future internet

MDPI

Article

An Intrusion Detection System for 5G SDN Network Utilizing Binarized Deep Spiking Capsule Fire Hawk Neural Networks and Blockchain Technology

Nanavath Kiran Singh Nayak and Budhaditya Bhattacharyya *

School of Electronics Engineering, Vellore Institute of Technology, Vellore 632014, India; nanavathkiran.nayak2020@vitstudent.ac.in
* Correspondence: budhaditya@vit.ac.in

Abstract: The advent of 5G heralds unprecedented connectivity with high throughput and low latency for network users. Software-defined networking (SDN) plays a significant role in fulfilling these requirements. However, it poses substantial security challenges due to its inherent centralized management strategy. Moreover, SDN confronts limitations in handling malicious traffic under 5G's extensive data flow. To deal with these issues, this paper presents a novel intrusion detection system (IDS) designed for 5G SDN networks, leveraging the advanced capabilities of binarized deep spiking capsule fire hawk neural networks (BSHNN) and blockchain technology, which operates across multiple layers. Initially, the lightweight encryption algorithm (LEA) is used at the data acquisition layer to authenticate mobile users via trusted third parties. Followed by optimal switch selection using the mud-ring algorithm in the switch layer, and the data flow rules are secured by employing blockchain technology incorporating searchable encryption algorithms within the blockchain plane. The domain controller layer utilizes binarized deep spiking capsule fire hawk neural network (BSHNN) for real-time data packet classification, while the smart controller layer uses enhanced adapting hidden attribute-weighted naive bayes (EAWNB) to identify suspicious packets during data transmission. The experimental results show that the proposed technique outperforms the state-of-the-art approaches in terms of accuracy (98.02%), precision (96.40%), detection rate (96.41%), authentication time (16.2 s), throughput, delay, and packet loss ratio.

Keywords: 5G SDN; intrusion detection system; binarized deep spiking capsule fire hawk neural network; blockchain; cybersecurity

Citation: Nayak, N.K.S.; Bhattacharyya, B. An Intrusion Detection System for 5G SDN Network Utilizing Binarized Deep Spiking Capsule Fire Hawk Neural Networks and Blockchain Technology. *Future Internet* **2024**, *16*, 359. https://doi.org/10.3390/fi16100359

Academic Editor: Jose I. Moreno Novella

Received: 18 August 2024
Revised: 21 September 2024
Accepted: 24 September 2024
Published: 3 October 2024

1. Introduction

SDN centralizes device management within a specified area of the network. SDN controllers oversee communication among a wide array of Internet of Things (IoT) devices. However, integration of these systems brings forth certain challenges [1,2]. Primarily, enhancements in quality-of-service metrics such as throughput, packet loss rate, delay, and jitter are limited. Additionally, security concerns persist, encompassing various attack vectors, privacy issues, and unaddressed vulnerabilities [3]. Research attention has been directed towards intrusion detection systems (IDS) to classify security breaches within the SDN framework [4,5]. Addressing latency and extensive IoT traffic in SDN is achieved by leveraging the capabilities of 5G and offering rapid and adaptable support to end users. Despite the benefits, IoT devices encounter challenges concerning security and limited resources amid substantial data traffic [6]. Consequently, the evolving concept of IDS plays a vital role in these integrated paradigms. Methods such as multi-stage intrusion detection are applied to detect attack packets operating within the data plane, control plane, or both. Despite the capability of several IDS techniques to differentiate diverse attacks, a common issue is the prevalence of false alarms [7,8].

Artificial intelligence (AI) shows promise in achieving low false alarm rates while detecting malicious data packets network-wide. However, existing deep learning algorithms face challenges in detecting intrusions and new attack types, which are affected by high computational complexity, inadequate training data, and lengthy operation times [9]. This is especially problematic for resource-constrained IoT devices. Blockchain technology offers a lightweight, effective solution to address these issues. Blockchain's immutable nature ensures transparency and resistance to tampering [10–12], making it ideal for enhancing security and trustworthiness in 5G networks. In an SDN controller, transactions are executed in a single location due to the global architecture [13]. The main role of blockchain in this work is to enhance the security and trustworthiness of the 5G network, particularly in the context of intrusion detection and prevention. In an SDN controller, transactions are executed in a single location due to the global architecture [13]. It provides a decentralized ledger for registering and authenticating mobile users, ensuring only authorized devices access the network [14]. This research introduces an innovative intrusion detection system to address cyber-security challenges in advanced network environments, especially in 5G SDN networks where security is paramount.

The suggested work flow diagram consists of a data acquisition layer, where users register and authenticate using the lightweight encryption algorithm through a trusted third party (TTP), which verifies user identity and records it on the blockchain ledger for network access. At the switches layer, the optimal switches are selected using the mud-ring algorithm, and the data flow rules are secured by using the blockchain ledger and searchable encryption algorithms. Blockchain also helps maintain the integrity of network switches and flow tables by recording and verifying transactions and configurations. The domain controller layer utilizes binarized deep spiking capsule fire hawk neural network (BSHNN) for real-time data packet classification within the control plane layer, while the smart controller layer uses enhanced adapting hidden attribute-weighted naive bayes (EAWNB) to identify suspicious packets during data transmission to the user's application. The proposed system addresses resource limitations in SDN controllers and switches, ensuring network efficiency and reliability [15]. An integrated security system with five layers is proposed for the protection of mobile users and network integrity. The flow diagram of the proposed work is illustrated in Figure 1. The major contributions are:

- The utilization of a lightweight encryption algorithm (LEA) in the data acquisition layer ensures data security during mobile user authentication, enhancing network integrity and user privacy.
- Implementation of the Mud-Ring Algorithm (MRA) for optimal switch selection, preventing flow table overloading, and dynamically managing switch resources for efficient network operation.
- The deployment of blockchain technology with searchable encryption and decryption techniques ensures the authenticity and reliability of flow rules, improving network resilience against malicious modifications. This innovation significantly strengthens network security.
- The application of the BSHNN method at the domain controller layer enables real-time data packet classification and management of network traffic based on dynamic conditions and requirements.
- Employing an enhanced adapting hidden attribute-weighted naive bayes (EAWNB) approach for identifying suspicious packets during data transmission.

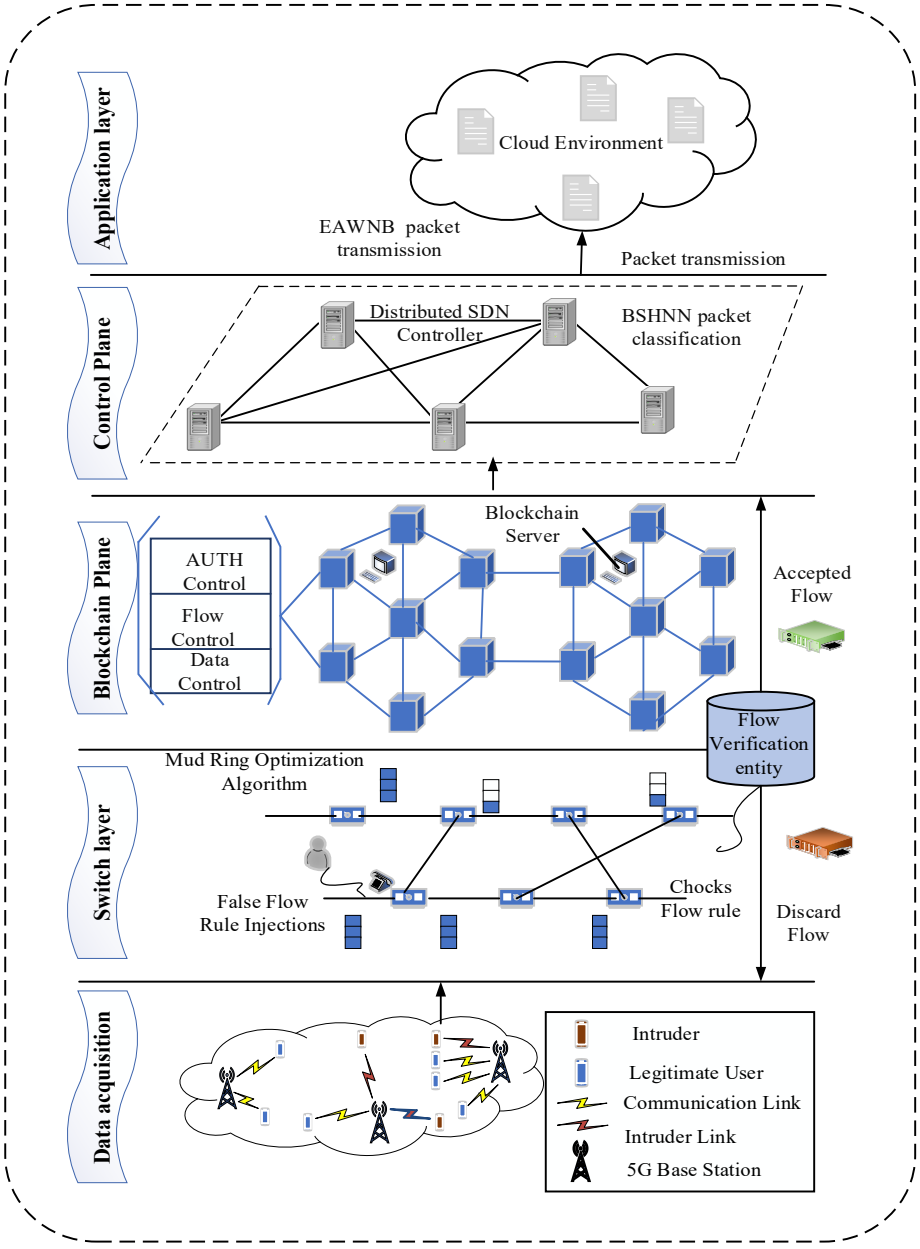

Figure 1. Flow diagram of the proposed work.

The remaining sections of the research work include: Section 2 gives an elaborate review related to intrusion detection in the SDN framework. Section 3 provides a detailed description of the proposed work, presenting comprehensive algorithms and mathematical models. Section 4 presents a concise discussion of simulations in the proposed work, along with a comparison to existing research. Section 5 concludes the research, outlining future scopes and implications.

2. Related Works

The recent study on intrusion detection systems in 5G SDN is evaluated in this section. Chaganti et al. [16] introduced a Long Short-Term Memory (LSTM)-based approach to improve IoT network security by integrating SDN technology, facing challenges adapting to evolving attacks and computational inefficiencies. Abdulqadder et al. [17] proposed the ML-IDP mechanisms for SDN and NFV-enabled 5G networks, despite potential performance impacts and reliance on training data quality. The study [18] proposes the use of the recurrent neural network classifier model, which incorporates LSTM and GRU, for

enhanced detection of DDoS attacks on SCADA systems, but its scalability beyond SCADA remains uncertain. The research paper [19] discusses improved intrusion detection systems (IDS) for detecting probe attacks, aiming to improve threat identification and response despite challenges in distinguishing benign from malicious threats. The study [20] introduces a reinforcement learning (RL) intrusion prevention system to tackle rank attacks in low-power IoT software-defined networks, addressing security challenges in applications such as smart cities, agriculture, and healthcare. RL-based approaches mitigate IoT rank attacks but face scalability issues. The article [21] presented a secure energy-efficient-based blockchain framework (SEE-BF) for the IoT networks, aiming to enhance network integrity, security, and energy efficiency. Blockchain models such as SEE-BF enhance IoT integrity but introduce latency. Saba et al. [22] introduced a CNN-based IoT security approach, achieving 92.85% and 99.51% accuracy on the NID and BoT-IoT datasets, respectively, demonstrating the potential for innovation in IoT device security. Yazdinejadna et al. [23] presented the Kangaroo-based IDS (KIDS), an optimization-based IDS designed for SDN. KIDS employs a zone-based structure for scalability, featuring flow-based and packet-based intrusion detection modules, promising SDN security but demanding computation.

Zainudin et al. [24] presented federated learning (FL)-based intrusion detection in 2023. Their approach targets low-complexity intrusion detection in SDN-enabled cyber-physical systems. The results show high accuracy, efficiency, and a streamlined model, albeit with computational demands on extensive datasets. Dat–Thinh et al. [25] presented MidSiot, a multiclass IoT IDS. It includes categorization, traffic differentiation, and attack identification stages, achieving an outstanding 99.68% average accuracy on common IoT attack types, surpassing existing IDS systems' performance. Serrano et al. [26] proposed the Blockchain Random Neural Network (BRNN) for IoT security, offering concealed user identity and decentralization. Despite computational complexity, BRNN enhances cybersecurity resilience and connectivity in both digital and physical security. Gharzadeh et al. [27] have developed a novel IoT anomaly detection method using a hybrid CNN for feature extraction and the Binary Multi-Objective Enhanced Capuchin Search Algorithm BME-CapSA for feature selection, but face computational challenges due to extensive security tests. Abdulqadder et al. [28] investigated the use of the Directed Acyclic Graph (DAG) blockchain to strengthen security in SDN 5G environments. It demonstrates that employing the DAG blockchain not only improves data integrity but also establishes a decentralized security approach. Almaraz–Rivera et al. [29] deployed machine learning to detect DDoS attacks targeting IoT devices. These models significantly improve the accuracy of detection and response mechanisms. Logeswari et al. [30] enrich the scientific discourse by proposing an IDS and mitigation system based on SDN and machine learning, which is unique but requires refinement for scalability, blockchain integration, and efficiency in 5G SDN and IoT. Thus, the proposed work in this paper facilitates advancements to the state-of-the-art (SOTA) by enhancing detection mechanisms, integrating blockchain for secure control, optimizing computational efficiency, strengthening SDN infrastructures and IoT security, and ensuring scalability. These innovations aim to address existing drawbacks and significantly improve IDS performance in 5G SDN networks.

3. Proposed Methodology

This section outlines the design of the proposed BSHNN approach, aimed at detecting and preventing diverse security attacks. Figure 1 provides a visualization of the proposed BSHNN within the SDN cloud of 5G networks. The data acquisition layer employs the lightweight encryption algorithm (LEA) for mobile user authentication. In the switches layer, the mud ring algorithm (MRA) is used for selecting the optimal switches, preventing flow table overloading attacks. It incorporates switch selection and flow rule security measures to uphold the integrity of network switches and flow tables. The blockchain-based approach with searchable encryption safeguards against the forged flow rules within the blockchain plane. At the control plane layer, the domain controller layer analyzes incoming data packets utilizing the binarized deep spiking capsule fire hawk neural

networks (BSHNN) method, while the smart controller layer uses enhanced adapting hidden attribute-weighted naive bayes (EAWNB) to identify suspicious packets during data transmission to the application layer.

3.1. LightWeight Encryption Algorithm

The lightweight encryption algorithm (LEA) is a 128-bit block cipher designed for high-speed environments such as big data and cloud computing, as well as lightweight environments such as IoT and mobile devices. It is ideal for devices with limited processing capabilities, embedded systems with memory constraints, and battery-powered devices such as sensors and IoT gadgets. It offers encryption and authentication capabilities, with a simple key schedule and operations that are computationally inexpensive. The key management mechanism involves a trusted third party (TTP) responsible for storing user credentials and generating unique secret keys for each mobile user upon registration. These keys are used for encryption and decryption during authentication with the AP. When a user registers with the network, the TTP generates a unique secret key for the user. This secret key is used to encrypt the user's data, including the MAC address, identity, and PUF information. The encrypted data is then transmitted to the AP, which uses the secret key to decrypt the data. If the decrypted data matches the expected values, the user is authenticated.

The pseudocode of the lightweight encryption algorithm begins with gathering essential user data such as MAC address, identity, password, PUF information, and location data. This data is initialized and requested for authentication, followed by preprocessing. The user is then registered with a TTP, which generates a unique secret key 'S_k'. This key is used to encrypt the user's data 'E_d', which is securely transmitted to the 5G AP. The AP decrypts the data using the secret key, verifying its validity. If the data is valid, the user is authenticated, enabling secure communication. If the data is invalid, the authentication is rejected, ensuring robust security throughout the process. In Algorithm 1 $G_{sk}()$ represent a key generation procedure. The TTP plays a pivotal role in the authentication process using LEA [31] for securing communication between mobile users and 5G APs. TTP stores mobile user credentials and generates public and private keys for each user, which are used to encrypt sensitive data. The encrypted data is then transmitted securely to the 5G AP, which uses the secret key to decrypt and verify the data. The overall steps involved in the authentication process are described in Algorithm 1.

Algorithm 1: LEA pseudocode for mobile user authentication

```
Input: 'D_i' (MAC address, identity, password, PUF information, location data)
Begin
Initialize 'D_i' /* "D_i" represents the data inputs */
Request 'D_i' for authentication
define preprocess user data
        return
define register with TTP
  S_k = G_sk /* "G_sk" indicates secret key generation process */
      return S_k /* "S_k" denotes the secret key */
define encryption and authentication process
  E_d = E(userdata, S_k) /* "E_d" specifies the encrypted key */
        decrypt and verify
        if(valid)
          authentication success
      else
        rejected
    end if
End
Output: Authentication successful
```

3.2. Mud-Ring Algorithm for Switch Selection

The motivation behind employing the MRA lies in its ability to optimize switch selection within a network, particularly focusing on identifying the most suitable "entry-point switch" for data originating from the data acquisition layer. In network architectures, efficient traffic management is critical to ensure seamless data flow and prevent congestion or overloading of switch flow tables. The MRA is inspired by the foraging behavior of dolphins (intelligent agents), which involves strategically encircling prey with a ring of mud. Similarly, in the network context, the MRA facilitates an intelligent agent, which refers to a virtual or software entity that operates autonomously within the network environment. These agents are programmed with decision-making capabilities and algorithms that enable them to interact with network components, such as switches, in an intelligent and adaptive manner. An intelligent agent interacts with switches (prey), evaluating factors such as proximity and flow table utilization.

The algorithm aims to select switches that can effectively manage incoming data flows by dynamically evaluating these parameters to optimize network performance and resource utilization. This approach not only enhances the efficiency of routing decisions but also mitigates potential issues related to flow table overloading, ensuring that the network remains responsive and capable of meeting operational demands effectively. Thus, the MRA serves as a proactive mechanism to intelligently manage switch selection, aligning with the overarching objective of optimizing data flow management within complex network environments. The detailed procedure of the MRA algorithm and its mathematical computations are explained in the following section.

Step 1: Initially, the software entities (intelligent searching agents) and switches ("prey") are distributed randomly in a ring topology network, with interconnections resembling "mud rings".

Step 2: The software entities explore the switches with random movements throughout the network to select an optimal switch. The new positions of these software entities are adjusted based on switch proximity, utilizing Equation (1).

$$N_P = C_P + RM \times D \tag{1}$$

where 'N_P' denotes the new position of the software entity; 'C_P' indicates the current position of the software entity, which can be expressed as a coordinate or a topology position; 'RM' represents the random movement factor, The 'RM' value represents a random number or vector that introduces uncertainty in software entity movement, preventing bias towards specific network areas and ensuring thorough exploration of all possible switches, with values ranging from 0 to 1; and 'D' represents the distance between the software entity and a switch. The exploration phase continues for a specified number of iterations.

Step 3: The software entities evaluate and update their positions based on their fitness function, which is determined by factors such as switch proximity and flow table usage. These updates are calculated to strategically position the software entities around the chosen switch, using Equation (2).

$$N_P = C_P + f - w_i \times TS_D \tag{2}$$

where 'w_i' denotes the weighted movement of iteration 'i', which adjusts the degree of movement toward or away from the target switch. 'f' represents the fitness function, which evaluates how well a switch meets the criteria based on proximity and other factors. 'TS_D' denotes the distance to the target switch, which is used to guide the entity toward or away from the switch based on the fitness value. The fitness function is mathematically expressed in Equation (3).

$$f = w_1 \times PF + w_2 \times FT \tag{3}$$

where 'w_1, w_2' are the weights that control the relative importance of each factor. 'PF' represents the proximity factor, which evaluates how close the switch is to the software

entity. '*FT*' represents the flow table usage factor, which asseses the switch flow table utilization rate. After evaluating the fitness function, it updates the new position to select an optimal switch; otherwise, it goes to check other switches in the network for '$i = i + 1$' times.
Step 4: Assess the software entities fitness function with respect to proximity to the selected switch and overall network conditions. Select the best switch for routing based on factors such as proximity, flow table usage, and network metrics to maximize flow table efficiency.
Step 5: The procedure concludes upon selecting an optimal switch for routing. Figure 2 shows the flow chart of the Mud-Ring algorithm.

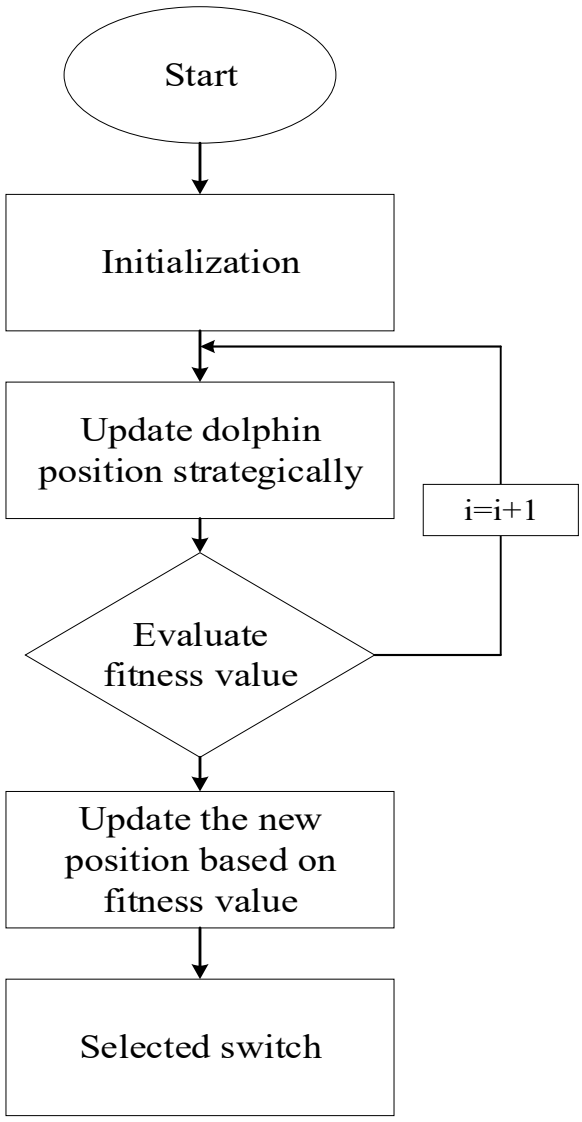

Figure 2. Flow chart of mud-ring algorithm.

The proposed research work uses a blockchain-based approach and searchable encryption algorithms to secure data flow rules and prevent security threats in an SDN environment. Searchable encryption ensures confidentiality and tamper-proof flow rules, allowing secure query processing without decryption, which is explained in Algorithm 2.

Algorithm 2: Searchable encrypted data query algorithm

Input: a protection metric '$\boldsymbol{\theta}$'.
Output: Document identifiers '$\boldsymbol{ID}(\boldsymbol{Doc})$'.

1. while generating random keys $\boldsymbol{M_{key}}$, $\boldsymbol{S_{key}}$, $\left(\boldsymbol{Pub_{key}}, \boldsymbol{Pr_{key}}\right) \leftarrow \{0,1\}^{\boldsymbol{\theta}}$, send $\boldsymbol{Pub_{key}}$ to the membership server do /* Cryptographic keys */
2. if certification is achieved by center verification, then
3. get Signature '($\boldsymbol{Sign}$)' /* "$\boldsymbol{Sign}$" symbol indicates digital signature */
4. else
5. Exit
6. end if
7. for $\mathbf{1} \leq \boldsymbol{g} \leq |\boldsymbol{a}|$, $\mathbf{1} \leq \boldsymbol{v} \leq |\boldsymbol{Doc}|$, $\boldsymbol{b} \leftarrow \boldsymbol{J_{key}}(\boldsymbol{A_g}) \bmod \boldsymbol{Q}$, Compute '$\boldsymbol{Encrp}(\boldsymbol{ID}(\boldsymbol{Doc_n}))$, store it in $\boldsymbol{B}[\mathbf{1}][\boldsymbol{g}]$, get index table '$\boldsymbol{T'}$ and encrypted set of documents do /* '$\boldsymbol{T'}$ **denotes** Index table */
8. if the client's identity is confirmed via $\boldsymbol{Sign}$
9. then the query function is updated with the '$\boldsymbol{T'}$.
10. else if the client proposes a transaction, then
11. start practical byzantine fault tolerance, the most recent transaction is verified and added to the blockchain ledger.
12. else if a trapdoor ($\boldsymbol{D_T}$) transmitted by the client, '$\boldsymbol{p} == \boldsymbol{J_{key}}\left(\boldsymbol{e}.\boldsymbol{b}^{-1}\boldsymbol{mod Q}\right)'$ then
13. $\boldsymbol{L}[\,] \leftarrow \boldsymbol{Encrp_{key}}(\boldsymbol{ID}(\boldsymbol{Doc_i}))$, /* Encrypted document identifiers '$\boldsymbol{Encrp_{key}}(\boldsymbol{ID}(\boldsymbol{Doc_i}))$'. */
14. else
15. get $\boldsymbol{Decrp_{key}}(\boldsymbol{L}(\boldsymbol{o}))$, $\boldsymbol{ID}(\boldsymbol{Doc})$ /* Decrypted document identifiers */
16. end if
17. end for
18. end while

Where 'M_{key}' is master key, 'S_{key}' is session key, 'Pub_{key}, Pr'_{key} are public and private keys respectively. '$|a|$' is total quantity of different keywords recognized, 'J_{key}' hash function 'J' uses the client's master key, 'g' is a threshold set. 'Q' is prime number used in modular arithmetic. 'B' is array storing encrypted document identifiers. Algorithms 2 and 3 show the encryption and decryption steps of the proposed algorithm. 'L' a set of encrypted document IDs, 'e' is encrypted value, 'p' is a decrypted value being checked against a condition. 'O' is an index within the set 'L'.

Algorithm 3: Decryption algorithm

Input: Master key '$\boldsymbol{M'_{key}}$ and a set '$\boldsymbol{L}$' of encrypted document '$\boldsymbol{ID}$'.
Output: Document identifiers '$\boldsymbol{ID}(\boldsymbol{Doc})$'.

1. while you start start decryption, do.
2. for $\mathbf{1} \leq \boldsymbol{o} \leq \boldsymbol{size\ of\ L}$ do /* "$\boldsymbol{L}$" denotes set of encryted IDs*/
3. $\boldsymbol{Decrp_{key}}(\boldsymbol{L}(\boldsymbol{o}))$ /* Decryption key for set of document IDs*/
4. end for
5. end while

Blockchain is a distributed ledger that records flow rule changes and access requests, ensuring transparency and immutability. It is crucial for traffic management and network security. The decentralized nature reduces the risk of a single point of failure. Consensus mechanisms validate flow rules, and blockchain provides a transparent audit trail for detecting unauthorized changes or anomalies. Searchable encryption is a secure method for storing and transmitting sensitive information. The authorized entities generate trapdoors for secure searches on these flow rules, which are stored in the blockchain. These authorized entities retrieve encrypted data from the blockchain for verification or updating a flow rule, which is then decrypted using their decryption key.

Data encryption converts data into an unreadable format using a searchable encryption algorithm before adding it to the blockchain. The encrypted data is then hashed with a cryptographic hash function such as SHA256, creating a unique digital fingerprint recorded on the blockchain. This hashed data, recorded as a transaction, is replicated across all network nodes, ensuring immutability and resistance to tampering. For data retrieval, the search query is encrypted with the same key, allowing searches for matching hashes without revealing content. This integration of searchable encryption with blockchain maintains user privacy and data integrity on the public ledger.

3.3. Binarized Deep Spiking Capsule Fire Hawk Neural Network Based Packet Classification

The binarized deep spiking capsule fire hawk neural network (BSHNN) algorithm combines state-of-the-art neural network architectures, including binarized neural networks, spiking neural networks, and capsule networks, for real-time, resource-efficient, and highly accurate data packet classification in 5G SDN environments. It enhances power efficiency, spatiotemporal sensitivity, and hierarchical data understanding, making it suitable for high-speed, low-latency, and large-scale 5G network demands. Binarized neural networks (BNNs) are deep learning models where the network's weights and activations are constrained to binary values (e.g., -1 and $+1$) instead of floating-point values. This reduces the computational complexity and memory usage, allowing the model to operate efficiently with minimal hardware resources. BNNs are particularly useful for mobile and internet-of-things devices with limited CPUs and large memories. Spiking neural networks (SNNs) are designed to mimic the biological neurons, processing information in discrete time steps, making them well-suited for handling temporal data such as network traffic flows. It allows real-time responses without the need to buffer large amounts of data. This event-driven approach reduces power consumption and enhances processing speed in 5G SDN networking environments. Capsule networks are a type of neural network that improves the accuracy of traditional convolutional neural networks by preserving spatial hierarchies between features. This is crucial for understanding data packet types and malicious patterns.

The data packets undergo a dual verification process for classification, involving header and content inspection in the controller plane with two layers: smart control and domain control. The domain control layer uses two classifiers to analyze header and content attributes, including flow-based characteristics and packet details. The suggested approach uses binary outputs instead of continuous outputs in CNN [32], enabling effective training on spatiotemporal data and simplifying the intricacies of conventional neural networks. The classification process integrates Capsule Network (CapsNet) [33] outputs with optimized parameters from Fire Hawk Optimization (FHO) [34], contributing to decision-making based on learned patterns and hierarchical relationships. The BSHNN is trained using a dataset of network packets labelled as normal or malicious and deployed in the network's controller layer to analyze incoming packets in real-time. BSHNN's robustness against adversarial attacks enhances network security, while its adaptability allows it to evolve with emerging threats, making it ideal for the dynamic 5G environment.

The FHO algorithm is an evolutionary optimization algorithm inspired by the behaviour of firehawks in nature. The detailed description of the FHO algorithm and its mathematical calculations are explained below. In the context of network optimization, it is used to optimize network parameters '(β, δ)' of the proposed neural network. This optimization process involves updating the weights of the neural network during backpropagation through a specified Equation (4).

$$W_i = \hat{W} - \eta \times \nabla loss \tag{4}$$

where 'W_i' represents the weight update of the neural networks utilizing gradient descent, '$\hat{W}$' represents the existing weight during the training of the network, 'η' is the learning rate, which controls the step size of the weight update of the proposed network. '∇' is the gradient

of the loss function with respect to the weights, which indicates the direction and rate of change of the loss function. The fitness function of solutions FHO is evaluated using Equation (5).

$$F = \frac{1}{1+J} \tag{5}$$

where 'F' represents the fitness function of the optimization method and 'J' is the loss function, which measures the error between the predicted output and the actual output.

BSHNN classifies packets considering three crucial factors: packet success rates, packet loss rates, and packet error rates. A packet that achieves a high success rate with minimal packet loss and low error rates is considered a standard packet within the BSHNN model. Notably, BSHNN employs a dynamic threshold that necessitates variation for different packets. This adaptive approach enables the effective identification of malicious packets solely based on their headers. The packets were classified into two distinct classes, normal and suspicious, based on the data packet features shown in Table 1.

The criteria that led to prioritizing BSHNN for packet classification over other models include its dual verification process, effective optimization techniques, adaptability to network conditions, application-specific requirements, and performance metrics such as throughput, packet loss ratio, accuracy, precision, and recall. These factors collectively contribute to the rationale behind choosing BSHNN as the preferred model for packet classification. The workflow of the BSHNN method is shown in Figure 3.

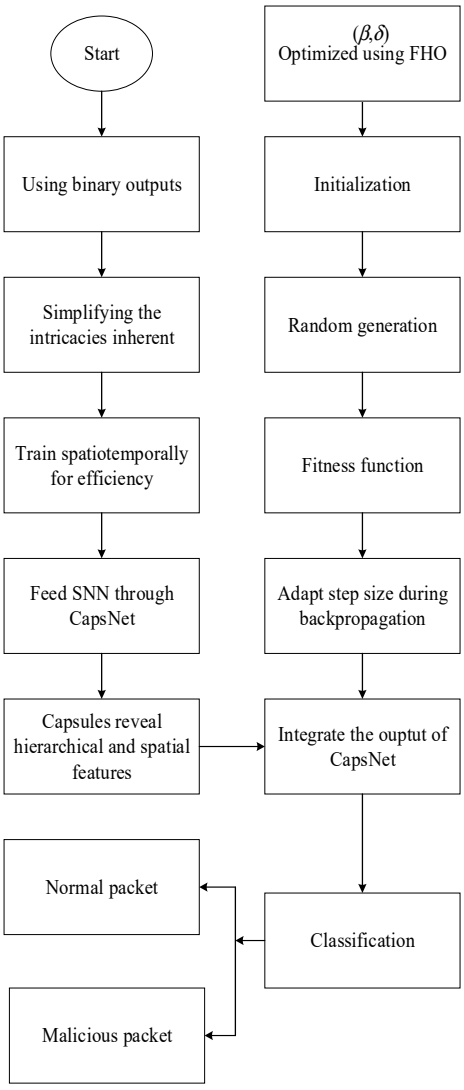

Figure 3. Workflow of the BSHNN approach.

Table 1. Data Packet Attributes.

Packet Features	Description
Service type	Service request to destination, e.g., HTTP
Packet header length	Size of packet header
Flags	Indicates connection status: normal (flag = 0) or error (flag = 1)
Epoch time	Epoch completion time duration
Protocol used	TCP/UDP
RRT	Response reply time
TTL	Time to live
Port Number	Port number of switches
Packet count	Packets per flow amount
Byte count	Bytes per flow amount
Duration	Alive time of switch (nanoseconds)

3.4. Enhanced Adapting Hidden Attribute Weighted Naïve Baye Based Data Packet Transmission

To showcase the effectiveness against flooding attacks of the control plane, various controllers are installed in the domain layer. They are managed by a smart controller (SC) with a hierarchical structure. The smart controller serves as a private key generator, creating secret keys for all switches using a consistent searchable algorithm. Switch authentication considers switch ID, location, and packet history stored in the control plane. Enhanced adapting hidden attribute-weighted naïve bayes integrates attribute-weighted naive bayes, which enhances traditional naive bayes by weighting attributes based on their relevance to identifying suspicious packets. This adaptation allows EAWNB to effectively prioritize and differentiate between normal and suspicious packet behaviors, thereby improving accuracy in threat detection.

The EAWNB approach enhances naive Bayes classification in 5G network security. It integrates attribute weighting, assigning varying importance to attributes such as MAC address, identity, password, physically unclonable function, and location [35]. This adaptability contributes to a more delicate and accurate classification of normal packets. Let 'v_i' represent the weight assigned to the attribute 'i'. The weighted probability of a feature 'Z_i' in the context of naive Bayes is expressed in Equation (6).

$$N(Z_i|D) = \frac{Count(Z_i, D) \times v_i}{Count(D)} \tag{6}$$

where '$N(Z_i|D)$' is the count of occurrences of feature 'Z_i' in class 'D', and '$Count(D)$' is the count of occurrences of class 'D'. EAWNB incorporates an adaptive hidden mechanism, dynamically adjust the weight assigned to each attribute based on the network's evolving conditions. This ensures that the model remains resilient to changes, adapts new patterns effectively, and faces potential security threats. The dynamic adjustment of attribute weights in the adaptive mechanism is formalized in Equation (7).

$$v_i(d+1) = v_i(d) + \eta \cdot \frac{\partial L}{\partial v_i} \tag{7}$$

where '$v_i(d)$' is the weight assigned to attribute 'i' at time 'd'; 'η' denoted as learning rate and 'L' denoted as loss function representing the difference between predicted and actual class labels. Various packet attributes are tracked and updated in the BSHNN model, mitigating security attacks effectively. Computation complexity is crucial for algorithm validation.

3.5. Computational Complexity Analysis

This section assesses the computational complexity of each proposed algorithm. The lightweight encryption algorithm is chosen for user authentication due to its simplicity, speed, and security (with a computational complexity of '$O(n, q)$', where 'q' represents the point (x, y)). The mud ring algorithm is utilized for switch selection and state update, offering an optimal solution with a computational complexity of '$O(n^2)$'. The BSHNN handles packet classification, and the Enhanced EAWNB deals with suspicious packet classification. The derivations of computational complexity are expressed in Equations (8)–(10).

$$T(n) = 2T(n/2) + n^2 \tag{8}$$

$$= n^2 + 2(2T(n/2) + n^2/4) \tag{9}$$

$$= O(n^2) \tag{10}$$

where '$T(n)$' represents the total work. The BSHNN for packet classification requires '$O(n^2)$' computational complexity, yielding a minimum complexity per round of '$O(n^{(log_2 n)})$'. EAWNB requires '$O(S)$', where 'S' is a nearly constant scale variable for all iterations. The aggregate processing duration of suggested BSHNN is significantly reduced, ensuring efficient and accurate results for applications, particularly due to sparse vectors in the total capacity.

4. Results and Discussion

The proposed work has been implemented in the Ns-3 network simulator, which is depicted in Figure 4. It utilizes the novel BSHNN algorithm with searchable encryption and blockchain technology. Initially, create a mobile ad hoc network structure in the NS-3 simulator, randomly assign malicious nodes to send or receive data, and authenticate incoming users using an append-only blockchain. Log the network traffic and other parameters in a CSV file, import it in Python, and classify vulnerabilities using the binarized deep spiking capsule fire hawk neural network. The details of the simulation parameters are presented in Table 2.

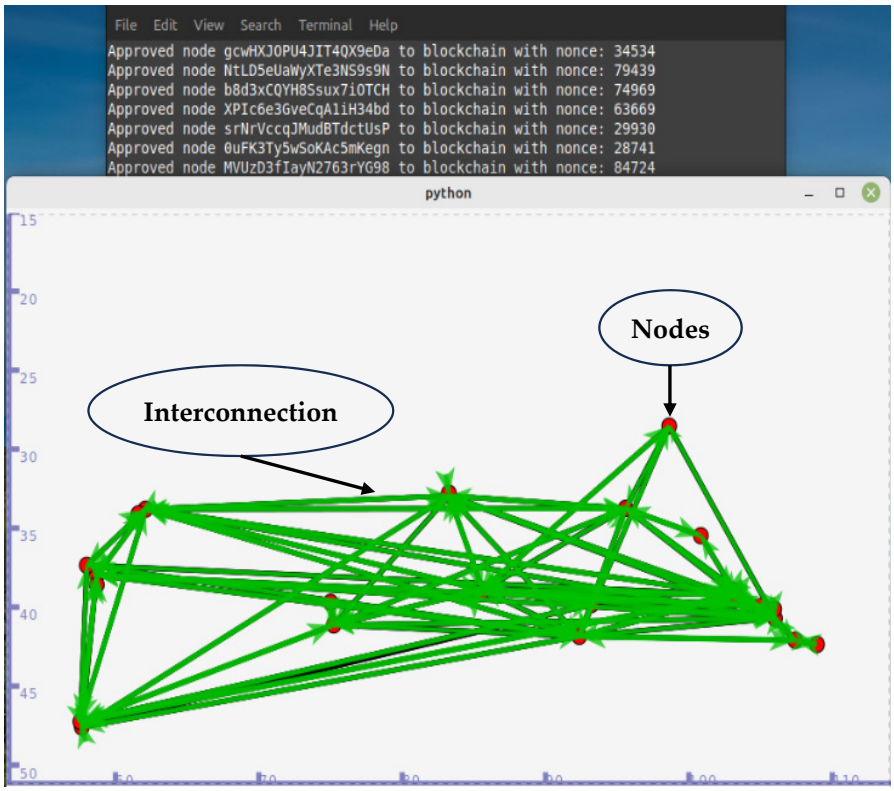

Figure 4. Network simulation environment.

Table 2. Simulation Parameters.

Parameters	Value
Simulation area	500 m × 500 m
Simulation time	45 s
Number of APs	8
Number of MUs	50
Number of controllers	1
Number of switches	7
Attacks	20%
Flow type	MAC-IP flow
SDN controller	OpenFlow
Protocol used	TCP
Packet length	1500
Mobility of MUs	0–2 m/s
Mobility model of MU	RandomWalk2D
Interval time	Random
Block size	100 Transactions
Block header	Previous block hash + timestamp + nonce + difficulty target
Number of transactions	600 in 45 s
Hash generation	SHA256

Figure 4 provides an intuitive view with respect to the approved nodes. A secure hashed number (hexadecimal) is assigned with each approved node as shown in the figure. Every blockchain is also referred to by a random or semi-random number (nonce) that is generated for a specific use, typically in cryptographic communication. The nonce is crucial because it ensures the randomness and uniqueness of each attempt at solving the cryptographic puzzle, making it computationally difficult to manipulate the blockchain. The above parameters comprehensively define the simulation environment by encompassing spatial dimensions, time duration, network components (Mobile User (MU), APs, switches, and controllers), mobility characteristics, communication protocols, and intricate details of the blockchain simulation (such as block size, header structure, proof type, and hash generation). The incorporation of these details in Table 3 serves to improve transparency and reproducibility in the simulation, facilitating a comprehensive grasp of the conditions under which the evaluation of the Ns-3 with the BSHNN algorithm takes place. The network simulation environment of the proposed method is depicted in Figure 4.

Table 3. Performance metrics for the proposed method.

Metrics	Proposed Approach
Detection rate (%)	96.41
Authentication time (s)	16.2
Delay (s)	0.09
Throughput (Mbps)	428
Packet loss ratio (%)	1.5
Accuracy (%)	98.02
Precision (%)	96.40
Recall (%)	97.53

The time required for detection and authenticating secure networks becomes a crucial aspect to determine the consistency of any proposal with blockchain networks. Keeping this in mind, the proposed algorithm demonstrates remarkable accuracy in classifying network events by achieving a detection rate of 96.41%. Particularly, the authentication time is impressively low at 16.2 s, underscoring the algorithm's efficiency in authenticating mobile devices. With a minimal delay of 0.09 s, encompassing propagation, access, switching, queuing, and controller delays, the proposed approach ensures swift and effective packet transmission. Thus, the proposal clearly assures its candidature to be used in a practical scenario.

The proposal accounts for a high throughput (i.e., 428 Mbps) and emphasizes its capability for successful data transmission. Additionally, the low packet loss ratio of 1.5% demonstrates the reliability of the blockchain-based IDS in maintaining data integrity during transmission. The algorithm's outstanding accuracy of 98.02%, along with precision and recall values of 96.40% and 97.53%, respectively, highlights its effectiveness in identifying positive cases. In fact, the results provide further evidence for the reason for the low detection rate and authentication time. In summary, these metrics together provide a comprehensive evaluation of the proposed method, highlighting its proficiency in decision-making, authentication, network efficiency, and maintaining data integrity. Table 3 provides the performance metrics of the proposed methodology.

4.1. Performance Analysis

This section presents a comprehensive analysis of simulation results for the proposed classification algorithm (BSHNN), comparing it with existing state-of-the-art (SOTA) methods.

4.1.1. Detection Rate

The detection rate metric plays an important role in IDS since it directly affects network performance. Figure 5a illustrates the performance of attack detection rate versus attack percentage. The detection rate represents the percent of correct detections in packet sorting. The proposed method obtains a detection rate of 96.41%. Prevailing approaches lack optimal classifiers for packet classification, which involve complex mathematical calculations and time-consuming processes. The analysis concludes that the BSHNN method outperforms the previous ML-IDP method in terms of detection rate.

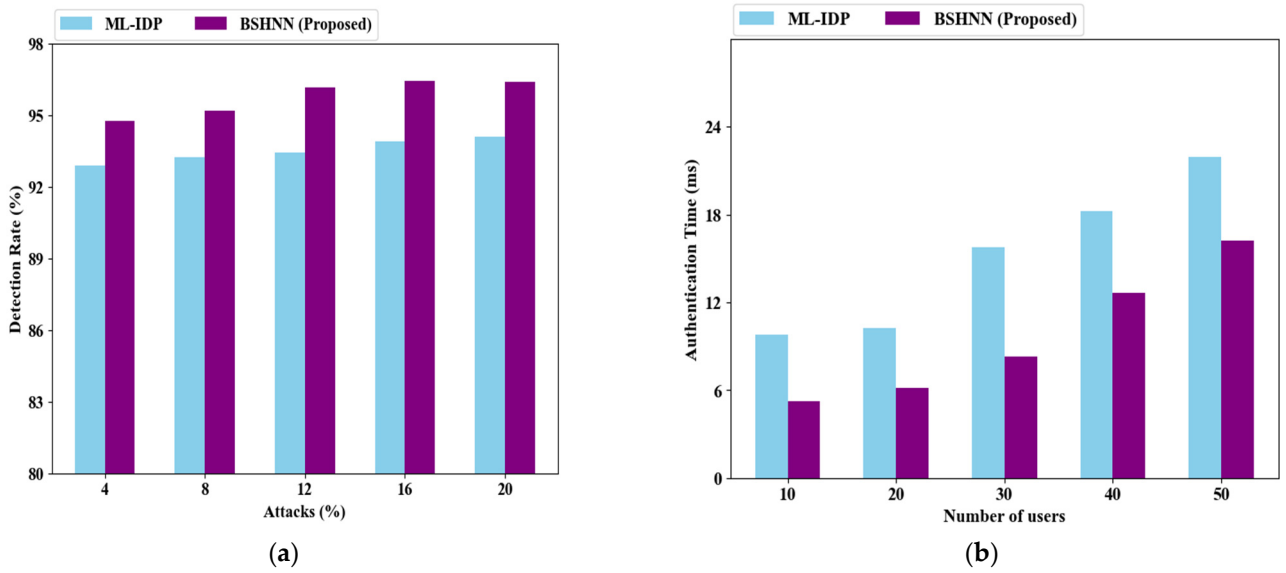

Figure 5. Simulation results for the BSHNN method (**a**) Detection Rate. (**b**) Authentication time.

4.1.2. Authentication Time

The authentication time plays a crucial role in authenticating and verifying mobile devices using a trusted third party to protect against network intrusions. Figure 5b shows

the authentication time performance considering the increasing number of mobile users. When comparing the performance of 50 mobile users, the ML-IDP technique requires 21.9 s, whereas the proposed BSHNN method takes 16.2 s. It is interesting to note that the proposed algorithm provides an average 26% improvement in the rate of authenticating legitimate users. In fact, a sufficient increase in detection rate and a significant dip in authentication time make BSHNN a best candidate under the given network condition.

4.1.3. Delay

In the current IoT context, packets arrive every 0.1 s, resulting in a high network traffic rate. Some packets get dropped at the base station (BS). To compute the delay, use Equation (11).

$$T = BT_j + ET_j + CT_j + KT_j + GT_j \tag{11}$$

where 'BT_j' represents the propagation delay of the packet 'j' and the time needed for the signal to travel from the base station to send the packet. 'ET_j' denoted as access delay, 'CT_j' denoted as switching delay, the time needed to send a packet to SDN. 'KT_j' is the queuing delay, the time spent waiting in the switch to transmit. 'GT_j' is the controller delay, the time for a controller to respond to a request and provide a response. The dynamic adaptability of BSHNN ensures minimal delays in complex network conditions, consistently outperforming ML-IDP as shown in Figure 6a. Initially, both ML-IDP and BSHNN demonstrate minimal delays of 0.001 s, indicating rapid processing with negligible impact on the network. As the number of switches increases to 7, ML-IDP experiences a delay escalation to 0.12 s, whereas BSHNN maintains a lower delay at 0.09 s. This consistent efficiency in processing, especially in scenarios with a higher number of switches, underscores the superior performance of BSHNN.

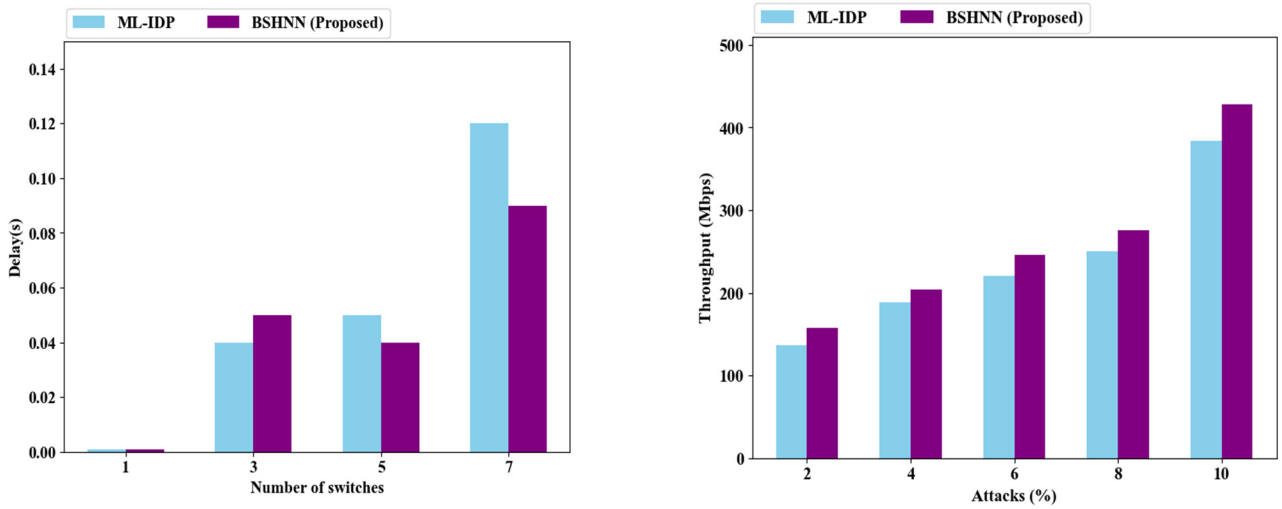

Figure 6. BSHNN method simulation results. (**a**) Delay. (**b**) Throughput.

4.1.4. Throughput

Throughput is the rate at which data is successfully transmitted or processed over a network within a given period. It is a measure of the efficiency and capacity of a system to handle data transfers. A robust attack detection and mitigation model should demonstrate efficient scalability in throughput to effectively handle varying levels of network attacks. Therefore, it is crucial to analyze the model's performance by assessing throughput in relation to the varying percentage of attacks.

Figure 6b shows the simulation results for the throughput. The proposed BSHNN approach significantly improves the throughput rate by 21.6%, even though the percentage

of attacks increases. The BSHNN method showcases its superior efficiency in handling elevated network security challenges compared to the ML-IDP [17] system.

4.1.5. Packet Loss Ratio (PLR)

The packet loss ratio measures the percentage of data packets that fail to reach their destination in a network. Along the same path to establish the success of BSHNN, a thorough look at PLR was executed. The acceptable rate ranges from 0 to 3%. The proposed method uses a single-spiking neural network optimized for packet classification, efficiently monitoring packet arrivals through distributed controllers. This method mitigates the risk of a single point of failure in a network. Figure 7a illustrates the observed packet loss rate across various percentages of attacks. It clearly depicts that even with an increasing number of switches, there is practically a drop in the failure ratio.

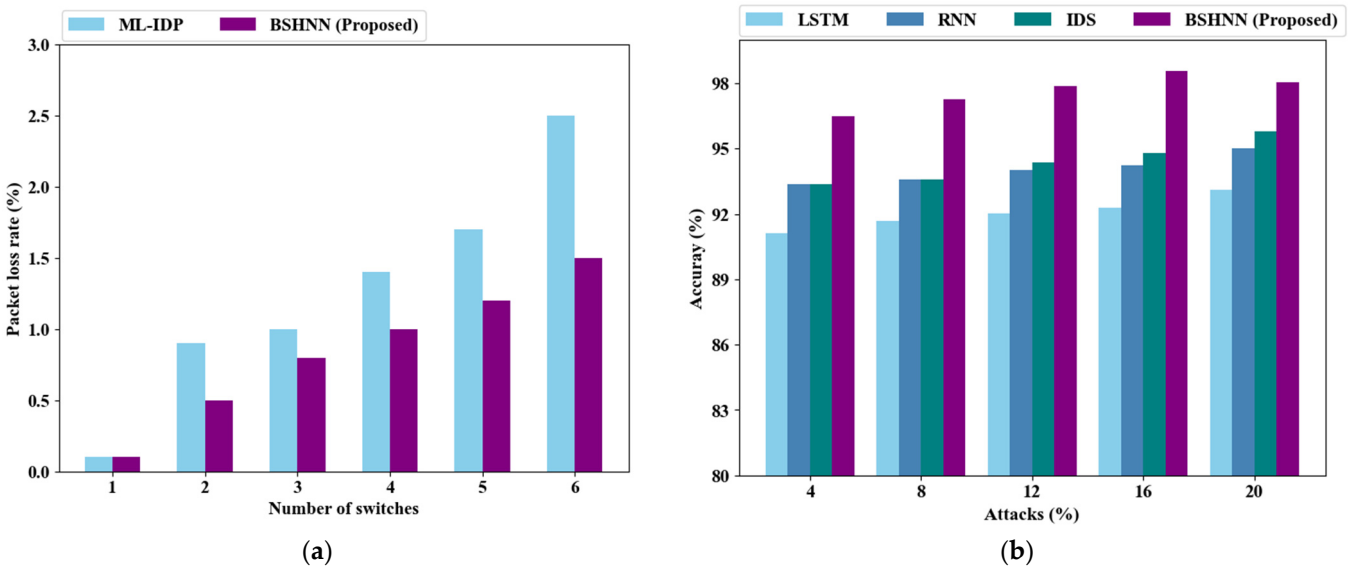

Figure 7. BSHNN approach simulation results. (**a**) Packet loss ratio. (**b**) Accuracy.

The proposed BSHNN method has a lower packet loss rate than the ML-IDP method, especially in scenarios with expanding network infrastructure. The distributed control and classification mechanisms contribute to a more resilient and reliable network, enhancing packet delivery and minimizing losses despite increasing attack percentages and network complexity. This makes the BSHNN approach particularly effective in minimizing packet loss.

4.1.6. Accuracy

The accuracy (*Accu*) rate serves as a primary metric for attack detection, offering insight into the algorithm's performance in identifying various types of attacks. The high accuracy rate indicates the algorithm's effectiveness in promptly detecting attacks. Evaluation metrics, including false positive (F_P), true positive (T_P), false negative (F_N), and true negative (T_N), are employed to compute accuracy. This metric is defined as the percentage of appropriately classified flows relative to the total flows arriving at each controller. The mathematical representation is shown in Equation (12).

$$Accu = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} * 100 \tag{12}$$

Figure 7b illustrates the accuracy results of LSTM, RNN, IDS, and BSHNN. The study demonstrates that the proposed BSHNN outperforms existing methods in terms of accuracy rates, with rates of 25%, 14%, and 10.26% higher. This success is attributed to hashing flow rules before switch deployment and introducing a classifier for packet content classification.

The study also reveals that BSHNN outperforms LSTM, RNN, and IDS subjected to a 20% attack scenario by attaining 98.02%, 93.1%, 95%, and 95.8%.

4.1.7. Recall

Recall, often referred to as the true positive 'T_P' rate, represents the percentage of correctly detected attack flows out of the total number of flows. In essence, it measures the algorithm's ability to capture and identify all instances of actual attacks. The mathematical representation is shown in Equation (13).

$$Recall = \frac{T_P}{T_P + F_N} * 100 \tag{13}$$

Figure 8 shows that the proposed BSHNN outperforms existing methods such as LSTM, RNN, and IDS in network classification. Its higher recall rate indicates its effectiveness in identifying more attack flows, which is shown in Figure 8a. This is due to its innovative approach, which streamlines the training process and reduces training time, making it an efficient choice for packet flow and content verification compared to traditional methods.

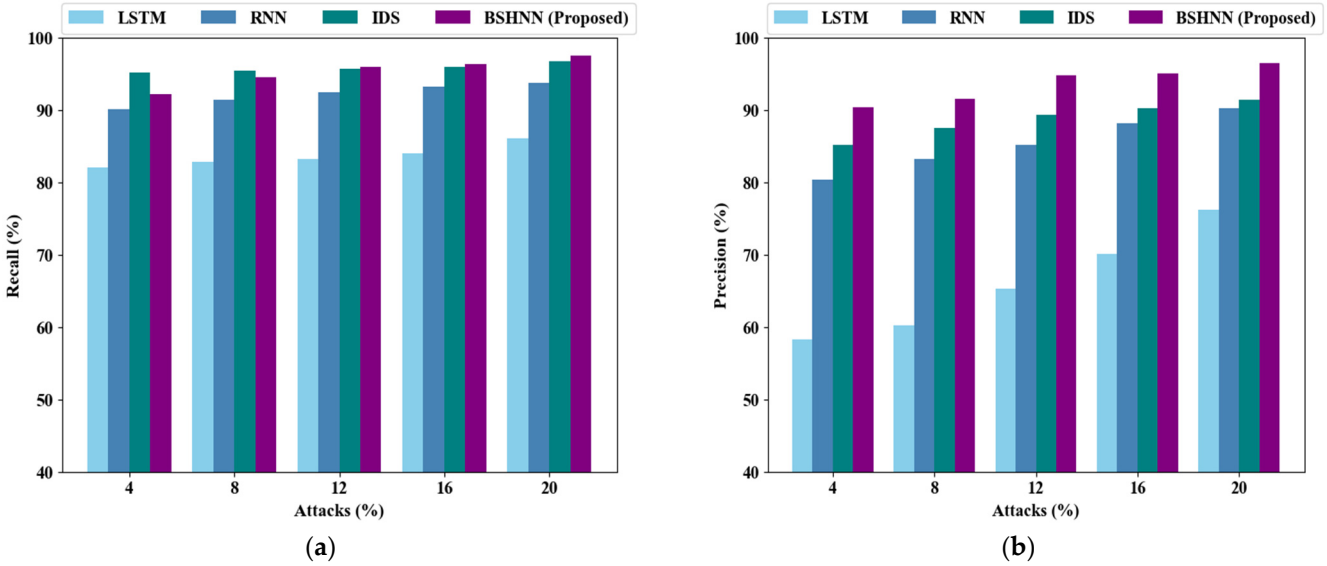

Figure 8. BSHNN method's simulation results. (**a**) Recall. (**b**) Precision.

4.1.8. Performance Metrics Comparisons: (Accuracy, Precision, Recall, and Delay)

Precision is a metric that measures the accuracy of a model's positive predictions, as shown in Figure 8b. It is the ratio of true positive predictions to the total number of positive predictions. The precision rates for LSTM, RNN, IDS, and BSHNN are 76.24%, 90.24%, 91.4%, and 96.40%, respectively, when the arrival packet rate (attacks) is 20%. This signifies that the BSHNN method is better for accurately identifying and permitting legitimate flows, contributing to enhanced precision in intrusion detection and prevention compared to other existing methods.

Table 4 shows the performance metrics of the proposed BSHNN method, which outperforms established methods such as LSTM, RNN, and IDS in accuracy, precision, and recall. The BSHNN efficiently manages feature extraction and optimization tasks, aiming to differentiate between normal and malicious packets using features extracted from packets and leveraging its unique capabilities.

The BSHNN algorithm's exceptional performance in accurate predictions, positive case identification, and capturing a high percentage of relevant events makes it a robust solution for intrusion detection and prevention. The detection rate and authentication time metrics are compared in Table 5.

Table 4. Performance metrics comparisons with respect to neural networks.

Methods	Evaluation Metrics		
	Accuracy (%)	Precision (%)	Recall (%)
LSTM [16]	93.1	76.24	86.13
RNN [18]	95	90.24	93.7
IDS [19]	95.8	91.4	96.7
BSHNN (Proposed)	98.02	96.40	97.53

Table 5. Comparison of detection rate, authentication time, and delay.

Detection Rate			Authentication Time			Delay		
% of Attacks	Methods		No. of Mobile Users	Methods		Number of Switches	Methods	
	ML-IDP (%)	BSHNN (%)		ML-IDP (s)	BSHNN (s)		ML-IDP	BSHNN
4	92.87	94.75	10	9.76	5.22	1	0.001	0.001
8	93.23	95.18	20	10.25	6.14	2	0.04	0.05
12	93.45	96.15	30	15.76	8.27	3	0.05	0.04
16	93.89	96.44	40	18.23	12.64	4	0.09	0.05
20	94.08	96.41	50	21.9	16.2	7	0.12	0.09

BSHNN uses parallel processing to expedite authentication by verifying multiple users simultaneously, resulting in faster results compared to ML-IDP. Its low computational overhead reduces the time required for authentication by 9.76 s for 10 mobile users, 10.25 s for 20 users, and 16.2 s for 50 users compared to ML-IDP's 21.9 s. Even the delay is drastically improved by an average of 20% with respect to the increasing number of switches. This efficiency is evident even with increased user quantities.

Table 6 provides a comprehensive comparison of throughput between ML-IDP and the proposed BSHNN method across different percentages of attacks and switches. Notably, at 2% of attacks, ML-IDP achieves a throughput of 137 Mbps, while BSHNN surpasses it with a higher throughput of 157 Mbps. As the attack percentage increases to 6%, ML-IDP achieves a throughput of 220 Mbps, and BSHNN demonstrates further improvement with a throughput of 246 Mbps. In more challenging scenarios with 10% attacks, ML-IDP attains a throughput of 384 Mbps, yet BSHNN maintains its superior performance with a higher throughput of 428 Mbps. Initially, both ML-IDP and BSHNN demonstrate minimal packet loss ratios of 0.1%, indicating highly efficient data transmission with negligible loss. As the number of switches increases to 6, ML-IDP experiences a notable rise in the packet loss ratio to 2.5%, highlighting potential challenges in data reliability. In contrast, BSHNN showcases increased resilience, maintaining a lower packet loss ratio of 1.5%.

Table 6. Comparison of throughput and packet loss ratio.

Throughput			Packet Loss Ratio		
% of Attacks	Methods		Number of Switches	Methods	
	ML-IDP (Mbps)	BSHNN (Mbps)		ML-IDP (%)	BSHNN (%)
2	137	157	2	0.9	0.5
4	189	204	3	1	0.8
6	220	246	4	1.4	1.0
8	250	276	5	1.7	1.2
10	384	428	6	2.5	1.5

The significant achievement of BSHNN is attributed to the proposal of the mud ring algorithm as an optimized switch selection process. The very selection is far superior to conventional and popular Dijkstra's algorithm. Thus, in Table 7, a comparison with respect to the performance attributes such as bandwidth, jitter, and delay has been presented. As can be readily observed, a sufficient improvement of bandwidth under a reduced jitter is presented by MRA. This is a significant achievement given that the success of BSHNN, as established in previous sections, heavily depends on the effectiveness of the MRA algorithm.

Table 7. Comparison of MRA with Dijkstra's shortest path algorithm.

Parameter	Dijkstra's Algorithm	Mud Ring Algorithm (Proposed)
Bandwidth (Mbps)	52.4679	69.325
Jitter (ms)	0.3501	0.241
Delay (ms)	1	0.09

MRA's dynamic switch selection is crucial for 5G's dynamic data flows, unlike Dijkstra's deterministic approach, which lacks flexibility and is less responsive to real-time network changes. MRA's load balancing prevents flow table overloading, demonstrating its superior performance over Dijkstra's algorithm [36], efficient resource use, and dynamic adaptability.

The BSHNN method outperforms existing methods in all evaluation metrics, achieving higher detection rates, lower authentication times, and increased throughput. Its advanced neural network architecture accurately learns and classifies data patterns, improving authentication and data processing speeds. Its optimized processing and decision-making capabilities enhance network performance and security compared to ML-IDP, CNN, and SEE-BF methods. Table 8 shows the performance comparison of detection rate, authentication time, and throughput with existing methods such as ML-IDP, CNN, and SEE-BF approaches.

Table 8. State-of-the-art comparison.

Methods	Evaluation Metrics		
	Detection Rate (%)	Authentication Time (s)	Throughput (Mbps)
ML-IDP [17]	94.08	21.9	384
CNN [22]	93	20	276
SEE-BF [21]	92.3	19.65	386
BSHNN (Proposed)	96.1	16.2	468

The BSHNN algorithm implemented in the NS-3 network simulator shows significant improvements in intrusion detection for 5G SDN environments. It achieves a high detection rate of 96.41% and reduces authentication time to 16.2 s for 50 mobile users. It also ensures minimal delay and a high throughput of 428 Mbps, surpassing traditional methods of data transmission. BSHNN maintains data integrity and accurately identifies network events, with precision and recall values of 96.40% and 97.53%, respectively. Its integration with blockchain technology enhances decentralized authentication and authorization for secure 5G operations.

4.2. Discussion of the Proposed Work's Scalability, Implications, and Insights

The proposed system is designed to handle large-scale high-speed 5G networks and has been validated through NS-3 simulations, demonstrating high accuracy and decision rates. It has a setup parameter of 50 mobile users and 8 access points spread across 500 m $\times$ 500 m, which can be extrapolated into real-world scenarios. The system integrates blockchain technology for secure, decentralized authentication and authorization and is energy-efficient for sustainable operation. It adapts to evolving threats using AI and

blockchain, ensuring effectiveness against new security challenges. Future steps include pilot testing in real-world 5G environments.

NS-3 is an abstract model of network protocols and components, which may not accurately represent real-world hardware and real-time conditions. It is effective for simulating network behaviour and performance but may oversimplify aspects of network operation and hardware limitations. Hardware-in-the-Loop (HIL) allows for interactive testing with actual hardware, providing a closer approximation of real-time behaviour. NS-3 can simulate large-scale networks with numerous nodes and complex topologies but may be limited by physical hardware and the cost of scaling up to large or complex networks. HIL is more suited for detailed testing of specific components or subsystems due to physical constraints and costs.

This research integrates user authentication, registration, and key generation into the data acquisition layer. The switch layer manages tasks such as switch selection, flow rule security, and hashing. In the domain control layer, activities involve packet analysis and classification. The smart control layer focuses on efficient packet transmission. The proposed system, incorporating blockchain and advanced neural networks, enhances 5G network security by addressing resource constraints and demonstrating superior attack mitigation, contributing to network resilience in a connected and evolving era.

5. Conclusions and Future Scope

The blockchain-based intrusion detection system addresses resource constraints in 5G SDN by merging blockchain security and advanced neural networks. This robust solution fortifies mobile users' information security, heightening the challenge for attackers. The decentralized blockchain ensures resilience by avoiding a single vulnerability point, promoting trust with its transparent, tamper-resistant design. The blockchain-based intelligent intrusion detection and prevention system overcomes the challenges of resource constraints faced by SDN controllers and switches in the 5G software-defined architecture. It enhances information security and broadens threat detection capabilities, shaping the future of 5G network security. The proposed BSHNN algorithm shows significant enhanced performance with respect to parameters such as accuracy, precision, recall, detection rate, and authentication time. The results also reveal that our proposal provides a substantial increase in throughput with a minimum packet loss ratio.

Further research in this field involves exploring potential challenges and limitations in integrating blockchain technology into 5G networks. It also includes investigating ways to minimize the overhead introduced by blockchain while maintaining its security benefits. Additionally, there is a need for ongoing research into more advanced AI techniques to achieve even more accurate packet classification and intrusion detection. These avenues of exploration will contribute to the continued enhancement of network security and the adoption of innovative technologies.

Author Contributions: Conceptualization, N.K.S.N.; Methodology, N.K.S.N.; Software, N.K.S.N.; Formal analysis, B.B.; Writing—Original draft, N.K.S.N.; Writing—Review & Editing, B.B.; Supervision, B.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ferrag, M.A.; Shu, L.; Friha, O.; Yang, X. Cyber security intrusion detection for agriculture 4.0: Machine learning-based solutions, datasets, and future directions. *IEEE/CAA J. Autom. Sin.* **2021**, *9*, 407–436. [CrossRef]
2. Siddamsetti, S.; Srivenkatesh, M. Implementation of Blockchain with Machine Learning Intrusion Detection System for Defending IoT Botnet and Cloud Networks. *Ing. Syst. D'inf.* **2022**, *27*, 1029–1038. [CrossRef]

3. Hernandez-Jaimes, M.L.; Martinez-Cruz, A.; Ramírez-Gutiérrez, K.A.; Feregrino-Uribe, C. Artificial intelligence for IoMT security: A review of intrusion detection systems, attacks, datasets, and Cloud-Fog-Edge architectures. *Internet Things* **2023**, *23*, 100887. [CrossRef]
4. Shrestha, R.; Omidkar, A.; Roudi, S.A.; Abbas, R.; Kim, S. Machine-learning-enabled intrusion detection system for cellular connected UAV networks. *Electronics* **2021**, *10*, 1549. [CrossRef]
5. Krishnan, P.; Jain, K.; Buyya, R.; Vijayakumar, P.; Nayyar, A.; Bilal, M.; Song, H. MUD-based behavioral profiling security framework for software-defined IoT networks. *IEEE Internet Things J.* **2021**, *9*, 6611–6622. [CrossRef]
6. Pandey, B.K.; Saxena, V.; Barve, A.; Bhagat, A.K.; Devi, R.; Gupta, R. Evaluation of soft computing in intrusion detection for secure social Internet of Things based on collaborative edge computing. *Soft Comput.* **2023**, 1–11. [CrossRef]
7. Kaur, J.; Singh, G. A blockchain-based machine learning intrusion detection system for internet of things. In *Principles and Practice of Blockchains*; Springer International Publishing: Cham, Switzerland, 2022; pp. 119–134.
8. De Souza, C.A.; Westphall, C.B.; Machado, R.B.; Loffi, L.; Westphall, C.M.; Geronimo, G.A. Intrusion detection and prevention in fog based IoT environments: A systematic literature review. *Comput. Netw.* **2022**, *214*, 109154. [CrossRef]
9. Sharma, P.; Jain, S.; Gupta, S.; Chamola, V. Role of machine learning and deep learning in securing 5G-driven industrial IoT applications. *Ad Hoc Netw.* **2021**, *123*, 102685. [CrossRef]
10. Malhotra, P.; Singh, Y.; Anand, P.; Bangotra, D.K.; Singh, P.K.; Hong, W.C. Internet of things: Evolution, concerns, and security challenges. *Sensors* **2021**, *21*, 1809. [CrossRef]
11. Arisdakessian, S.; Wahab, O.A.; Mourad, A.; Otrok, H.; Guizani, M. Survey on IoT intrusion detection: Federated learning, game theory, social psychology, and explainable AI as future directions. *IEEE Internet Things J.* **2022**, *10*, 4059–4092. [CrossRef]
12. Zhang, D.; Shi, W.; St-Hilaire, M.; Yang, R. Multiaccess edge integrated networking for Internet of Vehicles: A blockchain-based deep compressed cooperative learning approach. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 21593–21607. [CrossRef]
13. Shams, R.; Suri, D.O.; Hanif, F.; Otero, P. Comparative Analysis of Intrusion Detection Systems in SDN. In Proceedings of the 2023 Global Conference on Wireless and Optical Technologies (GCWOT), Malaga, Spain, 8–10 October 2023; pp. 1–9.
14. Nayak, N.K.S.; Bhattacharyya, B. MAC protocol based IoT network intrusion detection using improved efficient shuffle bidirectional COOT channel attention network. *IEEE Access* **2023**, *11*, 77385–77402. [CrossRef]
15. Singh, S.K.; Sharma, S.K.; Singla, D.; Gill, S.S. Evolving requirements and application of SDN and IoT in the context of industry 4.0, blockchain and artificial intelligence. In *Software Defined Networks: Architecture and Applications*; Scrivener Publishing LLC: Austin, TX, USA, 2022; pp. 427–496.
16. Chaganti, R.; Suliman, W.; Ravi, V.; Dua, A. Deep learning approach for SDN-enabled intrusion detection system in IoT networks. *Information* **2023**, *14*, 41. [CrossRef]
17. Abdulqadder, I.H.; Zhou, S.; Zou, D.; Aziz, I.T.; Akber, S.M.A. Multi-layered intrusion detection and prevention in the SDN/NFV enabled cloud of 5G networks using AI-based defense mechanisms. *Comput. Netw.* **2020**, *179*, 107364. [CrossRef]
18. Polat, H.; Türkoğlu, M.; Polat, O.; Şengür, A. A novel approach for accurate detection of the DDoS attacks in SDN-based SCADA systems based on deep recurrent neural networks. *Expert Syst. Appl.* **2022**, *197*, 116748. [CrossRef]
19. Almazyad, A.; Halman, L.; Alsaeed, A. Probe Attack Detection Using an Improved Intrusion Detection System. *Comput. Mater. Contin.* **2023**, *74*, 4769–4784. [CrossRef]
20. Miranda, C.; Kaddoum, G.; Boukhtouta, A.; Madi, T.; Alameddine, H.A. Intrusion prevention scheme against rank attacks for software-defined low power IoT networks. *IEEE Access* **2022**, *10*, 129970–129984. [CrossRef]
21. Al Ghamdi, M.A. An optimized and secure energy-efficient blockchain-based framework in IoT. *IEEE Access* **2022**, *10*, 133682–133697. [CrossRef]
22. Saba, T.; Rehman, A.; Sadad, T.; Kolivand, H.; Bahaj, S.A. Anomaly-based intrusion detection system for IoT networks through deep learning model. *Comput. Electr. Eng.* **2022**, *99*, 107810. [CrossRef]
23. Yazdinejadna, A.; Parizi, R.M.; Dehghantanha, A.; Khan, M.S. A kangaroo-based intrusion detection system on software-defined networks. *Comput. Netw.* **2021**, *184*, 107688. [CrossRef]
24. Zainudin, A.; Akter, R.; Kim, D.S.; Lee, J.M. Federated learning inspired low-complexity intrusion detection and classification technique for sdn-based industrial cps. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 2442–2459. [CrossRef]
25. Dat-Thinh, N.; Xuan-Ninh, H.; Kim-Hung, L. MidSiot: A multistage intrusion detection system for internet of things. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9173291. [CrossRef]
26. Serrano, W. The blockchain random neural network for cybersecure IoT and 5G infrastructure in smart cities. *J. Netw. Comput. Appl.* **2021**, *175*, 102909. [CrossRef]
27. Asgharzadeh, H.; Ghaffari, A.; Masdari, M.; Gharehchopogh, F.S. Anomaly-based intrusion detection system in the Internet of Things using a convolutional neural network and multi-objective enhanced Capuchin Search Algorithm. *J. Parallel Distrib. Comput.* **2023**, *175*, 1–21. [CrossRef]
28. Abdulqadder, I.H.; Zou, D.; Aziz, I.T. The DAG blockchain: A secure edge assisted honeypot for attack detection and multi-controller-based load balancing in SDN 5G. *Future Gener. Comput. Syst.* **2023**, *141*, 339–354. [CrossRef]
29. Almaraz-Rivera, J.G.; Perez-Diaz, J.A.; Cantoral-Ceballos, J.A. Transport and application layer DDoS attacks detection to IoT devices by using machine learning and deep learning models. *Sensors* **2022**, *22*, 3367. [CrossRef]
30. Logeswari, G.; Bose, S.; Anitha, T. Designing a SDN-Based Intrusion Detection and Mitigation System Using Machine Learning Techniques. In Proceedings of the International Conference on Advanced Communications and Machine Intelligence, Karur, India, 9–11 December 2022; Springer Nature: Singapore, 2022; pp. 303–314.

31. Panahi, P.; Bayılmış, C.; Çavuşoğlu, U.; Kaçar, S. Performance evaluation of lightweight encryption algorithms for IoT-based applications. *Arab. J. Sci. Eng.* **2021**, *46*, 4015–4037. [CrossRef]
32. Kheradpisheh, S.R.; Mirsadeghi, M.; Masquelier, T. BS4NN: Binarized spiking neural networks with temporal coding and learning. *Neural Process. Lett.* **2022**, *54*, 1255–1273. [CrossRef]
33. Vijayakumar, T. Comparative study of capsule neural network in various applications. *J. Artif. Intell.* **2019**, *1*, 19–27.
34. Shishehgarkhaneh, M.B.; Azizi, M.; Basiri, M.; Moehler, R.C. BIM-based resource tradeoff in project scheduling using fire hawk optimizer (FHO). *Buildings* **2022**, *12*, 1472. [CrossRef]
35. Zhou, X.; Wu, D.; You, Z.; Wu, D.; Ye, N.; Zhang, L. Adaptive Two-Index Fusion Attribute-Weighted Naive Bayes. *Electronics* **2022**, *11*, 3126. [CrossRef]
36. Abdelghany, M.; Zaki, W.; Ashour, M. Modified Dijkstra shortest path algorithm for SD networks. *Int. J. Electr. Comput. Eng. Syst.* **2022**, *13*, 203–208. [CrossRef]