. The purpose behind using this approach is to improvise the defense activities. This approach learns the strategies of attackers to ensure the best route hopping with the help of Markov decision-making processes. Besides, for realizing network conditions precisely, a context-aware module is utilized in this paper.

DDoS attacks in SDN were detected through a fused threshold technique by [30] in which threshold values are diminished and attack detection speed is increased. Here, fused threshold techniques encompass the log energy threshold and information threshold where log energy has significant speed in detecting attacks when compared to information threshold. yet it has been limited to understandable. For the sake of this limitation, they have fused both entropy techniques. For recognizing DDoS attacks, the diversity of events that takes place in the network is considered. The authors in this work [31] introduced a distributed denial of service attacks that influence the network with the least traffic was investigated. Here, two class of slow DDoS attack has been illustrated such as forwarding rules manipulation at the data plane and switches overloading attack which gets exhausted most slowly. In this case, attackers send data packets around 38 to 40 per second instead of sending a bulk number of data packets to the targeted switches. Here, this type of DDoS attack is mitigated by considering different aspects such as observation of RAM/CPU level of switches and controllers, the installation rate of flow rules, identification of unpaired flows, and rule timeout configurations.

The authors in [32] have performed DDoS detection and mitigation by using a triggering framework and ML algorithms. Primarily, first-level intrusion detection is performed at the data layer where switches point out the suspicious flows using a triggering framework by considering packets in messages. Here, the combination of KNN and K-means is taken into account for classifying the suspicious flows at the controller level, and based on this SDN controllers take equivalent remedies. For this, controllers extract the traffic features, detect the attack and propagate rules to the data plane. In [33], the authors have utilized a time series investigation to detect and mitigate the DDoS attacks in the SDN environment by considering spontaneous network changes. Here, a model which is aware of future network traffic is employed. Initially, flow table features are extracted for investigating the time series by the SDN controller which looks into the fields of irregular IP addresses. In addition to that, for detecting network changes, different techniques such as adaptive entropy, chaos theory, and filter are preferred in this work.

In [34], the authors have used GAN for training the SDN environment to detect distributed denial of attacks. By training the system with the help of GAN, they have realized low sensitivity when an attack takes place. Here, both anomalies and signature-based attackers were analyzed using the dataset named CICDDoS 2019. Initially, the controller obtains the details related to flows from the forwarding devices then data packets are processed. Besides GAN is used to detect the anomaly patterns to identify the DDoS attack occurrence. As a result of DDoS attack detection, mitigation (designated at the application layer) is taken place correspondingly. The authors in [35] have presented a framework that depicts packer parser which is aided with blockchain to offer security at the data layer. The packet parser at the data plane is analyzing the data packets with the help of the SDN controller. For identifying the forged attack flows, here a correlation technique is taken into account. Here 5 types of attacks were considered such as reconnaissance attack, root attack, remote attacks, denial of services, and other normal attacks. Eventually, the P4 programming language is used to offer the safety measures in the data plane.

3. Problem statement

The main problem considered in this work are poor security measures and inefficient management of data. However, other than that some of the specific problems faced by the state of the works and its corresponding solutions are provided below,

Existing Problems Background: In [36], proposed game theory model based MTD to ensure security in SDN. Here, trilateral game analysis was used in which users, attackers, and defenders are involved. The markov decision process was adopted to select the best MTD approach. Authors in [37], adopts machine learning algorithm to detect low-rate distributed denial of service attacks in flexible SDN environment. Here both the intrusion detection and prevention were performed by considering flow-based features. An efficient MTD was proposed for the SDN environment by considering the market utility was proposed by authors in [38]. For application migration, market-based optimization principles (VM market economics) are taken into account to choose the best VM location by considering both users QoE and cloud resource utilization. After migration, the reactive and proactive MTD was performed. The problems encountered by these approaches are mentioned as below,

- From the above works, users of the network are considered to be legitimate ones yet there might be a chance to be compromised by attackers. So, the admission of both malign and benign users results in weak security as well as increases the high congestion rate.
- The attack surfaces are only prevented by hopping port and IP addresses whereas detection surface is left as static, hence it can be recognized by potential hackers so that it will result in security issues.
- All the attacker behaviours are characterized and detected. However, the absence of logging attacker behaviours will significantly increase the computational overhead by frequently analysing the known patterns as well as resulting in ineffective prevention.
- Flow investigation is done for HTTP flows but in real time, a network can get different service-oriented flows thus this classification is limited and inefficient as adversaries can intrude in the network through different flows so that less attack detection accuracy exists here

Authors in [39], resist against reconnaissance attack by providing security to the controller and data plane using proactive and reactive approaches. Here, control layer security is ensured by deploying replica controllers and data layer security is ensured by IP hopping and port hopping. On the other hand, event-based attacks are tackled by using decoy nodes, which is used to respond the malicious inbound traffics. The problems faced by this work are,

- Here, security is offered at data layer by changing the port numbers and IP addresses Yet, improper switches utilization causes high energy consumption. Besides, static data plane configuration can be vulnerable to overloading attacks.
- The multiple and replica controllers are used to offer security. However, the static configurations of the SDN controllers can highly rely on security issues thus attackers will learn the environment so that they can surpass the security schemes.

In [40], proposed graph based moving target defence is SDN. The purpose of using GMTD which is a hierarchical multi-tier graph 3 to evaluate the probability of hosts being compromised for highly prioritizing them. In addition to that, critical resources in the network are identified as well as vulnerable traversing paths are also predicted with help of the same graph. The problems encountered by this approach are,

- The implementation of MTD in this work is offered by considering critical resources probability rather than considering event-based MTD thus it could lead to ineffective MTD implementation and less accuracy.
- The data plane organization is not focused on which scalability issues in terms of effective resource utilization as well as energy consumption will significantly increase

Research Solutions: The problems faced by the existing works are addressed by the proposed solution which are illustrated as follows. Firstly, the users in the environment are authenticated to the blockchain to reduce the primary vulnerable threats caused by malicious network traffic. The proposed work adopts NTRU based Nyberg Rueppel Digital Signature Algorithm (NTRU NR-DSA) for authentication which considers MAC address, ID, finger vein biometric, and a picture password. Only the authenticated users' flows are forwarded to the forwarding layer where the optimization-based forwarding device activation is performed. The Forwarding Layer Organizer (FLO) manages the forwarding layer by activating and idling the switch using the Enhanced Spotted Hyena Optimization (ESHO) algorithm based on the metrics such as residual resource, trust rate, history of successful packet transmission, and high energy. Further, the first level of IDS is performed in which flows are classified as normal malicious, and suspicious using a deep learning algorithm Finally, in the controller layer the controllers are constructed in DAG structure to ensure scalability. The scalable environment ensures BSDN-HMTD (i.e., time and event based MTD) using deep learning in which both the attack and detection surface is changed dynamically. For both MTD Quantum Convolutional Neural Network (QCNN) is adopted. Further, the pattern of the attackers is logged securely in the blockchain to ensure security.

Proposed work

3.1. System model

We aim to enable security to the SDN environment using MTD and blockchain technology against DDoS attacks. Whereas to provide a robust and secure framework, there must be a need of set of apparatuses to achieve the proposed aim. This sub-section, illustrates the proposed architecture description on BSDN-HMTD. Fig. 1. shows the proposed BSDN-HMTD architecture in detail. The proposed work is composed of three layers such as user layer, forwarding layer, and control layer. The responsibilities of each layer are provided as below,

(i) **User Layer:** In user layer, both the normal and malicious users are resided. Generally, users in the user layer tries to continuously seeks services to the applications by utilizing SDN technology.
(ii) **Forwarding Layer:** In this layer, SDN switches are involved. The proposed work utilizes the SDN switches for flow forwarding, and first level intrusion detection. This layer is wholly managed by the Forwarding Layer Organizer (FLO) who aims to reduce the energy consumption issues among the SDN switches. Further, the virtual switches deployment in case of overloading, and honeypot placement during flow migration also managed by the FLO.
(iii) **Control Lay**er: In control layer, multiple controllers are presented in which global controller is used to manage the rest of the local controllers. The local controller is responsible for proactive MTD whereas the global controller is to perform reactive MTD (i.e., second level IDS). Here, the controllers are placed and constructed in form of a Directed Acyclic Graph (DAG) by the global to improve the scalability and connectivity respectively.

In addition, the blockchain is juxtaposed with all the layers to ensure the tamper-proof and privacy of the user data. For all the transactions including user credentials, first level IDS results from the forwarding layer, and MTD results from the controller layer, attacker pattern logging is securely stored in the blockchain. The benefits of utilizing blockchain for logging and data privacy are provided as below,

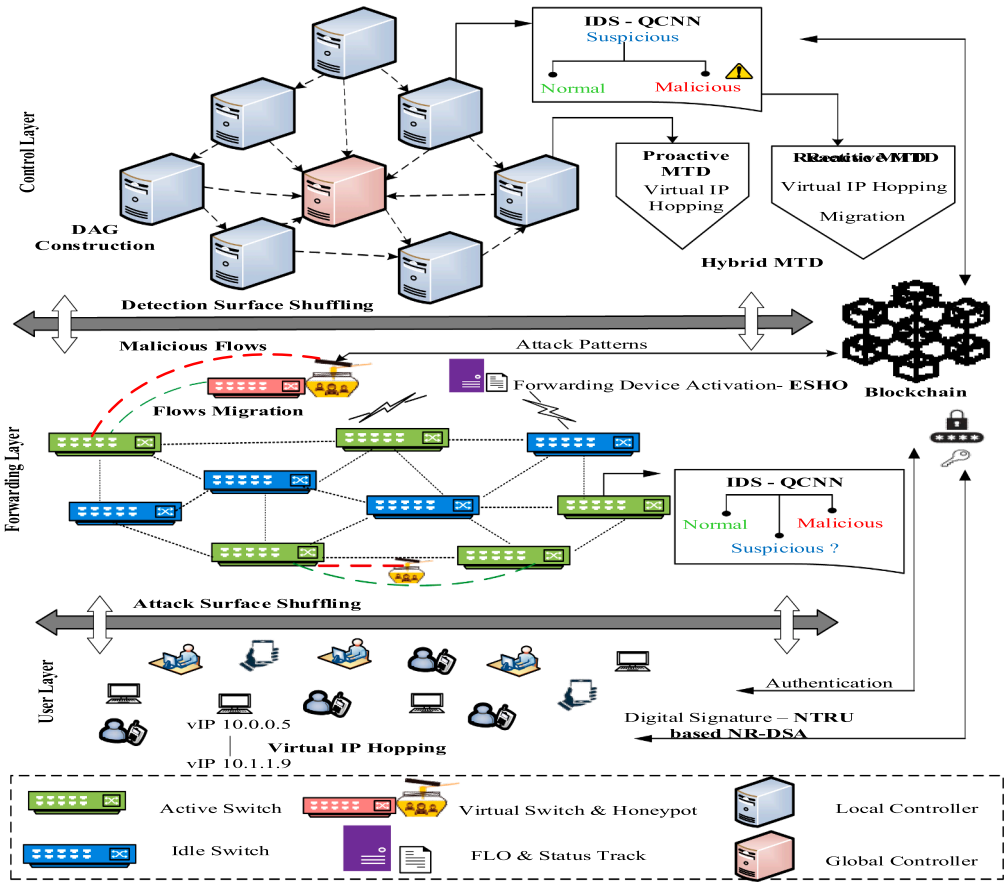

Fig. 1. Overall Architecture of BSDN-HMTD.

- Real Time Logging: Blockchain enables real time logging facility for effectively monitoring the network activities. With that, the security incidents are quickly responded that amplifies the overall network security.
- Trustless Environment: The insiders threats and potential corruption are resolved by utilizing consensus by replacing centralized trusted authority for data logging and transaction which makes the blockchain as trustless environment.
- Privacy Preservation: Even though there is transparent transaction in a blockchain, the transaction identities are pseudonymous which are effectively protected by advanced cryptographic functions.
- Immutability and Data Integrity: The blockchain immutability can be regarded as the data cannot be altered or modified. The immutable nature enables blockchain to be tamper-proof for activity logging in terms of packet validation, and authentication.

Apart from the architectural design, the proposed work made some assumptions to enhance the reliability of our framework. Some of the assumptions made by the proposed work are given below,

- It is assumed that the number of malicious and legitimate users are dynamically varied based on the network conditions.
- The attacker in the proposed environment may launch attacks in either any of the three layers (i.e., user, forwarding, and control layers respectively).
- The controller in the proposed work cannot be compromised to launch illegitimate activities rather, it is possible to take down the controllers by DDoS attacks.
- It is assumed that, the FLO is considered as the completely trusted entity and cannot be compromised or taken down by anyone.

Finally, the last assumption is the communication among the SDN environment are held over a secure channel and cannot be eavesdrop by anyone.

3.2. Threat model

This sub-section illustrates the proposed threat model in which the behaviour of DDoS attackers and the defences technique of the proposed work are explained. The diagrammatic representation of the threat model is realized in Fig. 2. By considering the above-mentioned system model and assumptions, the threat model is designed which is presented as follows,

(i) Behaviour of DDoS Attacker: The malicious tactical attackers aim to take down the resource in the SDN network to get sensitive information about the user and network privacy. Initially, the attacker tries to learn the environment and its sensitive vulnerable paths. After obtaining the desired information about the network (i.e., a DDoS attacker scans the IP of the vulnerable users, and ports of the vulnerable entities to launch attacks). From that learning, the attacker poses attacks which can be listed as,

- An Attacker may compromise the normal users and utilize them as bots to take down the network entities by launching continuous traffic.
- An attacker may sniff the network paths to capture the data traffics (i.e., passive attacks) to inject malicious information to compromise the network entity (i.e. SDN switch). The compromised entity acts as a bot for DDoS attackers and performs malicious activities.
- An attacker may directly launch the attack on vulnerable network entities (i.e., SDN switches, and controllers) by pinging them to malicious hosts (i.e., active attacks).

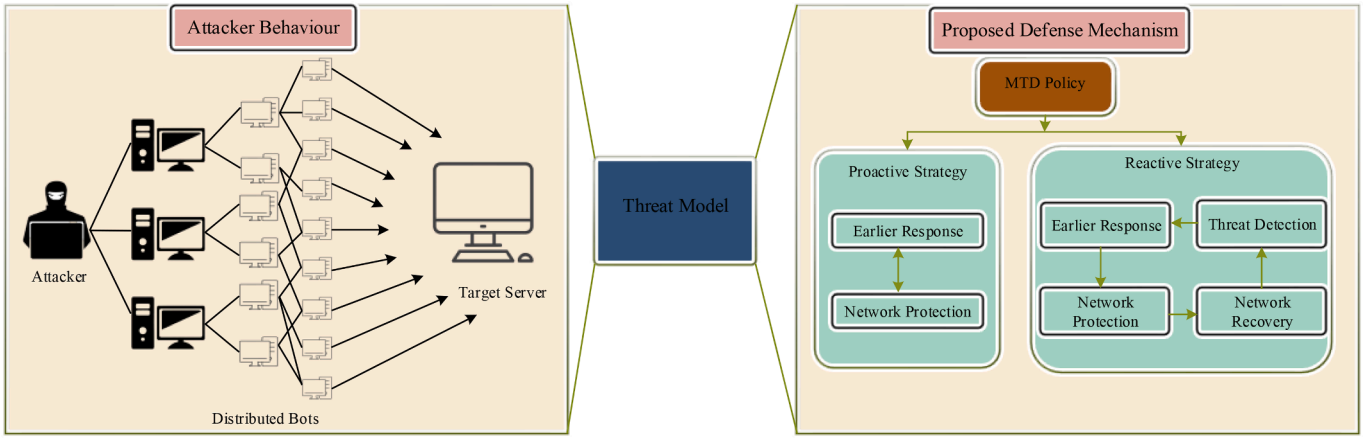

Fig. 2. Threat Model of the Proposed BSDN-HMTD.

(ii) Proposed Défense Technique: The proposed defences system detects and mitigates the attack launched by the DDoS attacker. The proposed work ensures security to all layers in the SDN environment. To be cautious, the proposed work ensures the authenticity of the users in the user layer to reduce the primary vulnerable threats and malicious traffics. In the forwarding layer, the first level IDS is performed to classify the user flows as normal, malicious, and suspicious. In addition, secure and optimal flow migration is realized by deploying honey pots. Adopting, Moving Target Defence (MTD) in the controller layer to ensure the network dynamicity. The MTD is performed in two ways such as,

- ***Proactive MTD (Time based MTD):*** The flows from the SDN switches are continuously reconnaissance by the FLO. The monitored information are forwarded to local controller which performs virtual IP hopping to the hosts.
- ***Reactive MTD (Event based MTD):*** Once the event occurs, the packets from the users are validated by performing second level IDS and classified as normal and suspicious. Based on the result, the controllers perform virtual IP hopping and migration to change their detection surface.

3.3. User authentication

Users involved in the SDN network are authenticated through blockchain. For this purpose, we have considered MAC address, ID, finger vein biometric, and a picture password. Here the picture password is taken into account which is handy and very effective against hackers because users are the only person who will be aware of where the gestures are made. When compared to conventional password schemes picture passwords are highly secure. Further, NTRU NR-DSA is used to provide keys and signatures. It is resistant to some attacks that take place in signature algorithms like key recovery, and quantum attacks to ensure high security. The reason for adopting NTRU NR-DSA over other authentication methods are listed as below,

- Post Quantum Security: Since the NTRU cryptography are based on the lattice problems that can have capability to resolve quantum computer attacks. By resolving quantum attacks, the proposed authentication methods ensure robustness and longevity than the conventional methods.
- Strong Security Guarantees: The signed messages integrity can be guaranteed by NR-DSA which can also resist against forgery attacks. By fusing NR-DSA with NTRU, the cryptographic signatures are secured thereby securing the authentication process.
- Scalability: Different from the conventional centralized authentication methods, the NTRU NR-DSA methods robustly scale to dynamic SDN environment. By such scalability, our method maintains constant performance and also able to high dynamic load conditions during DDoS attacks.
- Integration with Blockchain: The utilization of DSA is effective and secure and cane fits well with the blockchain technology. Ther NTRU NR-DSA methods enable notable performance and security for this blockchain integration which requires frequent verification and signing.

On the whole, authentication eliminates the participation of unauthorized users in the network thus reducing the computational overhead and congestion rate. There are two phases in the user authentication stage which are the registration and authentication phases. The steps involved in registration and authentication are provided below,

(i) Registration Phase

- Step 1: Firstly, the users with their credentials $(\mathbb{C})$ including MAC address, ID, and finger vein biometric are registered to the blockchain which can be formulated as,

$$\mathbb{U}_{\mathbb{C}}^{i} \rightarrow B \tag{1}$$

where, $\mathbb{U}_{\mathbb{C}}^{i}$ represents the i-th user with a credential such as MAC address, ID, and finger vein biometric respectively, and B denotes the blockchain entity.

- Step 2: After acquiring the $\mathbb{C}$ from the users, a challenge is provided to the users for creating the public and private key pairs. The challenge is provided in form of a picture. A 5×5 grid display consists of various pictures (i.e., dolphins, flowers, football, etc..,). The user had to select any of the pictures from the display and was advised to not disclose the picture to anyone.
- Step 3: Once the picture is selected, the NTRU based NR-DSA algorithm generates signed public and private key pairs based on the $\mathbb{C}$ and parameters $(M, r, w, w1, H)$ that can be formulated as,

$$pri(x) = f_w^{-1}, g_r^{-1} \tag{2}$$

$$pub(x) = f_w^{-1}(x) * g(x) \tag{3}$$

where $pri(x)$, and $pub(x)$ denotes the private and public key respectively, H denotes the constraint of the norm with $H = \lceil r^2 M/4 \rceil$, w1 and w are the integers which satisfy $w > (2r^2 + 4r)w1 + r$, and $w \gg r$, $gcd(r, w) = 1$ and$gcd(r, M) = 1$ respectively, r is the prime number which is a smaller odd number, and M is the integer of the ring dimension.

- Step 4: After computing the pri(x), and pub(x), both are signed by the B to ensure the successful registration which can be formulated as,

$$B[E,(j,k) \in \mathbb{R} \times L_h\left(\frac{w}{2}-H,\frac{w}{2}-H\right)] \rightarrow \mathbb{U}_{\mathbb{C}}^{i} \quad (4)$$

Where, E, (j, k) represents the signature parameters, $\mathbb{R}$ is the polynomial ring of the proposed NTRU NR-DSA, h is the hash function used for hashing the digital signature and L is the convolution modular lattice.

(ii) Authentication Phase

Upon completing the successful registration, the users are authenticated to enter the network. Only, the users whose credentials are correctly verified are considered legitimate user and allowed to the network. The steps involved in authentication are,

- Step 5: The $\mathbb{U}_{\mathbb{C}}^{i}$ is authenticated via ID and $\mathbb{R} \times L_h\left(\frac{w}{2}-H,\frac{w}{2}-H\right)$ to the blockchain. The blockchain then performs a verification process by extracting the information from the signed key pairs to validate the already registered and current key pairs as,

$$\mathbb{R} \times L_h\left(\frac{w}{2}-H,\frac{w}{2}-H\right) = \left[\mathbb{R} \times L_h\left(\frac{w}{2}-H,\frac{w}{2}-H\right)\right]' \quad (5)$$

If both the key pairs are same, then the next level of verification is provided to the user else revoked at this stage.

- Step 6: Once the key pairs are verified, the blockchain further checks the user legitimacy by validating the picture password where the user registered during registration. The user is considered as a legitimate user $L(\mathbb{U}_{\mathbb{C}}^{i})$ validated only if the user enters the correct picture else revoked from the network.

Pseudocode for NTRU NR-DSA based User Authentication

Input: $\mathbb{C}$, and (M, r, w, w1, H)
Output: Legitimate User $L(\mathbb{U}_{\mathbb{C}}^{i})$
Registration Phase
\\ Private Key Generation \\
$f \in r\mathbb{R}\left(\frac{3}{2}\right)$ and $g \in \mathbb{R}\left(\frac{r}{2}\right)$ is the initial private key of $\mathbb{U}_{\mathbb{C}}^{i}$
The keys satisfying $f_w^{-1} * f = 1 (\mathrm{mod}\, w)$, $g_r^{-1} * g = 1 (\mathrm{mod}\, r)$
The pri(x) of $\mathbb{U}_{\mathbb{C}}^{I}$ is f_w^{-1}, g_r^{-1}
\\ Public Key Generation \\
The public key is generated based on f_w^{-1}, g_r^{-1}
$Pub(x) = f_w^{-1}(x) * g(x)$ is the public key of $\mathbb{U}_{\mathbb{C}}^{i}$
\\ Digital Signature Generation \\
Input provided as key pairs [pri(x), Pub(x)]
B chooses the transient key in random manner $z \in \exists(w1, w1)$
Calculate $E \equiv rz * Pub(x) + m (\mathrm{mod}\, w)$
Compute hash values $h(\mathscr{H}, z) = (i_r, j_r)$
Generate signature $E,(j,k) \in \mathbb{R} \times L_h\left(\frac{w}{2}-H,\frac{w}{2}-H\right)$
End
Authentication Phase
\\ Legitimacy Validation\\
B initially validates the signature parameters
Validates current hashed signature
If both are the same
Accepted for next step validation;
Else
Revoked
End
Validate the picture password
If both are the same
The user is authenticated;
Else
Revoked
End
End

The pseudocode provides the registration & authentication of the proposed NTRU NR-DSA algorithm. On the whole, the conventional authentication methods are purely centralized and often faced with single point of failure issues and highly exacerbating with DDoS attacks. In addition to that, the conventional authentication methods are highly subjected to authentication failures and delays as they cannot handle enormous traffic volume generated by attackers. So that, we utilize NTRU NR-DSA based distributed authentication method based on blockchain technology. By incorporating blockchain technology, the proposed work attain several specific advantages on authentication in DDoS attack scenarios such as decentralization, integrity and immutability, transparency, distributed trust, and resiliency to DDoS attacks. Overall, the proposed authentication method allows only authenticated users to validating and forwarding the user flows which makes adds difficulty to attackers to spoof the identities and malicious traffic injection.

3.4. Forwarding devices activation

In this layer, switches are the forwarding devices in which some of them are kept active as well as some of them kept idle for improving utility rates. This layer is managed by an FLO who is keeping the switches active and idle. The deployment of FLO is handled by the controllers. Here active switches are carefully chosen by FLO based on the ESHO algorithm considering residual resource, trust rate, history of successful packet transmission, and high energy. The proposed ESHO algorithm is capable of offering greater accuracy which improves the local & global search by probing for the best candidate results around each switch. The major reason for adopting EHSO over other optimization methods is that dynamic adaptability due to its balanced exploitation and exploration, robustness due to its local optima avoidance, efficiency due to its minimized energy consumption, scalability due to its decentralized nature, and enhanced attack mitigation rate due to its proactive defense mechanism. In this case, active switches only accept the flows whereas the rest of the switches are kept idle for reducing energy consumption. Here, the switches are the hunter whereas the user flows are the prey. The hunters with optimal features may serve as an active switch for the user flow.

Initially, the population of the switches is initialized as SW_i in which $i = 1, 2, \cdots, n$. The parameters are initialized such as residual resource, trust rate, history of successful packet transmission, and high energy collectively called as ∀ to set the number of maximum iterations. The solution set of the SW_i in vector space can be formulated as,

$$SW_i = \left[SW_i^1, SW_i^2, \cdots, SW_i^U\right] \quad (6)$$

where, SW_i^j is the j-th dimension of the switches, the cost function of the SW_i can be computed as $F(SW_i)$.

The fitness weight can be computed based on SW_i with considerably finest ∀ and nastiest ∀. The finest and nastiest switches are represented by И and Θ respectively. The weights are linear mathematical form as,

$$We_i = 1 - \left(\frac{F(SW_i) - И}{\Theta - И}\right) \quad (7)$$

The above equation denotes the weigh computation of every SW_i in a linear manner. Whereas, in real time the SW_i posses' non-linear behaviour. Hence, non-linear We_i can be computed for SW_i that can be formulated as,

$$We_i = 1 - \left(\frac{F(SW_i) - И}{\Theta - И}\right)^{powe} \quad (8)$$

From the above equation, powe denotes the non-linear weight of the switches. In EHSO, utilizing cost function the amount of solutions is given by the switch itself. When the powe function is greater, the weight of the fitness function also larger. The position of each SW_i can be formulated as,

$$N_i = \left[1 - \left(\frac{F(SW_i) - И}{\Theta - И}\right)^{powe}\right].se_{maxi} \quad (9)$$

where, N_i is the amount to optimal position the SW_i generate around itself to be selected as active switch, and se_{maxi} is the candidate around the SW_i for searching. During iteration, the radius of search space is decreased for reducing the search rate which can be formulated as,

$$SW_i^{new} = SW_i + \frac{rad(iter-1)}{iter} * ran(-1+1) \quad (10)$$

From the above equation, $rad(t-1)$ is the search radius of every SW_i to become an active switch for a while, and iter is the iterations. In this work, every SW_i performs a local search to become an active switch that can be formulated as

$$SW_i^{new} = И - (2.v.rand - v).|2.rand.И - SW_i| \quad (11)$$

where, SW_i^{new} is the finest position on a search space while SW_i is the present position of the switch. И is the finest position of the switch to become an active switch, and v is a parameter of the spotted hyena optimization algorithm.

Gaussian and V-conversion functions are the main functions to convert the EHSO solutions into binary. The reason for converting binary space is to ease the selection process. The binary value is represented as 0 and 1. Both the function (i.e., V-conversion and Gaussian) can be formulated as,

$$\mathfrak{T} = \left|\frac{2}{\pi}\arctan\left(\frac{2}{\pi}x\right)\right| \quad (12)$$

$$\mathfrak{T} = \frac{1}{1+e^{-x}} \quad (13)$$

Every SW_i a component can be taken as an input for binary conversion. The adoption of the Gaussian function for converting the SW_i problem space into binary can be formulated as,

$$SW_i^j = \begin{cases} 1 rand(0.1) < T(SW_i^j) \\ 0 rand(0.1) \mp T(SW_i^j) \end{cases} \quad (14)$$

Similarly, converting the problem space into a V-conversion function can be formulated as,

$$SW_i^j = \begin{cases} -SW_i^j rand(0.1) < T(SW_i^j) \\ SW_i^j rand(0.1) \mp T(SW_i^j) \end{cases} \quad (15)$$

Pseudocode for ESHO based Forwarding Device Activation

Input: $SW_i(i = 1, 2, .., n)$
Output: Optimal SW_i^j for activation
Start
Initialize the ∀ and iter
Generate a solution set using (6)
Determine the И and Θ based on ∀
Compute linear and non-linear weights using (7) and (8)
The candidate search factor for SW_i using (10)
The local search of every SW_i using (11)
Compute binary function for every problem space
Compute Gaussian binary function using (13)
Compute V-conversion binary function using (12)
If iter < max_{iter} do
  Perform eqn from (7) to (13)
Else
  Select the optimal SW_i^j with optimal ∀
End
End

The reason behind selective optimum switches activation is to obtain equal resource utilization, improve attack detection speed, lessen computational complexity and shuffle the attack surface. The pseudocode for ESHO based forwarding device activation is presented below.

After a certain threshold time, another set of optimum switches is selected and they are activated. The threshold time is set by the FLO based on the energy consumption, utilization of CPU of the active switch respectively during flow forwarding and detection which can be formulated as,

$$SW_i^j[Ene, Cpu] \rightarrow Th[\blacksquare]$$

where, Ene, Cpu is the energy consumption and CPU utilization of the active switch, and $Th[\blacksquare]$ denotes the threshold value. If any of the active switches, go below this threshold $SW_i^j \leq Th[\blacksquare]$ would be set as idle and a new set of switches $(S_1W_i^j)$ are activated and replaced. Simultaneously flows are migrated to the $S_1W_i^j$ from the former switches to avoid service disruptions. During migration, the honey pots are deployed by the FLO. The honeypots attract the attackers to perform malicious activities in their fake environment to log the patterns for avoiding passive and active attacks. The attackers' logs are stored in the blockchain.

Fig. 3 represents the EHSO based active switch selection and flow validation. The active switches accept and classify the flows (fL_i) in terms of normal, malicious, and suspicious by extracting the flow features which are listed below. The fL_i are processed in terms of bits/s or packets/s (i.e., $\sum_{i=1}^{|t|} fL_i^{bits}, \sum_{i=1}^{|t|} fL_i^{packets}$).

- SRC_IP- Source IP address ($\mathbb{U}_i^{SRC\_IP}$), from where the traffic comes from.
- DST_IP- Destination IP address ($\mathbb{U}_j^{DST\_IP}$), for which IP the traffic goes.
- SRC_Port- Source Port ($\mathbb{U}_i^{SRC_{Port}}$), is a header in the TCP/UDP packets to sent data.
- DST_Port- Destination Port ($\mathbb{U}_j^{DST_{Port}}$), it is also residing in the header of TCP/UPD packets.
- Percentage of flow directed to the single entity out of the total incoming flow can be formulated as,

$$\frac{fL_i(SW_i)}{\sum fL_i} \quad (16)$$

where, $fL_i(SW_i)$ is the single SDN entity, and $\sum fL_i$ is the total number of flows.

For flow classification, Quantum CNN (QCNN) is used. The hand-picked algorithm is effective in yielding the best accuracy even when few parameters are given for training. The extracted features are provided as input to the QCNN. The QCNN is composed of three phases such as data encoding of quantum phase, convolutional filters phase, and pooling phase. Data encoding of quantum inputs is the static phase whereas the convolutional filters phase, and pooling phase utilizes Q-gates to perform classification operations. For the convolutional filters phase, and pooling phases precise assumptions are made to solve the classification problem. To be clearer, the QCNN composed of circuits and quantum gates that helps to examine the input data. Those gates and circuits, firmly extracts the network complex features and allowed to capture subtle and anomaly patterns. Furthermore, the parallelism model (i.e. quantum entanglement and superposition) allows QCNN to process multiple data simultaneously with accurate identification of flow characteristics when compared to conventional CNNs. Finally, quantum measurements are applied to classify the flows in three classes as regular, malicious, and dubious flows respectively.

To convert the fL_i into the quantum state, data encoding is essential in which initial states are prepared. The proposed work adopts q-bit encoding method for input flow features. The input fL_i can be normal-

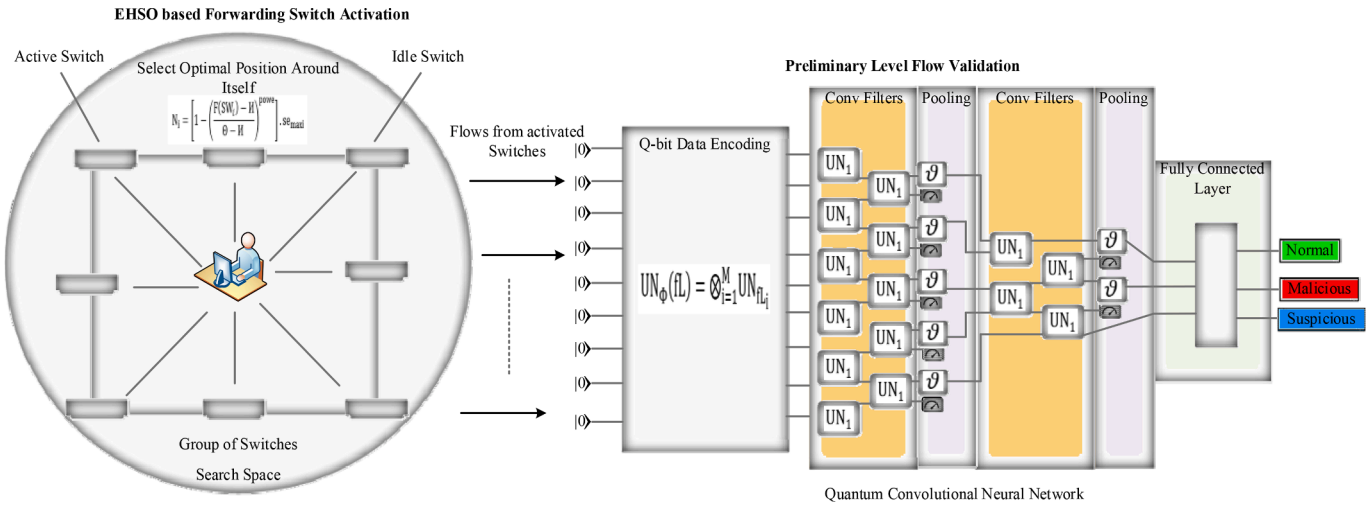

Fig. 3. EHSO based Forwarding Switch Activation & Flow Validation.

ized from 0toπ to solo q-bit as $\left|\Phi(fL_i)\rangle = \cos\left(\frac{fL_i}{2}\right)|0\rangle + \sin\left(\frac{fL_i}{2}\right)|1\rangle\right|$, where $i = 1, \cdots, M$. The input fL_i can be mapped as q-bit data of $fL_i = (fL_1, fL_2, \cdots, fL_N)^T$ which can be formulated as,

$$UN_\Phi(fL) = fL \in \mathfrak{w}^M \rightarrow |\Phi(fL)\rangle = \bigotimes_{i=1}^{M}\left(\cos\left(\frac{fL_i}{2}\right)|0\rangle + \sin\left(\frac{fL_i}{2}\right)|1\rangle\right) \tag{17}$$

From the above equation, UN is the unitary operator, and $fL_i \in |0, \pi)$ for all the flows. The final expression of the unitary operator can be formulated as,

$$UN_\Phi(fL) = \bigotimes_{i=1}^{M} UN_{fL_i} \tag{18}$$

The encoded data is used for flow classification. During classification, the encoded data is passed through three layers. In each layer, there are convolutional filters, and pooling layers are involved which perform quantum operations (qN). Every layer output is provided as the input for successive layer. The classification results are provided and results can be formulated as,

$$|\Phi_i(\vartheta_i)\rangle\langle\Phi_i(\vartheta_i)| = Tra_{\&_i}(UN_i(\vartheta_i)|\Phi_{i-1}\rangle\langle\Phi_{i-1}|UN_i(\vartheta_i)^\dagger) \tag{19}$$

From the above equation, ϑ denotes the parameters of the CNN model, $\&_i$ is the trace operator, UN_i is the unitary q-bit encoded operation that runs in every layer which consist of convolution filters and pooling parts. The final optimized classification results based on gaussian function with dissimilarity between actual and predicted traffic can be formulated as,

$$fL(y_t) = e^{\frac{-(x_t - y_t)^2}{2\varphi_t^2}} \tag{20}$$

where, x_t is the actual traffic, y_t is the forecasted traffic by QCNN, and φ_t is the threshold value for computing the maliciousness score (ב). Based on the ב, the user fL_i are classified into three classes which can be formulated as,

- Normal class if, $\sum_j^M fL_j(y_t) < ב$
- Malicious class if, $\sum_j^M fL_j(y_t) \geq ב$
- Suspicious class if, $\sum_j^M fL_j(y_t) < ב > \sum_j^M fL_j(y_t)$ (i.e., lies in between normal and suspicious class)

Then, the flows underlying the category of normal are routed to the requested services, and the malicious flows are dropped. Further, the suspicious flows are forwarded to the control layer for further investigation. The pseudocode for QCNN based flow classification is provided for better understanding to the readers.

QCNN based flow classification

```
Input: User fL_i
Output: Three classes (Normal, Malicious, Suspicious)
  For all the fL_i do
  Perform flow features extraction
  // Flow Classification //
  Perform q-bit encoding for fL_i from (17) and (18)
  Obtain QCNN based classification results (19)
  Compute dissimilarity of actual and predicted data using (20)
  If ∑_j^M fL_j(y_t) < ב, then
    User flow is normal (i.e., allowed for requested service)
  Elif ∑_j^M fL_j(y_t) ≥ ב, then
    User flow is malicious (i.e., dropped from the network)
  Else ∑_j^M fL_j(y_t) < ב > ∑_j^M fL_j(y_t), then
    User flow is suspicious (i.e., provided for anomaly detection)
End
End
```

3.5. BSDN-HMTD implementation

The controllers are constructed using the Directed Acyclic Graph (DAG) in which rearranging is very flexible, as well as it offers greater scalability, and easy adaptation and its speed will increase corresponding to the network enlargement. Here, reconstruction of controllers is performed by the global controller to reduce the attack success rates and to shuffle the detection surface when every set of switches gets activation. In this layer, a BSDN-HMTD implementation is performed by the decentralized controllers such as proactive (time-based) and reactive (event-based) to defeat the hacker strategies and increase the attack cost for them.

The time-based MTD implementation is incorporated to thwart the attacker's effort from the reconnaissance stage itself. During the reconnaissance stage, the flows from the switches are continuously monitored by the FLO. The monitored information is forwarded to the local controllers. With that monitored information, the dynamic IP hopping is taking place at a particular period which causes frustration as well as confusion to the attackers who are weaponized for an environment before the virtual IP (VIPs) hopping takes place. The local SDN controller (Loc_{Cont}), and the domain name server is responsible for providing the VIPs to the hosts. The shuffling of VIPs is managed by the Loc_{Cont}, and original IP address is managed by the domain name server. The procedure involved in virtual IP hopping are,

- The host/ user ($\mathbb{U}_i$) provides requests to the domain name server, the SDN switches of the corresponding $\mathbb{U}_i$ forwards to Loc_{Cont} in form of the packet in message.
- Upon receiving the packet in message, the Loc_{Cont} makes request to the domain name server for acquiring original IP address and stored in the database.
- After getting original IP of the host, the mapping of the original IP to VIP is taken place by the Loc_{Cont}. Since the mapping of the original IP to VIP is not static, the Loc_{Cont} manages the map track record of the host address. Once another domain name server request is arrived, based on the Time to Live value the shuffling delay and probability are done dynamic manner.
- Besides, the Loc_{Cont} manages the mapping table where the records about the original IP and shuffled VIPs records are maintained to provide the VIP in complex free manner.
- The hosts are assigned with VIPs in dynamic manner (i.e., not a fixed period, since attacker may sense the fixed period). The VIPs are selected randomly from the unused VIP pools so that the detection of VIP also impossible.
- Subsequently, the attacker poses their malicious eyes towards the benign host. In aware of that, the Loc_{Cont} proactively shuffles the IP of the host which waste the attacker reconnaissance cost and time. The pseudocode provides the proactive MTD in terms of VIP hopping for the host provided below.

```
Proactive MTD by VIP hopping for Host Security
SDN Network with normal users, malicious users, and controllers
For all monitored info do
  FLO reconnaissance all the flows before forwarding to the controller
  Forwards the monitored information to Loc_Cont
    Starts IP hopping
      Set the dynamic delay based on time to live of DNS request
  Acquire the original IP from the DNS server
  Update to VIP map record of Loc_Cont
  Randomly select the VIP from VIP pools
  Assign VIP to the hosts
End
End
```

Fig. 4 illustrates the BSDN-HMTD implementation of the proposed work. Further, the event-based MTD is performed as secondary level attack detection by considering the flows routed from the forwarding layer in terms of suspicious. These flows are immensely analysed by the controllers for labelling them as normal and malicious using QCNN. The procedure of analysing the suspicious packet is the same as the preliminary phase of attack detection. Only the features extracted from the suspicious flows are changed such as,

- Pac_size- Packet sizes of the DDoS attacker may vary randomly. The Pac_size of the attacker is differentiated by considering the normal packet size (i.e., 1514 bytes). The packet size also varies due to traffic

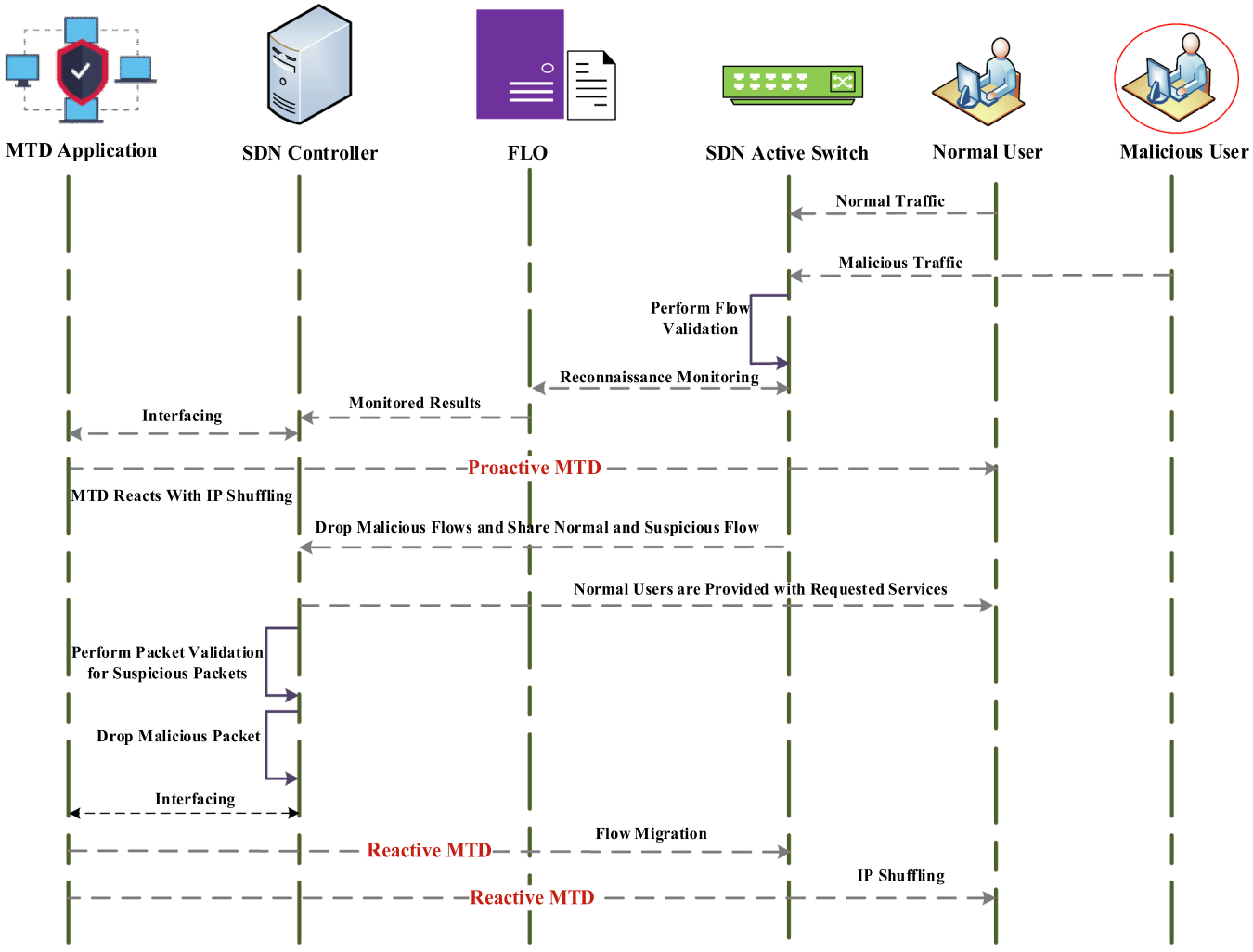

Fig. 4. BSDN-HMTD Implementation.

randomness. The proposed work also considers the packet size rate base on entropy function that can be formulated as,

$$\mathrm{ENT}'_{\mathbb{U}_i^{\mathrm{Pac_{size}}}}(\mathbf{x}) = \frac{\mathrm{ENT}_{\mathbb{U}_i^{\mathrm{Pac_{size}}}}(\mathbf{x})}{\mathrm{des_n}} \tag{21}$$

where, $\mathrm{ENT}_{\mathbb{U}_i^{\mathrm{Pac_{size}}}}(\mathbf{x})$ is the entropy of the user packet size, and $\mathrm{does_n}$ is the number of destination packet flows. Under, DDoS attacks the $\mathrm{des_n}$ is gradually decreasing which realizes the proposed detection framework to detect the packet as malicious packet.

- Relationship between packet arrival time and rate- The rate of packets arrived at the receiver at a certain period. Generally, the relationship between the packet arrival time and rate for the normal users is less which is expected in range of $(-1 \geq \mathrm{X} \geq 1)$. For malicious users, the relationship between them is higher than expected.
- Time to Live of the packets- It is defined as the packet lifetime in an SDN environment. The normal users have a defined time to live value while the malicious users have a higher time to live value than defined.

After extracting the malicious feature from the user packets. The extracted feature packets are provided as the input to CNN. Similar to the preliminary level attack detection, the proposed work performs q-bit encoding (17) to (18), quantized classification result (19), and dissimilarity rate computation (20). Finally, the anomaly score $(\aleph)$ is calculated to find the flows as normal and malicious can be formulated as,

- Normal Packet if, $\sum_{\mathrm{j}}^{\mathrm{M}} \mathrm{Pac_j}(\mathrm{y}_t) < \aleph$
- Malicious Packet if, $\sum_{\mathrm{j}}^{\mathrm{M}} \mathrm{Pac_j}(\mathrm{y}_t) \geq \aleph$

Once the flows are identified as malicious, the normal flows from that particular switch are migrated to the available active switches. On the other hand, the FLO tracks the status of available switches based on this status controllers takes respective actions. If the active switches possess no available resources, then new virtual switches are created by the controllers for avoiding switch overloading in terms of realizing effective event-based MTD implementation. The event based MTD includes virtual IP hopping for the hosts and flow migration of active switches. The procedure of virtual IP hopping is the same as the virtual IP hopping at the proactive stage. The migration of flows is taken place in which the flows are migrated either t an optimal active switch, or to a virtual switch. Furthermore, the possible scenarios for proactive MTD is regular network operation. At that time of regular network operation, the proactive MTD continuously changing the IP address and routing paths which makes difficult to the attackers to execute DDoS attacks. The possible scenario for reactive MTD is detected DDoS attacks. By positioning honeypots in the network environment, the malicious traffic can be effectively isolated to protect legitimate users.

The optimal active switch is selected based on three metrics such as

- **Trust-** The trust of the switch is acquired either directly from the corresponding switch or collected from the FLO. The trust value of the active switch must be higher.
- **Resource Availability-** The number of tasks currently running in the active switch is known as resource availability. The mathematical formulation of resource availability is provided as,

$$\mathrm{RA_{SW}} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{Tsk_i} \tag{22}$$

- **Load of the switch-** The number of tasks in the queue for the active switch determines the load of the switch. Mathematically, the load of the switch can be formulated as,

$$\mathrm{Lo_{SW}} = \frac{\mathrm{RA_{SW}}}{\mathrm{Tsk_{wait}}} \tag{23}$$

Based on the above three metrics, the switch migration is taken place. During searching, if no switch was found to be optimal (i.e., all the switches had no available resources in it) then the virtual switches (VSW) would be placed by the controller of the same configuration to balance the load and energy consumption. For improving the security during the migration stage, the virtual honeypot is created and deployed by the global controller for attracting hackers as it acts as vulnerable to exploitations. The reason behind deploying honeypots is that attackers will assume that they are attacking the target. But attacker behaviours are logged for reducing future attack success rates. Along with this, the behaviours are captured by the virtual honeypots and stored as transactions in the blockchain. The pseudocode for reactive MTD is provided below,

Reactive MTD by VIP hopping & Switch Migration

```
Initialize the SDN network with controllers, switches and users
For all suspicious flows do
  Perform suspicious feature extraction
  // Anomaly Packets Classification //
  Perform q-bit encoding for fL_i
  Obtain QCNN based classification results
  Compute dissimilarity of actual and predicted data using
  If ∑_j^M Pac_j(y_t) < ℵ then
    Normal Packet
  Else ∑_j^M Pac_j(y_t) ≥ ℵ then
    Malicious Packet
  End
End
// Perform VIP Hopping //
    Set the dynamic delay based on time to live of DNS request
    Acquire original IP from the DNS server
    Update to VIP map record of Loc_Cont
    Randomly select the VIP from VIP pools
  Assign VIP to the hosts
End
// Perform Switch Migration //
    Deploy virtual honey pots
    FLO tracks the switch status
    If optimal switch is found then
      Enable migration of flows to the selected switch
    Else
    Create a virtual switch by controller
    Enable migration of flows to the virtual switch
End
End
```

3.6. Effectiveness of proposed model on preventing DDOS attacks

(a) Enhanced Complexity for Attackers: The complexity of attackers gets significantly enhanced by adopting proactive MTD techniques (virtual IP hopping). By performing MTD, our proposed environment gets constantly changing due to the strategy adoption that enhances the attacker's operational cost and reduces the attack success rate.

(b) Reconnaissance Activities Disruption: The attacker reconnaissance efforts on the controller gets resolved by simultaneously changing the network topologies and IP addresses. By constantly changing the network topologies, the attackers are lacked with poor information gathering thereby preventing the well co-ordinated DDOS attacks.

(c) Amplified Network Resilience: By combining proactive MTD techniques and FLO, amplifies the network resilience. Our proposed work had an advanced ability to firmly responds the threats that reduces the DDOS impact and guarantees network availability.

4. Experimental results

The proposed BSDN-HMTD performance is evaluated in this section. The section undergoes four sub-sections including simulation setup,

comparative analysis, attacker cost analysis, and research summary. In every sub-section, the proposed work outperforms better than the existing works. The detailed description of every sub-section is as follows.

4.1. Simulation setup

The proposed BSDN-HMTD is adopted in an SDN environment to reduce the security threats caused by DDoS attackers. The proposed work is experimented with using Network Simulator 3.26 (NS-3.26) simulator tool by realizing several entities such as users, SDN switches, FLO, local and global controllers, and blockchain. To set up the environment, the system requirements and simulation parameters need to be configured and tuned. Fig. 5 illustrates the proposed simulation environment with all entities. Tables 4 and 5 represents the system requirements and simulation parameters of the proposed work.

4.2. Comparative analysis

This sub-section illustrates the comparative analysis of the proposed work in which the proposed work is validated with existing work such as Lowrate DDoS [37], FMU-MTD [38], and Trilateral MTD [36] in terms of validation metrics such as defender success rate (%), survival rate attack success rate (%), energy consumption (J), malicious traffic (bytes/second), and computation overhead (%).

(i) Comparison of Defender Success Rate.

The defender success rate is defined as the number of attempts (i.e., Scans) by the defender to detect the DDoS attack in the environment. The defender success rate (DSR) can be formulated as,

$$\mathrm{DSR} = \frac{\text{No. of Scans}}{\text{Malicious Attacks}} \times 100 \tag{24}$$

Fig. 6 shows the comparison of defender success rate with the number of scans. From the figure, it is represented that when the number of scans increases defender success rate also increases. Among that, our proposed

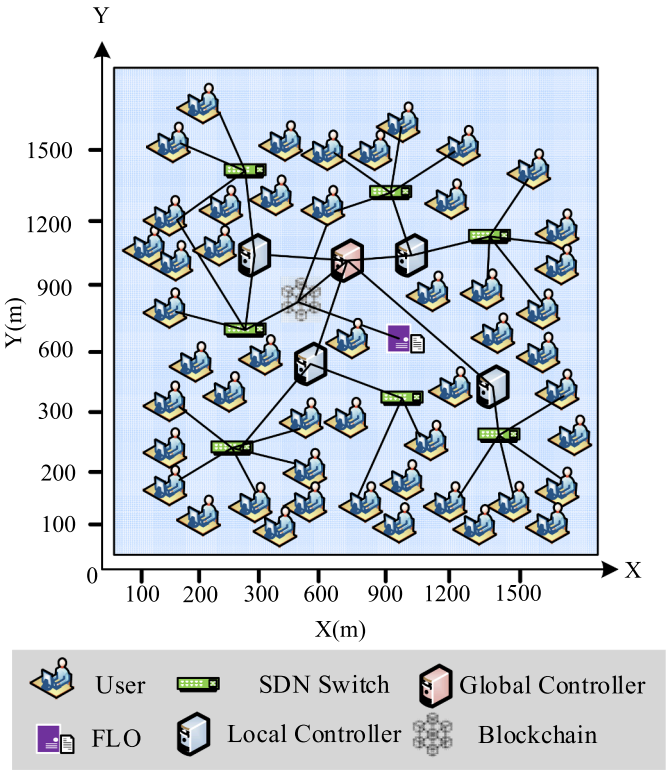

Fig. 5. Simulation Area of BSDN-HMTD.

Table 4
System requirements.

Hardware requirements	RAM	8 GB
	Hard disk	1 TB
Software requirements	Simulation tool	NS-3.26
	RAM OS	Ubuntu 14.04 LTS
	Central Processing Unit	Intel(R) Core (TM) i5-4590S CPU @ 3.00 GHz 3.00 GHz

Table 5
Simulation Parameters.

Simulation Parameters		Description
No. of Users		100
No. of SDN Switches		12
FLO		1
Controller	Local Controller	5
	Global Controller	1
Blockchain		1
Size of the Packet		2^6 KB
Time for Simulation		150 s
Area of Simulation		1500 × 1500
Rate of Traffic		20–140 packets/second
Mobility Model		Random Way Point
Packet Interval Time		0.2s
The bandwidth of the Link		8Mbps
Forwarding Delay		0.3ms

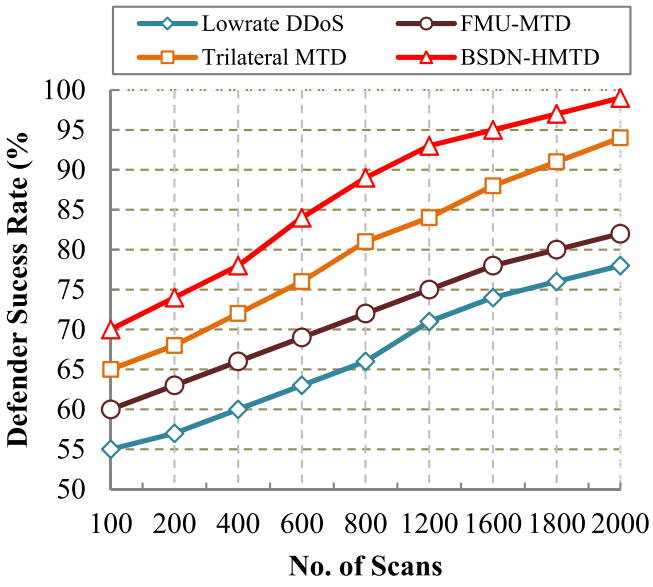

Fig. 6. No. of Scans Vs Defender Success Rate.

work achieves a higher defender success rate than the existing even though the number of scans increases. The reason for achieving a high defender success ratio of the proposed work is that the proposed work defends the attacks in the primary stage itself. The adoption of user authentication, and reconnaissance of flows in the earlier manner by FLO improves the defender success rate. For authentication, this works adopt the NTRU NR-DSA algorithm which validates the users based on security credentials such as MAC address, ID, finger vein biometric, and picture password. In addition to that, flows from the SDN users to SDN switches are monitored in terms of reconnaissance. The monitored results are forwarded to the local SDN controller performs proactive MTD thereby improving the defender success rate. In contrast, the existing works of Trilateral MTD and FMU-MTD limit with detection of the primary vulnerable threats thereby leading to less defender success rate as the rate of the attacker in the existing work environment gets increases.

The numerical results show that when the number of scans increases to 2000 the proposed work achieves a defender success rate of 99 %. The defender success rate of existing works trilateral MTD, FMU-MTD, and Lowrate DDoS for the same number of scans are 94 %, 82 %, and 78 % respectively which is higher than the existing works Trilateral MTD. On the whole, the proposed work achieves 12 %-14 % higher than the existing works.

(ii) Comparison of Survival Rate.

The survival rate is defined as the number of SDN users whose data are not explored over a time when the DDoS attacker poses a round of attacks. Generally, the survival rate of the node must be increased over time.

Fig. 7 shows the comparison of the survival rate of proposed BSDN-HMTD and existing works Trilateral MTD, FMU-MTD, and Lowrate DDoS respectively. In that figure, it is inferred that when the amount of time increases the survival rate gets decreases. The proposed work achieves a higher survival rate than the existing works. The reason behind for increasing success rate is that the proposed work performs virtual IP hopping of the users in two ways. At first, the Virtual IP hopping is performed based on proactive MTD (i.e., time based MTD in which the flows in the forwarding layer are proactively monitored), and reactive MTD (i.e., event based MTD) once the event has happened the controller detects the event as normal and malicious If the event is found to be malicious, then the virtual IP hopping for the normal user is taken place along with switch migration. By performing MTD two times, the nodes in the environment realize a higher survival rate. Whereas the existing works Trilateral MTD and FMU-MTD lacks with either performing MTD in reactive stage and proactive stage leads to poor survival rate.

The numerical results show that, the proposed work achieves higher success rate of 0.7 when the amount of time increases to 500 s. Whereas the existing works trilateral MTD, FMU-MTD, and Lowrate DDoS achieves less survival rate of 0.5, 0.2, and 0.05 respectively. The results show that the proposed work survival rate is 0.2–0.65 higher than the existing works.

(iii) Comparison of Attack Success Rate.

The robust and reliable defending methods of the proposed methods in DDoS attacks determine the attack success rate. It is defined as the ratio of attacks successfully held to the corresponding attacks. The formulation of the attack success rate is provided as below,

$$\mathrm{ASR} = \frac{\text{Successful Attacks}}{\text{No. of Attacks}} \times 100 \tag{25}$$

Fig. 8 shows the comparison of attack success rate Vs several DDoS attacks for the proposed and existing works respectively. From the graphical inference, the proposed work achieves a lesser attack success rate than the existing works. The reason behind such a lesser attack success rate is that the proposed work performs user authentication and forwarding switch activation respectively. In user authentication, the legitimacy of the user is validated by adopting a cryptographic algorithm named NTRU NR-DSA algorithm based on credentials such as MAC address, ID, and finger vein biometric. Further, the legitimacy is firmly validated by adopting the picture password method which greatly reduces the primary vulnerable threats. In forwarding switch activation, the attack surface is changed periodically by activating some set of optimal switches while the remaining switches are kept idle using the EHSO algorithm. By performing forwarding activation, the cause of successful DDoS accomplishment is greatly reduced as the attack surface is changed periodically, and also reduces energy consumption. In the case of existing work Trilateral MTD, only the detection surface is changed which may cause an attacker to easily reconnaissance the environment and perform successful attacks.

From the numerical results, the attack success rate of the proposed work decreased to 65 % even though the rate of DDoS in the environment increases to 15. In contrast, the existing works Trilateral MTD, FMU-MTD, and Low rate DDoS achieve a higher attack success rate of 77 %, 91 %, and 96 % respectively for the same number of DDoS. On the whole, it is inferred that the proposed work outperforms 12 %-31 % lesser than the existing works.

(iv) Comparison of Energy Consumption

The amount of energy consumed by an entity (i.e., switch) in an SDN environment is known as energy consumption. The formulation of energy consumption (E_C) is computed as,

$$\mathrm{E\_C} = \frac{\mathrm{E\_C}(\mathrm{SW}_i^j)}{\mathrm{ToT}_{\mathrm{E\_C}}} \tag{26}$$

Fig. 9 represents the energy consumption comparison of proposed and existing works. It is inferred that when the number of switches gets increased the rate of energy consumption also increased. From that, our proposed work achieves less energy consumption than the existing works. The major reason for realizing such less energy consumption is due to performing the forwarding switch activation method. The proposed work performs forwarding switch activation in the forwarding layer where the SDN switches reside. In that, the optimal switches from the set of switches are selected based on metrics such as residual resource, trust rate, history of successful packet transmission, and high

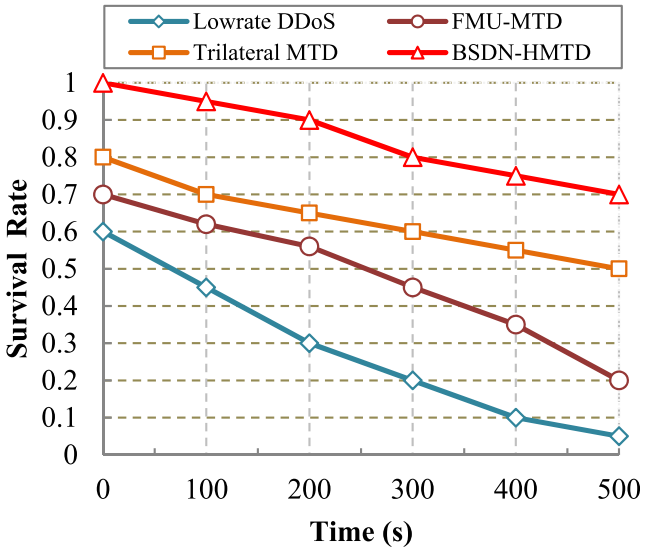

Fig. 7. Amount of Time Vs Survival Rate.

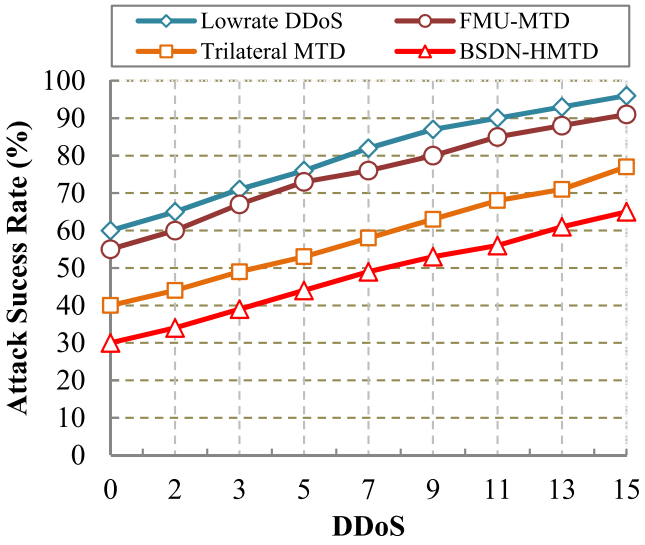

Fig. 8. DDoS Vs Attack Success Rate.

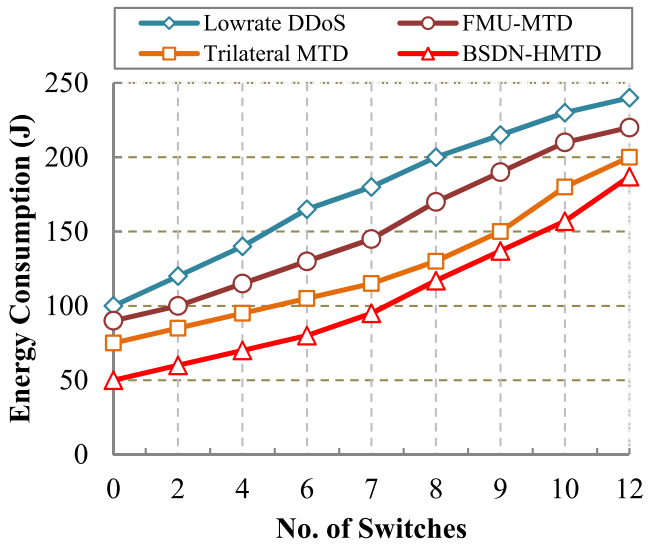

Fig. 9. No of Switches Vs Energy Consumption.

energy using the EHSO algorithm. The selected switches are activated for a certain period while other sets of switches are kept idle. Once the certain period got completed, another set of switches is activated which greatly reduces the energy consumption issue. In addition, security is also provided for the forwarding layer by performing honeypot based switch migration which also reduces energy consumption. In contrast to that, the existing work FMU-MTD adopts all the switches for flow forwarding leads to high energy consumption and complexity issues.

The numerical results show that the proposed work achieves lesser energy consumption of 187 J when the number of switches increases to 12. While, the existing works consume a higher amount of energy of 200 J, 220 J, and 240 J respectively for the same number of switches. The results confirm that the proposed work energy consumption rate is 13 J-43 J lesser than the existing works.

(v) Comparison of Malicious Traffic Rate.

Malicious traffic (MTR) is defined as the amount of traffic imposes by the malicious nodes in the environment on the actual traffic of the network. The formulation of malicious traffic is provided as below,

$$\mathrm{MTR} = \frac{\sum \mathrm{MT}^{rr}}{\mathrm{AT}^{rr}} \tag{27}$$

where, $\sum \mathrm{MT}^{rr}$ is the total malicious traffic in the network, and AT^{rr} is the actual network traffic.

Fig. 10 denotes the malicious traffic comparison or proposed and existing works. In the figure, when the number of scans gets increases the malicious traffic rate also increased. The proposed work realizes less malicious traffic than the existing works when the number of scans increases to the peak due to performing user authentication, flow classification, and packet classification. The proposed work adopts user metrics such as MAC address, ID, finger vein biometric, and picture password for registration and authentication respectively using the NTRU NR-DSA algorithm. During authentication, the users need to validate on two levels such as key verification and picture password verification. By ensuring user legitimacy, the rate of malicious users in the environment gets reduced thereby reducing malicious traffics in the network. Further malicious traffics are reduced by classifying the user data in terms of flows and packet in the forwarding and controller layer respectively. Both the flows and packet features are validated and classified using the QCNN algorithm. Whereas, the existing works Trilateral MTD, FMU-MTD, and Lowrate DDoS are not considered user legitimacy, and user data validation results in high malicious traffic in the network.

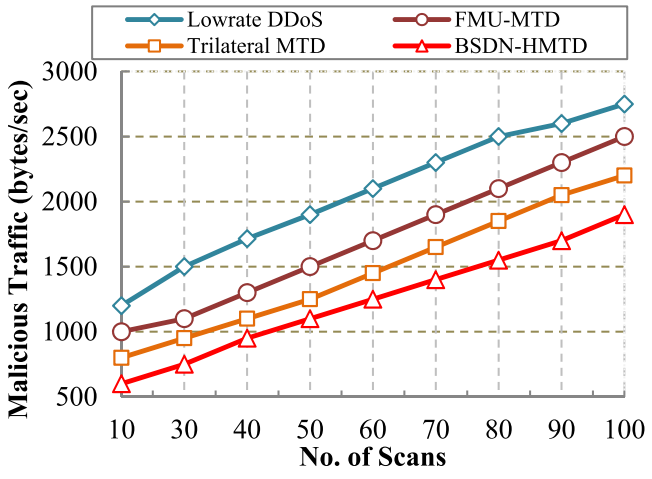

Fig. 10. No. of Scans Vs Malicious Traffic (bytes/sec).

From the numerical results, the proposed BSDN-HMTD achieves higher malicious traffic of 1900bytes/sec when the number of scans increases to 100. In contrast with that, the existing works Trilateral MTD, FM-MTD, and Low rate DDoS realized higher malicious traffic rates of 2200bytes/sec, 2500bytes/sec, and 2750bytes/sec respectively which is 300bytes/*sec*-850byte/sec lesser than the existing works. The final results show that the proposed work archives less traffic than the existing works.

(vi) Comparison of Computation Overhead.

The computation overhead is defined as the period required for a node to process the packets. The computation overhead is calculated in terms of computation time, bandwidth, etc..,. The formulation of computation overhead (COV) is,

$$\mathrm{COV} = \frac{\text{Packet reception time}}{\text{Node completion time}} \times 100\% \tag{28}$$

Fig. 11 represents the computation overhead comparison of proposed and existing works trilateral MTD, FMU-MTD, and Lowrate DDoS respectively. From the figure, it is shown that when the number of switches increases the computation overhead also increases. Among many existing methods, our proposed work achieves lesser computation overhead. The foremost reason for achieving such lesser computation overhead is due to energy efficiency and trusted flow forwarding, and secure migration respectively. The energy efficient and secure flow forwarding is done by adopting the forwarding switch activation

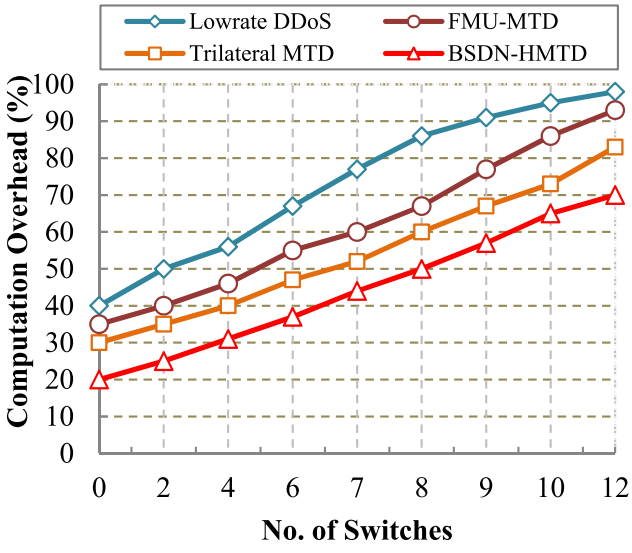

Fig. 11. No. of Switches Vs Computation Overhead.

method in which an optimal and trusted set of switches are activated for a certain period and the remaining are put idle until another period gets activated and vice versa. Further, dynamic changing of attack surface reduces the unwanted computation overhead in the network. For existing works Trilateral MTD, FMU-MTD, and Lowrate DDoS limits with optimal energy efficiency and secure flow forwarding lead to high computation overhead.

The numerical results show that the proposed work achieves less computation overhead of 70 % when the number of switches increased to 12. In a similar case, the existing works Trilateral MTD, FMU-MTD, and Low rate DDoS achieve lesser computation of 83 %, 93 %, and 98 % respectively. From the numerical results, the proposed work achieves 13 %–28 % more than the existing works.

4.3. Attacker cost analysis

The attacker invests some cost to thwart the defense mechanism to launch the DDoS attacks to gain fruitful information from the server/network. In this sub-section, the cost rate of the proposed BSDN-HMTD and existing works Trilateral MTD, and FWD-MTD are analyzed in the form of graphical results in Fig. 12 and numerical results. The attacker cost can be computed as,

$$\mathrm{Cost}_{\mathrm{Att}} = \mathrm{Detection}_{\mathrm{Traffic}} + \mathrm{M}_{\mathrm{Scans}} + \mathrm{MTD} + \mathrm{Auth} + \mathrm{FSA} \quad (29)$$

The above equation represents the computation of attacker cost. The attacker cost can be calculated by summing the traffic reconnaissance detection rate $\mathrm{Detection}_{\mathrm{Traffic}}$, number of scans by the attacker $\mathrm{M}_{\mathrm{Scans}}$, and the defensive techniques countered by the defenders such as MTD, authentication (Auth), and forward switch activation (FSA).

From the attacker cost analysis graph, it is shown that the comparison of attackers cost of proposed and existing methods such as trilateral MTD, and FMU-MTD respectively. The proposed work makes the attacker with high cost due to the following reasons,

- The proposed work periodically shuffles the attack surface by forwarding layer activation technique. The switches in the forwarding layer are shuffled periodically by FLO based on several metrics using an optimization algorithm. The periodic shuffling of attack surface thwarts the DDoS attacks in full swing thus increasing the cost of the attacker. In addition, the user flows are validated as normal, malicious, and suspicious using a deep learning algorithm.

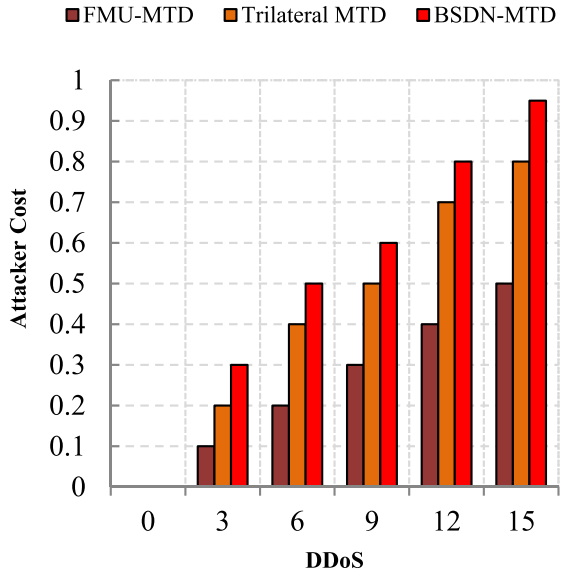

Fig. 12. DDoS Attack Rate Vs Attacker Cost.

- The detection surface (i.e., controllers) of the proposed work is also reconstructed by the global controller based on the forward switch activation which also increases the attacker cost.
- Further in the detection surface, proactive and reactive MTD posed by the proposed work confuses the attacker in terms of high cost and resource wastage to them. In proactive MTD, the flows from the SDN switches to the controller are secretly reconnaissance by the FLO and forwards the status of the controller. The local controller upon receiving the status, the controller performs Virtual IP hopping. On the other hand, in reactive MTD, the user suspicious packets are validated as normal or suspicious using deep learning. Based on the results, the controller performs Virtual IP hopping, and honeypot secure switch migration.

In contrast to the proposed work, the existing work trilateral MTD performs proactive and reactive MTD only for the detection surface by performing port and IP shuffling, and migration respectively. While the surfaces (i.e., attack and detection) are kept static paves an easy way for attackers to take down the surfaces. In FMU-MTD, only migration was performed between the controllers in a static environment leading to high vulnerability to attacks. The existing works even though provide a considerable MTD framework however they lack efficient approaches which make the attacker pose DDoS attacks at a low cost.

From the numerical results, when the rate of DDoS attacks by the attacker increases to 15, the proposed work defends the attacker strategies and thwarts the attacker ambiguous aim which increases the cost by 0.95. Whereas the existing works Trilateral MTD, and FU-MTD reduce the cost of the attacker which is lesser than the proposed work of 0.8 and 0.5 respectively. The proposed work reduces the cost of an attacker to 0.15–0.45 than the existing works.

4.4. Research summary and limitations

This sub-section provides a summary of the experimental results. The proposed work is simulated using NS-3.26 simulation tool with suitable hardware and software configurations. Further, the parameters for simulation are tuned to obtain better results. The proposed work outperforms better than the existing works by adopting secure and energy efficient proactive and reactive methodologies. Table 6 represents the average numerical results comparison of proposed and existing works. Table 7 shows the scalability comparison of the DAG with the conventional methods such as hierarchical and flat configuration. In addition, some of the research highlights are provided below,

- For eliminating the admission of external attackers into the network and for increasing the security measures, user authentication is performed which provides keys as well as digital signatures using the NTRU NR-DSA algorithm. This ensures that only authenticated user flows are validated and forwarded, leveraging blockchain technology to securely log authentication and validation data.
- To improve scalability and for reducing energy consumption, we have optimally selected switches that serve services only in the active state using the ESHO algorithm. By doing so, security is also enhanced because attackers' efforts are thwarted by making switches idle and active. The dynamic activation and deactivation of switches based on real-time demands are crucial in software-defined networking (SDN) environments.
- For accurately classifying the flows, QCNN is used in this work. Further, to ensure service availability of users at the attacked switches are migrated to the available switches and for attracting the attackers' virtual honeypots are created which are integrated with blockchain to avoid information violations. The blockchain ensures the integrity and immutability of the logged data related to honeypot activities.
- For reducing attack success rate, here we have performed a BSDN-HMTD implementation which takes place when a DDoS attack is

Table 6
Average numerical results comparison of proposed vs existing.

Validation Metrics	Low-rate DDoS [37]	FMU-MTD [38]	Trilateral MTD [36]	BSDN-HMTD
Defender Success rate (%)	66.6 ± 0.4	71.66 ± 0.2	79.88 ± 0.3	86.55 ± 0.2
Survival Rate	0.28 ± 0.5	0.48 ± 0.4	0.63 ± 0.2	0.85 ± 0.1
Attack Success Rate (%)	80 ± 0.3	75 ± 0.5	58.12 ± 0.1	47.88 ± 0.2
Energy Consumption (J)	176.66 ± 0.5	152.23 ± 0.4	126.12 ± 0.3	105.89 ± 0.4
Malicious Traffic Rate (bytes/sec)	2062.78 ± 0.1	1711.12 ± 0.3	1477.78 ± 0.4	1244.45 ± 0.5
Computation Overhead (%)	73.34 ± 0.4	62.12 ± 0.5	54.12 ± 0.3	44.34 ± 0.1

Table 7
Scalability Analysis Comparison.

Metric	Falt Configuration	Hierarchical Configuration	DAG Configuration
Throughput	As the network size increases throughput drops. It also lacks with scalability due to its simple structure.	Moderate decrease in throughput due to network expansion. Management is easy but suffers from bottleneck problems.	Easily scalable even though there is network growth. Routing can be performed well in distributed fashion.
Latency	Poor routing structure leads to enormous latency.	As the network size increase, latency increases. The hierarchical configuration allows centralized control leads to delays and single point of failure.	The latency attained is low even though on larger network size. DAG allows easy path finding leads to lesser latency.
Resource Utilization	Uneven	Unbalanced	Balanced
Performance	Low	Moderate	High

detected as well as a proactive MTD by continuously hopping the virtual IP addresses. Along with this, both the attack surface and detection surface are dynamically configured in terms of forwarding devices activation and controllers' construction using DAG. The combination of SDN and blockchain technology in this framework enhances the overall network security by providing robust, real-time defense mechanisms against sophisticated DDoS attacks.

Furthermore, the limitations of this work are listed as below,

- Complexity and Overhead: By integrating QCNN and blockchain, the proposed work faced with higher computational network overhead issues. The requirements of blockchain needed extraordinary latency and reduced throughput through cryptographic processes. Furthermore, QCNN also requires powerful computational resources which can also be introduced network overhead.
- Energy Consumption: The MTD processes introduced by the proactive and reactive MTD which includes continuous monitoring, frequent IP hopping, and validation can leads to enormous energy consumption when compared to straightforward methods.

5. Conclusion

The existing issues such as increased malicious traffic and energy consumption, and poor security measures proliferate the DDoS attack rate. The proposed BSDN-HMTD model resolves those issues by providing deep learning and blockchain based BSDN-HMTD solutions for the SDN environment. The proposed model is composed of three layers as user layer, the forwarding layer, and the control layer. In the user layer, the SDN users are authenticated by the blockchain using NTRU based NR-DSA algorithm based on metrics such as MAC address, ID, finger vein biometric, and a picture password. In the forwarding layer, the normal flows of the authenticated users are further validated by the SDN switches using the QCNN algorithm based on several flow features such as SRC_IP, DST_IP, SRC_Port, DST_Port, and the percentage of flow directed to the single entity out of the total incoming flow which classifies the flows as normal, malicious, and suspicious flows. Besides, the SDN switches are carefully activated and idled for the period for flow forwarding and validation using the EHSO algorithm. In the controller layer, the proactive and reactive MTD is taken place. For proactive MTD, the host virtual IP hopping is performed based on the monitored reconnaissance flows by the FLO. For reactive MTD, further, the suspicious flows are classified using the QCNN algorithm based on packet features such as packet size, the relationship between packet arrival time and rate, and time to live of the packets. Based on the results, the proposed work performs honeypot-based switch flow migration, and virtual IP hopping. All the transactions are securely stored in the blockchain. The proposed work is tested and simulated in NS-3.26 simulation tool and validated with several simulation metrics such as defender success rate, survival rate, attack success rate, energy consumption, malicious traffic rate, and computation overhead. The comparative results shows that the proposed work outperforms better than the existing works.

Theoretical Foundations

- Our research pinpoints with explicit discussion of blockchain technology and its uses in SDN environment security and privacy. The cryptographic algorithm (i.e. NTRU NR-DSA) and decentralized consensus mechanism are included in guaranteeing the data integrity.
- We have also elaborated the QCNN theoretical aspects by eluding its functionality, architecture, and advantages over existing works on flow classification.
- In addition to that, we have also discussed the EHSO algorithm in detailed manner by including their mathematical formulation, biological inspiration, and their advancements in forwarding switch activation by resolving energy consumption issues.

Supporting Evidence

- We have provided a detailed literature review section which compares existing and proposed work by highlighting their research gaps in terms of security and privacy. From that, out proposed work address those gaps.
- From simulation results we have provided more empirical evidence by comparing the proposed and existing works based on performance metrics. The effectiveness of proposed work is showcased in terms of tables and graphs with metrics such as QoS, energy efficiency, and security.

Funding

The work was funded by Deanship of Research and Graduate Studies at King Khalid University for funding through Large Research Project under grant number RGP2/179/45.

CRediT authorship contribution statement

Parthasarathy Ramadass: Conceptualization, Writing – original draft. **Raja shree Sekar:** Conceptualization, Writing – original draft. **Saravanan Srinivasan:** Methodology, Supervision, Writing – review & editing. **Sandeep Kumar Mathivanan:** Conceptualization, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Basu Dev Shivahare:** Methodology, Validation, Visualization. **Saurav Mallik:** Data curation, Investigation, Methodology, Validation, Visualization, Writing – review & editing. **Naim Ahmad:** Funding acquisition, Methodology, Supervision, Writing – review & editing. **Wade Ghribi:** Data curation, Funding acquisition, Methodology, Supervision, Writing – review & editing.

Conflict of Interest

The authors have declared that no competing interests exist.

Acknowledgement

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/179/45.

References

[1] Almadani B, Beg A, Mahmoud AS. DSF: A distributed SDN control plane framework for the east/west interface. IEEE Access 2021;9:26735–54.
[2] Jiménez MB, Fernández D, Rivadeneira JE, Bellido L, Cárdenas A. A survey of the main security issues and solutions for the SDN architecture. IEEE Access 2021;9:122016–38.
[3] Snehi M, Bhandari A. Vulnerability retrospection of security solutions for software-defined cyber-physical system against DDoS and IoT-DDoS attacks. Comput Sci Rev 2021;40:100371.
[4] Ahuja N, Singal G, Mukhopadhyay D, Kumar N. Automated DDOS attack detection in software defined networking. J Netw Comput Appl 2021;187:103108.
[5] Mahmood H, Mahmood D, Shaheen Q, Akhtar R, Changda W. S-DPS: An SDN-based DDoS protection system for smart grids. Secur Commun Netwk 2021. 6629098:1-6629098:19.
[6] Banerjee S, Chakraborty P. To detect the distributed denial-of-service attacks in SDN using machine learning algorithms. In: 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS); 2021. p. 966–71.
[7] Karki, Diwos & Dawadi, Babu. (2021). Machine Learning based DDoS Detection System in Software-Defined Networking.
[8] Valizadeh P, Taghinezhad-Niar A. DDoS attacks detection in multi-controller based software defined network. In: 2022 8th International Conference on Web Research (ICWR); 2022. p. 34–9.
[9] Huang Z, Huang X, Li J, Xue K, Sun Q, Lu J. LLDM: Low-latency DoS attack detection and mitigation in SDN. In: 2022 IEEE 23rd International Conference on High Performance Switching and Routing (HPSR); 2022. p. 169–74.
[10] Shayshab Azad KM, Hossain N, Islam MJ, Rahman A, Kabir S. Preventive determination and avoidance of DDoS attack with SDN over the IoT networks. In: 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI); 2021. p. 1–6.
[11] Yoon S, Cho J, Kim DS, Moore TJ, Free-Nelson F, Lim H. DESOLATER: deep reinforcement learning-based resource allocation and moving target defense deployment framework. IEEE Access 2021;9:70700–14.
[12] Xu X, Hu H, Liu Y, Zhang H, Chang D. An adaptive IP hopping approach for moving target defense using a light-weight CNN detector. Secur Commun Netwk 2021. 8848473:1-8848473:17.
[13] Mercado-Velázquez AA, Escamilla-Ambrosio PJ, Ortiz-Rodríguez F. A moving target defense strategy for internet of things cybersecurity. IEEE Access 2021;9:118406–18.
[14] Azab M, Samir MA, Samir E. "MystifY": A proactive Moving-Target Defense for a resilient SDN controller in Software Defined CPS. Comput Commun 2022;189:205–20.
[15] Ge M, Cho J, Kim D, Dixit G, Chen I. Proactive defense for internet-of-things: moving target defense with cyberdeception. ACM Trans Internet Technol (TOIT) 2022;22:1–31.
[16] Xu X, Hu H, Liu Y, Tan J, Zhang H, Song H. Moving target defense of routing randomization with deep reinforcement learning against eavesdropping attack. Digit Commun Networks 2022;8:373–87.
[17] Hyder MF, Waseemullah, Farooq M, Ahmed UT, Raza W. Towards enhancing the endpoint security using moving target defense (shuffle-based approach) in software defined networking. Eng Technol Appl Sci Res 2021.
[18] SongQun, YanZhenyu, & TanRui (2021). DeepMTD: Moving Target Defense for Deep Visual Sensing against Adversarial Examples.
[19] Jalowski Ł, Zmuda M, Rawski M. A survey on moving target defense for networks. A practical view. Electronics 2022.
[20] Santos RR, Viegas EK, Santin AO. Improving intrusion detection confidence through a moving target defense strategy. In: 2021 IEEE Global Communications Conference (GLOBECOM); 2021. p. 1–6.
[21] Sharma DP, Enoch SY, Cho J, Moore TJ, Free-Nelson F, Lim H, et al. Dynamic security metrics for software-defined network-based moving target defense. J Netw Comput Appl 2020;170:102805.
[22] Hyder MF, Fatima T. Towards crossfire distributed denial of service attack protection using intent-based moving target defense over software-defined networking. IEEE Access 2021;9:112792–804.
[23] Dass S, Siami Namin A. Reinforcement learning for generating secure configurations. Electronics 2021.
[24] Tan J, Zhang H, Zhang H, Hu H, Lei C, Qin Z. Optimal temporospatial strategy selection approach to moving target defense: A FlipIt differential game model. Comput Secur 2021;108:102342.
[25] Hyder MF, Ismail MA. Toward Domain Name System privacy enhancement using intent-based Moving Target Defense framework over software defined networks. Trans Emerg Telecommun Technol 2021;32.
[26] Ribeiro MA, Fonseca MSP, de Santi J. Detecting and mitigating DDoS attacks with moving target defense approach based on automated flow classification in SDN networks. Comput Secur 2023;134:103462.
[27] Xu X, Hu H, Liu Y, Zhang H, Chang D. An adaptive IP hopping approach for moving target defense using a light-weight CNN detector. Secur Commun Netwk 2021. 8848473:1-8848473:17.
[28] El Sayed MS, Le-Khac NA, Azer MA, Jurcut AD. A flow-based anomaly detection approach with feature selection method against ddos attacks in sdns. IEEE Trans Cognit Commun Netwrk 2022;8(4):1862–80.
[29] Xu C, Zhang T, Kuang X, Zhou Z, Yu S. Context-aware adaptive route mutation scheme: a reinforcement learning approach. IEEE Internet Things J 2021;8:13528–41.
[30] Fan C, Kaliyamurthy NM, Chen S, Jiang H, Zhou Y, Campbell CE. Detection of DDoS attacks in software defined networking using entropy. Appl Sci 2021.
[31] Swami R, Dave M, Ranga V. Mitigation of DDoS attack using moving target defense in SDN. Wirel Pers Commun 2023;131(4):2429–43.
[32] Kumar S, Kumar Keshari A. An effective DDOS attack mitigation of IoT using optimization-based adaptive security model. 112052 Knowl-Based Syst 2024.
[33] Alashhab AA, Zahid MS, Isyaku B, Elnour AA, Nagmeldin W, Abdelmaboud A, et al. Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model. IEEE Access 2024.
[34] Novaes MP, Carvalho LF, Lloret J, Proença ML. Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments. Future Gener Comput Syst 2021;125:156–67.
[35] Jmal R, Ghabri W, Guesmi R, Alshammari BM, Alshammari AS, Alsaif H. Distributed blockchain-SDN secure IoT system based on ANN to mitigate DDoS attacks. Appl Sci 2023;13(8):4953.
[36] Zhou Y, Cheng G, Jiang S, Zhao Y, Chen Z. Cost-effective moving target defense against DDoS attacks using trilateral game and multi-objective Markov decision processes. Comput Secur 2020;97:101976.
[37] Pérez-Díaz JA, Valdovinos IA, Choo KR, Zhu D. A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. IEEE Access 2020;8:155859–72.
[38] Debroy S, Calyam P, Nguyen M, Neupane RL, Mukherjee B, Eeralla AK, et al. Frequency-minimal utility-maximal moving target defense against DDoS in SDN-based systems. IEEE Trans Netw Serv Manag 2020;17:890–903.
[39] Hyder MF, Ismail MA. Securing control and data planes from reconnaissance attacks using distributed shadow controllers, reactive and proactive approaches. IEEE Access 2021;9:21881–94.
[40] Zhou Y, Cheng G, Ouyang Z, Chen Z. Resource-efficient low-rate ddos mitigation with moving target defense in edge clouds. IEEE Trans Netw Serv Manag 2024.