computers

MDPI

Article

Explainable AI-Based DDOS Attack Identification Method for IoT Networks

Chathuranga Sampath Kalutharage *,†, Xiaodong Liu †, Christos Chrysoulas †, Nikolaos Pitropakis *,† and Pavlos Papadopoulos †

School of Computing, Engineering & the Build Environment, Edinburgh Napier University, Edinburgh EH10 5DT, UK
* Correspondence: c.kalutharage@napier.ac.uk (C.S.K.); n.pitropakis@napier.ac.uk (N.P.)
† These authors contributed equally to this work.

Abstract: The modern digitized world is mainly dependent on online services. The availability of online systems continues to be seriously challenged by distributed denial of service (DDoS) attacks. The challenge in mitigating attacks is not limited to identifying DDoS attacks when they happen, but also identifying the streams of attacks. However, existing attack detection methods cannot accurately and efficiently detect DDoS attacks. To this end, we propose an explainable artificial intelligence (XAI)-based novel method to identify DDoS attacks. This method detects abnormal behaviours of network traffic flows by analysing the traffic at the network layer. Moreover, it chooses the most influential features for each anomalous instance with influence weight and then sets a threshold value for each feature. Hence, this DDoS attack detection method defines security policies based on each feature threshold value for application-layer-based, volumetric-based, and transport control protocol (TCP) state-exhaustion-based features. Since the proposed method is based on layer three traffic, it can identify DDoS attacks on both Internet of Things (IoT) and traditional networks. Extensive experiments were performed on the University of Sannio, Benevento Instrution Detection System (USB-IDS) dataset, which consists of different types of DDoS attacks to test the performance of the proposed solution. The results of the comparison show that the proposed method provides greater detection accuracy and attack certainty than the state-of-the-art methods.

Keywords: explainable AI; DDoS attack; IoT network; feature influence; anomaly detection; supervised learning

check for updates

Citation: Kalutharage, C.S.; Liu, X.; Chrysoulas, C.; Pitropakis, N.; Papadopoulos, P. Explainable AI-Based DDOS Attack Identification Method for IoT Networks. *Computers* **2023**, *12*, 32. https://doi.org/10.3390/computers12020032

Academic Editor: Paolo Bellavista

Received: 1 January 2023
Revised: 27 January 2023
Accepted: 1 February 2023
Published: 3 February 2023

1. Introduction

In the current era of rapid development of the Internet, the services available are expanding, and the Internet is now inseparable from all aspects of modern life. As a result of this trend, users rely more and more on the Internet for everything from travel to online shopping. By 2025, there will be over 21 billion Internet of Things devices [1]. However, despite the rapid and thorough development of the Internet, online threats are still around and constantly changing. Internet-connected applications have been targeted due to the wide range of purposes for which they are used. Common types of attacks are DDoS attacks, cross-site scripting attacks, and request forgery attacks [2]. DDoS attacks pose a serious risk to the availability and reliability of Internet services. A successful DDoS attack uses malicious traffic from multiple sources and attempts to exhaust an online service's resources and prevent regular users from accessing it. According to a North American service provider in 2018, the amount of DDoS attacks has increased at an alarming rate. The largest known attack has a target of 1.7 Tbps, and 400 Gbps attacks are already commonplace [3]. Most DDoS attacks since 2016 have been caused by the Mirai botnet. Mirai attacks a large number of Internet of Things devices, primarily older routers and closed-circuit video systems. By injecting traffic into DNS providers, Mirai targeted numerous well-known websites, including Shopify, SoundCloud, Twitter, and Netflix. GitHub suffered its worst DDoS attack ever in March 2018 with 1.35 Tbps of peak traffic. This attack suddenly stopped network services and caused significant financial losses. DDoS attacks are considered the greatest threat to the stability of the entire Internet and the operation of individual companies and organizations. DDoS attacks can come in a variety of ways, ranging from mild to severe [4]. It is also difficult to defend against the wide range of attack types, including volumetric attacks, TCP state-exhaustion attacks, and application-layer attacks that target multiple vulnerabilities in a victim [5]. Various security systems have been developed to detect DDoS attacks [6–8].

A number of IDS have been presented in the literature [9]. They can be categorized into three types, signature-based, artificial intelligence-based, and hybrid IDS. Signature-based IDS are incapable of detecting zero-day attacks, since they only analyze traffic for predefined attack patterns. However, IDSs based on artificial intelligence are becoming more and more attractive and continue to display exceptional performance in detecting attacks due to their ability to identify unseen attacks. The majority of proposed AI-based approaches are supervised learning techniques that require labeled training data showing both malicious and benign behaviors and absolute ground truth. However, obtaining labeled attack data is costly, and legal, ethical, and privacy concerns may prevent the sharing of realistic data within research communities. Therefore, anomaly-based detection methods are recommended for implementation in the security industry, as these models may be trained using only benign data. Two strategies have been proposed to identify attack streams: classification based on payload inspection and classification based on machine learning [10,11]. Payload inspection depends on analyzing packet payloads (e.g., the content of an HTTP message) to detect the attributes of an attack [10]. Due to the small amount of information in the packet payloads, the classification performance of attacks targeting protocol vulnerabilities such as synchronization packets (SYN) flooding and Internet Control Message Protocol packets (ICMP) flooding is poor. In addition, this method causes privacy issues as it inspects the packet content. Existing machine-learning-based methods are time-consuming because they require a large number of training features (e.g., the long short-term memory (LSTM) network [11]). Even after the model has been trained offline, the identification phase's significant parameter adjustments are expensive, especially while a victim is being attacked, and IoT devices have resource limits.

However, artificial intelligence-based IDS can detect completely unknown attacks by detecting anomalies associated with high network latency, traffic on unusual ports, large network volume, etc. Thus, here we try to detect the attack traffic flow in the network level when an attacker performs an attack. Therefore, the proposed method can effectively identify DDoS attacks. Compared to the current state of the art, the proposed method does not depend on the packet payload and will result in protecting user privacy. The proposed method significantly reduces the time required for attack detection and provides more accurate results. In addition, this method is based on explainable AI, which provides a better explanation of the anomalous behavior with the highest influencing features. Additionally, this will provide attack certainty for the detected anomalies. We need to address the following challenges to achieve this goal. First, we need to find a method to detect anomalies and explain it and then to find the highly influenced features. To solve this problem, we have developed a combined auto-encoder and XAI model for anomaly detection and determining the most influential features and their influence. The next challenge is identifying the most significant features which can distinguish DDoS attack flows from benign flows. To address this challenge, we extract features in which the attack flows exhibit anomalies from three categories: based on application layer, based on volumetric, and based on TCP state exhaustion. Then, we define a threshold for each feature. After that, we can map the most informative features with the most influential features. Then, we find DDoS attacks based on common features (to informative and most influential) that exceed the threshold. In particular, the main contributions of this paper are the following:

- We propose and implemented a novel method that consists of two key components: anomaly detection using autoencoder and XAI-based explanation of the most influential features for each anomalous instance.
- We suggest a method for selecting features for DDoS attack flow detection. By deciding which features are independent and most important for a DDoS attack, the methodology can reduce the amount of features.
- We present a comprehensive evaluation of the proposed method on the USB-IDS dataset and implemented a lightweight model, as it needs to deploy on IoT devices.

The rest of the paper is structured as follows: Section 2 describes background and related work. The proposed method description is given in Section 3. Section 4 describes the experimental setup, and Section 5 discusses the results obtained using the USB-IDs benchmark dataset. Finally, Section 6 concludes the paper.

2. Background and Related Work

The purpose of this research was to combine deep autoencoders with XAI to produce a DDoS attack identification mechanism for IoT networks. The work related to each area is therefore covered separately in this section.

2.1. Explainable Artificial Intelligence

An XAI system tries to describe its behavior to make it more understandable to humans. Several XAI concepts can be applied to create AI systems that are more effective and understandable for humans [12]. The XAI model must be able to explain its concepts and capabilities, what it has achieved, what it is doing, and what will happen next. It must also be able to disclose the critical details on which it bases its actions. In the past few decades, a number of ML-based IDSs have been presented to protect IoT networks from malicious attackers and have shown excellent performance [13]. However, these sophisticated AI-based solutions are usually considered "black box models" and are difficult for end users to interpret. A single wrong prediction in security exposes systems and networks to significant cyber risks. Consequently, AI-based security mechanisms should improve with XAI.

2.2. SHapley Additive Explanation

The Shapley additive explanation (SHAP) methodology integrates previously proposed explanation techniques, such as local interpretable model-agnostic explanations (LIME) and deep learning important features (DeepLIFT), within the domain of additive feature attribution techniques [14]. Shapley values are utilized by SHAP to explain a particular prediction. The values of Shaply are obtained from game theory:

- Local accuracy;
- Missingnes;
- Consistency.

The SHAP framework proposes Kernel SHAP, a model-independent approximation for SHAP values. Kernel SHAP builds a local explanation model using linear LIME and Shapley values. The local explanation model is built on a background set and a sample of the potential coalition of data items, and it uses weighted linear regression.

2.3. DDoS Attack

A DDoS attack uses multiple sources to attack the target system, as shown in Figure 1, making it difficult to identify illegitimate IP addresses making a request to the system. A DDoS attack consists of several steps. The attacker begins by acquiring multiple agent machines. This process is usually carried out automatically by checking remote workstations for vulnerabilities that could allow subversion. The vulnerability is then exploited to gain access to the recruited machines and infect them with malicious code. Computers that have been infected can be used to continue recruiting new agents in the exploit/infect process, which is typically automated [15].

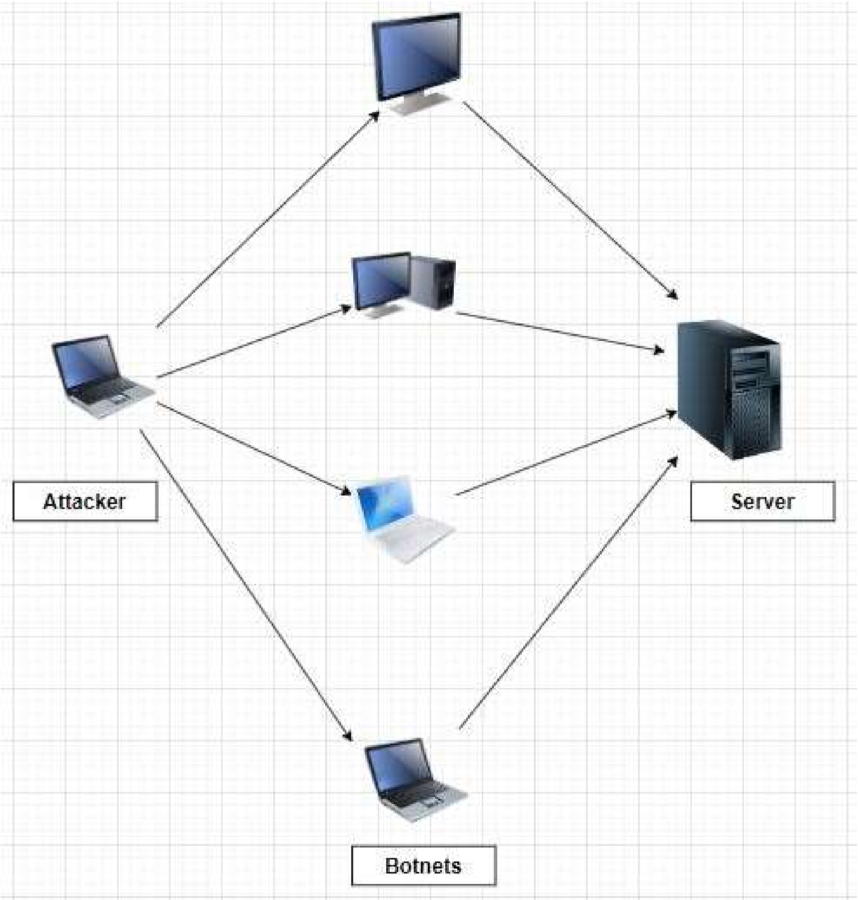

Figure 1. Visual representation of a DDoS attack. The attacker can send a large number of connection requests using a number of PCs in a botnet.

2.4. DDoS Detection

Extensive research has been conducted on DDoS attack detection, particularly anomaly detection and machine learning-based detection. A DDoS attack can be identified when an abnormal deviation in a given feature exceeds a threshold, which is the purpose of anomaly detection [16]. Yang Xiang et al. presented an extended entropy-based method to recognize a low-rate DDos attack. On the basis of the difference between the probability distributions of regular traffic and low-rate traffic, traffic is categorized as a low-rate attack if the normalized entropy decreases [17]. Bhuvaneswari et al. presented a method for anomaly detection in IoT. For learning, they applied the VCDL (vector convolutional deep learning) approach. The convolutional neural networks used in the VCDL models have two levels: PL (pooling layer) and CL (convolutional layer). There are two modules included in it: VCN (vector convolutional network) and FCN (fully connected network). The first module's purpose is to extract the features. The function of the second module is to learn the extracted features to identify the type of IoT traffic. The numbers of PLs, CLs, and HLs (hidden layers) in the FCN and nodes in the HLs are the hyperparameters used to build the VCDL structure. The BoT-IoT dataset was used to evaluate the deployed solution [18].

Liu et al. presented a multi-layer security model to defend DDoS attacks. Amplification-based DDoS attacks are specifically stopped by the flood limiter layer. Users could impose their own traffic management policies due to the per-user layer protecting against DDoS attacks. This method was implemented on Linux and was able to handle large-scale attack flows [1]. Jing Zheng et al. presented RADAR, a method based on adaptive correlation analysis, to detect and prevent SYN flooding attacks. A SYN flooding attack is discovered by the approach when the ratio of anomalous SYN packets to ACK packets reaches a particular threshold [19]. Khraisat et al. developed the hybrid intrusion detection system (HIDS), which combines a signature-based intrusion detection system (SIDS) and an anomaly-based intrusion detection system (AIDS). The C5 decision tree classifier is used by SIDS as the initial stage of attack detection. Pattern matching is used to handle unidentified traffic. The request will be sent to AIDS if it is not detected as an attack. A single-class SVM is used to train AIDS, which only learns the attributes of normal packets. It does not receive instruction from other classes. As a result, the AIDS assumes that any anomalies that fall outside of the usual will be recognized as zero-day attacks. For improving prediction accuracy, ensemble methods are used at the third stage [20]. Nagarathna Ravi et al. proposed a learning-based detection strategy to identify and mitigate DDoS in IoT. To detect and reduce traffic from DDoS attacks, they employed a semi-supervised machine learning method [21]. Xiaoyong Yuan et al. suggested a method for identifying DDoS attacks based on deep learning. They designed a bi-directional recurrent deep neural network to identify DDoS attacks and learn patterns [22]. Cagatay Ates et al. presented a method for detecting DDoS attacks based on the support vector machine (SVM) and community clustering. Two metrics were used for analysis, namely, modularity and normalized entropy [23]. Mengmeng Ge et al. presented a method for detecting DOS, DDOS, and reconnaissance attacks based on the FFNN (feed-forward neural network). The research focuses mostly on Message Queuing Telemetry Transport (MQTT), an Internet of Things (IoT) communication protocol that performs publish–subscribe functionalities between IoT devices and centralized brokers or base stations [24]. Gaganjot Kaur et al. proposed a new hybrid approach to detecting DDoS attacks in software define networks (SDN). They used a support vector machine (SVM) and artificial neural networks (ANNs). The system achieved a higher accuracy than the KNN algorithm [25]. A stacked autoencoder (SAE) model, presented by Raja Majid and team, is a deep-learning-based model that can efficiently detect DDoS attacks. Adaptive polling sampling was utilized to classify benign and malicious traffic with the SAE model after processing traffic samples [26]. Saif ur Rehman et al. evaluated a number of classification techniques. They employed gated recurrent units (GRU), recurrent neural networks (RNN), and Naive Bayes (NB) algorithms to detect DDoS attacks, and determined that gated recurrent units provide the most accurate detection [27]. Jie Cui et al. presented a novel method for DDoS attack detection based on cognitive inspired computing. The mechanism chose a few features and applied the SVM to distinguish between legitimate and malicious traffic [28]. Lu Zhou et al. proposed a technique for categorizing DDoS attack flow based on feature selection. They classified based on the area under the roc curve (AUC) [29].

However, since the purpose of these methods is to determine if a DDoS attack is in progress, they are unable to identify the attackers or distinguish attack traffic from regular traffic. In addition, a number of the approaches rely heavily on the analysis of packet content, such as HTTP content, to determine the number of different messages, which creates a privacy concern. Although AI-based methods have achieved great detection accuracy, it is difficult to understand and interpret why they work so effectively, and we cannot guarantee the decision that the AI model will make due to its black-box nature. Consequently, evaluating anticipated results in terms of feature contribution using feature-based impact analysis might boost the AI model's decision confidence. However, none of these works focus on improving the DDoS attack detection systems using the explanations of XAI tools.

3. Methodology

We propose an XAI-based method to identify DDoS attacks based on feature influence and mainly on explanation of unsupervised learning due to the lack of realistic attack data for the supervised model. In this section, we present an attack detection method and an explanation for the detected attacks. According to the engineering pipeline, we first extract the features with the CIC flow meter from the bi-directional network traffic. Then, we detect anomalies. Third, we explain the detected anomalies and find the most influential features. Then, as the purpose of the proposed research is to detect DDoS attacks, we find the DDoS-attack-related feature set. Finally, we map the most influential features to the DDoS-related features. Then, we identify DDoS attacks separately from each detected attacks. Figure 2 shows the main steps of the engineering pipeline of the proposed method.

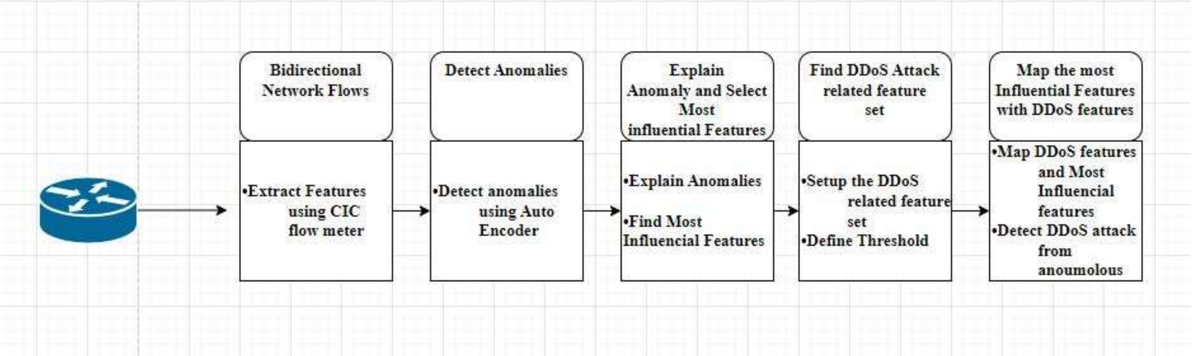
Bidirectional Network Flows
•Extract Features using CIC flow meter
Detect Anomalies
•Detect anomalies using Auto Encoder
Explain Anomaly and Select Most influential Features
•Explain Anomalies
•Find Most Influencial Features
Find DDoS Attack related feature set
•Setup the DDoS related feature set
•Define Threshold
Map the most Influential Features with DDoS features
•Map DDoS features and Most Influencial features
•Detect DDoS attack from anoumolous

Figure 2. The overview of the engineering pipeline.

To detect the DDoS attacks, we must first identify the attack flow over the benign flows. A flow is a collection of packets that all contain the same five pieces of data: source IP address, destination IP address, source port number, destination port number, and protocol [30]. Here, presuming that there is N number of flow samples and y classes, Equation (1) denotes X flow sample. where f_i is the ith flow, d represents the original features, and N is the number of flows. The true label of flow f_i is denoted as $y_i = 0, 1$. Our first goal is to develop a method to predict a label $y_pred(i)$ that is exactly the true label y_i.

$$X = [F_1, F_2, \dots, F_N] \epsilon R^d N \tag{1}$$

3.1. Feature Extraction from Network Traffic

The potentially useful features are extracted from the packets of each flow. We used the CIC flow meter to extract features from the traffic flow. CICFlowMeter is a tool distributed to create 84 different types of network traffic features. It reads a pcap file, extracts the features, and creates a report with visuals and a CSV file [31]. We extracted bidirectional statistical characteristics from network traffic. Min-max normalization was applied to all features [32].

3.2. Anomaly Detection

Recent research has placed significant emphasis on anomaly-based intrusion detection security systems, as these methods outperform signature and rule-based detection approaches to detect unknown attacks [33]. Therefore, traffic-flow-based anomaly detection is used for software define networks (SDN) in intrusion detection [34]. Unlike most DDoS detection methods that use supervised approaches, we employed autoencoder models to identify anomalies. To achieve this, we used an autoencoder to identify anomalies based on the reconstruction error (anomaly score). We define an anomaly score as the difference between the input value and the (reconstructed) output value. Equation (2) [35] shows the reconstruction error calculation in our work. Given an input row (A) with an array of features (ai) and its output row (A') with reconstructed feature values (ai'), and employing an anomaly detection model (f), the sum of the reconstruction errors for each feature that is specific to a certain row produces the reconstruction error for that row. If the reconstruction error exceeds the input value, it is identified as an anomaly.

$$L(A, A') = \sum_{i=1}^{n} (a_i - a_i')^2 \tag{2}$$

3.3. Explain Anomalies

Then, to identify the top-R features that include a set of selected features for which the total associated errors define a modifiable percentage of $L(A, A')$, the features in the error list must be rearranged so that $|a1 - a1'| > |an - an'|$. The autoencoder model uses SHAP values to identify which top-R features contributed to each of the significant reconstruction errors. We used Kernel SHAP to obtain the SHAP values of each feature (i.e., a_i) in the list—i.e., the importance of each feature $a_1, a_2, \ldots, a_n$ in predicting the examined feature a_i'. The pseudocode for the process is shown in Algorithm 1.

Algorithm 1 Calculate SHAP values for top-R features.

```
Require: X—Anomaly instance that need to explain, X1..j—instances used by kernel SHAP,
  Reconstruction errorList—a ranked list of errors for each feature, f—autoencoder model
Ensure: shaptopRfeatures—SHAP values for each feature within topRfeatures
  $topRfeatures \leftarrow$top value from Error List
  for each i $\epsilon$ $topRfeatures$ do
     $explainer \leftarrow shap.KernelExplainer(f, X1..j)$
     $shaptopRfeatures[i] \leftarrow explainer.shapvalues(X, i)$
  end for
  return $shaptopRfeatures$
```

3.4. Most Informative Features for DDoS Attacks

Since the goal of a DDoS attacker is to reduce a target's resources, resource exhaustion techniques are generally classified into three types. These are volumetric, TCP state exhaustion, and application layer-based [5].

3.4.1. State Exhaustion Attack Based Features

State-exhaustion attacks typically focus on shutting down the supporting infrastructure and services that deliver content to end users. These attacks attempt to overload TCP state tables while establishing three-way handshakes with spoofed connections, disrupting legitimate users' connections. There are few types of state-exhaustion attacks: SYN flood, TLS/SSL, DNS flood, etc. As an example, an SYN flood attack makes a number of half-open TCP connections by sending SYN packets and keeping the accompanying subsequent ACK to decrease the SYN-queue resources [36]. Therefore, we extracted five features to assess anomalous behavior related to TCP state-exhaustion attacks. The features selected in this study were extracted from the packet header rather than looking at the packet payload, and we selected features listed in Table 1. Figure 3 shows extraction of features in the packet header fields. Since none of these functions depend on packet payload inspection, users' privacy is guaranteed.

Table 1. TCP state-exhaustion attack-based features.

Feature	Details
fwd-TCP-num	amount of TCP packets forwarded
fwd-ACK-num	Amount of ACK flags on forwarded packets
fwd-max-ACK	Maximum ACK interval for forwarded packets
fwd-SYN-num	SYN flag amount for forwarded packets
fwd-SYN-rate	SYN flag transmission rate of forwarded packets

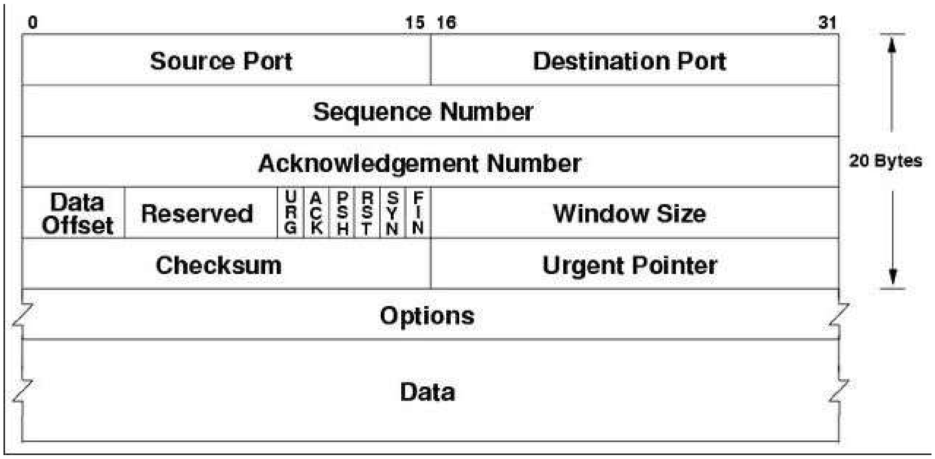

Figure 3. TCP packet header for feature extraction.

3.4.2. Application Layer-Based Attack Features

Application layer DDoS attacks analyze the vulnerability of the attack due to a specific open service port (e.g., flooding attacks) [37]. By establishing a regular connection bypassing firewalls, an attacker could target these open ports and launch flooding attacks. Therefore, a particular port number is used to track traffic by feature, source port, and destination port. Additionally, we have to notice that HTTP (Hyper Text Transfer Protocol), TCP (Transmission Control Protocol), UDP (User Datagram Protocol), and ICMP (Internet Control Message Protocol) are the primary protocols used by the attackers. Application layer-based Features are shown in Table 2.

Table 2. Application-layer-based attack features.

Feature	Details
src-port	Source port
dst-port	Destination port
protocol	protocol

3.4.3. Volumetric-Based Attack Features

The goal of volumetric DDoS attacks is to flood internal networks with a large amount of malicious traffic. Still 42% of attacks are volumetric attacks [1]. Volumetric attacks generate a large number of network layer packets. Based on packet number, packet length, and packet time interval, we extracted the network flows' statistical features and split them into three groups. Extracted features are shown in Table 3. After extracting features from all three categories need to find the threshold value for each feature. To choose the best threshold for each chosen feature, we use a simple threshold-tuning technique. The threshold tuning approach examines a series of thresholds. The optimal threshold can be identified as the one that increases the F1 score as shown in Equation (3).

$$T = arg\ max(F_1) \tag{3}$$

Table 3. Volumetric-based features.

Feature	Details
flow-duration	Flow Duration
tot-fwd-pkts	Total amount of packets forwarded
tot-bwd-pkts	Total Number of back/forward packets
tot-len-bwd-pkts	Total length of backward and forward packets
fwd-pkts-len-min	The minimum forward packet length
Fwd-Packet-Length-Std	Length variation of forward packets
bwd-pkts-len-max	The maximum back-forward packet length
flow-packets/s	Packets transferred per second
bwd-IAT-tot	Total time interval of the back-forward packets
bwd-IAT-mean	Mean time interval of the back-forward packets
fwd-psh-num	Number of packets with the PSH flag in forward packets
Bwd-Packets/s	back-forward packets transferred per second
fwd-ent-int	Time interval entropy of forwarded packets

3.5. Mapping the Most Influential Features with DDoS Features

The final phase of the method is to map the most influential features (top-R features) to DDoS features (most informative features for DDoS attack detection) and find the DDoS attack with the greatest attack certainty. If the most influential features in the SHAP top-R features list are a_1, $a_2 \ldots a_r$ for the detected anomalous instance, it matches them with the DDoS feature list (with 21 features) da_1, $da_2 \ldots da_2 1$ to find most influential DDoS attack features, as shown in Algorithm 2. If the most influential DDoS features exceed the threshold, it will be identified as a DDoS attack with an explanation, which can lead to attack certainty instead of receiving a decision from a typical black-box detection architecture.

Algorithm 2 Finding the most influential DDoS features.

```
Require: shaptopRfeatures, DDoS features
Ensure: Most Influential DDoS features
  for $a_i$ in shaptopRfeatures do
      for $da_i$ in DDoS features do
        if $a_i == da_i$
            Most Influential DDoS features ← $a_i$
      end for
  end for
  return $MostInfluentialDDoSfeatures$
```

4. Experimental Evaluation

4.1. Dataset

For our experimental evaluation, we used the USBIDS [38] dataset because, unlike other potential datasets, it contains explicit feature explanations. It consists of 17 labeled CSV files containing network traffic data. There are 16 files in total, including a benign (unaltered by an attack) flow traffic data file with a combined denial of service (DoS) attack and defense module. The network flows in the dataset were determined using CIC FlowMeter2. The 16 non-normative CSV files' naming pattern assist in identifying the collection context. For instance, HULK-NoDefense.csv lists the flows that were received when HULK was run without any defenses.

4.2. Experimental Environment

We simply used benign data to train the model, and we combined benign data with two sets of attack data to test the model's performance. A fully connected autoencoder model with RELU enabled was used for this. Only 2 hidden layers are used in the network to reduce the weight of the model. The hidden layers each included 10 and 32 neurons. Using benign data, the highest mean squared error (MSE) was selected as the anomalous data threshold. Python, TensorFlow light, and the Keras library were used to implement the proposed algorithm. The Adam optimizer was utilized with 40 epochs and a learning rate of 0.01. Experiments were conducted on a 2.30 GHz Intel Core i7-equipped ASUS ZenBook with 16 GB of RAM and a Raspberry pi model B with 4 GB of RAM.

5. Results and Discussion

In order to select the model with the lightest design and highest performance, we tested a variety of models and evaluated their detection efficiency and accuracy. Among the models tested, the above model showed the best results: Attack Hulk No Defense 0.98, Attack Hulk Evasive 1.0, and Attack Hulk Reqtimeout 1.0. These are superior results compared to the current state of the art: decision tree (DT), 0.97, 0.06, and 0.97; random forest (RF) 0.98, 0.00, and 0.98; deep neural network (DNN) 0.67, 0.05, and 0.66, respectively, for each attack [39], as shown in Table 4. This accuracy comparison was based on the USBIDS dataset, which is a DDoS attack dataset. We used the same dataset for other experiments because the goal of this model is to detect a DDoS attack.

Table 4. Proposed model comparison with the current state of the art (for the HULK attack of USB-IDS dataset) [39].

Detection Method	Hulk No Defense	Hulk Evasive	Hulk Reqtimeout
DT [39]	0.97	0.06	0.97
RF [39]	0.98	0.00	0.98
DNN [39]	0.67	0.05	0.66
Proposed Method	0.98	1.0	1.0

Explainability

Many tools and libraries for opening black box models have been released in recent years. To compare the effectiveness of such algorithms, there are no recognized performance indicators. There is no single explainability technique that is better than the others. Therefore, to further evaluate this model, we need to evaluate the explainability of this model. For this purpose, the explainability of this model in the individual phases must be shown. As a proof of concept, we reduced the model's explainability for five anomalous instances (384574, 602902, 686625, 718029, 124930). If we consider each anomalous instance, the explainability of anomaly detection was as follows. Anomalous instance 384574 in Figure 4, and according to the explainability of instance 384574, forward packets per second (fwd packets/s), flow packets per second (flow packets/s), backward packets per second (bwd packets/s), packet length max, average packet size, and backward packet length standard (bwd packet length std) are the most influential features (features with the highest SHAP values) for the anomalous behavior. Anomalous instance 718029, as shown in Figure 5, showed 10 features as the most influential features: flow packets/s, bwd packets/s, bwd packet length max, packet length max, packet length mean, bwd segment size avg, fwd packet length std, subflow bwd bytes, bwd packet length std, and packet length std. However, the 602902, 686625, 124930 anomalous instances shown in Figures 6–8 show 8, 7, and 4 features as the number of the most influential features for each instance. Thus, there is no standard number of features to find anomalous behavior. It will be one or more, but we can confirm the detected anomaly as an attack or normal anomalous behavior based on explainability.

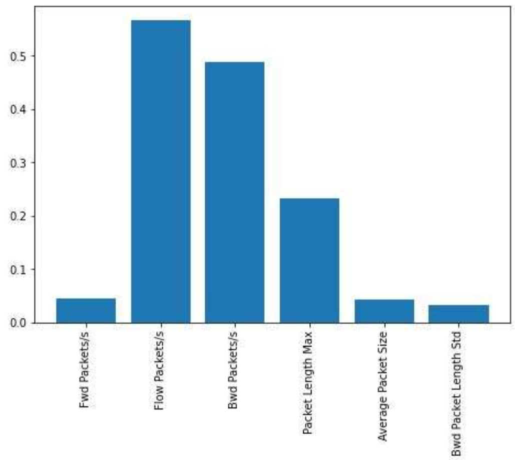

Figure 4. Anomalous instance 384574.

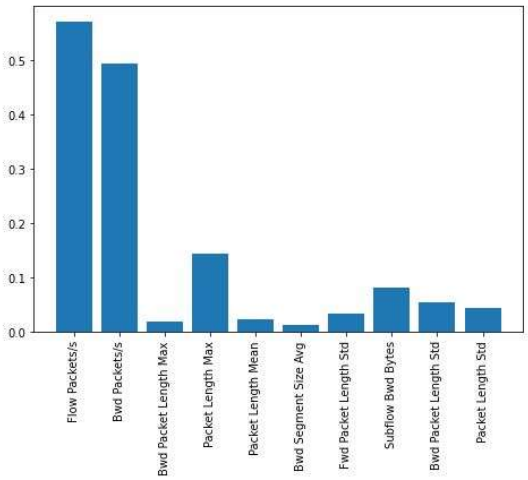

Figure 5. Anomalous instance 718029.

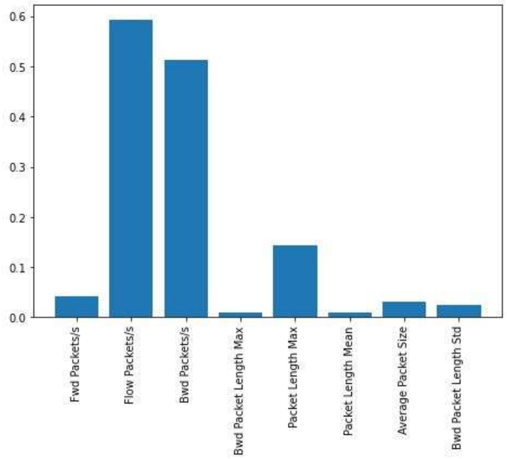

Figure 6. Anomalous instance 602902.

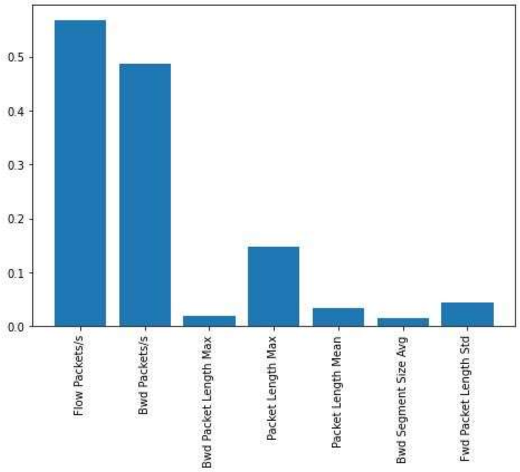

Figure 7. Anomalous instance 686625.

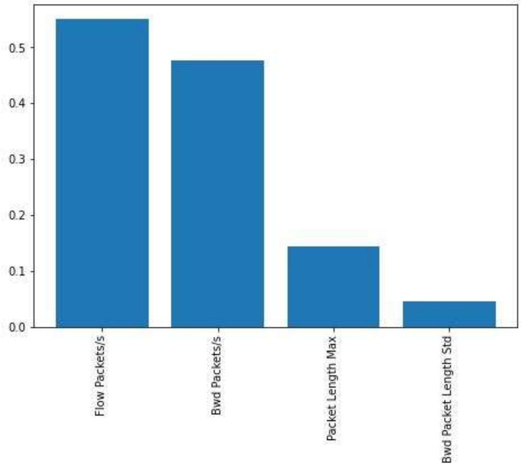

Figure 8. Anomalous instance 124930.

Here we discussed the explainability of the detected anomaly, and it is possible to gain attack certainty based on the cybersecurity domain knowledge. After deploying our model, we need to identify DDoS attacks. Thus, the next phase of the model is to map the most influential features with the the most informative DDoS features we defined earlier. We have defined the 21 most informative features for DDoS attacks, and after mapping these features to the most influential features ($ShaptopRfeatures$), we obtained the flow packets per second (flow packets/s), backward packets per second (Bwd Packets/s), the maximum length of the back-forward packets (bwd packet length maxx), and the length variance of the forward packets (fwd packet length std) as DDoS detection features. Among them, flow packets/s and bwd packets/s are the most influential features for identifying DDoS attacks for the above five anomalous instances. Figure 9 shows the explanation of DDoS attack detection related to feature impact with weights. According to the DDoS attack detection explanation, flow packets/s and bwd packets/s are the features that exceed the DDoS identification threshold.

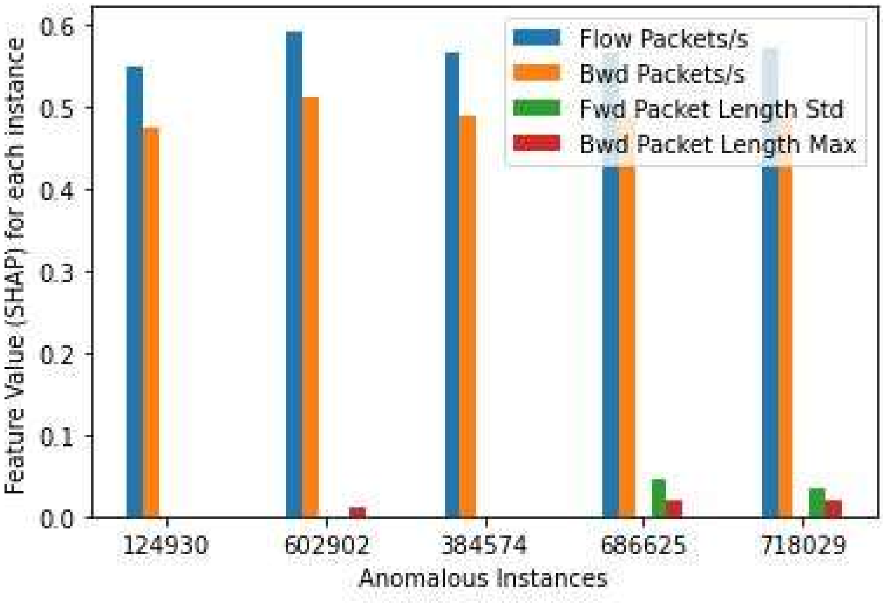

Figure 9. Explanation of DDoS detection features.

Considering these facts, we can confirm that we can effectively use explainable artificial intelligence to detect DDoS attacks. To further confirm this method, we took advantage of our dataset. The USBIDS dataset based on a DDoS attack and another dataset consists of label data. As further confirmation of our method, we could analyze the labeled data in relation to these two features, corresponding to the comparison feature values of flow packets/s or benign and attack classes. The flow packets' function has a value between 0 and 6000 for the benign class, but in the attack state, this feature value increases up to 15,000 per second, as shown in Figures 10 and 11. Bwd packs/s feature values vary from 0 to 3500 benign class and go up to 8000 when attacked, as shown in Figures 12 and 13. Based on the analysis of the feature values, we can confirm that we detected an attack that our method is designed to detect. In summary, the suggested method is suited to detecting DDoS attacks, since it detects attack flow more effectively and efficiently than DNN, RF, and DT and achieves greater accuracy in attack detection than other methods. In addition, the proposed technique can provide greater security against attacks. The explanation part is more unique than the proposed method, leading to attack certainty and reduced false positives.

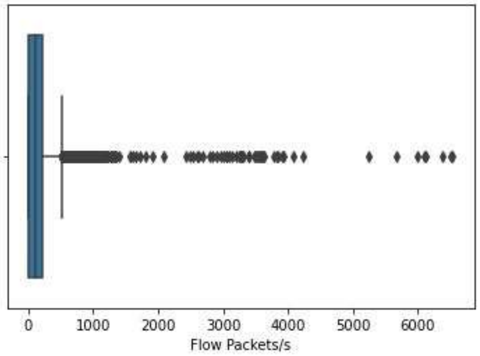

Figure 10. Flow packets per second—benign state feature value.

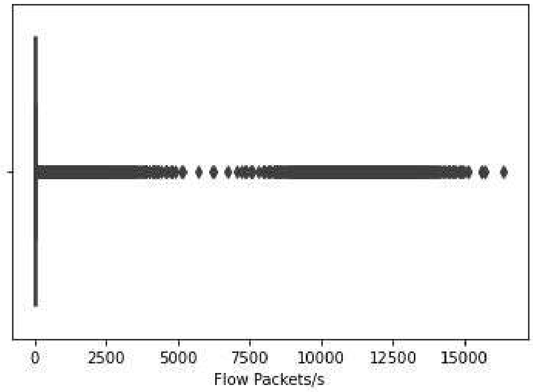

Figure 11. Flow packets per second—attack state feature value.

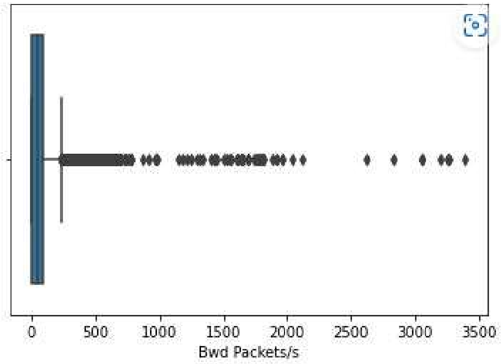

Figure 12. Backward packets per second—benign state feature value.

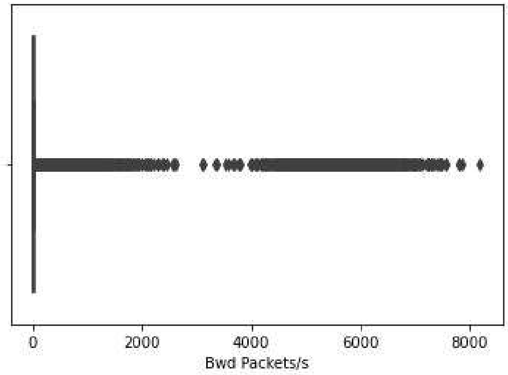

Figure 13. Backward packets per second—attack state feature value.

6. Conclusions

Security researchers are very interested in ML-based IDSs, but due to the black-box nature of these systems, they are not widely deployed in operational environments. Most anomaly detection methods find the anomalies, but there is no confidence in the attack. It is uncertain what factors influence their decisions. A system must be efficient if it is to quickly distinguish between attack flows and benign flows. In this paper, we proposed a method to detect DDoS attacks using anomaly detection that overcomes traditional AI-based problems. The proposed method provides instance-by-instance explanations, local and global explanations, and feature correlations. The outcomes help identify important decision-making criteria that finally enable determining the certainty of a detected DDoS attack. First, we extracted the features from the network traffic with the CIC flow meter, trained the model, and detected the anomalies. Then, we explained the detected anomaly and found the most influential features for each anomaly. After that, we created the list of most informative DDoS attack detection features and customized the threshold for each feature. Finally, we matched both feature lists and found the most informative features for DDoS attacks with feature impact weights. If the selected feature exceeds the threshold, this instance will be identified as a DDoS attack. We evaluated the method with three attack types and conducted experiments on Windows and Raspberry Pi 4. The results of the comparison experiment show that the proposed method can identify attack flow more effectively and quickly than the current state of the art. Currently, most DDoS attack detection methods are implemented and tested using static datasets. In future work, we will deploy this system with the simulated attacks in real-time IoT networks. This will lead to finding a more accurate, reliable, and realistic method of detecting DDoS attacks.

Author Contributions: Conceptualization, C.S.K., X.L., C.C., N.P. and P.P.; methodology, C.S.K., X.L. and C.C.; software, C.S.K.; validation, C.S.K., X.L., C.C., N.P. and P.P.; formal analysis, C.S.K.; investigation, C.S.K.; resources, X.L. and C.C.; data curation, C.S.K.; writing—original draft preparation, C.S.K.; writing—review and editing, X.L., C.C., N.P. and P.P.; visualization, C.S.K.; supervision, X.L., C.C., N.P. and P.P.; project administration, X.L. and C.C.; funding acquisition, X.L. and C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Liu, X.; Ren, J.; He, H.; Zhang, B.; Wang, Q.; Zheng, Z. All-Packets-Based Multi-Rate DDoS Attack Detection Method in ISP Layer. *Secur. Commun. Netw.* **2022**, *2022*, 7551107. [CrossRef]
2. Kaur, D.; Kaur, P. Empirical Analysis of Web Attacks. *Procedia Comput. Sci.* **2016**, *78*, 298–306. [CrossRef]
3. *Network Security Infrastructure Report: NETSCOUT*; NETSCOUT: Westford, MA, USA, 2019.
4. Alzahrani, S.; Hong, L. Generation of DDoS attack dataset for effective IDS development and evaluation. *J. Inf. Secur.* **2018**, *9*, 225–241. [CrossRef]
5. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; Halderman, J.A.; Invernizzi, L.; Kallitsis, M.; et al. Understanding the mirai botnet. In Proceedings of the 26th USENIX Security Symposium (USENIX Security 17), Vancouver, BC, Canada, 16–18 August 2017; pp. 1093–1110.
6. Kalkan, K.; Altay, L.; Gür, G.; Alagöz, F. JESS: Joint Entropy-Based DDoS Defense Scheme in SDN. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2358–2372. [CrossRef]
7. Ahmed, M.E.; Ullah, S.; Kim, H. Statistical Application Fingerprinting for DDoS Attack Mitigation. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1471–1484. [CrossRef]
8. Wang, A.; Chang, W.; Chen, S.; Mohaisen, A. Delving Into Internet DDoS Attacks by Botnets: Characterization and Analysis. *IEEE/ACM Trans. Netw.* **2018**, *26*, 2843–2855. [CrossRef]
9. Jemal, I.; Haddar, M.A.; Cheikhrouhou, O.; Mahfoudhi, A. Performance evaluation of Convolutional Neural Network for web security. *Comput. Commun.* **2021**, *175*, 58–67. [CrossRef]
10. Matta, V.; Di Mauro, M.; Longo, M. DDoS Attacks with Randomized Traffic Innovation: Botnet Identification Challenges and Strategies. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1844–1859. [CrossRef]
11. Jia, Y.; Zhong, F.; Alrawais, A.; Gong, B.; Cheng, X. FlowGuard: An Intelligent Edge Defense Mechanism Against IoT DDoS Attacks. *IEEE Internet Things J.* **2020**, *7*, 9552–9562. [CrossRef]
12. Dwivedi, R.; Dave, D.; Naik, H.; Singhal, S.; Omer, R.; Patel, P.; Qian, B.; Wen, Z.; Shah, T.; Morgan, G.; et al. Explainable AI (XAI): Core Ideas, Techniques, and Solutions. *ACM Comput. Surv.* **2023**, *55*, 1–33. [CrossRef]
13. Salih, A.A.; Abdulazeez, A.M. Evaluation of classification algorithms for intrusion detection system: A review. *J. Soft Comput. Data Min.* **2021**, *2*, 31–40. [CrossRef]
14. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4768–4777.
15. Verma, V.; Kumar, V. DoS/DDoS attack detection using machine learning: A review. In Proceedings of the International Conference on Innovative Computing & Communication (ICICC), Delhi, India, 20–21 February 2021.

16. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Network Anomaly Detection: Methods, Systems and Tools. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 303–336. [CrossRef]
17. Xiang, Y.; Li, K.; Zhou, W. Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 426–437. [CrossRef]
18. N.G., B.A.; S., S. Anomaly detection framework for Internet of things traffic using vector convolutional deep learning approach in fog environment. *Future Gener. Comput. Syst.* **2020**, *113*, 255–265. [CrossRef]
19. Zheng, J.; Li, Q.; Gu, G.; Cao, J.; Yau, D.K.Y.; Wu, J. Realtime DDoS Defense Using COTS SDN Switches via Adaptive Correlation Analysis. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1838–1853. [CrossRef]
20. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J.; Alazab, A. A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks. *Electronics* **2019**, *8*, 1210. [CrossRef]
21. Ravi, N.; Shalinie, S.M. Learning-Driven Detection and Mitigation of DDoS Attack in IoT via SDN-Cloud Architecture. *IEEE Internet Things J.* **2020**, *7*, 3559–3570. [CrossRef]
22. Yuan, X.; Li, C.; Li, X. DeepDefense: Identifying DDoS Attack via Deep Learning. In Proceedings of the 2017 IEEE International Conference on Smart Computing (SMARTCOMP), Hong Kong, China, 29–31 May 2017; pp. 1–8. [CrossRef]
23. Ateş, Ç.; Özdel, S.; Anarım, E. Clustering based DDoS attack detection using the relationship between packet headers. In Proceedings of the 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), Izmir, Turkey, 31 October–2 November 2019; pp. 1–6.
24. Ge, M.; Fu, X.; Syed, N.; Baig, Z.; Teo, G.; Robles-Kelly, A. Deep Learning-Based Intrusion Detection for IoT Networks. In Proceedings of the 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC), Kyoto, Japan, 1–3 December 2019; pp. 256–25609. [CrossRef]
25. Kaur, G.; Gupta, P. Hybrid approach for detecting ddos attacks in software defined networks. In Proceedings of the 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 8–10 August 2019; pp. 1–6.
26. Ujjan, R.M.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]
27. ur Rehman, S.; Khaliq, M.; Imtiaz, S.I.; Rasool, A.; Shafiq, M.; Javed, A.R.; Jalil, Z.; Bashir, A.K. DIDDOS: An approach for detection and identification of Distributed Denial of Service (DDoS) cyberattacks using Gated Recurrent Units (GRU). *Future Gener. Comput. Syst.* **2021**, *118*, 453–466. [CrossRef]
28. Cui, J.; Wang, M.; Luo, Y.; Zhong, H. DDoS detection and defense mechanism based on cognitive-inspired computing in SDN. *Future Gener. Comput. Syst.* **2019**, *97*, 275–283. [CrossRef]
29. 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]
30. Callado, A.; Kamienski, C.; Szabo, G.; Gero, B.P.; Kelner, J.; Fernandes, S.; Sadok, D. A Survey on Internet Traffic Identification. *IEEE Commun. Surv. Tutor.* **2009**, *11*, 37–52. [CrossRef]
31. Lashkari, A.H.; Draper-Gil, G.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of tor traffic using time based features. In Proceedings of the ICISSp, Porto, Portugal, 19–21 February 2017; pp. 253–262.
32. Friedman, L.; Komogortsev, O.V. Assessment of the Effectiveness of Seven Biometric Feature Normalization Techniques. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2528–2536. [CrossRef]
33. Singh, J.; Nene, M.J. A survey on machine learning techniques for intrusion detection systems. *Int. J. Adv. Res. Comput. Commun. Eng.* **2013**, *2*, 4349–4355.
34. Haider, S.; Akhunzada, A.; Mustafa, I.; Patel, T.B.; Fernandez, A.; Choo, K.K.R.; Iqbal, J. A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks. *IEEE Access* **2020**, *8*, 53972–53983. [CrossRef]
35. Kalutharage, C.S.; Liu, X.; Chrysoulas, C. Explainable AI and Deep Autoencoders Based Security Framework for IoT Network Attack Certainty. In Proceedings of the International Workshop on Attacks and Defenses for Internet-of-Things, Copenhagen, Denmark, 30 September 2022; pp. 41–50.
36. Kumar, P.; Tripathi, M.; Nehra, A.; Conti, M.; Lal, C. SAFETY: Early Detection and Mitigation of TCP SYN Flood Utilizing Entropy in SDN. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 1545–1559. [CrossRef]
37. Xie, Y.; Yu, S.Z. Monitoring the Application-Layer DDoS Attacks for Popular Websites. *IEEE/ACM Trans. Netw.* **2009**, *17*, 15–25. [CrossRef]
38. Catillo, M.; Vecchio, A.D.; Ocone, L.; Pecchia, A.; Villano, U. USB-IDS-1: A Public Multilayer Dataset of Labeled Network Flows for IDS Evaluation. In Proceedings of the 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Taipei, Taiwan, 21–24 June 2021; pp. 1–6.
39. Catillo, M.; Del Vecchio, A.; Pecchia, A.; Villano, U. Transferability of machine learning models learned from public intrusion detection datasets: The CICIDS2017 case study. *Softw. Qual. J.* **2022**, *30*, 955–981. [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.