electronics

Article

A Multi-Layered Defence Strategy against DDoS Attacks in SDN/NFV-Based 5G Mobile Networks

Morteza Sheibani, Savas Konur *, Irfan Awan and Amna Qureshi

School of Computer Science, Artificial Intelligence and Electronics, Faculty of Engineering and Digital Technologies, University of Bradford, Bradford BD7 1DP, UK; m.sheibani@bradford.ac.uk (M.S.); i.u.awan@bradford.ac.uk (I.A.); a.qureshi19@bradford.ac.uk (A.Q.)
* Correspondence: s.konur@bradford.ac.uk

Abstract: Software-defined networking (SDN) and network functions virtualisation (NFV) are crucial technologies for integration in the fifth generation of cellular networks (5G). However, they also pose new security challenges, and a timely research subject is working on intrusion detection systems (IDSs) for 5G networks. Current IDSs suffer from several limitations, resulting in a waste of resources and some security threats. This work proposes a new three-layered solution that includes forwarding and data transport, management and control, and virtualisation layers, emphasising distributed controllers in the management and control layer. The proposed solution uses entropy detection to classify arriving packets as normal or suspicious and then forwards the suspicious packets to a centralised controller for further processing using a self-organising map (SOM). A dynamic OpenFlow switch relocation method is introduced based on deep reinforcement learning to address the unbalanced burden among controllers and the static allocation of OpenFlow switches. The proposed system is analysed using the Markov decision process, and a Double Deep Q-Network (DDQN) is used to train the system. The experimental results demonstrate the effectiveness of the proposed approach in mitigating DDoS attacks, efficiently balancing controller workloads, and reducing the duration of the balancing process in 5G networks.

Keywords: 5G mobile networks; distributed denial-of-service attacks; SDN; network functions virtualisation; controller burden balancing; deep reinforcement learning

Citation: Sheibani, M.; Konur, S.; Awan, I.; Qureshi, A. A Multi-Layered Defence Strategy against DDoS Attacks in SDN/NFV-Based 5G Mobile Networks. *Electronics* **2024**, *13*, 1515. https://doi.org/10.3390/electronics13081515

Academic Editor: Martin Reisslein

Received: 21 February 2024
Revised: 3 April 2024
Accepted: 12 April 2024
Published: 16 April 2024

1. Introduction

As a promising network paradigm, software-defined networking/network functions virtualisation (SDN/NFV) is intended to provide security, efficiency, flexibility, dynamism, and cost-effectiveness to emerging networks like 5G networks. SDN networks comprise two primary devices: controller and forwarding devices or switches. Controllers are located in the control plane and are responsible for controlling and managing network devices and forwarding control commands to the switches. Switches, however, are responsible for forwarding packets to their destination in the data plane. NFV is proposed to virtualise network functions and provision services for users. Notably, as a result of NFV and virtual network functions (VNFs), network management is encouraged. The combination of SDN and NFV has the benefit of two distinct worlds and is more beneficial for complicated networks like the fifth generation of mobile networks.

Though the combination of SDN and NFV offers several advantages, such as increased flexibility, scalability, and efficiency, it also introduces new security challenges. One of the main security challenges with SDN/NFV-based 5G networks is that they are more complex and distributed than traditional networks. SDN/NFV-based 5G networks also rely on several new technologies, such as cloud computing and virtualisation. These technologies can introduce new security vulnerabilities. Another security challenge with SDN/NFV-based 5G networks is that attackers are increasingly targeting them and are vulnerable to various attacks, including distributed denial-of-service (DDoS) attacks, man-in-the-middle

attacks, and malware attacks [1]. These attacks can disrupt or disable SDN/NFV-based 5G networks and compromise mobile users' data [2].

Effective intrusion detection and mitigation systems are crucial for safeguarding SDN/NFV-based 5G networks from malicious attacks [3,4]. Traditional intrusion detection and mitigation systems are ineffective at monitoring and protecting SDN/NFV-based 5G networks because they are typically designed for centralised networks. Detecting and preventing intrusions in SDN/NFV-based networks is a motivating issue that has attracted significant research attention. Researchers are actively developing new intrusion detection and mitigation systems [2–4] designed explicitly for SDN/NFV-based 5G networks, and these systems are expected to play a vital role in securing the next generation of cellular networks.

NFV provides key capabilities like the chaining of service functions and personalisation of services, among others. The centralised controller in SDN brings many benefits to the control and management of the network, but in the meantime, it poses severe security threats. In the control plane, DDoS attackers will threaten the network, produce a considerable amount of traffic in a short duration, send an enormous number of unmatched traffic flows to the controller, and make the legitimate flow impossible.

SDN networks can use a central controller to manage and control them. However, this approach is only effective for small networks. For large-scale networks, a single controller can become a bottleneck [5], causing the whole network to fail. Therefore, using multiple controllers for large-scale networks is recommended to avoid the system becoming overloaded and non-functional [6].

The centralised SDN/NFV architectures in 5G networks face challenges due to controllers' inherent scalability limitations. In other words, as 5G networks expand to provide coverage for users and the ubiquity of services, the centralised (single) controller may be overburdened due to controlling too many switches. Furthermore, controller failure may cause the whole network to shut down because of a single point of failure problem. Distributed control architectures offer a critical and efficient solution for building scalable SDN networks.

To address the challenges of single controller deployment, we propose a new method leveraging multi-controllers in SDN networks. This involves using distributed controllers instead of a single controller to manage the network. It not only enhances the reliability and scalability of the network but also allows for better coordination among the controllers [7,8]. However, this approach presents a new challenge: relying on the static mapping of switches to distributed controllers may hinder the controller's ability to adapt to changes in traffic. Real networks are enormously variable in various dimensions, and as a result, the unequal traffic of forwarding devices in the distributed control network is likely to occur. In the case of the static allocation of switches to controllers, and when the traffic varies abruptly, some distributed controllers may be overburdened, while others might have unused resources. In this work, to solve the issue of the unbalanced burden of controllers in SDN/NFV-based 5G networks, a switch relocation approach based on deep reinforcement learning is presented, where a deep reinforcement learning framework is proposed to work with the whole network and train how to relocate switches for achieving the maximum amount for reward.

Contributions and plan of the paper: Our paper introduces a three-layered framework that builds on our previous work [8] on detecting DDoS attacks in SDN-based 5G networks. In this new framework, we use entropy detection to categorise incoming packets as normal or suspicious. Suspicious packets are then sent to a centralised controller for further processing using a self-organising map (SOM). Although entropy is a valuable solution, its effectiveness can be affected under bursty traffic conditions which may raise the false positive rate. To mitigate some limitations and improve entropy's effectiveness, we used traffic filtering to reduce the false positive rate. We also introduce a dynamic OpenFlow switch relocation method that uses deep reinforcement learning to balance the load among controllers and allocate OpenFlow switches dynamically. We tested the system using the

Markov decision process and trained it using a Double Deep Q-Network (DDQN). More specifically, the main contributions of this paper are as follows:

- A three-layered framework is proposed to address two distinct and crucial problems in 5G networks with multiple controllers based on SDN/NFV. The issues include balancing the burden of switches in a multi-controller scenario and presenting an intrusion detection system. Generally, a three-layer architecture is considered an effective perimeter security measure for all communication networks because it provides a balanced approach to addressing security concerns at different network architecture levels. Through the employment of security at the application layer, control layer, and infrastructure layer, various types of security threats can be mitigated. They can effectively ensure confidentiality, integrity, and availability in SDN-enabled networks.
- The proposed framework comprises three layers: forwarding and data transport, management and control, and virtualisation. The management and control layer is divided into main (centralised) and distributed controllers. In the distributed controllers' sub-layer, entropy detection is employed to classify incoming packets as normal or suspicious. The main controller then forwards the suspicious packets to the virtualisation layer for further processing using an SOM. This process results in the detection and mitigation of DDoS attacks.
- The static allocation of OpenFlow switches (OF-switches) and distributed controllers in SDN/NFV-based 5G networks featuring multiple controllers may overload some controllers and underuse others. This paper proposes an OF-switch relocation approach based on deep reinforcement learning to address this challenge.

The rest of the paper is organised as follows. Section 2 summarises related work. Section 3 presents the main building blocks of the framework. Section 4 details the proposed method for mitigating DDoS attacks in multi-controller SDN/NFV architectures. The results of the experiments and an evaluation of the proposed algorithm are presented in Section 5. Section 6 presents concluding remarks for the paper.

2. Related Work

This section reviews existing research on distributed control approaches for SDN networks. We categorise multi-controller SDN networks into two groups: physically distributed with logically centralised control and physically and logically distributed architectures [7,8].

In SDN networks with logical centralisation, multiple controllers are physically deployed, regardless of whether they are functionally or logically centralised. For example, HyperFlow [9] supports SDN networks that are logically centralised but physically distributed. While there are other examples of logically centralised and physically distributed SDN networks, some limitations of this approach have led to the presentation of logically distributed structures. This is particularly relevant for 5G networks, where mobility and scalability are crucial factors.

Multi-controller SDN networks can be further categorised based on the distribution of control logic: flat and vertically distributed. In flat architectures, each controller manages a specific network segment and cooperates using pre-defined mechanisms for overall network control. There is no single central controller dictating commands. References [9,10] showcase examples of flat-distributed control. In [9], a network is divided into two subnetworks, each with its controller and IP address. In [10], a switch transfer protocol helps distribute the processing load across multiple controllers.

Unlike flat-distributed control, hierarchical SDN networks employ multiple controller levels. The authors in [11] proposed a multi-service algorithm based on OpenFlow controllers. This algorithm leverages the FlowVisor application to manage distributed servers and dynamically switches between different load balancing methods to optimise processing across controllers. Similarly, [12] presents a HybridFlow-based SDN algorithm utilising distributed controllers in a hierarchical structure. Several studies have explored security challenges in multi-controller SDN networks. The authors in [13] investigated various

distributed denial-of-service (DoS) attacks and proposed machine learning-based detection and mitigation algorithms for flooding attacks. Focusing on performance optimisation, Wang et al. [14] proposed an efficient SDN latency monitoring solution to ensure accurate delay measurement and maximise network performance.

Numerous studies have explored the integration of SDN into wireless networks. For example, [15] introduced OpenFlow and FlowVisor to enhance control plane operations in wireless networks. NEC's approach in [16] focuses on virtualising base stations, enabling the dynamic allocation of radio resources at the Medium Access Control (MAC) layer. Ericsson's Cloud EPC adapted the Long-Term Evolution (LTE) control plane to manage OpenFlow switches. Alcatel-Lucent's SoftRAN architecture, detailed in [17], explores a logically centralised control plane with a distributed data plane for the scalable enforcement of Quality of Service (QoS) and firewall policies.

The literature review shows a need for research on detecting and mitigating DDoS attacks in multi-controller SDN-based 5G networks. The existing literature has not yet addressed this issue.

3. Materials and Methods

3.1. Markov Decision Process

The state of a typical SDN system at the current moment relates only to the state at the previous moment and is not associated with the state before the previous moment. Regarding the Markov characteristic of the system, we utilise the Markov decision process to analyse the system. The model is demonstrated as follows: $M = \{S_t, A, P, R_t, \alpha\}$, where S_t denotes the state of the system state at time t, A is the OpenFlow (OF) switch relocation action set, P is the probability of state transitions, R_t is the system reward (describing the feedback of the environment), and α represents the attenuation coefficient. Furthermore, the relocation approach $\delta(a, S_t)$ denotes the probability of action a in S_t. To analyse the system state that utilises the Markov decision process, the required terminologies and descriptions are presented in Section 3.1.1.

3.1.1. Terminologies and Descriptions

This section presents the terminologies and descriptions used in the model.

State of the System

The network is described as a graph with no direction, $\boldsymbol{G} = (\boldsymbol{O}, \boldsymbol{E})$, where $\boldsymbol{O} = \{\boldsymbol{o_1}, \cdots, \boldsymbol{o_n}\}$ is the set of OF-switches, and n is the number of switches. $\boldsymbol{C} = \{\boldsymbol{c_1}, \cdots, \boldsymbol{c_m}\}$ denotes the set of controllers, and m presents the number of controllers. The controller burden capacity set is denoted by $\boldsymbol{B} = \{\boldsymbol{b_1}, \cdots, \boldsymbol{b_m}\}$, and $\boldsymbol{b_i}$ denotes the maximum burden of the i-th controller. The request rate of the packet-in message sent by the j-th switch, $\boldsymbol{o_j}$, to the i-th controller, $\boldsymbol{c_i}$, at time t is stated as $\boldsymbol{q_{ij}(t)}$. The association between the i-th controller and the j-th OF-switch is expressed as the following (1):

$$\boldsymbol{u_{ij}(t)} = \begin{cases} \boldsymbol{1} & \boldsymbol{o_i} \text{ controlled by } \boldsymbol{c_i} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{1}$$

When the request rate of the packet-in message produced by the j-th OF-switch is found out, the state of the system state is expressed according to the following (2):

$$\boldsymbol{S_t} = \begin{bmatrix} \boldsymbol{q_{11}(t)u_{11}(t)} & \boldsymbol{q_{12}(t)u_{12}(t)} & \cdots & \boldsymbol{q_{1n}(t)u_{1n}(t)} \\ \boldsymbol{q_{21}(t)u_{21}(t)} & \boldsymbol{q_{22}(t)u_{22}(t)} & \cdots & \boldsymbol{q_{2n}(t)u_{2n}(t)} \\ \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{q_{m1}(t)u_{m1}(t)} & \boldsymbol{q_{m2}(t)u_{m2}(t)} & \cdots & \boldsymbol{q_{mn}(t)u_{mn}(t)} \end{bmatrix} \tag{2}$$

Action Set

In an SDN architecture for a 5G network with m controllers and n OF-switches, there are $m \times n$ kinds of OF-switch relocation actions, in which n kinds are "non-relocation" actions. Relocation actions are enumerated from 1 to $m \times n$.

Controller Burden

There exists a variety of sources of burden for controllers, among which are receiving messages of packet-in, transmitting flow entries, and getting in touch with other controllers. The primary reason for the burden of SDN controllers can be attributed to the process of messages of packet-in [18] and as a result, the request rates of packet-in messages received by the i-th controller as the controller burden (3):

$$\boldsymbol{b_i(t)} = \sum_{j=1}^{n} \boldsymbol{q_{ij}(t) u_{ij}(t)} \tag{3}$$

Controller Burden Ratio

Due to the difference in the burden capacities of controllers, it is imprecise to use the dispersal of the controller burden to reveal the effect of burden balancing. The burden ratio of the i-th controller (denoted by $\beta_i(t)$) is defined in a way to reveal the consumption ratio of the resources of the controller according to the following (4):

$$\boldsymbol{\beta_i(t)} = \frac{\boldsymbol{b_i(t)}}{\boldsymbol{d_i}} \tag{4}$$

where d_i shows the burden capacity of the i-th controller. The average burden ratio of controllers can be defined according to the following (5):

$$\overline{\boldsymbol{\beta}}\boldsymbol{(t)} = \frac{\sum_{i=1}^{m} \boldsymbol{\beta_i(t)}}{\boldsymbol{m}} \tag{5}$$

Rate of Balancing Burden of Controller

The rate of balancing the burden of the controller (denoted by $D(t)$) is used to illustrate the scattering of the burden ratio of the controller and the average of the burden ratio of controllers and is expressed according to the following (6):

$$\boldsymbol{D(t)} = \frac{\left(\sum_{i=1}^{m} \frac{\left(\beta_i(t) - \overline{\beta}(t)\right)^2}{m} \right)^{\frac{1}{2}}}{\overline{\boldsymbol{\beta}}\boldsymbol{(t)}} \tag{6}$$

It should be noted that $D(t)$ measures the level of balancing the burden of the controller. Therefore, $D(t)$ signifies the performance of the strategy recruited for balancing the burden among the controllers.

Switch Relocation Cost

The cost of relocation in OF-switches is affected by two parameters: (1) the controller sends the flow-mod messages to the OF-switches that require to be relocated; (2) the OF-switch sends the relocation requests to the target controller. The OF-switch relocation cost of the j-th OF-switch (denoted by $\boldsymbol{E_j(t)}$) is defined as follows (7):

$$\boldsymbol{E_j(t)} = \sum_{i=1}^{m} \left(\gamma_{ij}(t) u_{ij}(t) e_{ij} \right) + \sum_{k=1}^{m} \lambda_{kj} u_{kj}(t+1) e_{kj}, \tag{7}$$

where $\gamma_{ij}(t)$, e_{ij}, λ_{kj}, $u_{kj}(t+1)$, and e_{kj} denote the number of flow-mod requests sent by the overburdened controller i to OF-switch j, the number of hops from the j-th OF-switch to the i-th controller, the number of request messages sent by the j-th OF-switch to the k-th

controller, the relation between the *k*-th controller and the *j*-th switch after relocation, and the number of hops from the *j*-th OF-switch to the *k*-th controller, respectively.

The Reward of the System

After the relocation of switches, the quality of the relocation is influenced by whether the state of the system improves or deteriorates. Therefore, the reward of the system must be identified. In order to prevent numerous OF-switch relocation occurrences, only considering the enhancement of the controller burden after relocation and to avoid relocating the OF-switch to a distant controller, the OF-switch relocation cost must be taken into account. Hence, the reward of the system can be expressed according to the following Equation (8):

$$\boldsymbol{R_t} = \begin{cases} \frac{\boldsymbol{D(t)-D(t+1)}}{\sum_{j=1}^{n} \boldsymbol{E_j(t)}} & \sum_{j=1}^{n} \boldsymbol{E_j(t)} \neq \boldsymbol{0} \\ \boldsymbol{0} & \sum_{j=1}^{n} \boldsymbol{E_j(t)} = \boldsymbol{0} \end{cases} \tag{8}$$

where $D(t)$ and $D(t+1)$ represent the rate of balancing the burden of the controller before and after the relocation of the OF-switch, and $\sum_{j=1}^{n} \boldsymbol{E_j(t)}$ denotes the switch relocation cost during the relocation process. If the relocation does not happen, then $\sum_{j=1}^{n} \boldsymbol{E_j(t)} = \boldsymbol{0}$, and as a result, the reward of the system will be zero.

Best Approach

The best approach $\widehat{\delta}(\boldsymbol{a}|\boldsymbol{S_t})$ is identified in a way to maximise the reward of the system according to the below (9):

$$\widehat{\delta}(\boldsymbol{a}|\boldsymbol{S_t}) = \underset{\delta}{\boldsymbol{argmax}} \boldsymbol{E}\left[\sum_{t} \boldsymbol{\alpha}^{t} \boldsymbol{R_t}\right] \tag{9}$$

where $E[.]$ denotes the expected value. When $\boldsymbol{\alpha}$ (attenuation coefficient) is close to zero, the instantaneous reward of the system will be more crucial. If it becomes close to 1, the average reward of the system becomes more important.

3.1.2. Deep Q-Network Model for SDN/NFV-Based 5G Networks

In SDN/NFV-based 5G networks with multiple controllers, the state of the system changes dynamically because of the time-varying characteristic of the flow, and it is not possible to employ conventional reinforcement learning methods like Q-learning directly. We propose using a Deep Q-Network (DQN) [19], a deep neural network mixed with a Q-learning method. In a DQN, the following applies to update the Q-value (10):

$$\boldsymbol{Q}(\boldsymbol{S_t}, \boldsymbol{a}, \boldsymbol{\phi_t}) = \boldsymbol{Q}(\boldsymbol{S_t}, \boldsymbol{a}, \boldsymbol{\phi_t}) + \boldsymbol{\sigma}\left(\boldsymbol{R_t} + \boldsymbol{\alpha} \underset{\sigma'}{\boldsymbol{max}} \boldsymbol{Q}(\boldsymbol{S_{t+1}}, \boldsymbol{a'}, \boldsymbol{\phi_{t+1}}) - \boldsymbol{Q}(\boldsymbol{S_t}, \boldsymbol{a}, \boldsymbol{\phi_t})\right) \tag{10}$$

where the function $\boldsymbol{Q}(\boldsymbol{S_t}, \boldsymbol{a}, \boldsymbol{\phi_t})$ is utilised to denote the output achieved when the state of the system is its input, $\emptyset_t$ is the DQN parameter, and $\boldsymbol{\sigma}$ is the learning rate. After the convergence of Equation (10), the optimal value of the Q-function is obtained, and the target value of the Q-function ($\boldsymbol{z_t}$) is described as follows (11):

$$z_t = R_t + \alpha \max_{a'} Q\left(S_{t+1}, a', \phi_{t+1}\right) \tag{11}$$

The state in SDN/NFV-based 5G networks is used to determine the Q-value of relocation actions. Moreover, using the back-propagation mechanism, the loss function can be defined for optimising the parameters of the DQN according to the following Equation (12):

$$\boldsymbol{Z}(\boldsymbol{\phi_t}) = (\boldsymbol{z_t} - \boldsymbol{Q}(\boldsymbol{S_t}, \boldsymbol{a}, \boldsymbol{\phi_t})) \tag{12}$$

3.1.3. DDQN-Based Switch Relocation Approach

The proposed DQN-based relocation approach of OF-switches for 5G networks provides the best Q-value of the Q-function using the training of the neural network. Nevertheless, exploiting only one Q-function to choose the relocation action and computing the target value of the Q-function may result in big or small $\boldsymbol{Q}(\boldsymbol{S_t}, \boldsymbol{a}, \boldsymbol{\phi_t})$ and $\boldsymbol{Q}(\boldsymbol{S_{t+1}}, \boldsymbol{a'}, \boldsymbol{\phi_{t+1}})$ concurrently and result in oscillation in the model. Simultaneously utilising Equation (11) to compute the target value of the Q-function may cause overestimation. Hence, two equal Q-networks are recruited: the target Q-network and the learning Q-network.

- The learning Q-network chooses relocation actions and renews the parameters of the model.
- The target Q-network computes the target value of the Q-function. Afterwards, the target Q-network is renewed occasionally with the parameters of the learning Q-network to boost the training process.

Dissimilar Q-networks are used for the computation of the target value of the Q-function and the selection of actions to prevent overestimation. Additionally, experience repeat is utilised to put the samples achieved by interaction with the environment in the memory unit due to the requirement of the independent distribution of samples by the deep learning method. Throughout the training stage, a portion of the samples are chosen randomly to optimise the parameters of the learning Q-network. The target value of the Q-function of the DDQN is stated as follows (13):

$$\boldsymbol{z_t} = \boldsymbol{R_t} + \boldsymbol{\alpha Q}\left(\boldsymbol{S_{t+1}}, \boldsymbol{\underset{a'}{argmax} Q}\left(\boldsymbol{S_{t+1}}, \boldsymbol{a'}, \boldsymbol{\phi_{t+1}}, \boldsymbol{\phi'_{t+1}}\right)\right) \tag{13}$$

where $\boldsymbol{\emptyset_{t+1}}$ and $\boldsymbol{\emptyset\prime_{t+1}}$ denote the parameters of the learning and target Q-networks, respectively.

As stated earlier, a main controller is used to manage the distributed controllers in our proposed architecture for multi-controller SDN/NFV-based 5G networks. The main controller gathers the state messages and cooperates with its surroundings in a periodic manner. In the training stage, the system's state is shaped into a matrix (Equation (2)) and supplied to the learning Q-network. Then, the learning Q-network utilises neural networks to obtain the value of the Q-function for relocation occurrences. The action is chosen based on the greedy method to gain the reward of the system and the next state. After that, the system's state, action, reward, and next state are accumulated in the memory to provide random samples for training the learning Q-network and the target Q-network. In the online stage, the parameters of the trained Q-network are uploaded. As soon as the main controller realises that the distributed controller is overburdened, the state of SDN is shaped as a matrix (a two-dimensional one) and supplied into the system to obtain the Q-value. The OF-switch relocation action matching the maximum Q-value is then the outcome. The proposed OF-switch relocation approach based on a DDQN for multi-controller SDN/NFV-based 5G networks is provided in Algorithm 1.

Algorithm 1: The DDQN-based OF-switch relocation approach with multiple controllers.

Initialisations and General Info

Inputs:

The system graph $G = (O, E)$, set of controllers (C), no. of iterations (T), set of actions (A), examination rate (μ) for greedy algorithm, learning rate (σ), and attenuation coefficient (α)

Output:

OF-switch relocation action (a)

Initialisations:

Initialising the memory, parameters of the target, and learning Q-networks

Training stage:

Obtain the state of the network and produce the state in a shape of a state matrix S_t periodically

Main Algorithm

for $t = 1$ to T, do the following:

(1) For relocation actions, the learning Q-network makes use of the convolutional version of neural networks to achieve the value of the Q-function.

(2) The action is chosen based on the greedy algorithm, in this manner as follows:
i-th probability $1 - \mu$ choose an arbitrary action a
otherwise choose $a = \underset{a}{argmax} Q(S_t, a, \phi_t)$

(3) To obtain the state at time $t + 1$ and reward (R_t), the selected action in the previous step runs the simulator.

(4) Store the state at time $t + 1$ and t, the reward (R_t), and action (a).

(5) Choose arbitrary sets of data from memory for training the learning Q-network.

(6) Set $z_t = \begin{cases} R_t + \alpha Q\left(S_{t+1}, \underset{a'}{argmax} Q\left(S_{t+1}, a', \phi_{t+1}, \phi'_{t+1}\right)\right) & S_{t+1} \text{ is not a final state} \\ R_t & S_{t+1} \text{ is a final state} \end{cases}$

(7) Based on the loss function, $(z_t - Q(S_t, a, \phi_t))^2$, and by back-propagating, the neural network parameters of the network (ϕ_t) are updated.

(8) ***If*** $t \,\%\, z == 0$, ***then***
Update the parameters of the target Q-network by $\phi_t = \phi'_t$. ***End if***

(9) Update the state of the network by $S_t = S_{t+1}$.

End for

Online Phase:

(1) Use the current state of the system as an input to the learning Q-network.

(2) Obtain the OF-switch relocation action (a) matching to the maximum value of the Q-function.

3.2. DDoS Attack Detection and Mitigation

This section presents the details of the proposed two-phased framework for detecting and mitigating DDoS attacks. An initial anomaly identification phase is performed by every distributed controller for the prompt detection of any anomalies in traffic flows in an early manner. As soon as any controller detects a suspicious traffic flow, it is forwarded to a main (centralised) controller for further process (identifying the DDoS attack and taking proper measures). The main components of this architecture are shown in Figure 1. A detailed description of the individual components illustrated in Figure 1 is provided in the subsequent sections.

The proposed scalable framework for an SDN-based 5G network with multiple controllers in Figure 1 includes three layers: the forwarding and data transport layer (including OpenFlow Switches, 5G Base Stations, UEs, PDN-GWs, etc.), management and control layer (including distributed and main SDN controllers), and intelligence layer (including machine learning techniques, Traffic Classification, Analysis and Decision Making, Big data centre, etc.).

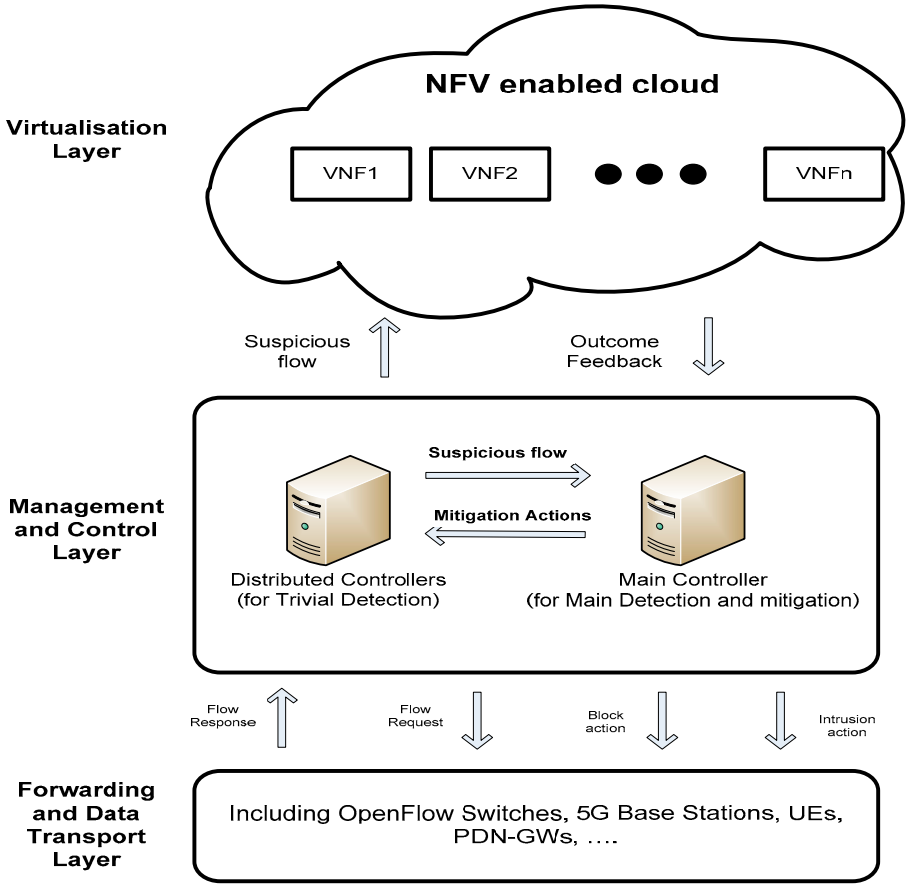

Figure 1. The main components of the proposed architecture.

The OpenFlow switches in the forwarding and data transport layer perform traffic monitoring, collecting, and uploading traffic flows to the upper layer (management and control layer) and blocking malicious traffic based on the instructions of the main controller. Identifying suspicious traffic and detecting DDoS attacks preliminarily in the uploaded traffic from the lower layer, generating mitigation strategies relying on decisions made by the upper layer (intelligence layer), and instructing the lower layer are the main objectives of the management and control layer. Performing further analyses through machine learning techniques, etc., is the main responsibility of the intelligence layer.

Figure 2 presents the proposed detection strategy, which will be explained in subsequent sections.

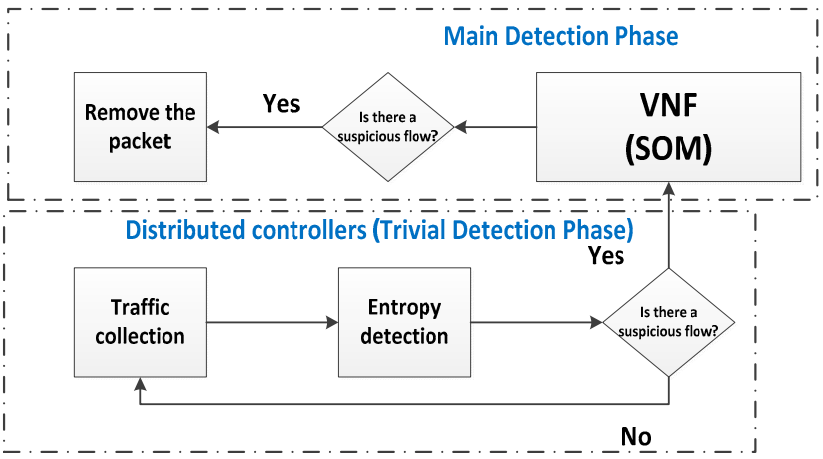

Figure 2. The proposed detection strategy.

3.2.1. Traffic Collection in Forwarding Layer

Although the basic OpenFlow traffic collection method can collect and analyse the required data regarding network traffic, it has some flaws, especially in dealing with vast amounts of data in scalable solutions in 5G networks [20]. Actually, due to the enormous number of packet-in events produced by the OpenFlow switch and sent to the OpenFlow controllers, an equal number of responses for flow entries needs to be passed on and kept in the flow table of the switch, and as a result, the resources of the OpenFlow switch and controllers will be entirely consumed. Additionally, the amount of overhead data in such SDN networks will be very unwieldy to handle.

To overcome the abovementioned restrictions, we propose to take advantage of sFlow (as the southbound API) and its packet sampling ability in the proposed scalable framework. It will be operational by incorporating the sFlow agent within switches [21]. The sFlow agent can be considered a procedure that combines counters of interface and flow samples into datagrams of sFlow, which are sent instantaneously to the sFlow collector module in the local dispersed controller. Then, all the required flow information is forwarded by the sFlow collector to the initial detection module in dispersed controllers.

3.2.2. Initial Anomaly Detection

We employ entropy to measure randomness and detect anomalies. The ability to measure the randomness in 5G traffic is the major reason for utilising entropy for DDoS detection in SDN-based 5G networks. We use two vital components for initial DDoS detection based on entropy: the window size and a threshold, where the entropy is calculated within this window. Moreover, a threshold is required for the initial DDoS attack detection. Assume n is the number of 5G traffic packets in a window, and p_i denotes the probability of each element in the window. Therefore, entropy (E) can be obtained by the following equation defined in [22]:

$$\boldsymbol{E} = -\sum_{i=1}^{n} \boldsymbol{p_i log p_i} \quad (14)$$

In SDN-based 5G networks, the controller recruits a flow for new incoming links to direct the packets to the destination 5G user without further processing. Considering that the packet is new and the destination 5G user is within the network, we can quantify the level of randomness by computing the entropy relying on window size. As a result, if each packet is intended for exactly one 5G user, we will have the maximum entropy. Similarly, suppose all the traffic in a window is intended for a single 5G user. In that case, the minimum amount of entropy happens. We will use this entropy characteristic to compute the randomness in the controller of SDN-based 5G networks. Therefore, entropy can be considered one of the suitable ways for DDoS detection in SDN-based 5G networks due to its capability of quantifying randomness and having minimum and maximum amounts. If a vast number of packets are attacking one 5G user or a subnet of them, the amount of entropy will drop.

In SDN-based 5G networks, the controller is responsible for collecting statistics from the switch tables to calculate the entropy. Distributed controllers monitor flows and remove any inactive flow that remains inactive for a certain amount of time. In this work, we will use this characteristic to attach another statistic to the controller. As long as we know the estimated number of 5G users in the network, the destination IP addresses of the newly arrived packets to be gathered into windows of size 50 are added. We will then compute and compare the entropy of each window with an experimental threshold. Low entropy (below a threshold) indicates a potential attack.

A window size of 50 has been selected due to the following reasons:

- The main reason is the limitation in the number of new connections arriving for each 5G user. In SDN-based 5G networks, as soon as a connection is established, packets will not cross distributed controllers if there are no new requests.
- Another reason is the limitation in the number of OF-switches and 5G users that can be connected to each controller.

- The third reason is the computational complexity and the number of calculations performed in every window. Clearly, 50 values can be calculated much quicker than 100, and attack detection in a window with 50 packets can be conducted much faster.
- Lastly, to determine the appropriate window size, we analysed and tested the entropy for five different window sizes and measured the CPU and memory usage. According to Table 1, the memory usage does not change substantially. However, the CPU usage grows as the window size increases.

Table 1. Window size and its effect on CPU and memory usage.

Window Size	CPU Usage	Memory Usage (GB)
5	55	1.5
50	61	1.5
100	66	1.5
500	75	1.5
5000	91	1.5

Therefore, we set the window size to be less or identical to the number of 5G users. Since the number of arriving new connections to each 5G user in the network is restricted and only a restricted number of switches and 5G users are eligible to be connected to each controller in SDN-based 5G networks with multiple controllers, we use 50 for the window size.

A preventive solution in the case of DDoS attacks on controllers in a 5G scenario is presented here. As soon as a controller receives a packet, a new flow is generated. Therefore, the controller constructs a new flow entry in the OF-switch, and the rest of the packets of the flow are forwarded to the destination user without further processing. Keeping in mind that the received packet is new and that the destination IP addresses exist in the network, it is possible to compute the entropy to determine the randomness amount of the destination IPs. The entropy method is a proper technique to detect DDoS attacks because, in the case of DDoS attacks, a vast amount of traffic is sent to a user or a collection of users. The threshold and window size are the two principal modules for detecting DDoS attacks based on entropy in the early stages.

The destination IP address of the arriving packets from the 5G user should be monitored to detect DDoS attacks in the controller. We added a function to dispersed controllers in the SDN-based 5G network with multiple controllers to build a hash table of the arriving packets. If a 5G user has recently been active in the network and its IP address is new in the table, its count in the table will be one. In the case of the existence of an example of it in the hash, its count will be increased by one. For every 50 packets, we compute the entropy of the window. The hash table is denoted by $W = \{(a_1, b_1), (a_2, b_2), \cdots\}$, in which a is the IP address of the destination, b is the number of times it showed up, and W denotes the window.

The probability of the occurrence of each IP address, p_i, can be computed by $p_i = \frac{a_i}{n}$, where n is the window size. We will have the maximum entropy if each IP address occurs only once and packets are distributed equally in the network. If more packets arrive to a user than others, its entropy will decrease. All entities must be capable of distributing traffic when the 5G network is in its normal working mode. A vast number of packets directed at the host indicates an attack. Such packets will occupy most of the window, reduce the number of unique 5G users, and decrease entropy. In the event of a DDoS attack, a large number of packets are sent to a single user or group of users, causing a reduction in entropy. This decrease in entropy can serve as an indication of a potential attack. To identify a DDoS attack, the entropy level is compared to a predetermined threshold within each window. If it is smaller than the threshold for five consecutive windows, it can be assumed that an attack is taking place. This process is completed within five consecutive windows

of 50 packets each, which is 250 packets in total. This method provides the early detection of a DDoS attack in 5G systems. In this work, we adjust an experimental threshold based on this fact.

Entropy is a measure of the probability of an event happening in relation to the total number of events. For instance, in a 5G network that has 64 mobile users, each user should have an equal chance of receiving new packets. This would result in a high entropy value. However, if some 5G users receive an excessive number of packets, the randomness is decreased and consequently, the amount of entropy drops. If the entropy value falls below a certain threshold, and it remains below that threshold for five successive windows, then an attack is likely to take place. By measuring the entropy in five intervals, we can detect DDoS attacks early, as it would require 250 packets for an attack to take place. Different amounts between one and five successive intervals were tested in this work, and it was proved that five results in the lowest number of false positives for the detection of DDoS attacks in an early manner. Moreover, using five windows has other benefits, such as resilience to potential network device losses or faults in 5G systems. This means that if such occurrences happen and disrupt service for certain 5G users or slow down the influx of new packets to controllers, the impact on system entropy and the likelihood of false positives are reduced. In other words, the use of five windows ensures that administrators of the 5G network have enough time to implement necessary security measures, which enhances overall system robustness and reliability.

3.2.3. Virtualisation Layer

The SOM will be used to classify suspicious flows (the outcome of the initial detection) into normal and malicious packets. The SOM is a tool that uses both supervised and unsupervised learning methods while also being dynamic and adaptive. We use an SOM to classify flows into normal and malicious ones. Normal flows are sent to the cloud for further processing, while malicious ones are eliminated. The virtualisation layer contains multiple VNFs that are used to classify the traffic for the possible detection of DDoS attacks. Suspicious flows are classified by the VNF based on factors such as the source and destination IP addresses, the duration of traffic flow, and other criteria.

In the following section, we discuss the proposed two-phased framework for detecting and mitigating DDoS attacks.

4. Proposed System

This section describes the proposed detection and mitigation framework for DDoS attacks within a multi-controller SDN-based 5G network. Our novel approach addresses the limitations of existing centralised architectures in handling DDoS threats. The framework operates at two key levels: (1) Switch-level processing: incoming traffic flows are first identified and then processed at the switch level and (2) Controller-level analysis: processed information is then forwarded to the relevant controllers for further analysis. Here, malicious activities are swiftly detected, and the controllers make informed decisions to either forward or block the suspicious packets.

In 5G mobile networks, OpenFlow switches play a crucial role in managing data flow [23]. These switches receive incoming packets from user equipment (UE). Each switch maintains a buffer flow table containing rules that define how to handle different types of traffic. When a packet arrives, the switch compares its header information (fields) against the rules in the table. If a matching rule is found, the switch knows exactly where to send the packet—it is forwarded to the appropriate output port based on the rule's instructions. If not, the switch does not have the necessary information to handle it independently. In this case, the packet is directed to the OpenFlow controller, a central entity responsible for making advanced routing decisions. The controller analyses the packet based on its pre-programmed conditions and decides whether to forward it within the network or block it for security reasons. This communication between OpenFlow switches and the

controller is facilitated by the OpenFlow protocol. Figure 3 shows the sample architecture of this system.

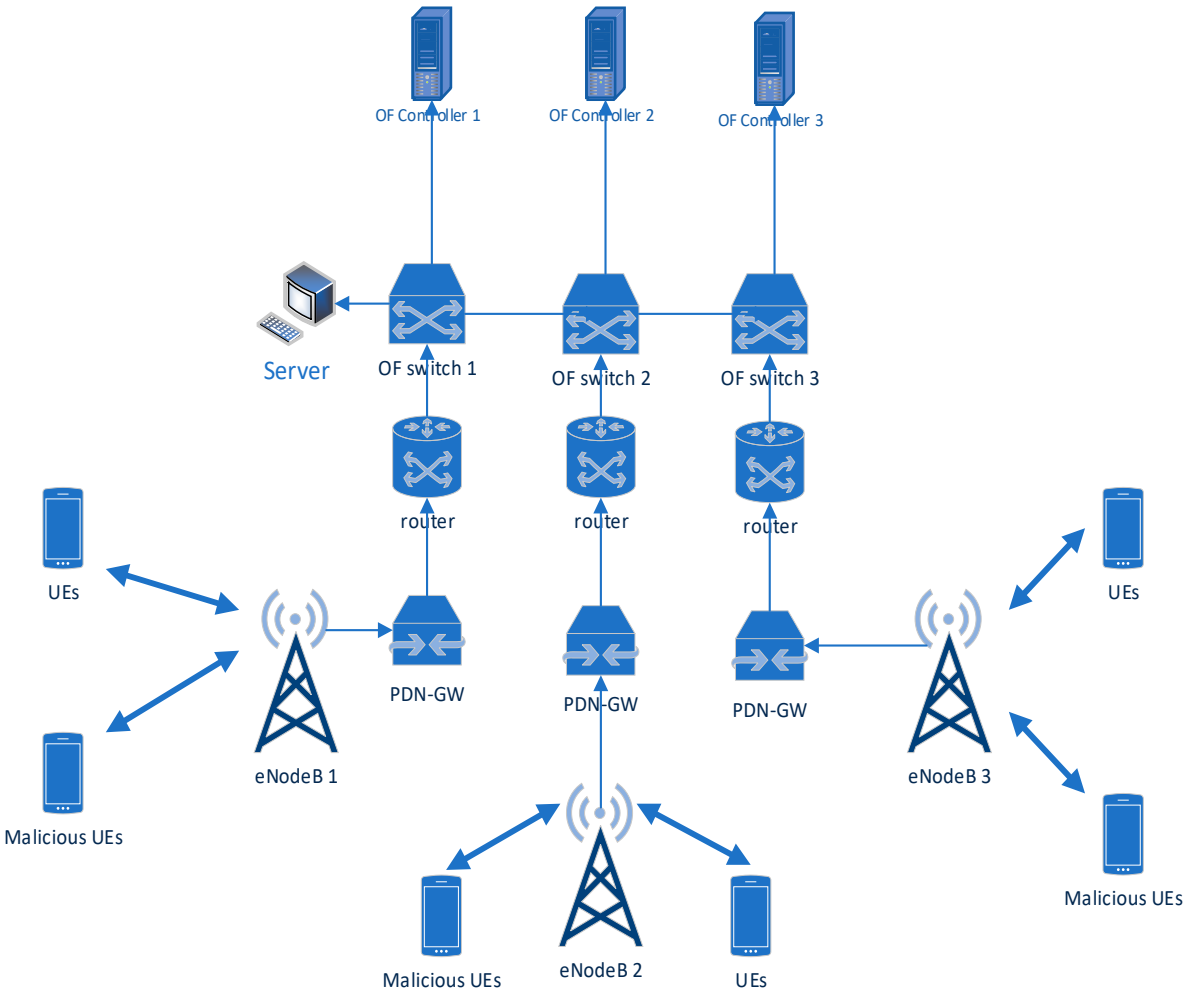

Figure 3. The proposed multi-controller (3 controllers) SDN management architecture.

Unlike traditional multi-controller SDN architectures, our approach utilises a single, central main controller to oversee the entire SDN/NFV-based 5G network. This main controller achieves network-wide management through real-time communication with several distributed controllers. Each distributed controller manages a designated cluster of base stations within the cellular network. To ensure efficient network control, the main controller dynamically creates groups of OpenFlow switches and informs each distributed controller about the members assigned to their group. Figure 3 depicts the interaction between the control plane (as shown in Figure 1) and the data plane within the system. Within the data plane, the Packet Data Network Gateway (PDN-GW) acts as the entry point for traffic entering the 5G network. UE, including potentially malicious ones, communicates with the server (the main controller). Importantly, OF-switches handle all data forwarding within the network but only communicate directly with the OF controller for control instructions.

Evaluation Metrics for Detection Algorithm

The application acts as a binary classifier, categorising network packets as either malicious or benign. Evaluating its classification accuracy is crucial to validate the proof of concept. To achieve this, we must analyse quantifiable data and statistics. A well-established and effective method for binary classifier evaluation is sensitivity and specificity analysis, as detailed in [24]. Sensitivity, also known as True Positive Rate (TPR), measures

the application's ability to detect malicious packets and provides the probability of detection. It can be calculated using the following Equation (15):

$$\boldsymbol{TPR} = \frac{\boldsymbol{TP}}{(\boldsymbol{TP} + \boldsymbol{FN})} \tag{15}$$

Specificity (True Negative Rate or TNR) reflects the likelihood of the application correctly identifying legitimate traffic. The likelihood of retaining legitimate traffic without dropping it can be computed using the following Equation (16):

$$\boldsymbol{TPN} = \frac{\boldsymbol{TN}}{(\boldsymbol{TN} + \boldsymbol{FP})} \tag{16}$$

The OMNeT++ simulation framework supports the generation and propagation of signals during simulations. These signals are subsequently logged for post-simulation statistical evaluation. The code has four signals that are implemented to record true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs), respectively.

- TPs refer to any packet that has been correctly identified as malicious by the switch and subsequently dropped.
- TNs refer to packets that have been correctly identified as benign by the switch and therefore forwarded through a port.
- FPs refer to packets that have been incorrectly identified as malicious and subsequently dropped.
- FNs refer to packets that have been incorrectly identified as benign and allowed to pass through the network instead of blocking them.

5. Performance Evaluation

This section presents the simulation results obtained using OMNeT++ to evaluate our proposed multi-controller SDN/NFV-based 5G network architecture. We focus on two key performance metrics: Firstly, we assess the effectiveness of the proposed methodology in distributing the processing burden evenly across the dispersed controllers within the multi-controller SDN architecture. Secondly, we analyse the performance of the proposed detecting and mitigating attack method in a simulated multi-controller environment representative of 5G networks.

5.1. Simulation Setup

The experiments conducted had specific parameters, which are mentioned in Tables 2–4. To optimise the execution time of the experiment, each network was run for a maximum of 50 s during the simulation. This allowed us to test a full SYN flood attack on the controller application with the maximum amount of connected UE calculated from a previous study [25], which determined that the number of active users who can be connected to and served by a given eNB simultaneously ranges from 60 to 100. In our simulation, we assume the majority of eNB-connected devices are malicious. We utilised a total of 60 UE nodes. Among these, 20 were categorised as benign UE, while the remaining 40 were classified as malicious UE. To ensure accuracy, we averaged outcomes from 100 independent, identically distributed (i.i.d.) random network simulations.

To distinguish between benign and malicious UE, we set non-realistic packet rates; instead, they were proportionally set, with the malicious packet rate notably higher than the benign packet rate. For TCP connection establishment and flow table management, we adopted the default values provided by OpenFlowOMNeTSuite, including those for the SYN-RECEIVED timer, SYN-ACK retransmissions (as outlined in [26]), and flow entry idle timeout. The anomaly detection threshold dynamically adjusted based on UE packet rates. Upon the receipt of an ACK before timer expiration, the connection was successfully established; otherwise, the connection attempt was terminated, and the allocated

resources were released. Additionally, the default idle timeout value for flow entries in OpenFlowOMNeTSuite was adopted. The simulation time for each experiment varied between 15 and 60 min, depending on the events generated and the nodes involved.

Table 2. Parameters used during simulation on OMNET++.

Parameters	Value
Network simulations	100
Runtime per network instance	50 s
Benign UE	20
Benign UE data throughput	0.1 data packets per second (pps)
Malicious UE	40
Malicious UE data throughput	3.5 pps
SYN-RECEIVED Timer	75 s
SYN-ACK Retransmissions	Loop: 3–6–12s
Flow cache timeout	10 s
Attack detection threshold	0.5 pps
TCP Algorithm	TCP Reno

Table 3. UE traffic parameters (benign vs. malicious) for simulation.

Parameters	Value
UE Mobility	Stationary
Packet Flow Direction	Omni-directional
UE Transmit Power	26 dBm
Delayed Acknowledgment (Enabled)	False
SACK Enabled	False
Multiple MIMO	True
Queue size	1 MiB
Max Payload (per TTL)	1 KiB

Table 4. eNB configuration parameters for simulation.

Parameters	Value
Resource block allocation	Distributed
Scheduling strategy	MAXCI
Traffic direction	Omni-directional
Transmit Power	26 dBm
Queue size	2 MiB
Max payload (per TTL)	3000 KiB
TCP App	TCPSessionApp

The properties of real network traffic are highly complex, making theoretical analysis unmanageable. In SDN, the statistical distribution of flow requests follows a Poisson distribution [27]. To ensure that our approach is suitable for statistical flow requests, we add a periodic perturbation to the Poisson distribution [28]. We use iperf to simulate packet-in messages with various rates. The attenuation coefficient is 0.2, which means that

the immediate reward is more important for the system. The learning rate value is set to 0.05. For the greedy algorithm, we start with 0.8 and gradually increase it until it reaches 1. Our simulations employ an SDN-controlled 5G network architecture with four distributed controllers and a central controller. The burden capacity for the four distributed controllers is 14,300 flows/s, 12,500 flows/s, 12,400 flows/s, and 14,400 flows/s. The burden threshold is set to 12,500 flows/s, 10,200 flows/s, 10,500 flows/s, and 12,200 flows/s.

5.2. Modelling Benign UE Traffic

The experiment incorporates 20 UE nodes simulating legitimate users generating realistic traffic. Each UE node initiates a single TCP connection with the server at a randomly chosen time within ten-second intervals (1–10 s, 11–20 s, etc.). The data size for each transmission also varies randomly between 100 and 2000 bytes, and the start time for transmission is chosen uniformly at random.

5.3. Attack Simulation

To simulate a DDoS attack, our simulation incorporates 40 malicious UE nodes that mimic the behaviour of a botnet. These nodes collaboratively launch a TCP SYN flood attack against the server. This attack aims to overwhelm the server with a multitude of uncompleted connection requests, ultimately exhausting its resources.

The attack unfolds in a specific pattern: each malicious UE node transmits a "wave" of 10 SYN messages in rapid succession, repeated every 3 s for a total duration of 29 s. This wave-like pattern is then repeated ten times throughout the entire attack. The first wave commences 2 s after the simulation begins, with each subsequent wave lasting approximately 1 s. By constantly sending these SYN messages and deliberately ignoring the server's response (SYN-ACK), the malicious nodes aim to deplete the server's resources and prevent legitimate users from establishing connections.

5.4. Simulation Results

This section analyses the performance of the proposed multi-controller 5G network through simulation results. First, we will present the results regarding the burden balancing issue. Following this, we will present the results on detecting and mitigating DDoS attacks.

5.4.1. Burden Balancing

In order to evaluate the effectiveness of our proposed methods, we compared the results of our burden balancing strategy with a classic switch relocation method commonly used in SDN networks and a reinforcement learning-based burden balancing approach (RLBBA) [19]. The DDQN serves as the central controller in our 5G load balancing scheme, dynamically relocating OpenFlow switches by analysing the current network state. The RLBBA uses a reinforcement learning model, which learns OF-switch relocation actions by interacting with its environment, to achieve burden balancing among distributed controllers.

Rate of Controller Burden Balancing

The rate of burden balancing is an important index for measuring the quality of the various burden balancing approaches. In our experiment, we utilised iperf to create flow requests with a Poisson distribution that lasted for 24 h. The average rate of the burden balance for various approaches is presented in Figure 4.

Figure 4 shows that the rate of burden balancing for the RLBBA is higher than that of the proposed method. There are two reasons for this difference. Firstly, the reinforcement learning approach used by the RLBBA relies on matrices to record states and actions, which is only suitable for limited and discrete circumstances. In real-world situations, however, states are infinite, making the RLBBA's approach ineffective. Secondly, the RLBBA lacks the ability to generalise, leading to a higher rate of burden balancing after OF-switch relocation. Our proposed method uses neural networks to fit the value of the Q-function, which significantly improves its generalisation capability and lowers the rate of burden balancing.

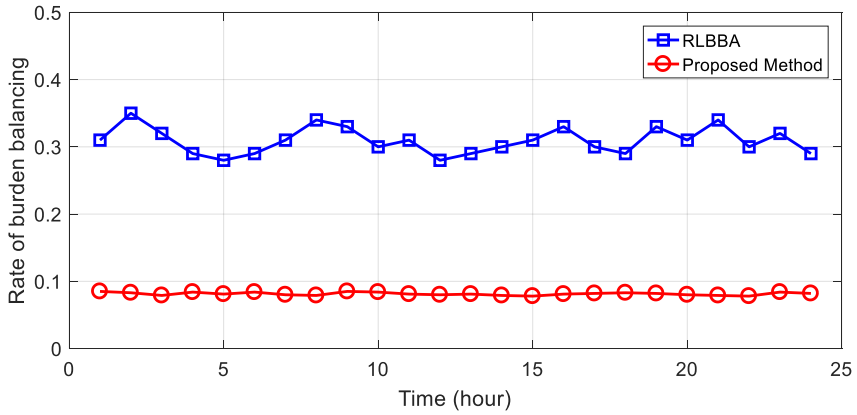

Figure 4. Comparison between proposed method for burden balancing and classic method in terms of rate of burden balancing.

Burden Ratio

The simulation result presented in Figure 5 quantifies the quality of burden balancing approaches by measuring the burden ratio of controllers.

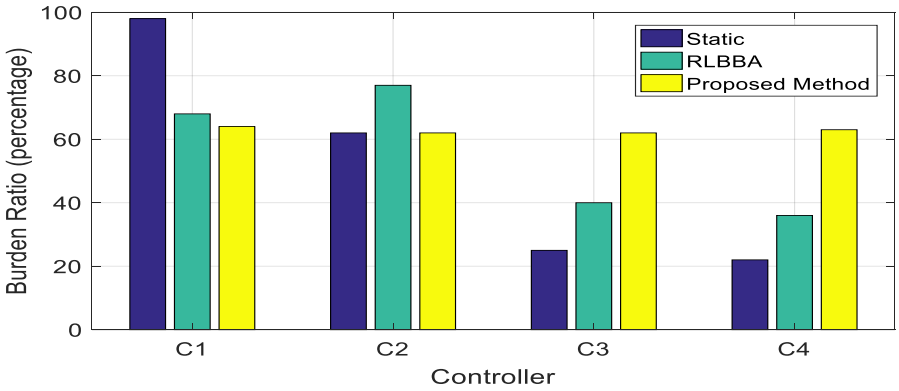

Figure 5. Burden ratio of distributed controllers for various burden balancing approaches.

Due to static burden balancing, the OF-switches in the 5G network cannot be relocated. Figure 5 shows a significant difference in the burden ratio of distributed controllers. To address this issue, the RLBBA has been used to relocate the OF-switches controlled by C1. However, due to the lack of generalisation capability, the burden ratio still varies after relocation. We address this challenge with a DDQN-based load balancing approach to extract the features of the system for decision-making. This approach reduces the required computation, boosts the generalisation capability, and minimises the difference in the burden ratio of distributed controllers. Table 5 presents the average controller burden ratio. The average burden ratio of our proposed method is higher and provides little variation in the burden ratio among distributed controllers.

Table 5. The average burden ratio of distributed controllers for various approaches.

Approach	Average Burden Ratio of Distributed Controllers
Static	0.5175
RLBBA	0.5525
Proposed Method	0.6275

Balancing Duration

We use the balancing duration to quantify the time it takes to perform the relocation approach. In this regard and in the simulations, we increase the request rate of the first distributed controller until it exceeds its threshold. Then, we implement burden balance approaches and compare the balancing duration and the burden of distributed controllers to measure the effectiveness of the approaches. The results of these comparisons are shown in Figure 6.

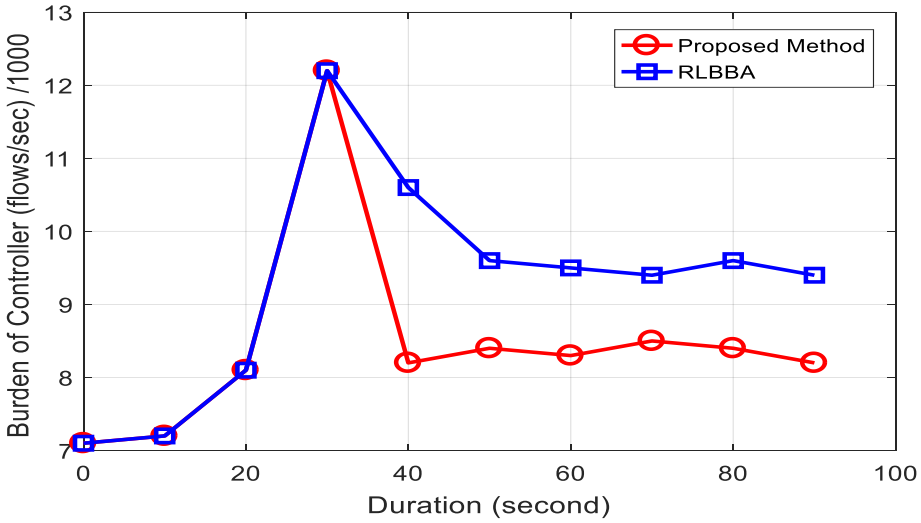

Figure 6. Balancing duration for various burden balancing approaches.

From Figure 6, it is clear that the burden on the first controller varies over time when running both the RLBBA and our proposed method. In the thirtieth second, the first controller experiences a significant number of requests from OF-switches, causing the controller's burden to exceed the threshold. At this moment, the proposed method relocates the burden and reduces the controller's burden to a normal level in the following seconds. Similarly, the RLBBA relocates the burden process around the fiftieth second. This experiment demonstrates the efficiency of our proposed method in balancing the burden in multi-controller SDN-based 5G networks while also reducing the duration of the balancing process.

5.4.2. DDoS Attack Detection and Mitigation

The experiment replicates the attack scenario and SDN controller application on each controller in the multi-controller 5G network. The evaluation metrics for the detection algorithm are presented in Equations (17) and (18). We evaluate the proposed system's performance using three key metrics: the detection rate, packet loss rate, and delay. The DDoS attack detection rate specifically evaluates how effectively the system identifies malicious activity under high attack loads. It is mathematically expressed as follows (17):

$$\textit{Detection Rate} = \frac{\textit{True Negatives}}{\textit{Attack Rate}} \times \mathbf{100}, \tag{17}$$

where the attack rate is equal to the total number of attacks. The packet loss rate in a 5G network refers to the rate at which packets are lost. There are many reasons why packets may get lost, such as traffic issues or collisions in the packet switched network. When packets get lost, they may not reach their intended destination. The packet loss rate is expressed as follows (18):

$$\textit{Packet Loss Rate} = \textit{Packet Forward Rate} - \textit{Packet Recieved Rate} \tag{18}$$

The delay metric in our work measures the delay in providing packet-in messages as a response to UE and is calculated by subtracting the time taken to send and receive packets to specific UE.

It is important to note that the following results represent performance in the steady state of the multi-controller scenario, achieved after balancing the workload across controllers. In this study, we compare our proposed framework with some existing works, namely the robust security scheme (RS) in [29] and SDN-cloud computing (SDN-CC) in [30] in terms of the detection rate, packet loss rate, and delay.

The detection rate plays a crucial role in measuring the ability of various attack detection frameworks and comparing their performance. Figure 7 shows the relationship between the DDoS attack detection rate and the false positive rate (FPR).

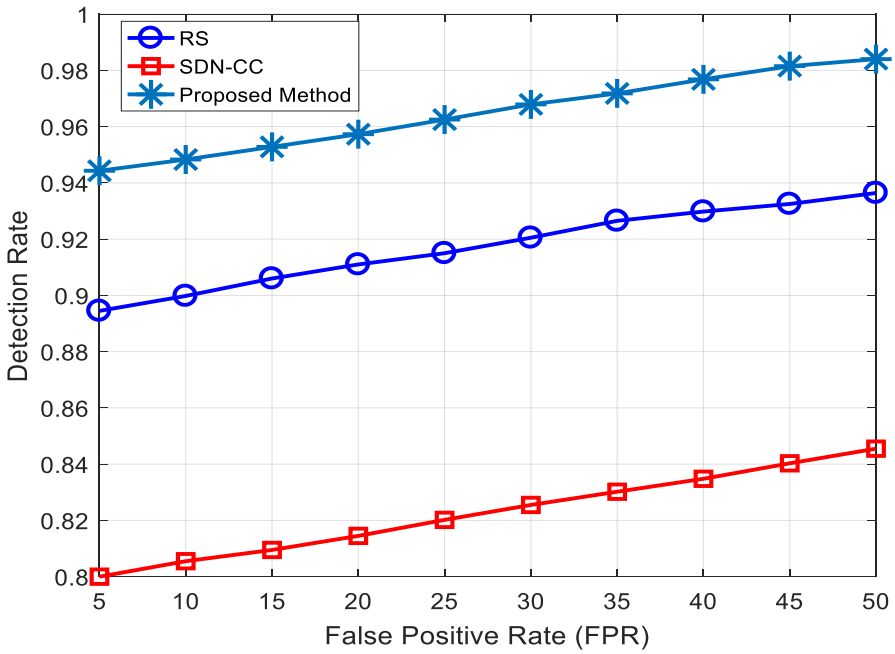

Figure 7. DDoS attack detection rate with FPR.

We have considered the detection rate for proving the efficacy of our proposed method in mitigating DDoS attacks in 5G networks and compared it with existing works in Figure 7. Our proposed method outperforms previous works significantly in terms of detection rate. This is because the classification methods used in [29,30] are not optimal and require a large amount of computational complexity.

We also evaluate the packet loss rate, comparing our results to existing work. Congestion occurring in OpenFlow switches and the network itself is the leading cause of packet loss. In SDN-based 5G networks, it is necessary to monitor the packet loss rate constantly. It is possible to calculate the packet loss rate through either per-packet flow or per switch. We employ a per-packet flow approach to calculate packet loss rate, as illustrated in Figure 8. The packet loss rate was compared with existing works, and it was found that SDN-CC's performance deteriorated as the number of OF-switches increased, in contrast to RS and the proposed method.

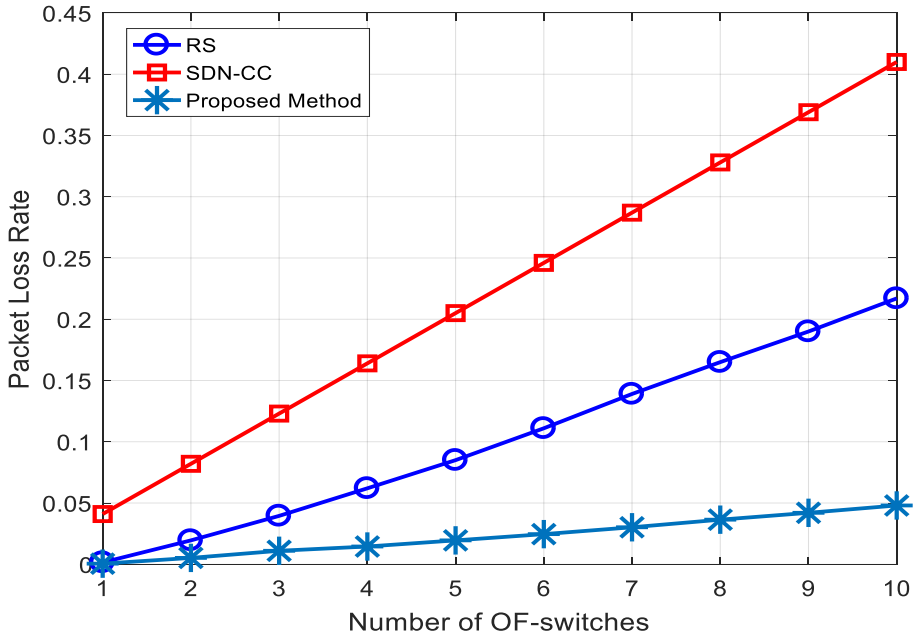

Figure 8. The packet loss in terms of the number of OF-switches.

Figure 9 shows the relationship between the packet loss rate and the data rate. The data rate refers to the total number of packets transmitted from the source to the destination within a given time frame. By comparing our proposed solution with existing methods, we found that our approach achieves better performance with less computational effort in terms of the packet loss rate.

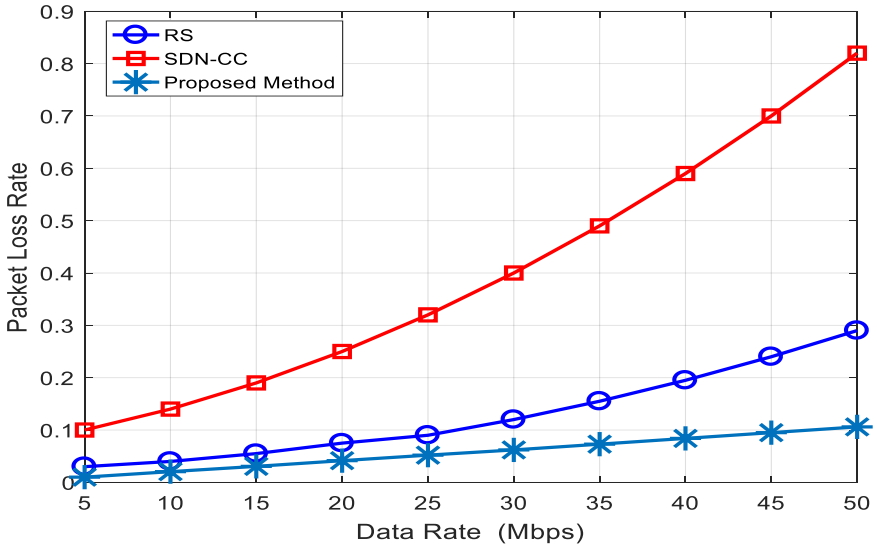

Figure 9. The packet loss in terms of the data rate.

The time taken for data to be transmitted from its source to its destination is known as a delay. This calculation takes into account the queuing delay as well. In Figure 10, we compare the delay of our proposed system to existing works. Our method achieves demonstrably lower delay compared to existing approaches.

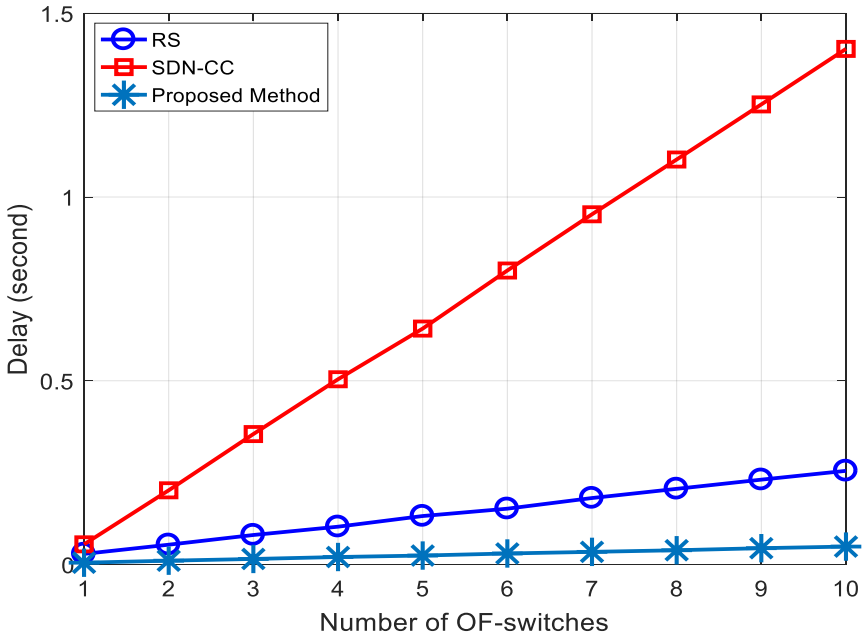

Figure 10. The delay in terms of the number of OF-switches.

6. Conclusions

This paper introduces a new intrusion detection system and burden balancing technique designed explicitly for multi-controller SDN/NFV-based 5G networks. The proposed framework consists of three layers. The two upper layers are responsible for detecting and mitigating DDoS attacks, while the second layer handles the adaptation and balancing of the burden of distributed controllers. The controllers are classified into two categories: distributed controllers and a main (centralised) controller. The distributed controllers leverage a lightweight entropy-based method to classify incoming packets into normal and suspicious. Suspicious packets undergo further categorisation into normal packets and malicious packets, which are then stored in NFV utilising an SOM. In multi-controller SDN scenarios, the static relationship between OF-switches and controllers can lead to burden imbalance in distributed controllers. To overcome this problem, it is necessary to relocate the OF-switches. The proposed solution uses a Markov decision process to describe the system's state, relocation action set, and other parameters. This is combined with deep reinforcement learning to present a novel OF-switch relocation approach for multi-controller SDN/NFV-based 5G networks.

We have assessed the effectiveness of the proposed methodology in distributing the processing burden evenly across the dispersed controllers within the multi-controller SDN architecture and analysed the performance of the proposed detecting and mitigating attack method in a simulated multi-controller environment representative of 5G networks.

Based on the experimental results, the proposed framework provides better performance in the burden balancing of distributed controllers, augmenting the mean burden rate, reducing the balancing duration, decreasing the packet loss rate and the overall system delay, and increasing the detection rate, compared to existing works.

In our future work, we plan to incorporate comparisons with the most recent studies to enhance and complement our evaluation results, particularly in relation to RS and SDN-CC methods. We will also explore other machine learning-based approaches and network slicing techniques. A multifaceted approach will also be investigated to enhance entropy's effectiveness in detecting DDoS attacks under bursty traffic conditions.

Author Contributions: M.S.: conceptualisation, investigation, methodology, software, validation, and writing—original draft preparation. S.K.: methodology, writing—review and editing, and supervision. I.A.: methodology, writing—review and editing, and supervision. A.Q.: writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets presented in this article are not readily available because the experiments were mainly conducted by the first author, who unfortunately passed away. The experimental data were stored on his personal machine, and there was insufficient time to make arrangements to share the data publicly. If access to the personal machine of the deceased author is permitted, we will make every effort to retrieve the data. We understand the importance of data accessibility for research transparency and are committed to finding a solution to address this issue. Requests to access the datasets should be directed to Savas Konur.

Acknowledgments: In memory of Morteza Sheibani, who made significant contributions to this work but sadly passed away before its submission. We are grateful for his dedication, hard work, and commitment.

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

References

1. Ayodele, B.; Buttigieg, V. SDN as a defence mechanism: A comprehensive survey. *Int. J. Inf. Secur.* **2024**, *23*, 141–185. [CrossRef]
2. Khan, S. Detection of DoS and DDoS Attacks on 5G Network Slices Using Deep Learning Approach. Ph.D. Thesis, University of Regina, Regina, SK, Canada, 2023.
3. Yungaicela-Naula, N.M.; Vargas-Rosales, C.; Pérez-Díaz, J.A. SDN/NFV-based framework for autonomous defense against slow-rate DDoS attacks by using reinforcement learning. *Future Gener. Comput. Syst.* **2023**, *149*, 637–649. [CrossRef]
4. Shoaib, F.; Chow, Y.-W.; Vlahu-Gjorgievska, E.; Nguyen, C. Mitigating Timing Side-Channel Attacks in Software-Defined Networks: Detection and Response. *Telecom* **2023**, *4*, 877–900. [CrossRef]
5. Wang, G.; Zhao, Y.; Huang, J.; Wang, W. The Controller Placement Problem in Software Defined Networking: A Survey. *IEEE Netw.* **2017**, *31*, 21–27. [CrossRef]
6. Chen, M.; Ding, K.; Hao, J.; Hu, C.; Xie, G.; Xing, C.; Chen, B. LCMSC: A lightweight collaborative mechanism for SDN controllers. *Comput. Netw.* **2017**, *121*, 65–75. [CrossRef]
7. Zhang, L.; Wang, Y.; Li, W.; Qiu, X.; Zhong, Q. A survivability-based backup approach for controllers in multi-controller SDN against failures. In Proceedings of the 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), Seoul, Korea, 27–29 September 2017; pp. 100–105.
8. Sheibani, M.; Konur, S.; Awan, I. DDoS Attack Detection and Mitigation in Software-Defined Networking-Based 5G Mobile Networks with Multiple Controllers. In Proceedings of the 9th International Conference on Future Internet of Things and Cloud (FiCloud), Rome, Italy, 22–24 August 2022; pp. 32–39.
9. Yazici, V.; Sunay, M.O.; Ercan, A.O. Controlling a Software-Defined Network via Distributed Controllers. *arXiv* **2014**, arXiv:cs.NI/1401.7651.
10. Krishnamurthy, A.; Chandrabose, S.P.; Gember-Jacobson, A. Pratyaastha: An efficient elastic distributed sdn control plane. In Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, Chicago, IL, USA, 22 August 2014; pp. 133–138.
11. Dixit, A.A.; Hao, F.; Mukherjee, S.; Lakshman, T.V.; Kompella, R.R. ElastiCon; an elastic distributed SDN controller. In Proceedings of the 2014 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), Marina del Rey, CA, USA, 20–21 October 2014; pp. 17–27.
12. Koerner, M.; Kao, O. Multiple service load-balancing with OpenFlow. In Proceedings of the IEEE 13th International Conference on High-Performance Switching and Routing (HPSR), Belgrade, Serbia, 24–27 June 2012; pp. 210–214.
13. Yao, H.; Qiu, C.; Zhao, C.; Shi, L. A Multicontroller Load Balancing Approach in Software-Defined Wireless Networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 4541–4559. [CrossRef]
14. Kalliola, A.; Lee, K.; Lee, H.; Aura, T. Flooding DDoS Mitigation and Traffic Management with Software Defined Networking. In Proceedings of the IEEE 4th International Conference on Cloud Networking (CloudNet), Niagara Falls, ON, Canada, 5–7 October 2015; pp. 248–254.
15. Wang, W.; Qi, Q.; Gong, X.; Hu, Y.; Que, X. Autonomic QoS management mechanism in software-defined network. *China Commun.* **2014**, *11*, 13–23. [CrossRef]
16. Gudipati, A.; Perry, D.; Li, E.L.; Katti, S. SoftRAN: Software defined radio access network. In *The Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13)*; Association for Computing Machinery: New York, NY, USA, 2013; pp. 25–30.
17. Bernardos, C.J.; de la Oliva, A.; Serrano, P.; Banchs, A.; Contreras, L.M.; Jin, H.; Zúñiga, J.C. An architecture for software defined wireless networking. *IEEE Wirel. Commun.* **2014**, *21*, 52–61. [CrossRef]

18. Pentikousis, K.; Wang, Y.; Hu, W. Mobileflow: Toward software-defined mobile networks. *IEEE Commun. Mag.* **2013**, *51*, 44–53. [CrossRef]
19. Li, Z.; Zhou, X.; Gao, J.; Qin, Y. SDN Controller Load Balancing Based on Reinforcement Learning. In Proceedings of the IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 November 2018; pp. 1120–1126.
20. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.A.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
21. Giotis, K.; Argyropoulos, C.; Androulidakis, G.; Kalogeras, D.; Maglaris, V. Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. *Comput. Netw.* **2014**, *62*, 122–136. [CrossRef]
22. Lee, S.; Kim, J.; Shin, S.; Porras, P.; Yegneswaran, V. Athena: A framework for scalable anomaly detection in software-defined networks. In Proceedings of the 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Denver, CO, USA, 26–29 June 2017; pp. 249–260.
23. Mehdi, S.A.; Khalid, J.; Khayam, S.A. Revisiting traffic anomaly detection using software defined networking. In *14th International Symposium on Recent Advances in Intrusion Detection (RAID)*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 161–180.
24. Yungaicela-Naula, N.M.; Vargas-Rosales, C.; Perez-Diaz, J.A. SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning. *IEEE Access* **2021**, *9*, 108495–108512. [CrossRef]
25. Sauter, M. *Beyond 3G-Bringing Networks, Terminals and the Web Together: LTE, WiMAX, IMS, 4G Devices and the Mobile Web 2.0*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
26. Schuba, C.L.; Krsul, I.V.; Kuhn, M.G.; Spafford, E.H.; Sundaram, A.; Zamboni, D. Analysis of a denial-of-service attack on TCP. In Proceedings of the IEEE Symposium on Security and Privacy (Cat. No. 97CB36097), Oakland, CA, USA, 4–7 May 1997; pp. 208–223.
27. Wang, T.; Liu, F.; Xu, H. An efficient online algorithm for dynamic SDN controller assignment in data center networks. *IEEE/ACM Trans. Netw.* **2017**, *25*, 2788–2801. [CrossRef]
28. Sun, P.; Guo, Z.; Wang, G.; Lan, J.; Hu, Y. MARVEL: Enabling controller load balancing in software-defined networks with multi-agent reinforcement learning. *Comput. Netw.* **2020**, *177*, 107230. [CrossRef]
29. Yao, J.; Han, Z.; Sohail, M.; Wang, L. A robust security architecture for SDN-based 5G networks. *Future Internet* **2019**, *11*, 85. [CrossRef]
30. Bhushan, K.; Gupta, B.B. Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1985–1997. [CrossRef]