network

MDPI

Article

Optimized MLP-CNN Model to Enhance Detecting DDoS Attacks in SDN Environment

Mohamed Ali Setitra [1,*], Mingyu Fan [1], Bless Lord Y. Agbley [2] and Zine El Abidine Bensalem [1]

[1] School of Computer Science and Engineering (Cyberspace Security), University of Electronic Science and Technology of China (UESTC), No. 2006, Xiyuan Ave., West Hi-Tech. Zone, Chengdu 611731, China; ff98@uestc.edu.cn (M.F.); z.bensalem@std.uestc.edu.cn (Z.E.A.B.)

[2] School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), No. 2006, Xiyuan Ave., West Hi-Tech. Zone, Chengdu 611731, China; agbleybless@uestc.edu.cn

* Correspondence: setitra@std.uestc.edu.cn

Abstract: In the contemporary landscape, Distributed Denial of Service (DDoS) attacks have emerged as an exceedingly pernicious threat, particularly in the context of network management centered around technologies like Software-Defined Networking (SDN). With the increasing intricacy and sophistication of DDoS attacks, the need for effective countermeasures has led to the adoption of Machine Learning (ML) techniques. Nevertheless, despite substantial advancements in this field, challenges persist, adversely affecting the accuracy of ML-based DDoS-detection systems. This article introduces a model designed to detect DDoS attacks. This model leverages a combination of Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN) to enhance the performance of ML-based DDoS-detection systems within SDN environments. We propose utilizing the SHapley Additive exPlanations (SHAP) feature-selection technique and employing a Bayesian optimizer for hyperparameter tuning to optimize our model. To further solidify the relevance of our approach within SDN environments, we evaluate our model by using an open-source SDN dataset known as InSDN. Furthermore, we apply our model to the CICDDoS-2019 dataset. Our experimental results highlight a remarkable overall accuracy of 99.95% with CICDDoS-2019 and an impressive 99.98% accuracy with the InSDN dataset. These outcomes underscore the effectiveness of our proposed DDoS-detection model within SDN environments compared to existing techniques.

Keywords: SDN; DDoS attacks; deep learning; CNN; MLP; optimization; feature selection

Citation: Setitra, M.A.; Fan, M.; Agbley, B.L.Y.; Bensalem, Z.E.A. Optimized MLP-CNN Model to Enhance Detecting DDoS Attacks in SDN Environment. *Network* **2023**, *3*, 538–562. https://doi.org/10.3390/network3040024

Academic Editors: Hakim Mellah and Filippo Malandra

Received: 23 October 2023
Revised: 27 November 2023
Accepted: 28 November 2023
Published: 1 December 2023

1. Introduction

A Distributed Denial of Service (DDoS) attack seeks to incapacitate a service or resource by inundating it with an overwhelming traffic volume, rendering it inaccessible to legitimate users. These attacks are typically orchestrated through a network of compromised devices known as botnets, unleashing a deluge of traffic upon the target [1]. DDoS attacks can wreak havoc on online services, and combating these threats can be exceptionally challenging due to the multifarious sources from which the traffic originates [2,3].

Recently, there has been a surge in high-profile DDoS attacks targeting various entities, employing many techniques, including amplification attacks that exploit vulnerabilities in Internet of Things (IoT) devices. This evolving landscape has posed significant challenges to cybersecurity. The advent of Software-Defined Networking (SDN), which enables a centralized controller to oversee and configure the entire network, has made SDN a prime target for DDoS attacks, as depicted in Figure 1 [3,4]. Attackers have leveraged the unique architecture of SDN infrastructures to develop increasingly sophisticated and potent DDoS threats, raising serious concerns for both the online community and service providers. Machine Learning (ML) has emerged as a crucial defense mechanism [5].

The integration of ML into DDoS detection has gained momentum, positioning it as an indispensable tool for safeguarding SDN environments. ML algorithms can assimilate insights from historical data and adapt to evolving SDN dynamics, substantially enhancing the efficiency and effectiveness of DDoS detection. Nevertheless, given the ever-evolving nature of these threats, refining ML-based DDoS detection is primary in addressing this monumental challenge [6,7].

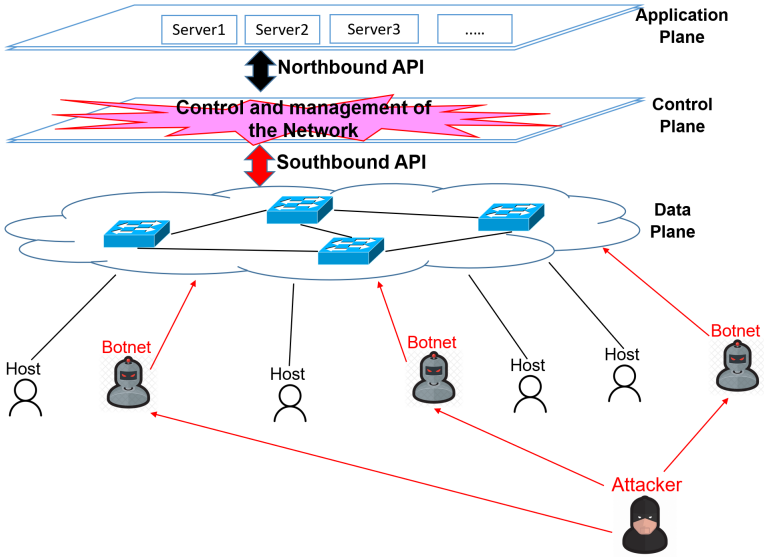

Figure 1. DDoS attacks in SDN [3].

Numerous approaches can be harnessed to enhance the effectiveness of ML in DDoS-attack detection. These strategies encompass feature selection, Deep Learning (DL), and optimization techniques. Feature selection aims to pinpoint the most relevant attributes within a dataset, thereby fortifying the ML model's capability to discern DDoS attacks [4]. Trimming down the dataset's dimensionality and discarding extraneous or duplicative features contributes to superior model performance and guards against pitfalls like overfitting or poor generalization. Furthermore, a hybrid DL model emerges as a formidable asset in ML [3]. By amalgamating their inherent strengths, it unites disparate neural network architectures or DL models to confront specific tasks and challenges. This amalgamated approach paves the way for heightened performance and resolving intricate problems. Moreover, these models harness the most effective methods from a pool of trained alternatives. Techniques such as stacking, bagging, and boosting empower the identification of the cream of the crop among these models, thereby elevating the efficacy of DDoS-detection systems to a new level [8].

This article introduces a highly effective model to enhance the detection of DDoS attacks within SDN environments. This method combines a feature selection technique with an optimized hybrid DL model to craft a robust classifier. The initial step involves the application of a feature-selection method, precisely the SHAP-feature-selection technique, to identify the most relevant attributes for DDoS detection. The core of this model lies in a hybrid neural network design that blends the capabilities of Multilayer Perceptron (MLP) and Convolutional Neural Networks (CNNs). This combination is adept at capturing the intricate components and temporal dependencies within network traffic data, making it an ideal choice for the task at hand. Furthermore, hyperparameter tuning and the Adaptive Moment Estimation (ADAM) optimizer are employed to fine tune and enhance the proposed method. The effectiveness of the OptMLP-CNN model is assessed through evaluations conducted on two diverse and widely used datasets: InSDN and CICDDoS-2019. This comprehensive evaluation framework validates the theoretical robustness of the proposed model, affirming its high degree of practicality within real-world SDN environments. The main contributions are:

- Introduction of OptMLP-CNN: This research introduces a novel DDoS-attack-detection method referred to as "OptMLP-CNN". This method combines two critical elements: SHAP-feature selection and a hybrid neural network architecture. The primary goal is to create an optimized and effective DDoS attack detector.
- Optimization Using Bayesian and ADAM Optimizers: We optimized the proposed model by applying two optimization techniques: the Bayesian optimizer and the ADAM optimizer. These optimization methods fine tune the model to enhance its performance and effectiveness in detecting DDoS attacks.
- In-Depth Analysis of DDoS-Attack-Detection Systems: The research extends beyond introducing a new model by conducting a detailed analysis of DDoS-attack-detection systems. This analysis encompasses systems that incorporate DL techniques and were selected based on specific criteria. The criteria include evaluating and comparing their DDoS-detection performance, the datasets used, optimization methods applied, and the types of systems they are designed to protect.
- Evaluation of Effectiveness of Using Public Datasets: This study evaluates the proposed method by applying it to two publicly available datasets. One of these datasets is deliberately constructed to simulate an SDN environment. The results of this evaluation demonstrate that OptMLP-CNN, the proposed method, achieves a high accuracy rate and outperforms other existing methods in detecting DDoS attacks in the context of SDN infrastructure.
- Promising Solution for SDN Security: This research's overarching contribution is the presentation of a promising solution for enhancing the security of SDN and addressing the growing threat of DDoS attacks. By introducing and optimizing the OptMLP-CNN method, this study aims to enhance the resilience of SDN environments against DDoS attacks, which have significant security implications in contemporary networked environments.

This paper unfolds as follows: In Section 2, an extensive survey of related research is presented, focusing on advancing DDoS detection through utilizing ML techniques. Section 3 explores the foundational concepts and overarching background within this domain. Section 4 introduces a novel model for DDoS detection in SDN, incorporating a feature-selection method and an optimized hybrid DL model. Section 5 presents an in-depth analysis of the proposed model's performance, along with a detailed comparative assessment of the experimental outcomes. Section 6 encapsulates the paper with final insights, synthesizing findings and delineating potential future research directions, providing a holistic framework for advancing DDoS detection in SDN environments.

Motivation

In network security, the motivation behind this research is driven by the pressing need to combat the increasing menace of DDoS attacks. These attacks have evolved in frequency, sophistication, and ability to disrupt essential network services, leading to severe financial and reputational repercussions for organizations. This alarming trend in DDoS attacks demands an urgent response in the form of improved detection and mitigation mechanisms.

One key area of concern is the vulnerability of networks operating under the paradigm of SDN. While SDN offers numerous advantages in centralized control and dynamic configurations, it also introduces novel vulnerabilities that make SDN environments attractive targets for DDoS attacks. Safeguarding SDN networks from such threats has become a vital concern, necessitating the development of advanced DDoS-detection solutions tailored to this unique network architecture.

ML and DL techniques have emerged as potent tools for enhancing network security, given their capacity to process vast volumes of network traffic data and identify anomalous patterns. However, to fully harness the potential of these models, they require meticulous optimization, customization, and adaptation to the specific dynamics of SDN environments.

Optimization plays a crucial role in the effectiveness of DDoS-detection systems. This entails fine tuning model hyperparameters, selecting relevant features, and refining

the model architecture. In SDN, where network behavior can significantly differ from traditional networking environments, this customization and rigorous optimization are vital to achieving heightened accuracy and efficiency in DDoS detection.

SDN networks introduce unique characteristics such as centralized control and dynamic reconfiguration. To effectively counter DDoS attacks in SDN, detection systems must be tailored to these distinctive attributes and demonstrate adaptability to evolving network conditions. Beyond accuracy, DDoS-detection models should offer interpretability, allowing network administrators to understand the rationale behind detection decisions. Furthermore, these systems must exhibit robustness in the face of rapidly evolving attack techniques and changes in network configurations.

Developing an effective DDoS-detection solution for SDN requires a comprehensive evaluation within a realistic SDN environment. The rigorous assessment provides critical insights into the model's capacity to distinguish legitimate traffic from DDoS attacks, ultimately determining its overall efficacy.

In light of these compelling motivations, this research embarks on developing an "Optimized MLP-CNN Model" explicitly designed to enhance the detection of DDoS attacks within SDN environments. By harnessing the capabilities of ML and DL and coupling them with meticulous optimization, this study aspires to furnish a robust, adaptable, and transparent solution that reinforces SDN networks against the mounting menace of DDoS attacks.

2. Related Works

2.1. Comprehensive Overview

This section delves into contemporary and extensively employed methodologies for detecting DDoS attacks by applying ML techniques. An exhaustive comparison of the recent related works on improving ML-based DDoS detection is summarized in Table 1.

Table 1. Comparison of recent literature.

Ref	Year	Technique	Optimizer	Feature Selection	Dataset	Network
Thakkar, A. and Lohiya, R. [9]	2023	DNN	NS	Fusion of statistical importance using standard deviation	NSL-KDD, UNSW_NB-15, CIC-IDS-2017	NS
Setitra, M. A. et al. [4]	2023	Autoencoder and XGBoost	Hyperparameter tuning	SHAP	SDNIoT, SDN-NF-TJ	SDN-IoT
Saha, S. et al. [10]	2022	Seven ML, four DL, five unsupervised	NS	15 individual methods and one ensemble method	UNSW_NB-15	NS
Türkoğlu, M. et al. [11]	2022	Bayesian optimize-DT	Bayesian optimization	MRMR	Self-generated	SD-VANET
Habib, B. and Khursheed, F. [12]	2022	LR, DT, SVM, KNN, GNB, RF, XGBoost, ANN, CNN	GridSearchCV and random sampling	Chi2, IG, merged Chi2-IG ranking ML classifiers; that is, DT, RF and XGBoost	KDD Cup 99, UNSW_NB-15	NS
Batchu, R. K. and Seetha, H. [13]	2022	Extreme ML	Memory optimization	Hybrid	CICDDoS-2019	NS
Wang, S. et al. [14]	2022	SVM, NB, FNN, KNN, GLM, DT, BT	DA, BT	Single feature based on counter within a time period	1999 DARPA, DASD, InSDN, Self-generated	SDN
Batchu, R. K. and Seetha, H. [15]	2022	KNORA-E, KNORA-U	Hyperparameter tuning	SHAP	CICDDoS-2019	NS
Chanu, U. S. et al. [16]	2022	MLP-GA, MLP, NB, RBF network	NS	IG, Gain ratio, Chi2, ReliefF, ensemble method	NSL-KDD	NS

Table 1. *Cont.*

Ref	Year	Technique	Optimizer	Feature Selection	Dataset	Network
Kshirsagar, D. and Kumar, S. [17]	2022	J48 classifier	NS	Combination of IG and correlation (CR)	CICDDoS-2019, KDD Cup 1999	NS
El Sayed, M. S. et al. [18]	2022	Autoencoder and LSTM	NS	IG, RF	InSDN, CICIDS2017, CICIDS2018	SDN
Akgun, D. et al. [19]	2022	DNN, CNN, LSTM	NS	IG	CICDDoS-2019	NS
Zhou, L. et al. [20]	2022	SVM, Guard, KNN	Threshold tuning	A proposed feature-selection method	IMPACT CAIDA	IoT
Saha, S. et al. [21]	2022	Majority Voting	NS	One from filter, wrappers, and embedded methods	UNSW_NB-15	NS
Fenil, E. and Kumar, P. M. [22]	2022	Six ML	NS	ANOVA	Self-generated	SDN
Golchin, P. et al. [23]	2022	RF, LR, SVM, NB, MLP	NS	A proposed multiaspect ensemble feature selection	CICDDoS-2019, InSDN	SDN
Setitra, M. A. et al. [3]	2022	Eight ML	NS	Adaptation and dimensionality reduction	SDN dataset	SDN
Batchu, R. K. and Seetha, H. [24]	2021	LR, GB, DT, KNN, SVM	Hyperparameter tuning	Hybrid based on Spearman's correlation and RF	CICDDoS-2019	NS
Bindra, N. and Sood, M. [25]	2020	LR, KNN, GNB, RF, SVM	NS	One from filter, wrappers, and embedded methods	CICIDS2017	NS
Polat, H., et al. [26]	2020	SVM, NB, ANN, KNN	NS	One from filter, wrappers, and embedded methods	Self-generated	SDN
Zaki, F. A. M. and Chin, T. S. [27]	2019	C4.5	NS	Hybrid based on filter and wrapper methods	Self-generated	SDN
Cauteruccio, F. et al. [28]	2019	Combining edge-based method with a cloud-based one	Multiparameter edit distance	Self-selected based on WSN behavior	Self-generated	WSN

In [9], Thakkar and Lohiya introduced a model that harnessed statistical importance, specifically standard deviation, to facilitate the selection of features within a deep neural network. Their pioneering work sought to bolster the efficiency of intrusion detection and classification. To assess the efficacy of their approach, they conducted evaluations by using three diverse datasets: NSL-KDD, UNSW_NB-15, and CIC-IDS-2017. In [4], the authors presented an optimized approach for precisely identifying DDoS attacks within Software-Defined Internet of Things (SDIoT) networks. This approach harnessed the power of Autoencoder and XGBoost algorithms' power, meticulous feature selection, and hyperparameter-tuning optimization. The method's robustness was evaluated through rigorous testing by using two specific SDN-IoT datasets: SDN-NF-TJ and SDNIoT.

The authors in [10] presented a DDoS-detection analysis by using seven ML algorithms, four DL, and five unsupervised models. They evaluated the performance of 15 methods to select features according to the UNSW_NB-15 dataset. The authors of [11] presented an approach to address the security of the SDN paradigm compared to the traditional Vehicular Ad Hoc Networking (VANET); they simulated a dataset and proposed a system based on the Minimum Redundancy Maximum Relevance process to deal with DDoS attacks. This approach was implemented in a Bayesian optimization-based decision tree classifier.

In [12], the authors evaluated the performance of detecting DDoS attacks by using logistic regression, a decision tree classifier, linear support vector machine, k-nearest neighbors, Gaussian Naive Bayes, Random Forest Classifier, XGBoost, ANN, and CNN. They also used hyperparameter optimization and applied various techniques to select features on KDD Cup 99 and UNSW-NB15. An extreme learning machine was used in [13] to ameliorate the performance related to the DDoS-attack-detection system; the authors

suggested a method based on memory optimization and a hybrid approach for selecting features. They conducted the experiments by using the CICDDoS-2019 dataset.

In the SDN infrastructures context, the article [14] simulated an SDN environment to evaluate DDoS-attack detection in SDN by using supervised learning techniques, and 1999 DARPA, the DDoS-attack SDN dataset (DASD), and InSDN datasets were also used for this evaluation. In [15], the authors employed the SHAP-feature weight within recursive feature exclusion by using five base classifiers to detect DDoS attacks.

In [16,25], many experiments were conducted to evaluate feature selection's impact on Ensemble Learning techniques' performance by combining several ML models to improve the overall performance. It is based on the principle that multiple models working together often achieve better performance than a single model alone. In [17,24,27], the authors proposed hybrid feature-selection methods to enhance the DDoS-detection systems. Selecting features according to the Information Gain technique was also evaluated in [18,19] to deal with DDoS attacks.

The authors of [20] proposed a method to select features and detect DDoS attacks based on flow classification and threshold tuning for optimization. The authors of [26] generated a dataset in an SDN-simulated network and evaluated the performances against DDoS attacks through different methods of selecting features and ML models. The paper [21] evaluated selecting-features methods by using Majority Voting and the UNSW_NB-15 dataset. The authors of [22,23] exploited the SDN infrastructures to evaluate the DDoS-attack-detection systems. In [3], the authors investigated a methodology based on dataset understanding, feature modeling, and dimensionality reduction to improve DDoS detection via ML-based systems in the SDN environment.

The approach introduced by Cauteruccio et al. [28] leverages edge-data analysis combined with cloud data analysis to autonomously detect anomalies within heterogeneous sensor networks. Their methodology utilizes a fully unsupervised artificial neural network algorithm for edge-data analysis, while cloud-data analysis involves the application of the multiparameterized edit-distance algorithm.

2.2. Key Findings

The realm of detecting DDoS attacks has witnessed a substantial surge in research endeavors. These investigations have honed in on the strategic fusion of SDN infrastructures and the capabilities of DL models within diverse domains. This collective exploration has produced a trove of invaluable insights, each adding a distinctive layer to the evolving landscape of DDoS detection. Let us delve into these pivotal findings:

- Network-Specific Innovations: A significant portion of this research body has been intricately tailored to address the intricate nuances of specific network types. Notably, SDN and SD-VANET networks have emerged as focal points of interest. Researchers have recognized the importance of customizing DDoS-detection approaches to suit these specialized network environments. These tailored strategies delve into the unique characteristics and challenges that SDN and SD-VANET networks present, paving the way for more effective ML-based DDoS detection within these domains.
- Feature Selection and Deep Learning's Prowess in SDN: In the context of SDN environments, a remarkable discovery has centered around the power of feature-selection techniques. These methodologies meticulously sift through data to pinpoint the most pertinent features, substantially elevating the performance of ML-based DDoS-detection systems. Furthermore, DL techniques have come under the microscope, with researchers diligently assessing their contribution to SDN's overarching DDoS-detection framework. DL models have showcased an unparalleled aptitude for capturing intricate data patterns and relationships, positioning them as valuable assets in the arsenal against DDoS threats.
- Fine Tuning via Hyperparameter Optimization: Research efforts have introduced a meticulous fine-tuning mechanism to optimize the performance of DDoS-detection systems. Hyperparameter-optimization techniques, including Bayesian optimization

and GridSearchCV, have played a central role in systematically adjusting the settings of ML models. This systematic calibration process bolsters the accuracy and efficiency of these models, ensuring that they are finely attuned to the intricate nuances of DDoS detection.

- The Rise of Hybrid Deep Learning Models: A prominent revelation in this landscape pertains to the emergence of hybrid DL models. These innovative architectures skillfully merge multiple DL frameworks or combine DL with traditional ML techniques. The overarching objective is to enhance the interpretability and robustness of DDoS-detection systems. By amalgamating diverse models, these hybrid entities provide a wealth of insights and information that guide the final decision making. This synergy augments the accuracy of DDoS detection and crystallizes the findings, making them more comprehensible and actionable.

In essence, the meticulous research endeavors in DDoS detection have coalesced into a compendium of strategies, each meticulously tailored to the nuances of SDN environments and propelled by the profound capabilities of DL models. These discoveries encompass network-specific tailoring, the precision of feature selection, the finesse of hyperparameter optimization, and the innovation of hybrid models. Collectively, these findings fortify the security of networks and equip them to confront the ever-evolving specter of DDoS attacks with heightened resilience.

3. Background

ML constitutes a pivotal realm within cognitive computing, focusing on creating intelligent systems capable of autonomously acquiring knowledge from data to make predictions and decisions without explicit programming. This field encompasses various ML categories: supervised, unsupervised, semisupervised, and reinforcement learning [29]. ML has gained significant prominence in detecting DDoS attacks in recent years, primarily due to its innate aptitude for continual learning and performance enhancement. This ongoing refinement is achieved through applying various techniques, such as feature selection, optimization, and incorporating hybrid DL methodologies [3,5].

3.1. Feature Selection

Feature selection is an important step in DDoS-attack detection, primarily due to the often high dimensionality of network data. This process is instrumental in improving the performance and interpretability of ML models by selecting a subset of the most essential features. The significance of feature selection arises from its ability to address various challenges within DDoS detection. High dimensionality can lead to computational inefficiencies and overfitting while including irrelevant or noisy features can obfuscate the model's decision-making process. Therefore, feature selection aims to mitigate these challenges by reducing dimensionality, eliminating noise, and enhancing interpretability. Feature-selection methods encompass three main categories: filter methods, which independently evaluate feature relevance; wrapper methods, which involve the ML classifier in exploring feature subsets; and embedded methods, which seamlessly integrate feature selection with the ML algorithm. The feature-selection method should be tailored to the dataset's characteristics, available computational resources, and the specific ML model in use, as it significantly impacts the effectiveness of DDoS-attack detection, particularly within SDN environments [3,4].

SHAP-feature selection represents a cutting-edge technique for assessing the significance of individual features within a dataset while constructing predictive models. Rooted in the principles of cooperative game theory, SHAP assigns a numerical value to each feature, quantifying its impact on model predictions by meticulously considering every possible combination of features. A higher SHAP value for a particular feature signifies its heightened influence on shaping the model's predictive outcomes. Armed with this knowledge, data scientists can meticulously rank features according to their SHAP values, enabling them to pinpoint the most essential attributes for model precision and inter-

pretability. This process facilitates strategically selecting a subset of the most informative features, streamlining the model-development process and elevating the overall predictive performance to new heights [4,13].

The Shapley value, formally defining the SHAP value for cooperative games, is calculated by summing the worth of all potential coalitions that do not include a specific player, denoted as i, and subtracting the value of coalitions that exclude this player, as mentioned in [4]. This calculation is divided by the total number of players, essentially quantifying the individual player's contribution to the coalition. The formula for the Shapley value is expressed as

$$\varphi i(v) = \frac{1}{n} \sum S \subseteq N \setminus i \left[n - 1 \, |S| \right]^{-1} (v(S \cup i) - v(S)) \tag{1}$$

In this equation, $|S|$ represents the size of the coalition, and N stands for the total number of players involved. The summation encompasses all feasible coalitions that exclude player i. Furthermore, $v(S)$ denotes the value attributed to the coalition S.

The Shapley value introduces two important concepts, namely "dummy" and "efficiency", which are described by Equations (2) and (3). In cooperative games, a player who contributes nothing to the final outcome and is essentially ignored in the Shapley value calculation is referred to as a "dummy" player. The principle of efficiency ensures that the summation of Shapley values for all players equals the total outcome of the game:

$$v(S \cup x_n) = v(S), \tag{2}$$

$$for\ all\ S \subseteq \{x_1, \ldots, x_m\},\ then\ \varphi_n = 0$$

$$\sum_{i \in N} \varphi_i(v) = v(N) \tag{3}$$

3.2. Optimization

Optimization is a fundamental aspect of enhancing the effectiveness and efficiency of ML models. When it comes to DDoS-attack detection, fine tuning model hyperparameters, selecting relevant features, and optimizing the model architecture are crucial. The optimization algorithm then iteratively adjusts the model's parameters to minimize the objective function by using various optimization techniques such as Bayesian optimization and ADAM optimizer.

Bayesian optimization is a potent ally among the arsenal of advanced techniques available. Bayesian optimization is a sophisticated strategy rooted in Bayesian statistical principles, designed to master the optimization of complex and computationally demanding functions. This technique shines when confronted with vast search spaces and resource-intensive evaluations, as commonly encountered in hyperparameter tuning and feature selection. At its core, Bayesian optimization revolves around creating a probabilistic model, often termed a "surrogate model," that approximates the target objective function. This surrogate model, which can take the form of a Gaussian process (GP) or even a deep neural network, plays a crucial role in estimating the objective function's behavior [30].

Surrogate Modeling: Bayesian optimization begins with establishing a prior distribution over the objective function. In conjunction with available data, this prior allows the surrogate model to approximate the objective function's behavior. The surrogate model is typically represented as a GP, which predicts accurate function values for different inputs. The prior and posterior distributions of the GP are defined as follows:

$$Prior : f(x) \sim \mathrm{GP}(\mu(x), \kappa(x, x')) \tag{4}$$

$$Posterior : f(x)|X, y, x' \sim \mathrm{GP}(\mu'(x), \kappa'(x, x')) \tag{5}$$

where

- $f(x)$ represents the function modeled by the GP.
- $\mu(x)$ is the mean function of the GP.
- $\kappa(x, x')$ is the GP's covariance function (kernel).
- X is the set of input points.
- y is the observed data.
- x' is the point for which we want to make predictions.
- $\mu'(x)$ is the mean function of the posterior GP.
- $\kappa'(x, x')$ is the covariance function (kernel) of the posterior GP.

Equations (4) and (5) capture the prior and posterior distributions used in Surrogate Modeling in Bayesian optimization, where the GP approximates the objective function based on available data.

The acquisition function: It guides the search for the optimum by quantifying the value of evaluating the objective function at different points. It balances exploration (sampling from areas where the surrogate model is uncertain) and exploitation (sampling where the surrogate model predicts high objective values). The acquisition function, often represented as $\alpha(x)$, guides the selection of the next evaluation point. One commonly used acquisition function is the Expected Improvement (EI):

$$\alpha_{\mathrm{EI}}(x) = \mathbb{E}[\max(f(x') - f(x_{\mathrm{best}}), 0)] \tag{6}$$

where

- $\alpha_{\mathrm{EI}}(x)$ is the Expected Improvement acquisition function.
- $f(x')$ is the predicted function value for a candidate point x'.
- $f(x_{\mathrm{best}})$ is the best observed function value so far.
- $\max(\cdot)$ returns the maximum of the values inside the parentheses.

The Point Selection: The acquisition function is employed to choose the next point for evaluation. This point is then assessed on the true objective function. The point to be evaluated next, x_{next}, is chosen by maximizing the acquisition function:

$$x_{\mathrm{next}} = \arg\max_{x} \alpha(x) \tag{7}$$

In Equation (7), x_{next} represents the next point selected for evaluation, and $\alpha(x)$ is the acquisition function, such as EI, that guides the choice of the next evaluation point. This equation plays a central role in the iterative process of Bayesian optimization, where new points are selected based on the acquisition function to iteratively find the optimal configuration.

On the other hand, ADAM is an optimization algorithm commonly used in DL to train ML models. It is particularly well suited for optimizing neural networks, which have become fundamental to many modern AI applications. The core idea behind ADAM is to adapt the learning rates for each parameter during training, which helps the algorithm converge faster and more efficiently. Traditional gradient-based optimization algorithms, such as Stochastic Gradient Descent (SGD), use a fixed learning rate for all parameters. ADAM, on the other hand, maintains a separate adaptive learning rate for each parameter. This adaptive learning rate is based on two key moving averages: the gradients' first moment (the mean) and the second moment (the uncentered variance) [31].

The process begins with a set of parameters, including the learning rate (α), the exponential decay rates for moment estimates (β_1 and β_2), the objective function to minimize ($f(\theta)$), and the initial parameter vector (θ_0).

Initialization: The process commences by initializing two-moment vectors: the first-moment vector, m_0, and the second-moment vector, v_0, both set to zero. The iteration counter, t, is initialized to zero as well.

Iteration: The algorithm iterates until the parameters converge. The algorithm progresses iteratively until the parameters converge. In each iteration, the counter increments ($t \leftarrow t + 1$), the objective function's gradient ($\nabla_\theta f_t(\theta_{t-1})$) computes based on the current

parameter vector, and the first-moment estimate (m_t) updates through an exponential moving average with β_1. Similarly, the second-moment estimate (v_t) updates by using β_2, incorporating the square of the gradient. Bias-corrected estimates, $\hat{m}_t$ and $\hat{v}_t$, adjust for the exponential moving average and iteration number for m_t and v_t, respectively. The parameter vector θ_t updates by using these moments to determine step sizes, scaled by α and adjusted for numerical stability with a small constant ϵ.

Convergence: The algorithm continues these steps until the parameters converge, returning the optimal parameter vector θ_t.

The mathematical equations for the ADAM optimizer can be summarized as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{8}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{9}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{10}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{11}$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{12}$$

These equations capture the essence of the ADAM optimizer's behavior and its ability to efficiently update the parameters during the training of deep neural networks.

3.3. Deep Learning

DL is a subfield of ML that aims to mimic how the human brain processes information by utilizing neural networks composed of multiple layers, referred to as DNNs. These networks are designed to automatically learn and extract hierarchical features from raw data, making them exceptionally well suited for tasks that involve large and complex datasets. DL models have demonstrated remarkable success in various applications, including image recognition, natural language processing, and speech recognition, owing to their capacity to uncover intricate patterns within data that may not be evident to traditional ML algorithms [4].

In the context of detecting DDoS attacks, DL offers several advantages. DDoS attacks are prevalent cyber threats that inundate a network or system with an overwhelming traffic volume, disrupting regular operations. They are particularly challenging to detect due to their ability to mimic legitimate traffic patterns, rendering traditional rule-based methods less effective. DL models excel in discerning nuanced and evolving patterns within network traffic data, which can aid in the early identification of DDoS attacks. DL techniques can be applied to DDoS-attack detection in various ways. One common approach is to utilize DNNs, such as CNNs, to process network traffic data. CNNs are adept at extracting spatial features from data, making them suitable for tasks involving structured inputs, like network packet headers. One of the critical advantages of DL in DDoS detection is its adaptability. DNNs can be trained to automatically learn and adapt to new attack strategies, making them highly effective against evolving threats. Additionally, feature engineering, a labor-intensive step in traditional ML, is often unnecessary with DL. These models can autonomously extract relevant features from the raw data, reducing the burden on security analysts. Nonetheless, DL models are not without their challenges. They typically require large volumes of labeled data for training, which can be scarce in the case of DDoS attacks. Ensuring the interpretability of deep models is another concern, as black-box models may not provide insights into why an attack was flagged. Despite these challenges, adopting DL in DDoS detection showcases the potential for more accurate and adaptive security systems to identify previously unseen attack patterns, strengthening network resilience and security.

4. Proposed Model

The proposed "OptMLP-CNN" is a comprehensive model developed to enhance the detection of DDoS attacks within SDN environments. It leverages a combination of advanced techniques, including SHAP-feature selection, a fused MLP and CNN architecture, and the application of Bayesian optimization and the ADAM optimizer.

4.1. MLP

MLP is a type of artificial neural network that serves as a foundational building block for DL models. MLPs are a class of feedforward neural networks, meaning that information flows in one direction, from the input layer to the output layer. They are widely used for various ML tasks, including classification, regression, and feature learning. Let us explore the structure, components, and workings of an MLP in more detail [32,33]. An MLP consists of multiple layers of interconnected neurons, and three main types of layers characterize its structure:

- Input Layer: This is the network's first layer, receiving the raw input data. Each neuron in the input layer corresponds to a feature or variable in the dataset, making it a fundamental representation of the data's dimensions.
- Hidden Layers: MLPs can have one or more hidden layers between the input and output layers. These layers are called "hidden" because they are not directly connected to the outside world (i.e., input or output). Neurons in hidden layers take the weighted sum of inputs from the previous layer, apply an activation function, and pass the result to the next layer. The number of hidden layers and the number of neurons in each layer are hyperparameters that can be adjusted to optimize the model's performance.
- Output Layer: The final layer produces the model's output or prediction. The structure of this layer depends on the problem the MLP is designed to solve. For example, in binary classification, there may be a single neuron with a sigmoid activation function, while in multiclass classification, there might be multiple neurons, each representing a class and using a softmax activation function.

Each neuron in an MLP performs the following operations:

Weighted Sum

For each neuron in a layer (except the input layer), the input is a weighted sum of the outputs from the previous layer. This weighted sum is often referred to as the "activation" of the neuron. Mathematically, the weighted sum $z_j^{(l)}$ of neuron j in layer l can be expressed as

$$z_j^{(l)} = \sum_i w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)} \tag{13}$$

where

$z_j^{(l)}$ is the activation of neuron j in layer l.

$w_{ij}^{(l)}$ is the weight of the connection between neuron i in layer $l-1$ and neuron j in layer l.

$a_i^{(l-1)}$ is the output of neuron i in layer $l-1$.

$b_j^{(l)}$ is the bias of neuron j in layer l.

Activation Function

The weighted sum is passed through a nonlinear activation function to introduce nonlinearity into the model. Common activation functions include the sigmoid function, Rectified Linear Unit (ReLU), and the hyperbolic tangent function (tanh). The output of one layer becomes the input for the next layer:

$$a_j^{(l)} = f(z_j^{(l)}) \tag{14}$$

where $a_j^{(l)}$ is the output of neuron j in layer l and f is the activation function.

Feedforward Propagation

During the feedforward process, the activations are computed for each neuron by applying the activation function to the weighted sum. The output of one layer becomes the input for the next layer, propagating information from the input layer to the output layer:

$$a_j^{(l)} = f\left(\sum_i w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right) \tag{15}$$

4.2. CNNs

CNNs are a class of DL models specifically designed for processing structured grid data, such as images. CNNs are widely used for tasks like image classification, object detection, and image segmentation [33–35]. The components of a CNN can be summarized as follows:

4.2.1. Convolution Layer

The fundamental operation in CNNs is the convolution operation. Given an input feature map I and a convolution kernel K, the convolution operation is defined as:

$$(I * K)(x, y) = \sum_i \sum_j I(x + i, y + j) \times K(i, j) \tag{16}$$

Here, (x, y) represents the spatial position in the output feature map, and $I(x + i, y + j)$ denotes the input at spatial position $(x + i, y + j)$. The convolution operation computes the weighted sum of the input values by sliding the kernel over the input feature map.

4.2.2. Activation Function

After the convolution operation, an activation function is applied element-wise to introduce nonlinearity. Common activation functions include ReLU and sigmoid. Mathematically, this operation is expressed as

$$A(x, y) = f(I * K)(x, y) \tag{17}$$

4.2.3. Pooling Layer

Pooling layers reduce the spatial dimensions of the feature map. The most common pooling operation is max pooling, where the maximum value in a local region is retained. The max-pooling operation can be defined as

$$P(x, y) = \max_{i,j} A(x \cdot s + i, y \cdot s + j) \tag{18}$$

Here, $P(x, y)$ represents the output of the pooling layer, s is the stride, and $A(x \cdot s + i, y \cdot s + j)$ denotes the input to the pooling layer.

4.2.4. Fully Connected Layer

In the final layers of the CNN, one or more fully connected layers are used for tasks such as classification. These layers flatten the output feature map and connect every neuron to the previous layer's neurons.

The entire operation of a CNN, from convolution to fully connected layers, can be mathematically expressed as

$$Y = f(W * X + B) \tag{19}$$

Here, Y is the output, W represents the weights (learned during training), X is the input data, B is the bias term, $*$ denotes the convolution operation, and f represents the activation function.

4.3. OptMLP-CNN Detector

The methodology behind the OptMLP-CNN Detector represents an advanced approach to detecting DDoS attacks, leveraging a combination of MLP, CNN, SHAP-feature selection, Bayesian optimization, and the Adam optimizer. This methodology outlines a systematic process for deploying and maintaining a robust cybersecurity model, making it a valuable asset for safeguarding network infrastructure.

Commencing with data preprocessing as illustrated in Figure 2, the flowchart encapsulates the initial phase where the input dataset undergoes normalization, augmentation, and formatting, ensuring its readiness for subsequent modeling. It seamlessly progresses into feature selection, incorporating SHAP to discern the pivotal features essential for robust DDoS-attack detection. This phase refines the dataset, laying the foundation for subsequent modeling endeavors. A crucial "Class Distribution Evaluation" step at this junction determines the following pathway: directly proceeding to SHAP-feature selection or opting for Up-Sampling before engaging in SHAP-feature selection.

The heart of the flowchart unfolds with the construction of the combined model, constituting both MLP and CNN architectures. This merger aims to harness the diverse strengths of both models, leveraging MLP's prowess in handling structured data alongside CNN's adeptness in spatial- and image-data analysis. The step-by-step delineation within the flowchart elucidates the intricate layers and connections within the combined architecture, emphasizing the fusion's intricacies.

The flowchart seamlessly transitions into the compilation and training phases. Here, crucial elements such as defining loss functions, optimizers, and evaluation metrics are introduced, followed by rigorous training by using the designated dataset. The subsequent evaluation using a separate testing dataset and calculating performance metrics ensue, validating the model's accuracy and efficacy.

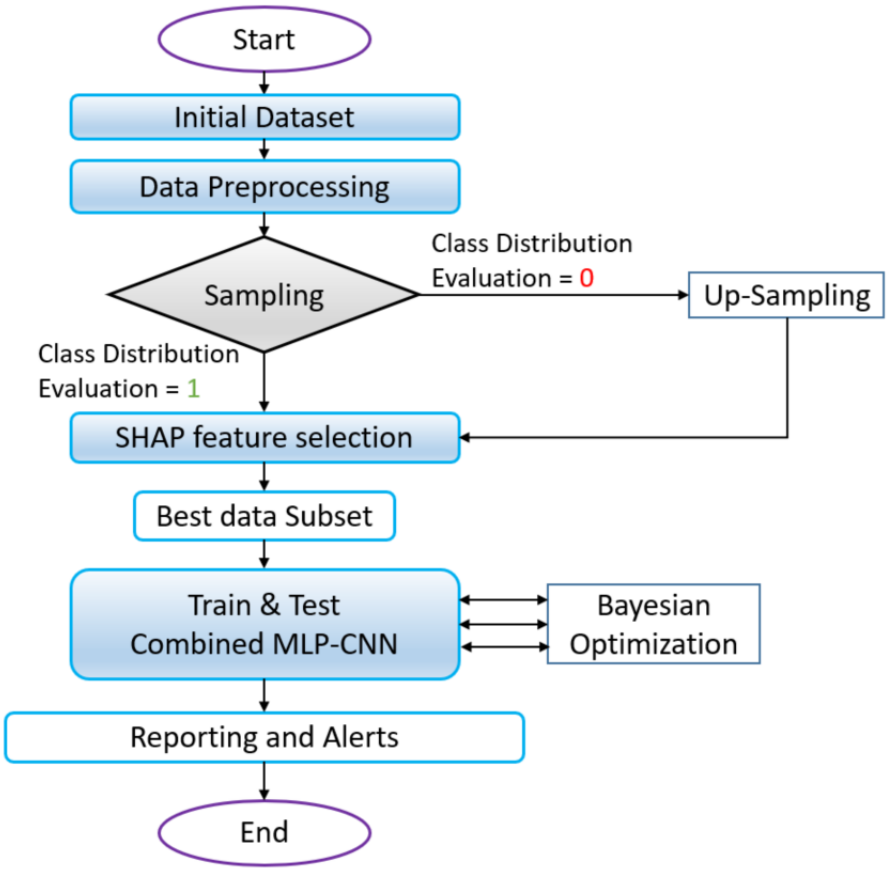

Figure 2. Summary of the OptMLP-CNN flowchart.

The flowchart does not conclude merely with model training but extends into the domain of hyperparameter optimization by using Bayesian techniques. This stage refines the model's efficiency by fine tuning hyperparameters, enhancing its precision in identifying and mitigating DDoS attacks. The option for model deployment, continuous monitoring, reporting, and alert mechanisms for detected attacks further solidifies the adaptive and proactive nature of the OptMLP-CNN Detector.

The flowchart is a visual representation of the intricate and comprehensive workflow that underlies the OptMLP-CNN Detector, offering a structured roadmap from data preprocessing and feature selection to model training, refinement, and potential deployment.

The proposed MLP Architecture algorithm outlines the design and implementation of our MLP model, which is fundamental for classification tasks (see Algorithm 1). Its workflow commences with data preprocessing, encompassing normalization, augmentation, and dataset formatting, crucial preparatory steps for model training. The MLP architecture consists of an input layer, two hidden layers, and an output layer. The input layer's size aligns with the dataset's feature count, while the hidden layers, comprising 128 and 64 units utilize the ReLU activation function. The output layer's configuration adapts to the classification task, employing softmax for multiclass or sigmoid for binary classification.

Upon model construction, the algorithm proceeds to compile the MLP model by defining appropriate loss functions, optimizers, and evaluation metrics. Subsequently, it undergoes training on the designated training dataset across specified epochs. After training, the model undergoes rigorous evaluation by using the testing dataset, assessing diverse performance metrics encompassing the accuracy, precision, recall, and F1-score. Additionally, the algorithm accommodates hyperparameter tuning and provides options for deploying the trained model. It strongly advocates continuous monitoring and iterative updates to ensure adaptability and performance refinement.

The proposed CNN Architecture algorithm delineates the creation of a CNN model, paramount for image-based classification tasks (refer to Algorithm 2). Aligning with the MLP Architecture, the algorithm initiates data preprocessing, ensuring dataset normalization, augmentation, and formatting for the CNN model. The CNN architecture encompasses two convolutional layers, each paired with ReLU activation and same padding. Max-pooling layers augment these layers to reduce feature map dimensionality. Further components include a flattening layer and two fully connected layers, employing softmax for multiclass or sigmoid for binary classification in the last layer.

Algorithm 1 Proposed MLP Architecture

Require: training data D_{train}, testing data D_{test}
Require: hyperparameters for MLP architecture
1: Preprocess the data: normalize, augment, and format the dataset
2: Build the MLP model:
3: Number of layers: 3
4: Units per layer:
5: Input layer: number of features in your data
6: Hidden layer 1: 128 units with ReLU activation
7: Hidden layer 2: 64 units with ReLU activation
8: Output layer: number of output classes with suitable activation function (e.g., softmax for multiclass or sigmoid for binary classification)
9: Compile the MLP model: define loss, optimizer, and evaluation metrics
10: Train the MLP model on D_{train} for a set number of epochs
11: Evaluate the model on D_{test} and calculate performance metrics
12: Fine tune hyperparameters as needed
13: Optionally deploy the model
14: Continuously monitor and update the model

Algorithm 2 Proposed CNN Architecture

Require: training data D_{train}, testing data D_{test}
Require: hyperparameters for CNN architecture
Preprocess the data: normalize, augment, and format the dataset
Build the CNN model:
Convolutional layer 1:
Number of filters: 32
Filter size: 3×3
Activation function: ReLU
Padding: same
Max pooling layer 1:
Pool size: 2×2
Convolutional layer 2:
Number of filters: 64
Filter size: 3×3
Activation function: ReLU
Padding: same
Max pooling layer 2:
Pool size: 2×2
Flatten layer: converts 2D feature maps to a 1D vector
Fully connected layer 1:
Number of units: 128
Activation function: ReLU
Fully connected layer 2 (output layer):
Number of units: number of classes
Activation function: softmax (for multiclass) or sigmoid (for binary classification)
Compile the CNN model: define loss, optimizer, and evaluation metrics
Train the CNN model on D_{train} for a set number of epochs
Evaluate the model on D_{test} and calculate performance metrics
Fine tune hyperparameters as needed
Optionally deploy the model
Continuously monitor and update the model

The algorithm compiles the CNN model postconstruction, specifying crucial components like loss functions, optimizers, and evaluation metrics. Subsequent training on the provided dataset occurs over a defined number of epochs. Upon completion, the model undergoes rigorous evaluation on the testing dataset, where diverse performance metrics are computed. Similar to the MLP Architecture, provisions for hyperparameter tuning, optional deployment, and continuous model monitoring and updating are integrated into this CNN model.

The combined MLP-CNN Detector, as detailed in Algorithm 3, merges the MLP and CNN architectures to optimize DDoS-attack detection. It integrates preprocessing for dataset refinement and employs SHAP for feature selection, which is crucial for identifying key DDoS-related features. This amalgamation paves the way for a robust combined MLP-CNN model that leverages MLP's structured data handling and CNN's image-analysis prowess. The algorithm initiates with meticulous data preprocessing, normalizing and enhancing the dataset, and SHAP-feature selection. This preparatory phase primes the data for modeling. The heart of the algorithm lies in merging the MLP and CNN architectures to craft a powerful model adept at decoding complex patterns associated with DDoS attacks.

Algorithm 3 Combined MLP-CNN Detector

Require: training data D_{train}, testing data D_{test}
Require: hyperparameters for MLP and CNN architectures
Require: SHAP-feature selection and Bayesian optimization parameters
Require: Adam optimizer hyperparameters
1: Preprocess the data: normalize, augment, and format the dataset
2: Select the most important features using SHAP
3: Build the combined model: combining MLP and CNN architectures
4: Compile the combined model: define loss, optimizer, and evaluation metrics
5: Train the combined model on D_{train} for a set number of epochs
6: Evaluate the model on D_{test} and calculate performance metrics
7: Perform Bayesian optimization to fine tune hyperparameters
8: Retrain the model with optimal hyperparameters
9: Optionally deploy the model
10: Continuously monitor and update the model
11: Implement reporting and alert mechanisms for detected DDoS attacks
12: Maintain documentation of the model architecture and training process
13: Update and maintain the model regularly

Subsequent steps entail model compilation, defining loss functions and metrics, and rigorous training. This training phase refines the model's parameters, improving its recognition of DDoS indicators. Post-training, the model undergoes a thorough evaluation by using a separate dataset and employs Bayesian optimization to fine tune hyperparameters, enhancing precision in identifying and mitigating DDoS attacks. The algorithm's adaptive nature includes provisions for model deployment in DDoS-detection systems, advocating continuous monitoring and updates to counter evolving threats effectively. It highlights the implementation of alert mechanisms for detected attacks, ensuring swift responses, and emphasizes meticulous documentation for comprehensive records of the model's architecture and training processes.

5. Experimental Results

In this section, we provide a detailed evaluation of the proposed model. We begin by outlining the experimental setup and introducing the evaluation metrics employed to assess the model's performance. Subsequently, the model's performance is examined under various scenarios, including exploring different activation functions and input-weight ranges. To gauge the effectiveness of our model, we compare the results with those of recent state-of-the-art methods. The experiments were conducted by using a Jupyter Notebook with Python, leveraging libraries such as scikit-learn, Pandas, and Matplotlib. The computational environment featured a 2.60 GHz Intel Core i7 10G processor, 4 GB NVidia GTX 1650 Ti graphics, 16 GB of RAM, and the Windows operating system. This rigorous evaluation validates our proposed model's robustness and effectiveness.

5.1. Dataset

In our study, we harnessed two datasets: InSDN [36] and CICDDoS-2019 [37]. The CICDDoS-2019 dataset [37] provides a contemporary repository of DDoS-attack traces that manifest at the application layer, employing TCP/UDP-based protocols. This dataset classifies DDoS attacks into two distinct categories: reflection based and exploitation based. Reflection-based attacks involve using a reflector server to redirect malicious traffic towards the target, masking the source IP address. In contrast, exploitation-based attacks directly target the victim without needing a reflector server. The CICDDoS-2019 dataset encompasses a rich feature set comprising 88 distinct features. The InSDN dataset, as detailed by the authors of [36], aligns its focus with SDN infrastructures. This dataset was meticulously curated by deploying four virtual machines and incorporating a broad spectrum of attack classes from the SDN network's internal and external sources. It also

includes a diverse representation of normal traffic, reflecting numerous application services. The InSDN dataset encompasses a substantial collection of 361,317 instances, including 68,424 benign or normal traffic and 292,893 instances of attack traffic. The dataset comprises 84 features that serve as the foundation for our research endeavors.

5.2. Performance Metrics

Evaluating the effectiveness and accuracy of the DDoS-detection model is paramount. The literature consistently employs a set of well-established evaluation metrics based on four fundamental elements: true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

Accuracy: This widely used metric provides a holistic assessment of the classifier's correctness. It is calculated by dividing the total number of correct predictions by the total number of predictions made. The accuracy is computed as

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{20}$$

Precision: Precision measures the proportion of true positive predictions out of all positive forecasts. It is defined as

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

Recall: Recall quantifies the percentage of actual positive events correctly predicted. It is calculated as

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

F1-Score: This metric harmonizes precision and recall, offering a balanced assessment of a model's performance. It is computed as the harmonic mean of precision and recall

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{23}$$

Area Under the Receiver Operating Characteristic Curve (AUC-ROC): The AUC-ROC is a graphical representation that showcases a classifier's performance. It plots the true positive rate against the false positive rate, providing a visual performance indicator. The AUC value quantifies the classifier's accuracy, with higher values indicating superior performance.

The choice of performance evaluation metrics, including accuracy, precision, recall, F1-score, and AUC-ROC, reflects a meticulous consideration of various assessment dimensions essential for a holistic model evaluation. Accuracy stands as a fundamental indicator, quantifying the ratio of correctly predicted instances to the total. Precision accentuates the model's prowess in identifying true positives among all positive predictions, while recall illuminates its capacity to capture all actual positives. The F1-score, a harmonic mean of precision and recall, offers a balanced assessment. Additionally, the AUC-ROC provides insights into the model's discriminative ability across classes. This suite of metrics ensures a comprehensive understanding of the model's performance, offering nuanced insights into its strengths and potential limitations across diverse performance aspects.

5.3. Detection Phase

Figures 3 and 4 portray the outcomes derived from conducting the SHAP-feature-selection experiment by using the CICDDoS2019 and InSDN datasets, respectively, within the framework of the OptMLP-CNN model. These visual representations encapsulate the fluctuating landscape of feature importance, quantified through Shapley values, across varying subset sizes. These graphics serve as powerful tools, offering deep insights into the pivotal features that significantly influence the OptMLP-CNN model's efficacy in identifying and

flagging DDoS attacks. Through meticulous scrutiny of these figures, we can discern the features exerting substantial influence on the model's predictive outcomes.

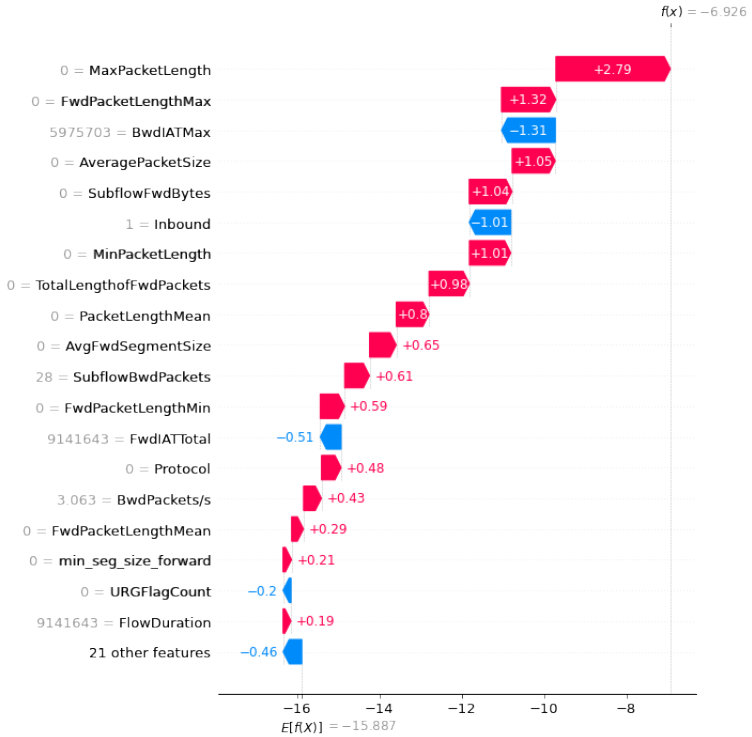

Figure 3. SHAP summary plot using CICDDoS2019 dataset.

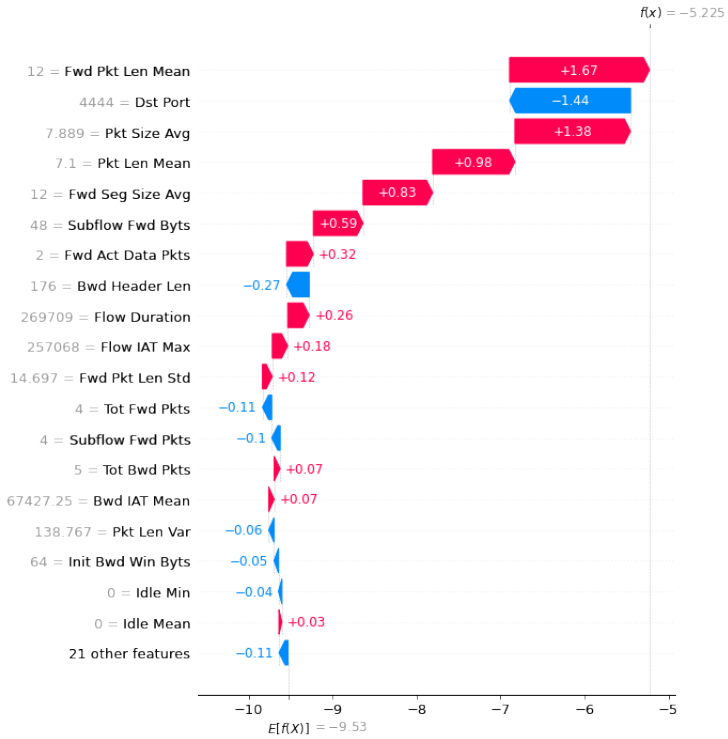

Figure 4. SHAP summary plot using InSDN dataset.

This scrutiny steers the strategic decision-making process regarding feature inclusion or exclusion, consequently bolstering the model's proficiency and accuracy in detecting DDoS attacks within the intricate realms of SDN environments. Figures 3 and 4 facilitate

the enhancement of feature-selection methodologies, thereby contributing substantively to the evolution of more robust and effective DDoS-detection models.

The correlation matrices depicted in Figures 5 and 6 offer comprehensive insight into the inter-relationships among the 20 selected features obtained through the SHAP-selection method applied to the CICDDoS-2019 and InSDN datasets, respectively. Each cell in these matrices represents the correlation coefficient between a pair of features, ranging from -1 to 1. A coefficient of -1 indicates a perfect negative correlation, signifying that as one feature increases, the other decreases in proportion. Conversely, a coefficient of one signifies a complete positive correlation, indicating that both features either increase or decrease simultaneously. A correlation coefficient of 0 denotes no linear relationship between the features. This visualization serves as a pivotal tool to unearth redundant features and discern the most influential ones in shaping the DDoS-detection model's performance within these datasets.

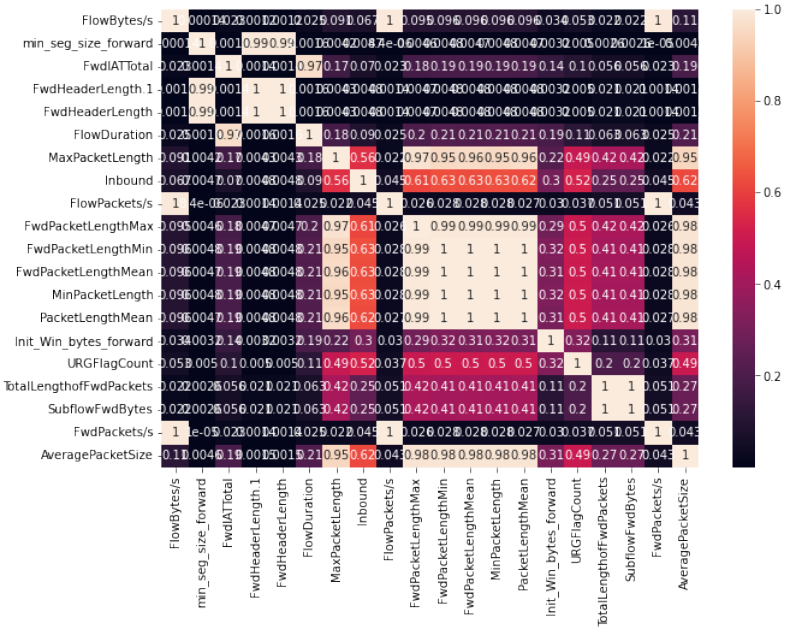

Figure 5. Correlation of selected features in CICDDoS-2019.

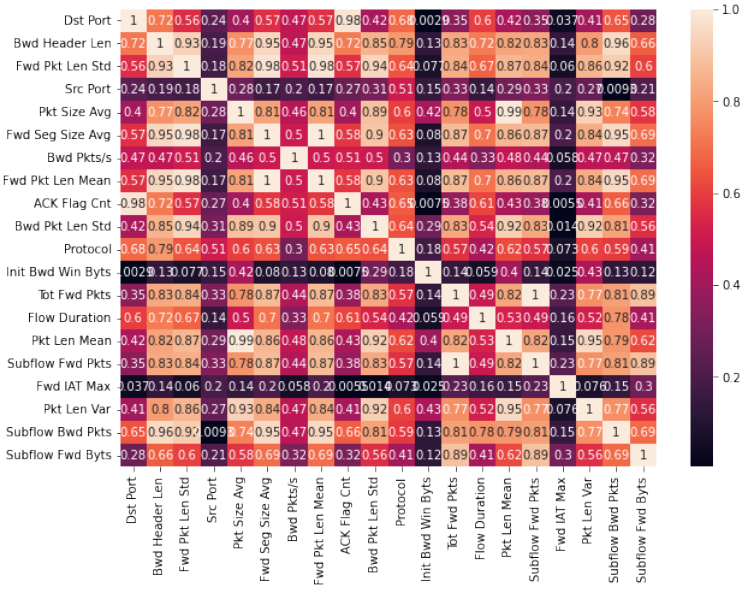

Figure 6. Correlation of selected features in InSDN.

The performance evaluation of the optimized MLP-CNN model, detailed in Table 2 and Figure 7, showcases its effectiveness in detecting DDoS attacks, a critical concern within network security, especially in SDN environments. Our assessment covers two distinctive datasets, CICDDoS2019 and InSDN, providing insights into attack patterns in conventional and SDN infrastructures. The chosen performance metrics offer a comprehensive overview of the model's accuracy and efficacy, and our model demonstrates remarkable prowess in identifying malicious network traffic.

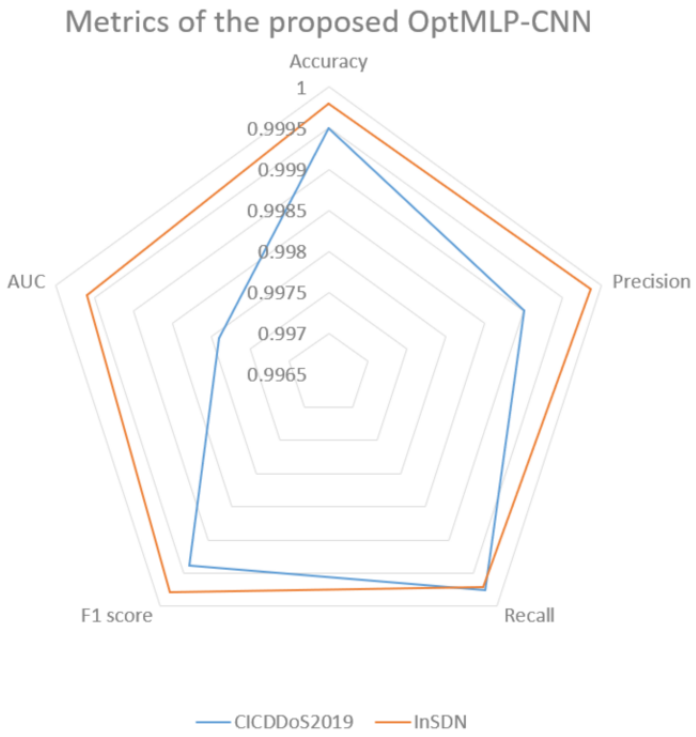

Figure 7. Metrics of the OptMLP-CNN.

Table 2. OptMLP-CNN evaluation.

Dataset	Accuracy	Precision	Recall	F1-Score	AUC
CICDDoS-2019	0.999504	0.999009	0.999752	0.999381	0.997901
InSDN	0.999802	0.999858	0.999717	0.999787	0.999601

For the CICDDoS2019 dataset, the model's accuracy score 0.999504 signifies its exceptional precision in predicting class labels with minimal classification errors. With a precision score of 0.999009, the model accurately identifies DDoS attacks while maintaining an impressively low rate of false positives, crucial for minimizing false alarms and ensuring accurate attack predictions. Its recall score of 0.999752, representing the true positive rate, showcases the model's efficiency in capturing genuine DDoS attacks while demonstrating very few false negatives.

Moreover, the F1-score of 0.999381, a balance between precision and recall, highlights the model's ability to classify attacks while minimizing false negatives accurately. This underscores its capability to achieve high accuracy while effectively managing false alarms. Additionally, the AUC score of 0.997901, observed in the AUC-ROC, emphasizes the model's robust discriminatory power in effectively discerning benign and malicious network traffic.

The model maintains its exceptional performance by transitioning to the InSDN dataset, which reflects the unique landscape of SDN network environments. With an accuracy score of 0.999802, the model demonstrates its efficacy in classifying network

traffic, even within the specialized realm of SDN. The precision score is equally impressive at 0.999858, highlighting the model's ability to classify instances as DDoS attacks within the SDN context accurately. Furthermore, the recall score of 0.999717 signifies the model's capability to identify most actual DDoS attacks in SDN networks while minimizing false negatives. The F1-score of 0.999787 strikes an optimal balance between precision and recall, reinforcing the model's robust performance in DDoS-attack detection. The AUC score of 0.999601 signifies its discriminatory ability within SDN environments. Table 3 provides a comprehensive assessment of various ML techniques, including DNN, ELM, GB, and our novel OptMLP-CNN, in their capacity to detect DDoS activities across different datasets, namely NSL-KDD, UNSW_NB-15, CIC-IDS-2017, CICDDoS-2019, and InSDN. The findings reveal that all models, both the well-established benchmark models [9,13,24] and the proposed methods, exhibit a commendable performance across various evaluation metrics, including accuracy, precision, recall, and F1-score.

Table 3. Comparison with previous studies.

Ref	Used Dataset	Accuracy	Precision	Recall	F1-Score	AUC	Approach
Thakkar, A. and Lohiya, R. [9]	NSL-KDD UNSW_NB-15 CIC-IDS-2017	99.84% 89.03% 99.80%	99.94% 95.00% 99.85%	98.81% 98.95% 99.94%	99.37% 96.93% 99.89%	NS NS NS	Deep neural network with fusion of statistical importance by using standard deviation
Batchu, R. K. and Seetha, H. [13]	CICDDoS-2019	99.94%	99.88%	99.99%	99.94%	99.94%	Extreme ML with hybrid feature selection
Batchu, R. K. and Seetha, H. [24]	CICDDoS-2019	99.97%	98.90%	99.99%	99.44%	99.97%	Gradient boosting with hybrid feature selection based on Spearman's correlation and RF
Proposed Model	CICDDoS-2019 InSDN	99.95% 99.98%	99.90% 99.99%	99.98% 99.97%	99.94% 99.98%	99.79% 99.96%	Optimized MLP-CNN with SHAP-feature selection

Notably, the OptMLP-CNN model achieves the highest accuracy of 99.98% when employed with the InSDN dataset, underscoring its proficiency in accurately classifying network traffic. Precision, a crucial metric for minimizing false positives, is also notably high across all models. Among the benchmark models, model [13] exhibits the highest precision at 99.88%, while the proposed OptMLP-CNN model, when applied to the InSDN dataset, boasts the highest precision at 99.99%. This indicates the model's ability to minimize misclassifications of benign traffic as DDoS attacks.

The recall metric, which measures the capacity to capture actual DDoS attacks effectively, is quite impressive across the benchmark models, ranging from 98.81% [9] to 99.99% [13]. Furthermore, the proposed OptMLP-CNN, when utilized with the CICDDoS-2019 dataset, demonstrates a recall of 99.98%, while it maintains a recall of 99.97% when applied to the InSDN dataset. This highlights the model's ability to detect the most genuine DDoS attacks while keeping false negatives minimal. The F1-score, a harmonious metric considering precision and recall, ranges from 99.37% [9] to 99.94% [13] in the benchmark models. In the case of the OptMLP-CNN model, the F1-score attains 99.94% when employed with the CICDDoS-2019 dataset and achieves an even higher F1-score of 99.98% when operating within the InSDN dataset.

It is important to note that the AUC, representing the classifier's discriminatory ability, was reported for all models except model [9]. The AUC values for the OptMLP-CNN model are notable, reaching 99.79% with the CICDDoS-2019 dataset and an impressive 99.96% within the InSDN dataset.

The presented results indicate that the proposed OptMLP-CNN model consistently exhibits a remarkable performance across diverse datasets, often outperforming or at least matching the capabilities of other established techniques. These findings attest to the model's robustness and potential to bolster network security against the evolving landscape of DDoS threats, with its most exceptional performance observed in the InSDN dataset.

The proposed model leverages a combined MLP-CNN architecture, amalgamating the strengths of both models. This fusion enables the model to capture intricate patterns in structured and spatial data, significantly enhancing its adaptability and accuracy across

diverse datasets. Additionally, our approach incorporates SHAP-feature selection, enabling the identification of critical features pivotal for accurate classification, particularly in DDoS-attack detection.

Furthermore, the model integrates Bayesian optimization techniques, refining hyperparameters to improve efficiency and precision. This iterative optimization process enhances the model performance and contributes to its adaptability to varying data distributions, ensuring robustness across evolving threat landscapes.

Moreover, the continuous monitoring and update mechanisms embedded within our method facilitate real-time adaptation to emerging threats, offering a proactive defense against DDoS attacks. The system's agility in detecting and responding to new attack patterns is a preventive measure, contributing to a more-resilient network infrastructure.

By underscoring these advantages, our manuscript now prominently emphasizes our proposed method's unique strengths and benefits, showcasing its potential to significantly improve DDoS-attack-detection strategies.

6. Conclusions and Future Work

In conclusion, this study introduced and evaluated the OptMLP-CNN model for detecting DDoS attacks in SDN environments. The model combines the innovative SHAP-feature-selection method to identify the most crucial features and a hybrid DL technique based on MLP and CNN architectures. This combination allows the model to efficiently process and analyze network traffic data, effectively identifying anomalies and DDoS attacks. The OptMLP-CNN model was optimized by using a Bayesian optimizer for hyperparameter tuning and the ADAM optimizer, which adapts its learning rates during training, enhancing the model's convergence and overall performance. The evaluation results indicate that the OptMLP-CNN model consistently achieves high accuracy, precision, recall, F1-score, and AUC across various datasets, showcasing its effectiveness and adaptability in enhancing network security within the dynamic realm of SDN environments. Future work in this domain may explore further model refinement, scalability, real-time adaptation, and expansion into broader cybersecurity applications, leveraging the OptMLP-CNN model's potential to address emerging security challenges effectively.

Author Contributions: M.A.S. contributed to the research's central idea, including conceptualization, methodology, formal analysis, writing the original draft, reviewing and editing it, as well as handling correspondence. M.F. contributed to the methodology, supervision, project administration, validation, visualization, and written work review and editing. B.L.Y.A. participated in the investigation, conducted a formal analysis, and contributed to reviewing and editing. Z.E.A.B. conducted a formal analysis and contributed to reviewing and editing. All authors have thoroughly read and given their approval of the final manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Two publicly available datasets were analyzed in this study: InSDN [36] and CICDDoS-2019 [37].

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

Abbreviations

The following abbreviations are used in this manuscript:

ADAM	Adaptive Moment Estimation
ANN	Artificial neural network
ANOVA	ANalysis Of VAriance
BT	Bagging Tree
CNN	Convolutional Neural Network
DA	Discriminant Analysis
DDoS	Distributed Denial of Service
DL	Deep Learning
DNN	Deep neural network
DT	Decision tree
EI	Expected Improvement
FNN	Feedforward neural network
GB	Gradient boosting
GLM	Generalized Linear Model
GNB	Gaussian Naive Bayes
GP	Gaussian process
IG	Information Gain
IoT	Internet of Things
KNN	K-nearest neighbor
LR	Logistic regression
LSTM	Long Short-Term Memory
ML	Machine learning
MLP	Multilayer Perceptron
MLP-GA	Multilayer Perceptron-Genetic Algorithms
MRMR	Maximum Relevance Minimum Redundancy
NB	Naive Bayes
NS	Not Specified
RBF network	Radial Basis Function
ReLU	Rectified Linear Unit
RF	Random Forest
SDIoT	Software-Defined Internet of Things
SDN	Software-Defined Networking
SGD	Stochastic Gradient Descent
SHAP	SHapley Additive exPlanations
SVM	Support vector machine
tanh	Hyperbolic tangent function
VANET	Vehicular Ad Hoc Networking
WSN	Wireless Sensor Network
XGBoost	Extreme gradient boosting

References

1. Ali, T.E.; Chong, Y.W.; Manickam, S. Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review. *Appl. Sci.* **2023**, *13*, 3183. [CrossRef]
2. Karnani, S.; Agrawal, N.; Kumar, R. A comprehensive survey on low-rate and high-rate DDoS defense approaches in SDN: Taxonomy, research challenges, and opportunities. *Multimed. Tools Appl.* **2023**, 1–54. [CrossRef]
3. Setitra, M.A.; Benkhaddra, I.; Bensalem, Z.E.A.; Fan, M. Feature Modeling and Dimensionality Reduction to Improve ML-Based DDoS Detection Systems in SDN Environment. In Proceedings of the 2022 19th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 16–18 December 2022; pp. 1–7.
4. Setitra, M.A.; Fan, M.; Bensalem, Z.E.A. An efficient approach to detect distributed denial of service attacks for software defined internet of things combining autoencoder and extreme gradient boosting with feature selection and hyperparameter tuning optimization. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4827. [CrossRef]
5. Benkhaddra, I.; Kumar, A.; Setitra, M.A.; Bensalem, Z.E.A.; Lei, H. Prevention of DDoS attacks using an optimized deep learning approach in blockchain technology. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4729.
6. Rashid, M.M.; Khan, S.U.; Eusufzai, F.; Redwan, M.A.; Sabuj, S.R.; Elsharief, M. A Federated Learning-Based Approach for Improving Intrusion Detection in Industrial Internet of Things Networks. *Network* **2023**, *3*, 158–179. [CrossRef]

7. Fox, G.; Boppana, R.V. Detection of Malicious Network Flows with Low Preprocessing Overhead. *Network* **2022**, *2*, 628–642. [CrossRef]
8. Shieh, C.S.; Nguyen, T.T.; Horng, M.F. Detection of Unknown DDoS Attack Using Convolutional Neural Networks Featuring Geometrical Metric. *Mathematics* **2023**, *11*, 2145. [CrossRef]
9. Thakkar, A.; Lohiya, R. Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System. *Inf. Fusion* **2023**, *90*, 353–363. [CrossRef]
10. Saha, S.; Priyoti, A.T.; Sharma, A.; Haque, A. Towards an Optimized Ensemble Feature Selection for DDoS Detection Using Both Supervised and Unsupervised Method. *Sensors* **2022**, *22*, 9144. [CrossRef]
11. Türkoğlu, M.; Polat, H.; Koçak, C.; Polat, O. Recognition of DDoS Attacks on SD-VANET Based on Combination of Hyperparameter Optimization and Feature Selection. *Expert Syst. Appl.* **2022**, *203*, 117500. [CrossRef]
12. Habib, B.; Khursheed, F. Performance evaluation of machine learning models for distributed denial of service attack detection using improved feature selection and hyper-parameter optimization techniques. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e7299. [CrossRef]
13. Batchu, R.K.; Seetha, H. On Improving the Performance of DDoS attack detection system. *Microprocess. Microsyst.* **2022**, *93*, 104571. [CrossRef]
14. Wang, S.; Balarezo, J.F.; Chavez, K.G.; Al-Hourani, A.; Kandeepan, S.; Asghar, M.R.; Russello, G. Detecting flooding DDoS attacks in software defined networks using supervised learning techniques. *Eng. Sci. Technol. Int. J.* **2022**, *35*, 101176. [CrossRef]
15. Batchu, R.K.; Seetha, H. An integrated approach explaining the detection of distributed denial of service attacks. *Comput. Netw.* **2022**, *216*, 109269. [CrossRef]
16. Chanu, U.S.; Singh, K.J.; Chanu, Y.J. An ensemble method for feature selection and an integrated approach for mitigation of distributed denial of service attacks. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6919. [CrossRef]
17. Kshirsagar, D.; Kumar, S. A feature reduction based reflected and exploited DDoS attacks detection system. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *1-13*, 393–405. [CrossRef]
18. El Sayed, M.S.; Le-Khac, N.A.; Azer, M.A.; Jurcut, A.D. A Flow-Based Anomaly Detection Approach With Feature Selection Method Against DDoS Attacks in SDNs. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 1862–1880. [CrossRef]
19. Akgun, D.; Hizal, S.; Cavusoglu, U. A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. *Comput. Secur.* **2022**, *118*, 102748. [CrossRef]
20. Zhou, L.; Zhu, Y.; Zong, T.; Xiang, Y. A feature selection-based method for DDoS attack flow classification. *Future Gener. Comput. Syst.* **2022**, *132*, 67–79. [CrossRef]
21. Saha, S.; Priyoti, A.T.; Sharma, A.; Haque, A. Towards an Optimal Feature Selection Method for AI-Based DDoS Detection System. In Proceedings of the 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2022; pp. 425–428.
22. Fenil, E.; Kumar, P.M. Towards a secure Software Defined Network with Adaptive Mitigation of DDoS attacks by Machine Learning Approaches. In Proceedings of the 2022 IEEE International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 28–29 January 2022; pp. 1–13.
23. Golchin, P.; Kundel, R.; Steuer, T.; Hark, R.; Steinmetz, R. Improving DDoS Attack Detection Leveraging a Multi-aspect Ensemble Feature Selection. In Proceedings of the NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 25–29 April 2022; pp. 1–5.
24. Batchu, R.K.; Seetha, H. A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning. *Comput. Netw.* **2022**, *200*, 108498. [CrossRef]
25. Bindra, N.; Sood, M. Evaluating the impact of feature selection methods on the performance of the machine learning models in detecting DDoS attacks. *Rom. J. Inf. Sci. Technol.* **2020**, *23*, 250–261.
26. Polat, H.; Polat, O.; Cetin, A. Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability* **2020**, *12*, 1035. [CrossRef]
27. Zaki, F.A.M.; Chin, T.S. FWFS: Selecting robust features towards reliable and stable traffic classifier in SDN. *IEEE Access* **2019**, *7*, 166011–166020. [CrossRef]
28. Cauteruccio, F.; Fortino, G.; Guerrieri, A.; Liotta, A.; Mocanu, D.C.; Perra, C.; Terracina, G.; Vega, M.T. Short-long term anomaly detection in wireless sensor networks based on machine learning and multi-parameterized edit distance. *Inf. Fusion* **2019**, *52*, 13–30. [CrossRef]
29. Setitra, I.; Iwahori, Y.; Meziane, A. Walking cycle and walking phases extraction from videos using transfer learning. *Procedia Comput. Sci.* **2020**, *176*, 2695–2704. [CrossRef]
30. González-Nóvoa, J.A.; Busto, L.; Campanioni, S.; Fariña, J.; Rodríguez-Andina, J.J.; Vila, D.; Veiga, C. Two-step approach for occupancy estimation in intensive care units based on Bayesian optimization techniques. *Sensors* **2023**, *23*, 1162. [CrossRef]
31. Hassan, E.; Shams, M.Y.; Hikal, N.A.; Elmougy, S. The effect of choosing optimizer algorithms to improve computer vision tasks: A comparative study. *Multimed. Tools Appl.* **2023**, *82*, 16591–16633. [CrossRef] [PubMed]
32. Taud, H.; Mas, J.F. Multilayer perceptron (MLP). In *Geomatic Approaches for Modeling Land Change Scenarios*; Springer: Cham, Switzerland, 2018; pp. 451–455.
33. Desai, M.; Shah, M. An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN). *Clin. eHealth* **2021**, *4*, 1–11. [CrossRef]

34. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [CrossRef]
35. Benkhaddra, I.; Kumar, A.; Setitra, M.A.; Hang, L. Design and Development of Consensus Activation Function Enabled Neural Network-Based Smart Healthcare Using BIoT. *Wirel. Pers. Commun.* **2023**, *130*, 1549–1574. [CrossRef]
36. Elsayed, M.S.; Le-Khac, N.A.; Jurcut, A.D. InSDN: A novel SDN intrusion dataset. *IEEE Access* **2020**, *8*, 165263–165284. [CrossRef]
37. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; pp. 1–8.