algorithms

MDPI

Article

Federated Learning-Based Security Attack Detection for Multi-Controller Software-Defined Networks

Abrar Alkhamisi [1,*], Iyad Katib [1] and Seyed M. Buhari [2]

[1] Computer Science Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; iakatib@kau.edu.sa

[2] UTB School of Business, Universiti Teknologi Brunei, Bandar Seri Begawan BE1410, Brunei; ismail.buhari@utb.edu.bn

* Correspondence: aalkhamisi0034@stu.kau.edu.sa

Abstract: A revolutionary concept of Multi-controller Software-Defined Networking (MC-SDN) is a promising structure for pursuing an evolving complex and expansive large-scale modern network environment. Despite the rich operational flexibility of MC-SDN, it is imperative to protect the network deployment against potential vulnerabilities that lead to misuse and malicious activities on data planes. The security holes in the MC-SDN significantly impact network survivability, and subsequently, the data plane is vulnerable to potential security threats and unintended consequences. Accordingly, this work intends to design a Federated learning-based Security (FedSec) strategy that detects the MC-SDN attack. The FedSec ensures packet routing services among the nodes by maintaining a flow table frequently updated according to the global model knowledge. By executing the FedSec algorithm only on the network-centric nodes selected based on importance measurements, the FedSec reduces the system complexity and enhances attack detection and classification accuracy. Finally, the experimental results illustrate the significance of the proposed FedSec strategy regarding various metrics.

Keywords: multi-controller SDN; data plane; deep learning; SDN attacks; federated learning; attack detection

check for updates

Citation: Alkhamisi, A.; Katib, I.; Buhari, S.M. Federated Learning-Based Security Attack Detection for Multi-Controller Software-Defined Networks. *Algorithms* **2024**, *17*, 290. https://doi.org/10.3390/a17070290

Academic Editors: Mario Rosario Guarracino, Laura Antonelli and Pietro Hiram Guzzi

Received: 25 April 2024
Revised: 21 June 2024
Accepted: 24 June 2024
Published: 2 July 2024

1. Introduction

In the digital world, SDN is emerging as a promising network paradigm that offers flexibility to the network infrastructure by clearly decoupling the data and control planes [1]. SDN simplifies the vertically integrated complex network by separating centralized network control and data forwarding. This process significantly enhances network flexibility and management by decomposing the components. The SDN revolutionizes network technology, and its inherent architecture offers enormous benefits. For instance, it enables global network visibility in a centralized control manner and can cost-effectively offer better coverage and capacity. Due to the rapid advancement of SDN technology, innovative SDN solutions are being developed and deployed, such as the Open Networking Automation Platform (ONAP), VMware NSX-based network virtualization platform, and Cisco ACI (Application Centric Infrastructure), to name a few. SDN has been adopted recently by many networks, such as the Internet of Things (IoT) and Wireless Sensor Networks (WSN), to improve their communication remarkably [2,3]. However, the main problem associated with the SDN is its centralized architecture, which leads to the possibility of a single point of failure and scalability issues. To avoid a single point of controller failure and efficiently handle large multi-domain networks, the MC-SDN is a promising solution, as each controller is responsible for managing a domain [4]. The robust MC-SDN design provides a powerful architecture that offers scalability, flexibility, and resilience to failures, making it a potential choice for large and complex network environments. The capabilities of SDN controllers are criticized for network availability, performance, and scalability. Thus, enabling multiple controllers is imperative for managing large-scale networks to improve the above issues, even though it increases network complexity and management [5,6]. The rapid deployment of SDN applications by multiple entities leads to complex network environments, which are highly vulnerable to several attacks.

The data plane attacks pose potential threats to the SDN, such as the DoS and side-channel attacks, and belong to a data-to-control plane saturation attack and time delay based on the network configurations, respectively. Most conventional research utilizes machine learning-based solutions [7,8] that can potentially analyze the network traffic and classify the attacks under different categories. Generally, machine learning solutions [9] necessitate large traffic to learn the environment for attack detection and classification. Among the ML-based solutions, federated DL algorithms have significant potential to improve classification accuracy, as they can obtain global attack knowledge with the assistance of federated model parameters from large-scale multi-domain networks [9]. Utilizing single DL-based attack detection algorithms leads to inaccurate classification in many scenarios due to inconsistent and erroneous dataset information. Hence, making attack detection decisions based on hybrid learning models is essential to improve classification accuracy. Moreover, federated DL improves the cybersecurity posture by gaining the patterns from different attempts of flow table manipulation with minimal communication overhead.

Therefore, the main research goal of the proposed FedSec attack detection is to utilize the advantages of key enabling technologies federated learning (FL) to improve the security level of MC-SDN against various attacks in data planes.

The main contributions of the proposed work are given as follows:

- The primary intention of the proposed system is to enhance the MC-SDN security and efficiency against various attacks by adopting a federated learning paradigm.
- Implementing the FedSec only on the network-centric nodes reduces the system complexity and overhead without affecting the intrusion detection accuracy.
- To customize the generalized network traffic into SDN flow traffic, this work targets to enrich the benchmark NSL-KDD dataset with unique MC-SDN-specific attack features, resulting in the improved effectiveness of the intrusion detection strategy.
- The proposed FedSec optimally selects the potential feature set using the hybrid heuristic algorithm and constructs the malicious activity-aware graph structure with the enriched edge weight computation, offering the generation of high-level abstractive features as the input to the classification model.
- The FedSec designs GCNN-GRU as a local model in the federated model, which collaboratively extracts spatial and temporal features in the large-scale and diversified SDN network traffic to determine the attack behaviors in the data plane layer.

2. Related Works

Recently, several works have been developed to provide SDN security using DL models and have listed the advantages and limitations of the SDN protocols, especially from the security perspective, including DoS and illegal intrusion attacks on SDN control nodes. This section reviews the DL model-based SDN security approach. Table 1 comparatively reviews several DL-based SDN security research works.

The research work in [10] presented a Deep Learning-based Intrusion Detection and Prevention (DL-IDPS) Scheme that protects an SDN against brute-force and DDoS attacks. Moreover, by obtaining the SDN packet length, the DL-IDPS identifies the malicious packets and the attacker by analyzing the gathered evidence. Temporal pattern or feature analysis is vital in investigating the attack behaviors in the network traffic. Hence, security researchers have increasingly applied RNN-based deep learning models in addition to spatial analysis. For instance, the SDN security approach uses the GRU model [11] to protect the flow records within the SDN by identifying intrusions and DDoS attacks. The impact of the attackers on the SDN is reduced by the GRU-based lightweight SDN security model design and the direct flow inspection. The work in [12] detects the DDoS attacks in the IoT network with the design of the data plane and control plane in the SDN. Initially, the deployed sFlow and adaptive polling-based sampling enforce the minimization of processing and network overheads in the data plane switches. Subsequently, the Snort IDS integrated with the Stacked Autoencoders model improves the detection accuracy in the control plane. The adversarial training-based detection and defense system [13] employs the Generative Adversarial Network to detect DDoS attacks in SDN. Continuously monitoring the network traffic from the IP flow analysis ensures real-time anomaly detection.

The SDN-enabled secure approach [14] employs the Deep Neural Network and LSTM models to identify new cyber risks in the Internet of Things (IoT). In an IoT network, the hybrid deep learning model identifies frequent and rarely occurred cyber threats like botnet attacks, port scanning attacks, DDoS attacks, and brute-force attacks. The work in [15] determines and classifies the DDoS attacks over a multi-controller SDN environment. In addition to detecting the attacks, it traces the paths utilized by the attackers. It initiates a mitigation step to safeguard the network devices with the assistance of feature selection and hybrid deep learning algorithms. Moreover, the work in [16] ensures accurate attack detection in the large-scale SDN, particularly for a flow-based intrusion detection system, by applying the GRU-LSTM and feature selection. Also, the SDN-enabled attack detection approach [17] utilizes the hybrid deep learning models involving the Cuda-Deep Neural Network GRU and Cuda-Bidirectional LSTM for a highly scalable and affordable solution for accurate defense mechanisms in an IoT context.

The research in [18] presents an online attack detection and mitigation system for the SDN infrastructure. To detect the anomalies in the network traffic, it designs a lightweight hybrid DL model, namely CNN and Extreme Learning Machine, and then, to filter the abnormal traffic during the flow table update, it exploits the IP traceback that locates the attacker in the SDN. To improve the intrusion detection accuracy in SDN, the work in [19] initially performs the hybrid feature selection for the large-scale network traffic data and applies the hybrid DL model, CNN and BiLSTM, for the binary and multiclass classification. The abnormal traffic detection approach [20] initially utilizes the port information to detect the abnormal traffic roughly and applies the wavelet transform along with the hybrid DL model in the hierarchical model to detect the intrusions in the SDN switches precisely. Conversely, as presented in [21], the spatial aspects of the SDN traffic flow have been investigated using the enhanced CNN model with graph features, namely Spatial–Temporal Graph Convolutional Network (ST-GCN). With the assistance of In-band Network Telemetry support, the ST-GCN model identifies the SDN switches impacted by DDoS attacks. The work in [22] presents the application of a Perceptron-based deep learning technique to improve Quality of Service (QoS) and security within software-defined networks. The limitations in this work dataset cannot reflect the characteristics of the SDN environment.

In order to enhance security and handle distributed network traffic, several security researchers have employed federated learning models against emerging attack types. For example, Deep Monitor [23] is an intelligent flow rule match-field control system that monitors fine-grained network traffic in SDN-based IoT edge nodes considering maximum flow table capacity. Moreover, to improve the learning ability of the edge nodes, the work in [23] applied a federated Double Deep Q network as the traffic monitoring mechanism. Another research work in [24] applied the Weighted Federated Learning to identify the Low-Rate DDoS attacks. By examining the model accuracy, a robust preference assignment mechanism in [24] leverages the federated server to prioritize the locally trained models. Table 1 comparatively reviews several learning-based SDN security research works.

Table 1. Comparison of Learning-based SDN Security Solutions.

Ref	Year	Work	Learning Model	Dataset	Limitations
[16]	2019	Flow-based Anomaly detection.	GRU and LSTM	NSL-KDD	Single point of controller failure affects the flow table security.
[10]	2020	Flow-based IDPS for preventing SSH brute-force and DDoS attacks.	MLP	Flow-based SDN dataset	Lack of providing detailed packet details to the SDN controller.
[11]	2021	GRU-based SDN Attack Detection.	GRU	CICDDoS 2019 and CICIDS 2018	Confronts with a single point of controller failure.
[17]	2021	Hybrid DL-based IoT Cyber threats detection.	CuDNNLSTM and CuDNNGRU	CICDDoS 2019	Lack of examining the spatial features of the network traffic.
[21]	2021	Spatial–temporal DL-based DDoS detection.	GCNN	CICDDoS	Efficiency is a major constraint due to the lack of feature selection in large-scale networks.
[14]	2022	Cyber threat detection for smart environment.	DNN and LSTM	CICIDS2018	Fails to learn the spatial features in the network structure.
[15]	2022	Multi-controller SDN for DDoS detection.	LSTM	CIC-DDoS2019	The lack of SDN structural relationship-based attack detection degrades the performance.
[18]	2022	DDoS attack detection and mitigation.	CNN and ELM	CICIDS-2017 and InSDN	Fails to learn the diversified attack behaviors.
[19]	2023	Network traffic classification with feature selection.	CNN and BiLSTM	NSL-KDD, UNSW-NB15, and InSDN	The lack of investigating structural relationships in the network traffic flows misleading the spatial feature learning.
[24]	2023	DDoS attack detection.	Weighted Federated Learning	-	Large-scale network traffic features affect the latency.
[22]	2024	A deep learning technique to detect distributed denial of service attacks in software-defined networks.	AE and BGRU	NSLKDD dataset	Data cannot reflect the characteristics of the SDN environment.

3. The Proposed Methodology

MC-SDNs revolutionize network management by separating the control plane from the data plane. Hence, both planes have different characteristics and functionalities, and the attacks faced by these layers are also different. The proposed FedSec only countermeasures attacks against data plane security. Its main intention is to detect data plane security attacks and improve data plane efficiency. Thus, in the system model, the FedSec made two assumptions. Firstly, the MC-SDN structure included for implementing FedSec among data plane nodes and a server is highly secure, and there are no control plane attacks. Hence, the controllers are only employed to establish communication with cloud servers. Secondly, the network-centric nodes in the data plane are highly trusted.

While the FedSec exploits secure MC-SDN structure, it is crucial to determine the data plane attacks such as traffic analysis, packet spoofing, false flow rule injection, man-in-the-middle, and distributed denial of service in SDN environments that impact and degrade the data plane performance level. Therefore, the FedSec intends to detect such attacks with the assistance of deep learning and federated learning strategies. Protecting the SDN nodes from various attacks is crucial to accomplishing the entire MC-SDN performance as high. Therefore, the FedSec-based security strategy utilizes federated learning to improve the learning efficiency and attack detection performance at the data plane level. As depicted in Figure 1, the proposed FedSec deploys the local model at SDN nodes, which are determined by the direct and indirect importance-based network-centric node selection in each domain, resulting in minimum resource consumption with wider coverage. The MC-SDN structure comprises main and sub-controllers within the control layer in which the main controller is only used to supervise all the devices in the SDN data plane. The main role of the sub-controllers is to monitor the activities of the main controller, thereby evaluating the honesty level of the main controllers. The sub-controllers do not engage in other operations like data plane monitoring but focus only on ensuring that the main controller acts in a reliable and trustworthy manner. Furthermore, the main controller within a domain is responsible for controlling the data plane activities and improving the data transfer efficiency via accurate flow table maintenance. The proposed work implements the federated learning structure among the network-centric nodes selected in the data plane and cloud server in which the nodes share the models with the cloud server through the main controller. The cloud server is responsible for aggregating the local model parameters of various network-centric nodes through FedAvg and shares the global model reversely to the corresponding nodes via the main controller. During the global model training in the Federated SDN, the widely learned normal and malicious traffic patterns facilitate the building of generalized attack detection with collaborative knowledge. By obtaining the global attack knowledge through controllers, the network-centric nodes retrain their learning models and create a wider attack knowledge for attack detection. Thus, it significantly improves attack detection during testing and proves the effectiveness of the proposed FedSec model.

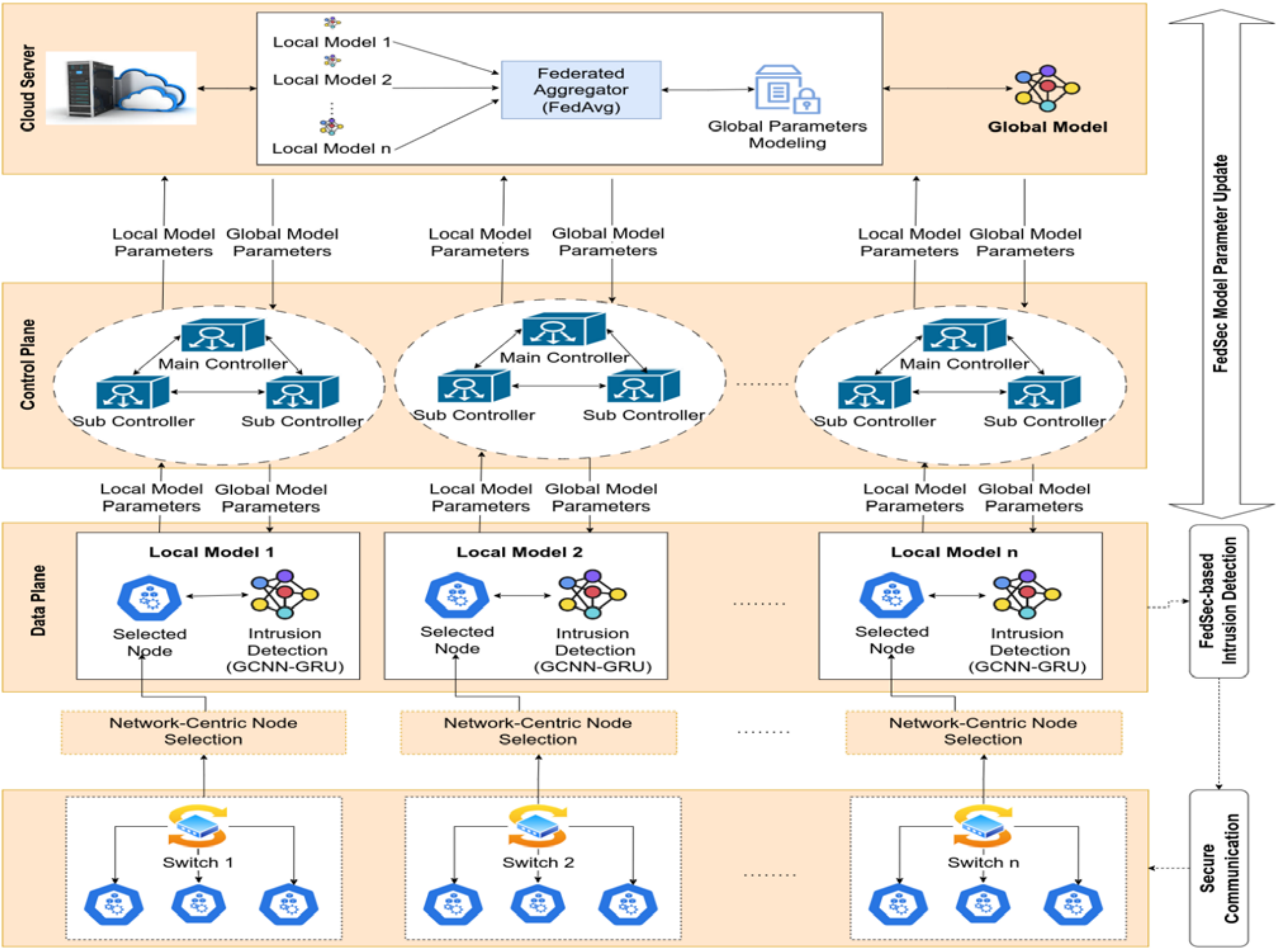

Figure 1. Design Overview of FedSec.

To secure the SDN nodes at the data plane, the proposed system designs the FedSec mechanism with four phases: Network-centric node selection, Hybrid heuristic-based feature selection, SDN-specific dataset enrichment, and Attack detection. In particular, attack detection or security mechanism through the classification model refers to the GCNN-GRU model in this work, accomplished by discriminating the normal and malicious traffic.

3.1. Network-Centric Node Selection

In the modeling of federated paradigm are 'C' controllers, {C1, . . ., CK}, in which each controller comprises different network traffic data, {D1, . . ., DK}. For training the learning model, the local data managed by each SDN controller is provided as the input matrix that consists of 'T' traffic flow samples as a row and features as columns. Hence, the sample ID is X, and the feature space is Y for the horizontal federated learning schema. In essence, the controllers share the same feature space for the different network traffic flows generated by hosts with different services. Wherein, $X_a \neq X_b, Y_a = Y_b, \forall T_a, T_b, a \neq b$, in which 'a' and 'b' refer to the different clients or controllers in the MC-SDN environment. The proposed approach develops the attack detection mechanism for the data plan security to protect the flow table. Hence, for deploying the proposed model's implementation, the FedSec attack detection algorithm is executed on selected network-centric nodes to ensure security coverage in the overall SDN. In this scenario, it is assumed that the selected network-centric nodes are trusted. These nodes are also federated clients that participate in the model sharing process and perform attack detection.

The main objectives behind the node selection for the implementation of the attack detection are that the nodes or switches can analyze the incoming traffic flows, update flow entries in the flow table, and perform actions regarding the traffic flows based on the rules mentioned by the controller. Inspired by the work in [25], the proposed attack detection associated with the deep learning model is executed on the selected nodes that can acquire the network data and leverage the network security. Hence, to minimize energy consumption and maximize communication security, the proposed approach only deploys the attack detection on network-centric nodes rather than executing all the switch nodes for SDN security. Accordingly, the network-centric nodes build the local model based on its local information and send it to the cloud server via the corresponding main controller. In this work, the node selection aims to identify the node that has the ability to protect its connected nodes within the network based on their importance value among all other nodes in the network. By deploying the attack detection mechanism on the selected node, the attack detection algorithm efficiently provides security coverage for the connected nodes in the network, thereby mitigating the security risks even when non-selected nodes are attacked by the intruder in the SDN. The proposed system's significant node selection depends on the number of connected hosts and network traffic within each controller. During the computation of importance value, a higher value indicates that the switch or node 'S' has established their connection with numerous hosts 'H', H ∈ N. The measurement of node importance values is the summation of all the connected host importance values. As formulated in Equations (1)–(3), the proposed approach estimates the importance value by combining each node's direct and indirect importance values. In essence, the Direct IMportance (DIM) value of the 'S' node is based on the total number of connected 'H' nodes within the controller network, whereas the Indirect Importance (IIM) value of the 'S' node is the maximum DIM value of node 'S' among all the Connected Nodes (CN).

$$\mathrm{DIM}(S) = \sum_{\mathrm{i}=1}^{\mathrm{H}} \mathrm{IM_i} \text{ or } |\mathrm{H}| \tag{1}$$

$$\mathrm{IIM}(S) = \mathrm{MAX}_{\mathrm{j}=1}^{\mathrm{CN}} \mathrm{DIM_j} \tag{2}$$

$$\mathrm{IM}(S) = \delta_1 \times \sum_{\mathrm{i}=1}^{\mathrm{H}} \mathrm{DIM_i}(S) + \delta_2 \times \sum_{\mathrm{j}=1}^{\mathrm{CN}} \mathrm{IIM_j}(S) \tag{3}$$

In addition to the spatial connectivity, the proposed approach assigns the importance value based on the traffic flows for each host 'H' as {0,0.5,1} when there is the {Elephant traffic flow, Mice traffic flow, Idle traffic flow}. In Equations (1)–(3), 'i' refers to the directly connected hosts, and 'j' refers to the indirectly connected hosts through host 'i'. During the IIM measurement, the proposed approach computes the maximum DIM score for $\mathrm{j^{th}}$ nodes rather than applying the summation operation for all the connected nodes, resulting in selecting nodes with higher IIM values. As mentioned in Equation (3), the proposed approach combines the weighted combination of direct and indirect importance values to provide significance to both the measures using equally δ_1 and δ_2 constant values as 0.5. In conclusion, the proposed network-centric node selection enables the security system to prevent intruders from launching attacks on high-priority or the most significant node in the controller network by deploying the security mechanism. In the subsequence of network-centric node selection, the proposed approach utilizes the selected node for the local model deployment in the federated environment.

3.2. Hybrid Heuristic-Based Feature Selection

In the SDN, the network traffic data comprise numerous features in different categories, such as the basic, content, and traffic categories-based 41 features in the NSL-KDD dataset. Owing to the vast number of features that describe different aspects of network traffic, superfluous, irrelevant, or noisy features increase the computational cost and overfitting. Moreover, the significance of each feature and the interrelationship between a set of features varies in the network traffic data. Accordingly, with the awareness of the interrelationship among the features, feature selection is imperative [26] for the hybrid learning-based traffic classification. To optimally select the features in the feature-rich network dataset, FedSec initially applies the hybrid heuristic algorithm [27], namely Mayflies-Harmony Search (MA-HS). By modeling the feature selection algorithm as the binary optimization problem, the proposed approach applies the wrapper-based method that selects whether an individual feature in a feature subset contributes to an accurate attack detection or not.

In the FedSec, instead of utilizing the global flight experience of all mayflies (g), the global experience gained only from the mayflies (b) that are highly associated with the corresponding mayfly (f) in achieving higher local fitness function. In this context, mayfly refers to the features in the network traffic dataset, and the fitness function implies the classification accuracy. Instead of selecting the features that highly impact the accurate attack classification, the proposed approach focuses on selecting the features with the global impact of classification accuracy improvement as well as the association among the features in improving the classification accuracy.

$$\mathrm{gbest}(\mathrm{b})_a^f = \begin{cases} \mathrm{x}_{\mathrm{b}}^{\mathrm{t}+1}(\mathrm{a}) \\ \text{if fitness}(\mathrm{a},\mathrm{b})^f > fitness(\mathrm{a},\mathrm{g})^f \end{cases} \tag{4}$$

As modeled in Equation (4), the enhanced MA-HS selects a set of globally experienced mayflies (b) based on the fitness score of any global mayfly with ith mayfly. Suppose the bth mayfly obtains a higher fitness score than the gth mayfly. The enhanced MA-HF selects that bth mayfly as the associative combination for the ath mayfly and updates the position of the ith mayfly to the bth mayfly through dynamic interactions. Accordingly, the feature is retained in a final feature subset based on the influence on all the attack classes; hence, the fitness score is validated across multiple classes.

$$\text{Fitness Function}(f) = \frac{\left|(\rho)_{\mathrm{f_s}}\right|}{\left|\varphi_{\mathrm{f_s}}\right|}, \text{where} \left|\varphi_{\mathrm{f_s}}\right| = |\mathrm{N_f}|/|\mathrm{N_F}| \tag{5}$$

In the enhanced MA-HS, the fitness function consists of accuracy and the number of features with the feature threshold of iterating the validation of the feature subset in the global search space, formulated in Equation (5). To avoid the compromise between the error and the number of selected features, the proposed approach contemplates the accuracy with the direct relationship between the number of selected features and the feature threshold. As formulated, the fitness function needs to be increased with the increase in accuracy ($(\rho)_{\mathrm{f_s}}$) as well as the decrease in the number of features (fs); in particular, the ratio ($\left|\varphi_{\mathrm{f_s}}\right|$) between the number of reduced features and the total number of features. The proposed approach initially aims to maintain 50% of network features to represent patterns on various attack classes. Hence, it selects the features to obtain better accuracy if the feature threshold

($|\varphi_{f_s}|$) is above 0.5; otherwise, the proposed approach selects an optimal set of features until reaching an improved accuracy over the accuracy on the entire feature set or feature set in its previous iteration. In this context, the assignment of the feature threshold is based on the mid-point of the fitness score that ranges from '0' to '1'; in the enhanced MA-HS, the accuracy of the classification model taken as the ratio, that is, $(\rho)_{f_s} \in [0, 1]$ and thus, the score of fitness function $\in [0, 1]$. In the MA-HS algorithm, harmony search selects a superior feature subset from the different solutions in the feature space. Further, the selected feature set is input to enrich the SDN-specific dataset.

3.3. SDN-Specific Dataset Enrichment

The proposed approach intends to enrich the benchmark NSL-KDD dataset in the context of SDN characteristics to assess the quality of attack detection in the federated environment. The study of work in [11] found that testing the context of flow-based attack detection from the packet-based generic network dataset, such as the NSL-KDD [28], is inappropriate. For the SDN environment, the Time-To-Live (TTL) and flow duration are essential to identify attacks, which is lacking in the benchmark NSL-KDD dataset. Hence, it is imperative to incorporate SDN-specific features into publicly available datasets to ensure accurate and consistent attack detection because the generic network dataset lacks features to reflect real-world network traffic in the SDN accurately. Accordingly, the proposed approach incorporates the dataset enrichment process as one of the objectives of improving intrusion detection accuracy. It aims to append prospective attributes that influence the attack detection ability in the benchmark NSL-KDD dataset from the influence of available attributes. The NSL-KDD dataset [28] has been increasingly examined by attack and intrusion detection to develop and assess security solutions. It comprises the network traffic for the attacks, such as DoS, Probe, U2R, and R2L. In essence, the number of additional packets or traffic flows generated for the attacks with the additional SDN-specific features of the Node ID, Flow duration, TTL, and Trust Value for all the instances are presented in the paper published in the IEEE International Conference on Consumer Electronics [29]. In addition to the DoS, Probe, U2R, and R2L attack categories in the NSL-KDD dataset, the FedSec mechanism appends one additional attack category as other categories with the attacks of time jack and peer flooding attack.

In the enriched dataset, Node ID refers to the identity of the connection for a specific service between the hosts and switches in the controller, indicating the IP address and Port identity. The flow duration and TTL values are updated for the normal and malicious traffic from the study of several existing SDN systems. Moreover, trust value implies the ability of traffic flow fluctuations for each node in its historical activities based on the immediate trust measurement. Accordingly, the trust value measurement is vital for detecting the attacks in the SDN data plane based on evaluating flow rules, as presented in Equations (6) and (7).

$$\mathrm{IT_{LFR}} = \sum_{k=1}^{|\mathrm{LFR}|} \Phi_k \Big/ |\mathrm{LFR}| \tag{6}$$

$$\mathrm{IT_{LFR}} = \sum_{k=1}^{|\mathrm{LFR}|} \Phi_k \Big/ |\mathrm{LFR}| \tag{7}$$

To enrich the NSL-KDD dataset with the trust value, FedSec designs the computation of the immediate trust value for the SDN nodes based on the flow rules to improve the classification accuracy in the dynamic network environment. The estimation of Immediate Trust (IT) measurement determines the average number of matching fields of all Legitimate Flow Rules (LFR) and Malicious Flow Rules (MFR) to the Suspected Flow (SF). If SF matches with many fields in legitimate flow entries, it is categorized as legitimate, compared to malicious flows. In Equations (6) and (7), Φ_k denotes the number of matched fields between the suspected flow, LFR, and MFR. Thus, the FedSec enriches the NSL-KDD dataset with additional features and samples. The enriched dataset is provided to the GCNN-GRU model for attack classification, wherein the graph-based features are computed and classification is performed.

3.4. Model Training and Attack Classification

The FedSec incorporates GCNN-GRU to construct the model parameters and classify the various attacks in the MC-SDN infrastructure or data plane layer. It is further divided into two parts that are GCNN-GRU-based model training, and model retraining and attack classification. The FedSec performs model training and attack classification using the enriched NSL-KDD dataset.

In the experiment, hyperparameters play a paramount role in the decision-making to build and train the GCNN-GRU model. To provide the generalization ability of the model, the parameters of the GCNN-GRU model are given in Table 2. For the implementation of the GCNN model, this work employs the GeLU and Tanh activation functions that enable the learning of complex and nonlinear input patterns. Moreover, the Adam optimizer in GCNN facilitates bias correction during the training of spatial dependencies. The application of adadelta in GRU aids in capturing the temporal dependencies over the timesteps. Moreover, the ReLU activation function in the GRU leverages the mitigation of the vanishing gradient problem in the recurrent networks.

Table 2. Model Parameters.

Learning Parameters	**Models**	
	GCNN	**GRU**
Dropout Rate	0.2	0.4
Hidden Units	32	64
Learning Rate	0.01	0.001
Activation	GeLU, Tanh	ReLU
Optimizer	Adam	adadelta
Loss Function	Categorical Cross-Entropy	
Epochs	100	
Batch Size	64	

3.4.1. GCNN-GRU-Based Model Training

In the FedSec, the GCNN-GRU relies on the features of an enriched NSL-KDD dataset for model training. The enriched NSL-KDD comprises an intelligent extraction of spatial and temporal features in the large-scale and dynamic SDN network traffic. The enriched NSL-KDD input features and local model construction of FedSec are depicted in Figure 2.

The FedSec implements the GCNN model at the network-centric nodes of various domains managed by controllers. Each node has its local dataset. They initially train their learning models using the local dataset information and construct a local model. The MC-SDN environment is modeled as a graph G = <V, E> in which 'V' denotes the records and 'E' denotes the type of flows that are either legitimate or malicious based on the nodes' interactions. To analyze the data flow between the nodes, the source and destination IP address and ports are the potential indicators for transmitting packets or flow traffic. To provide the input to the GCNN model, the feature matrix and adjacency matrix are created from the 'T' number of traffic samples or records and the 'F' number of feature dimensions. The construction of the input or feature matrix comprises the standardized network traffic values with the dimension of XT × F that is referring the standardized format of the input-enriched dataset with the impact of selected features by the hybrid heuristic algorithm. With the target of establishing the edge relationship between the network traffic flows, the proposed approach considers the source IP address, destination IP address, source port, and destination port as the key elements. For instance, in the NSL-KDD dataset, 'same_srv_rate', 'diff_srv_rate', and 'srv_diff_host_rate' features are utilized as the key indicators to create the edge relationships between the traffic flows, indicating the port and IP address in the aspect of service (srv) and hosts. As per the hypothetical rules, the proposed approach constructs the graph with the connection establishment between two traffic flows, TF1 and TF2.

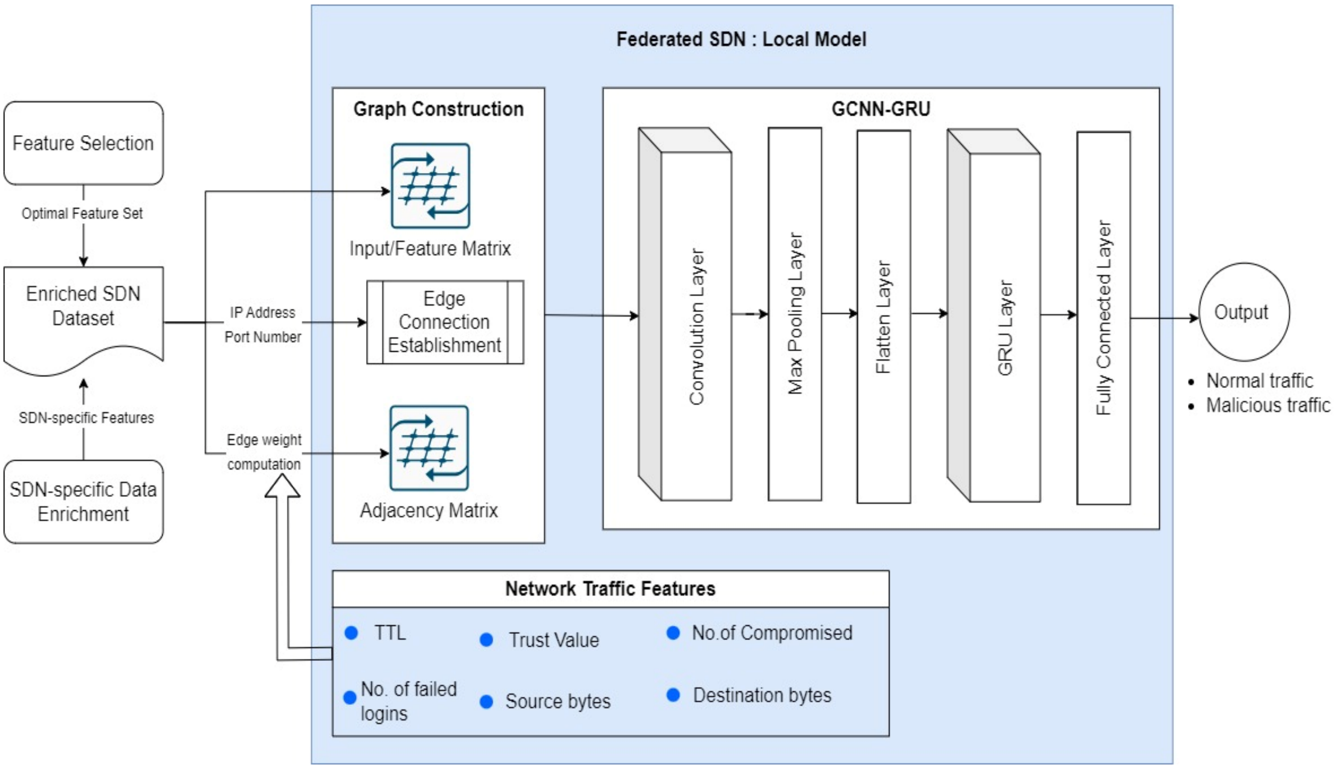

Figure 2. Federated Local Model in FedSec.

$$1 \rightarrow \text{TF1[Source IP]} = \text{TF2[Source IP] and TF1[Source Port]} = \text{TF2[Source Port]}$$

$$2 \rightarrow \text{TF1[Source IP]} = \text{TF2[Dest IP] and TF1[Source Port]} = \text{TF2[Dest Port]}$$

$$3 \rightarrow \text{TF1[Dest IP]} = \text{TF2[Source IP] and TF1[Dest Port]} = \text{TF2[Source Port]}$$

$$4 \rightarrow \text{TF1[Dest IP]} = \text{TF2[Dest IP] and TF1[Dest Port]} = \text{TF2[Dest Port]}$$

Similarly, the connection between the traffic flows is built for the graph model in which the connection is established if any one of the rules is true. In the subsequence of feature matrix construction and connection establishment, the adjacency matrix is constructed based on the edge weight computations. Instead of building the adjacency matrix from the connection establishment, the proposed approach computes the edge weight based on the attack behavior analysis. For any traffic flows in the graph connection, if the minimum three criteria are satisfied, the proposed approach assigns the edge weight as '1'; otherwise, '0' in the adjacency matrix in which '1' refers to the malicious entry. The proposed approach computes the edge weight based on the factors of 'number of compromised', 'number of failed logins', 'logged in', loss computation from 'source bytes' and 'destination bytes', and false injection computation from 'source bytes' and 'destination bytes', 'TTL', and 'Trust Value'. Equation (8) computes the edge weight to emphasize legitimate and malicious behaviors.

$$(\mathrm{EW})_{1,2} = \begin{cases} 1, \text{ if } \mathrm{num}_{\mathrm{compromised}} \geq 1 \\ 1, \text{ if } \mathrm{num}_{\mathrm{failed}_{\mathrm{logins}}} \geq 3 \\ 1, \text{ if } \mathrm{logged}_{\mathrm{in}} == 0 \\ 1, \text{ if } \mathrm{src}_{\mathrm{bytes}} > \mathrm{dst}_{\mathrm{bytes}} \\ 1, \text{ if } \mathrm{src}_{\mathrm{bytes}} < \mathrm{dst}_{\mathrm{bytes}} \\ 1, \quad \text{ if TTL} > 80 \\ 1, \quad \text{if Trust Value} < 0.5 \end{cases} \tag{8}$$

In the SDN, the attack detection heavily relies on the trust of the nodes in the transmission path. Hence, the adjacency matrix computation primarily focuses on the trust-based approach during packet transmission. In Equation (8), '1' indicates malicious activity, whereas '0' indicates normal behavior. To build the adjacency matrix, computing the trust relationship between the pair of nodes is vital for the implementation of the graph model. Hence, to highlight the malicious behavior in the network traffic patterns, the proposed approach assigns the edge weight as '1' for the adjacency matrix when there are malicious risks in the various factors of trust-based assessment.

For the trust-based approach, the examination of compromised nodes indicates that there is the existence of malicious activity, particularly the DDoS attacks. Hence, this work explores the 'number of compromised nodes' as one of the significant factors for trust-based attack detection.

In Equation (8), if the number of compromised nodes is greater than 1, there is the possibility of attack. Within the network, too many failed logins, and also transmitting or accessing packets without a logged-in state, emphasize the occurrence of the malicious activity. Moreover, the number of source and destination bytes are to be equal in the scenario of legitimate traffic; otherwise, there is the occurrence of either 'packet drop' or 'packet flood' attacks. Hence, if there are increased or decreased quantities between the source and destination bytes, the proposed approach assigns the edge weight as '1', which indicates the malicious behavior in terms of 'packet drop' or 'packet flood' attacks associated with any one of the connected nodes in the graph network. Similarly, TTL and trust values are formulated for the edge weight modeling based on the standard value range of the node with attack behaviors.

The proposed intrusion detection aims to diminish the false positives by fulfilling at least three criteria of the seven, resulting in the balancing of sensitivity, that is, actual attack detection and specificity in false alarm mitigation. Moreover, the selection of only three criteria instead of seven diversified criteria leverages the detection of known patterns without failing to detect the unknown anomaly patterns. Accordingly, the TTL criteria modeling in this work is referred from several previous research works to fix TTL values for normal and attack categories. For instance, in the work in [30], the TTL value is described for normal and attack behaviors with the unit as number of hops passed by the packets before discard, irrespective of time during the IP packet transmission in the SDN rather than the Domain Name System (DNS), which indicates the TTL value in seconds. Among multiple factors, the proposed approach concludes that a minimum of three criteria provide nearly half of the importance of the seven factors. The attack detection emphasizes the attack's existence when any one of the factors is violated. Hence, to optimally balance and strengthen the tracing of attack behavior, this work formulates the attack behavior with the satisfaction of three minimum factors.

In particular, the proposed system builds the local model through the learning parameters of the GCNN-GRU model. By providing the constructed feature matrix and adjacency matrix of the input network traffic dataset as the input, the GCNN model precisely learns the spatial relationship in the SDN that facilitates the accurate recognition of attack behaviors with the aid of the GRU model. The proposed GCNN-GRU model comprises the graph convolution layers, max pooling layer, flatten layer, and GRU layer. Initially, the graph convolution layers are responsible for capturing the neighborhood information from the nodes in terms of the spatial relationships in the graph network. In subsequence, the max pooling layer compresses the feature representation with highly informative features, preventing overfitting as well as reducing the dimensionality. Also, the pooled feature representation is provided as the input to flatten layer to transform the 2D output of the GCNN model into 1D input for the GRU model. Finally, the GRU layer learns the spatially aggregated features in sequential patterns to capture the temporal relationships using the gating mechanism, resulting in normal and malicious traffic with the integration of a fully connected layer. Here, n number of local models are generated and updated to the global server through main controllers, facilitating the global model generation. Algorithm 1 explains the local model generation of FedSec using GCNN-GRU.

Algorithm 1 Local Modeling using GCNN-GRU

```
Input: Local datasets of different network-centric nodes
Output: n number of LM generation for FL aggregation
for each client do
    Construct local dataset in its area
    Create a graph with Feature matrix and adjacent matrix
    for all the pairs of nodes do
        Compute edge weight using Equation (8)
        if EW = 1 minimum 3 criteria for a pair of nodes then
            Assign weight as 1 in adjacency matrix
        else
            Assign weight as 0 in adjacency matrix
        end if
    end for
    Initialize the GCNN-GRU training process as the local model for each client
    for each GCNN-GRU do
        Implement convolution, max pooling, flatten layer, and GRU layer
        Learn the spatial and temporal relationships from enriched NSL-KDD dataset
    end for
end for
for all the clients do
    Feed the 'n' number of local models' parameters to global server for aggregation
end for
```

3.4.2. Model Retraining and Attack Classification

Each network-centric node starts to update its local model to the cloud server for aggregated global model generation. Consequently, the global server generates a global model by aggregating the n number of local models using the FedAvg algorithm. The FedSec exploits the following Equation (9) to aggregate the local models at the global server and generate the global model.

$$\mathrm{GM} = \sum_{i=1}^{n}\sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{W}_i * \mathrm{LM}_j \tag{9}$$

In Equation (9), the terms GM and LM represent the global model generated at the global server and the local model updated by a network-centric node. The term W is the weighting factor of a local model. The terms I and j are varied from 1 to n. Here, n denotes the total number of LMs the global server receives. The proposed model combines the local models of different edges using this equation. After generating the GM, the global server reversely updates it via the control plane to the network-centric nodes. After local model construction and update, the network-centric nodes retrain their learning models with globally shared parameters. By employing the globally shared knowledge, the proposed FedSec customizes the initial local learning parameters of GCNN-GRU and improves the attack detection efficiency. The global model consists of various novel attack patterns of various SDN domains. Thus, it effectively enhances the learning knowledge of GCNN-GRU without impacting the MC-SDN network performance. Moreover, the FedSec enhances the local learning accuracy of the hybrid learning model by incorporating the distributed federated learning strategy. Finally, the FedSec initiates the attack detection and classification process. During the attack detection phase, the network-centric nodes can determine various attacks and classify them under different classes like DoS, Probe, U2R, R2L, time jack, and peer flooding according to the local learning knowledge of GCNN-GRU during the testing phase. The following Algorithm 2 explains the model retraining and attack classification of FedSec.

Algorithm 2 GM Generation and Attack Classification

```
input: Local models of different network-centric nodes
Output: GM generation and attack classification
for each network-centric node do
        Feed the output of GCNN-GRU as an LM input to the global server
        Global server do:
            Aggregate the LMs of high-context edges using Equation (9)
            Generate the GM with global attack patterns
            Feed the GMs to the network-centric node
      Retrain the GCNN-GRU using the GM update
   Improve the learning accuracy with wider attack knowledge
   Initiate the attack detection and classification
  Compare the testing data with learned data
  Categorize the attacks under different categories
end for
```

4. Experimental Evaluation

To implement the FedSec algorithm, the experimental model conducts the experiments using Python. The experimental model conducts the experiments of the proposed and existing comparative models on the NSL-KDD dataset [28]. This work assumed the enriched NSL-KDD dataset as the SDN flow traffic dataset to assess the performance of attack detection in the SDN data plane [29]. In contrast to implementing the network traffic flows obtained from the multi-controller SDN environment, the proposed attack detection algorithm is evaluated for three different subsets of traffic flows as three clients' data in the enriched NSL-KDD dataset. The publicly available multi-controller SDN network dataset is limited, which is not a constraint to assessing the traffic flow security. Hence, the experiments are performed for the core of the NSL-KDD dataset. The experimental work employs Python libraries with Python to test the deep learning model-based attack classification. To evaluate the FedSec performance in the classification problem, the experimental model utilizes the precision, detection rate or recall, specificity, and accuracy metrics computed from the confusion matrix. The experimental framework implements four existing SDN security research works, including the hybrid deep learning-based [19], GCNN-based [21], and federated learning-based [24] SDN security solutions to assess the performance of the proposed FedSec model comparatively.

4.1. Study of Experimental Dataset

The experimental model has tested the proposed and existing attack classification algorithms on the enriched NSL-KDD and NSL-KDD datasets, respectively. The enriched NSL-KDD dataset assumes that a set of nodes are selected by the DIM and IIM score, and the immediate trust measurement computes the trust value to reflect the outcome of the proposed SDN model implementation on the test dataset for further deep learning model implementation. The target result of the deep learning model indicates that the system enforces the dropping of malicious traffic in the MC-SDN environment. The number of data samples in the test datasets is shown in the paper published in the IEEE International Conference on Consumer Electronics [29].

4.2. Data Visualization

The deep learning model requires several data analysis and data preprocessing operations to build a secure intrusion detection model for the SDN data plane. Figure 3 illustrates the data distribution of the TTL feature for different attacks in the enriched NSL-KDD dataset.

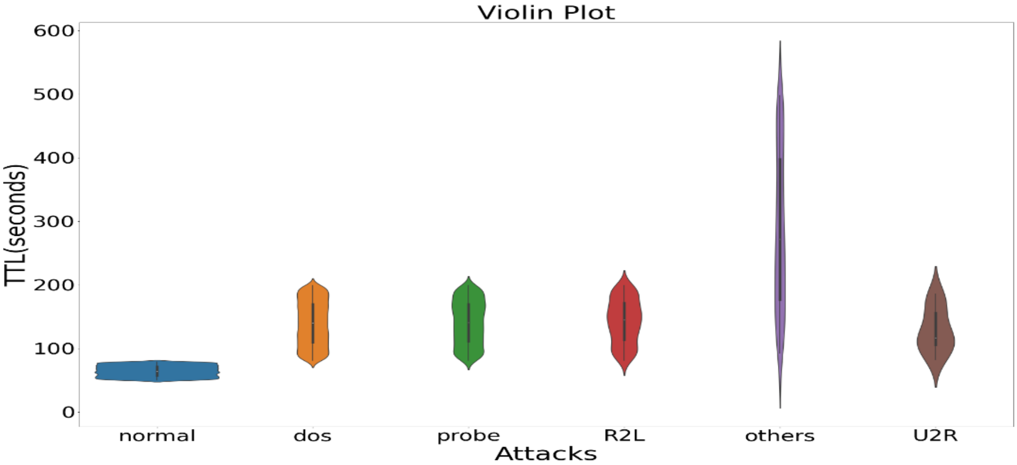

Figure 3. TTL Distribution over SDN attacks.

From the analysis of Figure 3, it is recognized that the TTL values are comparatively low for the normal class that is legitimate traffic flow compared to the various malicious network traffic in the SDN. TTL criterial modeling is based on the work in [30]. The TTL value is described for normal and attack behaviors with the unit as a number of hops passed by the packets before discard, irrespective of time during the IP packet transmission in the SDN rather than the Domain Name System (DNS), which indicates the TTL value in seconds. In the SDN environment, the default value of TTL is set to 64, and it is near integer values for normal traffic. Hence, the range from 50 to 80 is assigned for the normal traffic in this work. TTL is based on the number of hops and routers, which indicates the count in its traversing path. According to Figure 3, the malicious traffic, including the SDN attacks, exists above 80, indicating that the packets traverse through a higher number of hops or routers before being discarded. As a result, in contrast to the default TTL range, traversing through numerous hops or routers implies the occurrence of malicious activity.

In the enriched NSL-KDD dataset, the correlation between the features of detecting legitimate and malicious traffic flows in the SDN is illustrated in Figure 4. The need for feature selection is proved in Figure 4 due to the existence of an uncorrelated feature set in the NSL-KDD dataset, which misleads the attack detection. Moreover, to illustrate the significance of each set of correlated features on the discrimination of normal and malicious traffic, Figure 4 plots the correlative relationships among the features, proved through the proposed feature selection mechanism.

To understand the data distribution, the visualization of high-dimensional data provides the guidelines for initiating the required data handling operations. Figure 5 shows that the attacks in the enriched NSL-KDD dataset are not spatially and temporally separated. In conclusion, traditional machine learning algorithms are unsuitable for dealing with this data.

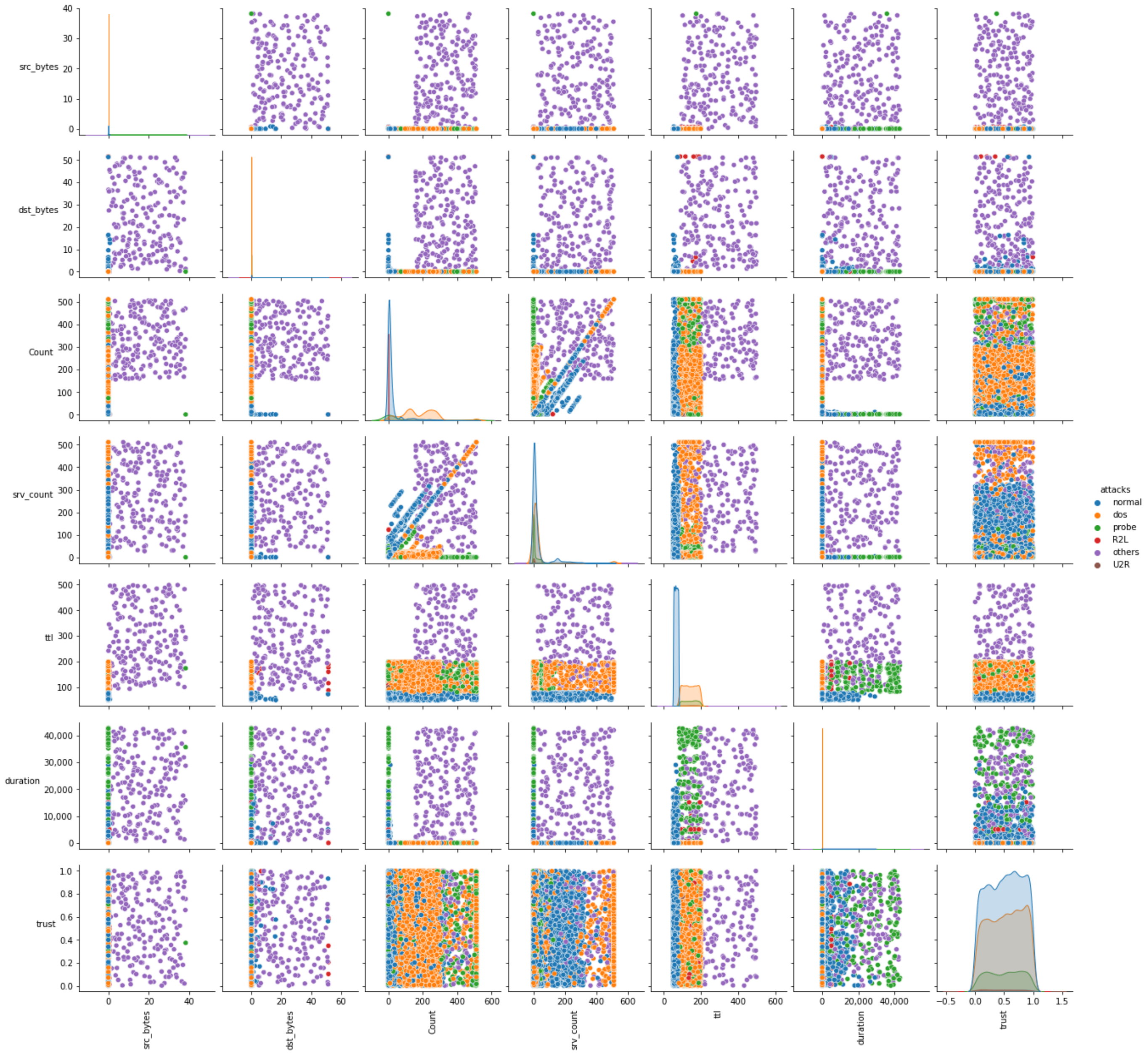

Figure 4. Feature Correlation in Enriched NSL-KDD Dataset.

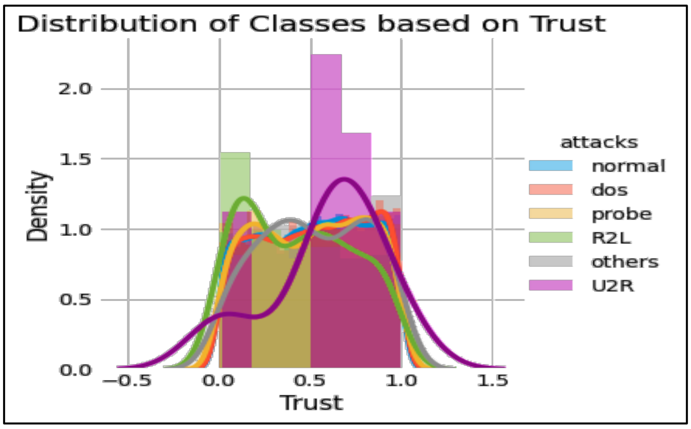

Figure 5. *Cont.*

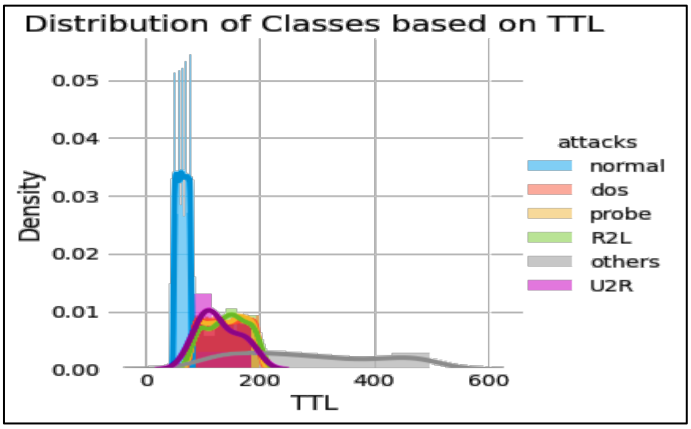

Figure 5. Density of Trust and TTL Features.

5. Experimental Results

5.1. Evaluation of Feature Selection

To detect malicious traffic flow in the SDN data plane precisely, feature selection becomes a prerequisite in deep learning-based traffic classification for a huge set of features in the NSL-KDD dataset.

Figure 6 depicts the feature importance analysis in detecting attack classes in the enriched NSL-KDD dataset before the feature selection. As shown in Figure 6, the statistical analysis reveals the feature importance in the dataset; however, to further efficiently and optimally select a set of salient features, the FedSec attempts to select the features using the enhanced MA-HS algorithm optimally. From the illustration of Figure 6, the effectiveness of the proposed feature selection algorithm with the standard assessment score is evaluated.

Table 3 compares the performance variations of the proposed FedSec with enhanced MA-HS algorithm-based feature selection and without feature selection. Among the performance metrics of accuracy, specificity, precision, and recall, the proposed feature selection plays a significant role, depicted as the improvements in all the metrics against the proposed FedSec without feature selection. Table 4 illustrates the influence of various Reduced Features (RF) in the enriched NSL-KDD dataset for different percentages of training data samples as 40%, 60%, 70%, and 80%.

Table 3. Comparative Analysis of Feature Selection Performance.

Metrics	FedSec with FS	FedSec without FS
Accuracy (%)	97.78	88.05
Specificity (%)	96.53	91.39
Precision (%)	98.24	89.44
DR or Recall (%)	98.01	89.07

Table 4. Impact of Reduced Features on GCNN-GRU Classification.

Training Size	Detection Rate (%)			
	RF = 8	RF = 20	RF = 30	RF = 41
40%	86.01	96.21	95.12	88.14
60%	85.14	96.53	97.31	90.02
70%	83.12	96.74	98.01	93.24
80%	88.07	95.02	98.12	91.31

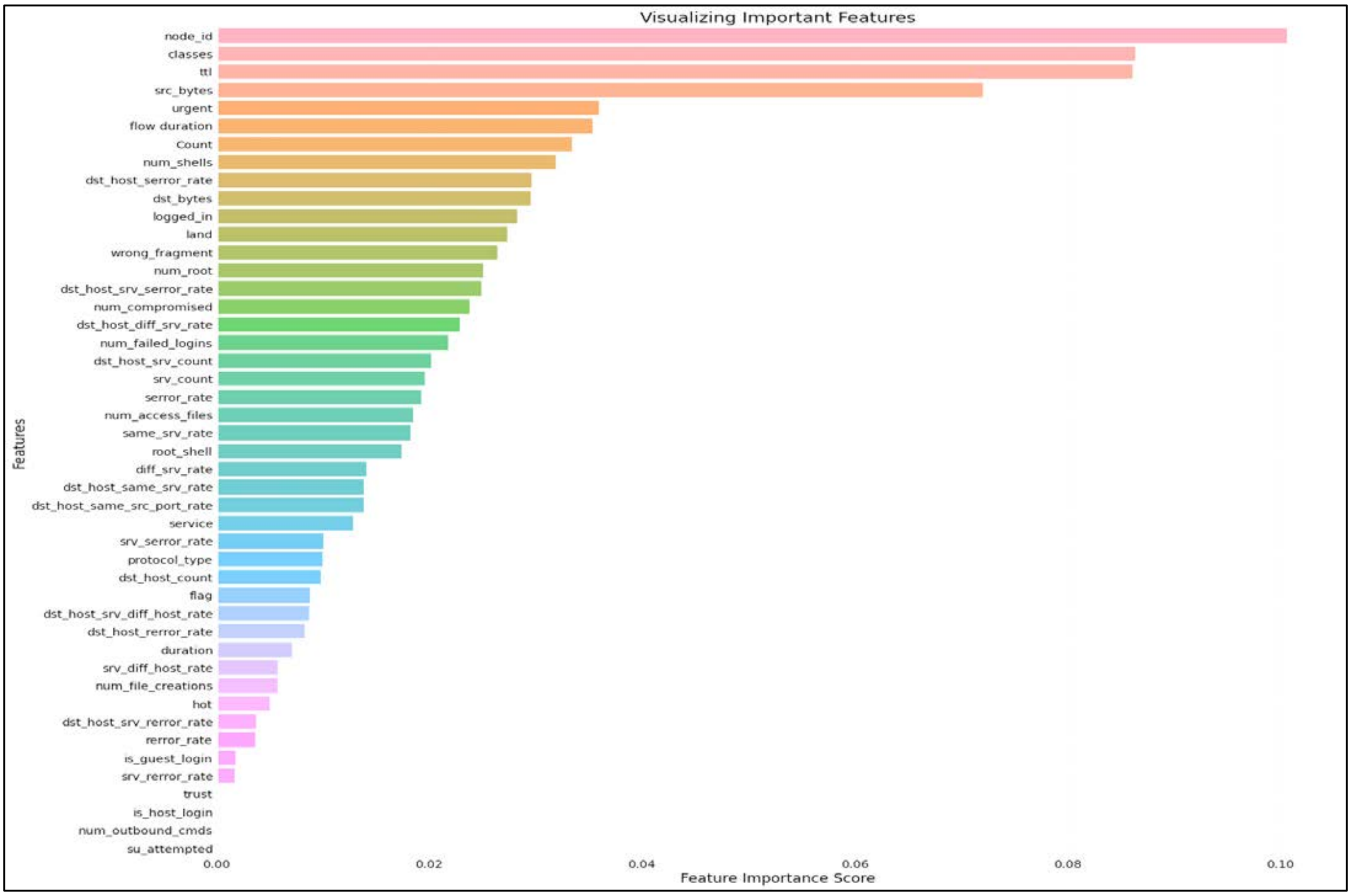

Figure 6. Feature Importance in Enriched NSL-KDD Dataset.

The analysis of Table 4 shows that the optimal number of features only provides a better detection rate than a large number of entire features and a minimal set of features. This is because reducing the number of features from the NSL-KDD dataset leads to information losses, and retaining many features misleads the detection accuracy and increases the training time.

5.2. Evaluation of Traffic Classification

The comparative works [19,21] have employed the hybrid deep learning models to investigate the network traffic flows from the spatial and temporal dimensions. However, the lack of examining the spatial node connectivity and its packet transmission instead of feature dependencies leverages the inaccurate discrimination of the normal and malicious traffic. Even though the outcome of the feature selection algorithm leverages the learning of abstractive features by the hybrid model [19], the deployment of a security mechanism on the attacker node or low-coverage node further degrades the intrusion detection performance. Also, the comparative work [21] fails to cope with the diversified malicious attack behaviors when applying multiple centralized security mechanisms on distributed networks due to the lack of collaborative training knowledge. In addition, the existing research [24] applies weighted federated learning to collaboratively examine the heterogeneous network traffic patterns to detect malicious network traffic in SDN accurately. Although research studies [21,24] have targeted detecting the DDoS attacks in the SDN, the suggested security solution is inappropriate for emerging SDN data plane attacks. Also, the graph-based security solution fails to build the feature matrix and adjacency matrix with the awareness of the malicious representative behaviors, misguiding the spatial feature analysis and inaccurate attack detection. To overcome these shortcomings in the comparative works, the proposed FedSec selects the representative features with minimal computation time and constructs the graph from the potentially reduced and enriched SDN-specific features with collaborative learning among the multiple controllers in the federated SDN. In the experiment, hyperparameters play a paramount role in the decision-making to build and train the GCNN-GRU model.

In addition to the enriched NSL-KDD dataset, the traffic classification and SDN security performance are tested on two other datasets, the UNSW_NB15 dataset [30] and the InSDN dataset [31].

- UNSW_NB15: The generation of raw network packets in the UNSW-NB 15 dataset is based on the combination of real modern normal activities and synthetic contemporary attack behaviors, developed by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). This dataset includes the nine attack types: fuzzers, analysis, backdoors, denial-of-service, exploits, generic, reconnaissance, shellcode, and worms. Twelve algorithms are built, and 49 features with the class label are generated using the Argus and Bro-IDS tools.
- InSDN Dataset: The InSDN dataset contains threats specific to SDNs and SDN-specific assaults on different types of standard traffic. The attacks include DoS, DDoS, web, botnet, probing, exploiting, and brute force. Furthermore, various external and internal attack routes are incorporated into the InSDN dataset to replicate real-time attack situations.

To examine the performance improvement of the GCNN-GRU in the FedSec on each performance metric, the results are briefly presented as follows.

Table 5 evaluates the proposed attack detection algorithm with three existing SDN security solutions [19,21,24] for three different datasets. The proposed FedSec approach accomplishes a Detection Rate (DR) of 98.01%, and the DR of the works [21,24] are 95.27% and 96.67%, respectively, while testing on the enriched NSL-KDD dataset. It is because, even though the federated learning model collaboratively learns the diversified attack knowledge from the multiple domains, the lack of investigating the intrusion-aware spatial correlations in the traffic flows degrades the attack detection performance [24] compared to the proposed approach. In addition, learning the small perturbations in the network traffic patterns by fine-tuning the local model based on the global update ensures the adaptability and robustness of the proposed intrusion detection without retraining on the entire data, even when there are diversified network traffic patterns. Moreover, the proposed security scheme learns the different attack patterns with the enhanced hybrid deep learning model's assistance in weight dispersion regularization in GCNN during feature extraction and adaptive ownership weight in GRU during traffic classification. Moreover, the FedSec enriches the NSL-KDD dataset with the additional impact of the SDN-focused attributes of trust value, TTL, selected node ID, flow duration, and malicious patterns possible in SDN, which tends to improve the accurate classification of multiple attacks in the enriched NSL-KDD.

Table 5. Comparative Traffic Classification Performance.

SDN Security Models	Enriched NSL-KDD		UNSW_NB15		InSDN	
	DR (%)	Accuracy (%)	DR (%)	Accuracy (%)	DR (%)	Accuracy (%)
Hybrid DL [19]	96.39	95.11	89.36	83.19	95.18	93.07
Graph-based [21]	95.27	92.13	89.15	87.22	95.02	92.49
Federated-based [24]	97.67	96.51	87.36	84.13	96.14	95.01
Proposed	98.01	97.78	94.31	92.15	97.56	96.12

The proposed FedSec yields higher accuracy in detecting the malicious and normal traffic from the network traffic samples, even when testing on the UNSW_NB15 and InSDN datasets. The adjacency matrix construction varies for each dataset based on the malicious behavior analysis. In particular, highly influencing features in the dataset are utilized for the rule generation during edge weight calculation for the adjacency matrix. In particular, the intricate relationships and interactions within the network are captured by the graph-based representation in the proposed work, which enables the intrusion detection model to learn the malicious behavior from the network traffic patterns. Moreover, the SDN-specific data enrichment phase does not apply to implementing UNSW_NB15 and InSDN datasets. However, the potential advantages of federated learning, hybrid heuristic feature selection, and the GCNN-GRU model leverage the security system towards improved performance compared to the existing security solutions. Thus, the comparison among the different datasets proved the ability of proposed intrusion detection on the diversified attack behaviors and network traffic patterns by the potential modeling of the hybrid heuristic algorithm-based feature selection and graph construction.

In Table 6, the proposed SDN security model outperforms the existing federated work [24] and the proposed approach without implementing federated concepts. The detection rate of the proposed approach has a 5.7% performance improvement compared to the centralized proposed approach (without FL), which executes the GCNN-GRU model in a centralized manner for each local network traffic dataset. It is because the collaboratively trained global model in the FL optimally fine-tunes the local model with the global parameters after the federated aggregation, which enforces the accurate discrimination of normal and malicious traffic even when there are new types of attacks in the SDN. Moreover, the federated model in the proposed intrusion detection is associated with the GCNN-GRU model, facilitating the adaptability in the attack detection regarding the spatial and temporal dynamics in the network traffic, and tends to improve detection rate compared to the federated model. Also, the decentralized model updates in the federated settings allow intrusion detection by collective intelligence regardless of centralized data sharing. However, federated learning guarantees an improved detection rate and accuracy even when there is an increased number of clients in a large-scale network. The objective of the FedSec leverages the IDS towards the accomplishment of security in the MC-SDN with the help of the federated model along with the GCNN-GRU. Accordingly, the effectiveness of attack detection heavily relies on the enrichment of the traditional deep learning model and the learning of diversified attack patterns. The FedSec accurately recognizes the attacks launched in the SDN, such as the time jack and peer flooding attacks in the enriched dataset. Even though the DoS and probe attacks only have comparatively high attack patterns, the FedSec distinguishes the R2L and U2R attack types from the normal, DoS, probe, and other traffic samples.

Table 6. Comparative Federated Learning Performance in Traffic Classification.

SDN Security Models	Enriched NSL-KDD	
	DR (%)	**Accuracy (%)**
Federated-based [24]	97.67	96.51
Proposed (without FL)	92.31	90.05
Proposed	98.01	97.78

5.3. Evaluation of SDN Security

To assess the traffic classification performance in the SDN environment, the experimental model measures several metrics, including the False Alarm Rate, error rate, and latency.

$$\text{False Alarm Rate (FAR)} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{10}$$

$$\text{Error Rate} = \frac{\text{FP} + \text{FN}}{\text{P} + \text{N}} \tag{11}$$

FP and FN refer to the number of incorrectly detected malicious and normal traffic samples, respectively. Moreover, TN denotes the number of correctly detected normal traffic, whereas 'P' and 'N' correspondingly imply the total number of malicious traffic and normal traffic samples. Also, latency measures the time the controller takes to process a single packet in the SDN, which is influenced by the efficient flow traffic verification in the data plane. In particular, data plane latency is based on the packet processing time and forward decision of packets in the SDN. The design of FedSec with feature selection does not affect the packet processing time during the inspection of the network traffic flows. The hybrid heuristic-based feature selection and federated settings leverage the optimal execution of the traffic classification and reject the malicious entry updating in the flow table without increasing the overhead.

In FedSec, the federated modeling of MC-SDN facilitates the accurate distinguishing among the multiple attack classes in the enriched NSL-KDD dataset, leveraging the dropping of malicious entries during the flow table updating by the collective intelligence from the distributed and diversified malicious traffic patterns. By training the GCNN-GRU model with the CNN-assisted extracted feature values under the categories of the DoS, Probe, R2L, U2R, and other attacks and normal samples on the TCP, UDP, and ICMP protocols with the collaborative attack knowledge, the FAR and error rate are comparatively minimal compared to the existing SDN security research while testing on three different network traffic datasets. It is accomplished by the balanced criteria-based edge weight computation for the adjacency matrix modeling in the proposed FedSec, precisely discriminating the normal and malicious patterns even when there are inherent variations in the attack features. As a result, as mentioned in Table 7, the FedSec accurately rejects the malicious traffic flows without compromising the latency in the large-scale network.

Table 7. Comparative SDN Security Models on Three Different Datasets.

SDN Security Models	Enriched NSL-KDD			UNSW_NB15			InSDN		
	FAR (%)	Error Rate (%)	Latency (ms)	FAR (%)	Error Rate (%)	Latency (ms)	FAR (%)	Error Rate (%)	Latency (ms)
Hybrid DL [19]	2.04	2.81	27	8.87	14.01	25	4.44	5.03	31
Graph-based [21]	1.99	2.73	32	9.04	11.73	34	4.78	5.77	40
Federated-based [24]	2.05	2.67	28	12.06	17.31	26	3.07	4.17	34
Proposed	1.75	1.98	25	5.34	7.32	24	2.54	3.12	28

6. Conclusions

In an SDN-based environment, the data plane is often confronted with severe threats due to the increased traffic flow from the attackers. For the secure SDN model, it is essential to identify various attacks across the massive data flows with the awareness of the behavior of the SDN nodes in the network structure. This paper proposed a FedSec security model to provide security against multiple MC-SDN attacks. The FedSec mechanism secures the MC-SDN nodes against malicious data flow entries. In the FedSec, the hybrid heuristic algorithm MA-HS optimally reduced the computational complexity and improved the intrusion detection accuracy by utilizing representative features. Moreover, FedSec has customized the NSL-KDD dataset with additional features and attacks. Moreover, the proposed FedSec has enhanced the distinguishing ability of normal and malicious traffic in SDN by potential features-based graph construction, particularly adjacency matrix for the GCNN. Thus, the proposed approach detected malicious traffic during the flow table updating in the data plane by the spatial–temporal correlation combination of the GCNN-GRU model. Finally, the evaluation results have reached a 98.01% recall or detection rate for the proposed FedSec while testing the enriched NSL-KDD dataset.

Author Contributions: Conceptualization, A.A.; Methodology, A.A.; Software, A.A.; Validation, A.A.; Resources, A.A.; Writing—original draft, A.A.; Supervision, I.K. and Supervision, S.M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Kreutz, D.; Ramos, F.M.V.; Veríssimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-Defined Networking: A Comprehensive Survey. *Proc. IEEE* **2015**, *103*, 14–76. [CrossRef]
2. Mohammed, A.H.; Khaleefah, R.M.; Abdulateef, I.A. A review software defined networking for internet of things. In Proceedings of the 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 26–28 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.
3. Modieginyane, K.M.; Letswamotse, B.B.; Malekian, R.; Abu-Mahfouz, A.M. Software defined wireless sensor networks application opportunities for efficient network management: A survey. *Comput. Electr. Eng.* **2018**, *66*, 274–287. [CrossRef]
4. Hu, T.; Guo, Z.; Yi, P.; Baker, T.; Lan, J. Multi-controller based software-defined networking: A survey. *IEEE Access* **2018**, *6*, 15980–15996. [CrossRef]
5. Zhang, Y.; Cui, L.; Wang, W.; Zhang, Y. A survey on software defined networking with multiple controllers. *J. Netw. Comput. Appl.* **2018**, *103*, 101–118. [CrossRef]
6. Haas, Z.J.; Culver, T.L.; Sarac, K. Vulnerability Challenges of Software Defined Networking. *IEEE Commun. Mag.* **2021**, *59*, 88–93. [CrossRef]
7. Ahmad, Z.; Shahid Khan, A.; Wai Shiang, C.; Abdullah, J.; Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4150. [CrossRef]
8. Aslam, N.; Srivastava, S.; Gore, M.M. A comprehensive analysis of machine learning-and deep learning-based solutions for DDoS attack detection in SDN. *Arab. J. Sci. Eng.* **2023**, *49*, 3533–3573. [CrossRef]
9. Taheri, R.; Ahmed, H.; Arslan, E. Deep learning for the security of software-defined networks: A review. *Clust. Comput.* **2023**, *26*, 3089–3112. [CrossRef]
10. Lee, T.H.; Chang, L.H.; Syu, C.W. Deep learning enabled intrusion detection and prevention system over SDN networks. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 7–11 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
11. Assis, M.V.; Carvalho, L.F.; Lloret, J.; Proença, M.L., Jr. A GRU deep learning system against attacks in software defined networks. *J. Netw. Comput. Appl.* **2021**, *177*, 102942. [CrossRef]
12. Ujjan RM, A.; Pervez, Z.; Dahal, K.; Bashir, A.K.; Mumtaz, R.; González, J. Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN. *Future Gener. Comput. Syst.* **2020**, *111*, 763–779. [CrossRef]
13. Novaes, M.P.; Carvalho, L.F.; Lloret, J.; Proença, M.L., Jr. Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments. *Future Gener. Comput. Syst.* **2021**, *125*, 156–167. [CrossRef]
14. Al Razib, M.; Javeed, D.; Khan, M.T.; Alkanhel, R.; Muthanna, M.S.A. Cyber Threats Detection in Smart Environments Using SDN-Enabled DNN-LSTM Hybrid Framework. *IEEE Access* **2022**, *10*, 53015–53026. [CrossRef]
15. Gebremeskel, T.G.; Adere, K.; Krishna, T.; Ramulu, P.J. DDoS Attack Detection and Classification Using Hybrid Model for Multicontroller SDN. *Wirel. Commun. Mob. Comput.* **2023**, *2023*, 9965945. [CrossRef]
16. Dey, S.K.; Rahman, M.M. Effects of machine learning approach in flow-based anomaly detection on software-defined networking. *Symmetry* **2019**, *12*, 7. [CrossRef]
17. Javeed, D.; Gao, T.; Khan, M.T. SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT. *Electronics* **2021**, *10*, 918. [CrossRef]
18. Wang, J.; Wang, L. SDN-Defend: A Lightweight Online Attack Detection and Mitigation System for DDoS Attacks in SDN. *Sensors* **2022**, *22*, 8287. [CrossRef] [PubMed]
19. Said, R.B.; Sabir, Z.; Askerzade, I. CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software Defined Networking with Hybrid Feature Selection. *IEEE Access* **2023**, *11*, 138732–138747. [CrossRef]
20. Wang, K.; Fu, Y.; Duan, X.; Liu, T.; Xu, J. Abnormal traffic detection system in SDN based on deep learning hybrid models. *Comput. Commun.* **2024**, *216*, 183–194. [CrossRef]
21. Cao, Y.; Jiang, H.; Deng, Y.; Wu, J.; Zhou, P.; Luo, W. Detecting and mitigating DDoS attacks in SDN using spatial-temporal graph convolutional network. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 3855–3872. [CrossRef]
22. Ismail, R.; Al-Turjman, F. Enhancing QoS and Security in Software Defined Networks Using Perceptron-Based Deep Learning. *Int. J. Intell. Syst. Appl. Eng.* **2024**, *9*, 39–45.
23. Nguyen, T.G.; Phan, T.V.; Hoang, D.T.; Nguyen, T.N.; So-In, C. Federated deep reinforcement learning for traffic monitoring in SDN-based IoT networks. *IEEE Trans. Cogn. Commun. Netw.* **2021**, *7*, 1048–1065. [CrossRef]

24. Ali, M.N.; Imran, M.; din MS, U.; Kim, B.S. Low rate DDoS detection using weighted federated learning in SDN control plane in IoT network. *Appl. Sci.* **2023**, *13*, 1431. [CrossRef]
25. Liu, L.; Lin, J.; Wang, P.; Liu, L.; Zhou, R. Deep Learning-Based Network Security Data Sampling and Anomaly Prediction in Future Network. *Discret. Dyn. Nat. Soc.* **2020**, *2020*, 4163825. [CrossRef]
26. Di Mauro, M.; Galatro, G.; Fortino, G.; Liotta, A. Supervised feature selection techniques in network intrusion detection: A critical review. *Eng. Appl. Artif. Intell.* **2021**, *101*, 104216. [CrossRef]
27. Bhattacharyya, T.; Chatterjee, B.; Singh, P.K.; Yoon, J.H.; Geem, Z.W.; Sarkar, R. Mayfly in harmony: A new hybrid meta-heuristic feature selection algorithm. *IEEE Access* **2020**, *8*, 195929–195945. [CrossRef]
28. NSL-KDD Dataset. Available online: https://www.kaggle.com/datasets/hassan06/nslkdd (accessed on 16 March 2024).
29. Alkhamisi, A.; Katib, I.; Buhari, S.M. Blockchain -Assisted Hybrid Deep Learning-Based Secure Mechanism for Software Defined Networks. In Proceedings of the 2023 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 6–8 January 2023; pp. 1–8. [CrossRef]
30. UNSW_NB15 Dataset. Available online: https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15 (accessed on 1 February 2024).
31. Elsayed, M.S.; Le-Khac, N.A.; Jurcut, A.D. InSDN: A novel SDN intrusion dataset. *IEEE Access* **2024**, *8*, 165263–165284. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.