applied sciences

MDPI

Article

Deep-Learning-Based Approach to Detect ICMPv6 Flooding DDoS Attacks on IPv6 Networks

Omar E. Elejla [1], Mohammed Anbar [2,*], Shady Hamouda [3], Serri Faisal [3], Abdullah Ahmed Bahashwan [2] and Iznan H. Hasbullah [2]

[1] Department of Computer Science, Al-Aqsa University, Gaza 4051, Palestine; oe.elejla@alaqsa.edu.ps
[2] National Advanced IPv6 (NAv6), Universiti Sains Malaysia, Gelugor, Penang 11800, Malaysia; bahashwan@student.usm.my (A.A.B.); iznan@nav6.usm.my (I.H.H.)
[3] Department of Business Information Technology, Liwa College of Technology, Abu Dhabi 51133, United Arab Emirates; shady.hamouda@ect.ac.ae (S.H.); serri.faisal@ect.ac.ae (S.F.)
* Correspondence: anbar@usm.my; Tel.: +60-4-653-4633

Abstract: Internet Protocol version six (IPv6) is more secure than its forerunner, Internet Protocol version four (IPv4). IPv6 introduces several new protocols, such as the Internet Control Message Protocol version six (ICMPv6), an essential protocol to the IPv6 networks. However, it exposes IPv6 networks to some security threats since ICMPv6 messages are not verified or authenticated, and they are mandatory messages that cannot be blocked or disabled. One of the threats currently facing IPv6 networks is the exploitation of ICMPv6 messages by malicious actors to execute distributed denial of service (DDoS) attacks. Therefore, this paper proposes a deep-learning-based approach to detect ICMPv6 flooding DDoS attacks on IPv6 networks by introducing an ensemble feature selection technique that utilizes chi-square and information gain ratio methods to select significant features for attack detection with high accuracy. In addition, a long short-term memory (LSTM) is employed to train the detection model on the selected features. The proposed approach was evaluated using a synthetic dataset for false-positive rate (FPR), detection accuracy, F-measure, recall, and precision, achieving 0.55%, 98.41%, 98.39%, 97.3%, and 99.4%, respectively. Additionally, the results reveal that the proposed approach outperforms the existing approaches.

Keywords: ICMPv6 flooding DDoS attacks; deep learning; machine learning; intrusion detection system

Citation: Elejla, O.E.; Anbar, M.; Hamouda, S.; Faisal, S.; Bahashwan, A.A.; Hasbullah, I.H. Deep-Learning-Based Approach to Detect ICMPv6 Flooding DDoS Attacks on IPv6 Networks. *Appl. Sci.* **2022**, *12*, 6150. https://doi.org/10.3390/app12126150

Academic Editors: Christos Bouras and Ángel González-Prieto

Received: 28 April 2022
Accepted: 14 June 2022
Published: 16 June 2022

1. Introduction

The Internet Engineering Task Force predicted as early as the 1980s that the IPv4 address pool would be exhausted within the next few decades. However, their predictions came earlier than expected after the unpredictable exponential growth of emerging technologies such as mobile devices, the Internet of Things (IoT), and cloud computing services.

Therefore, the next-generation IPv6 protocol was engineered to overcome the shortcomings of IPv4 [1]. The shift from IPv4 occurred slowly, but with millions of intelligent internet-facing devices entering the market daily, the need for the new Internet protocol also increased [2]. As a result, there is a need for a faster transition to the IPv6 protocol. Additionally, IPv6 has introduced better security features than IPv4 [3], making the optional IPv4 security protocol known as IPsec mandatory. However, recent changes may also present new challenges due to unknown vulnerabilities or inherited ones [4].

Moreover, IPv6 introduced Internet Control Message Protocol version 6 (ICMPv6), one of the essential protocols in IPv6. IPv6 cannot function without the ICMPv6 protocol's functionality since ICMPv6 messages perform many vital functions that enable data routing along network paths via nodes. However, the ICMPv6 messages lack authentication by default, making them vulnerable to exploitation. Furthermore, the IPv6 node does not validate ICMPv6 messages because it considers them legitimate. Thus, each node is susceptible to fake ICMPv6 messages and is vulnerable to ICMPv6 flooding DDoS attacks [5]. In an ICMPv6 flooding DDoS attack, adversaries deliberately make the victim's

server unavailable to users by sending myriad fake messages, which exhaust the server's resources (i.e., CPU and RAM), making it unresponsive to legitimate user requests.

Recently, many organizations have suffered from intruders' attacks. Nearly 60% of enterprises have been subjected to many attacks, such as social engineering, phishing, and DDoS attacks. Based on Juniper's investigation, small businesses account for 13% of cybercrime victims. Nowadays, not all attacks are motivated by monetary gain or data theft. Instead, some adversaries find satisfaction with just denying users from utilizing their desired service for as long as possible by using DDoS flooding attacks [6].

Therefore, attacks on network services should be prevented since any incidents would cause significant losses for both users and organizations. Even though security systems are constantly improved and updated regularly, attackers also pursue a similar strategy with their attack tools. As a result, security solutions such as intrusion detection systems (IDS) must be updated and improved continuously.

There are two types of IDS: signature-based (SIDS) and anomaly-based (AIDS). AIDS is typically more efficient than a signature-based IDS because it depends on predefined signatures that make them ineffective in detecting novel attacks. Thus, there is a pressing need for more research effort on AIDS because it has shown better performance in detecting anomaly attacks. Meanwhile, machine-learning-based approaches are considered adequate for IDSs because they can automate the process and create an efficient detection system. In addition, more human effort is also spared [7].

Moreover, the machine learning (ML) approaches are remarkably efficient in this domain. At the same time, advancement in ML algorithms leads to a new subset called deep learning (DL) algorithms. DL is an advanced subgroup of ML algorithms since it mimics the functions of the human brain and consists of multiple neural layers. The human brain is a source of inspiration for deep network researchers, who use various techniques to train hierarchical network layers [8].

IPv6 is vulnerable to DDoS and DoS attacks as long as the ICMPv6 messages remain unsecured, which could result in degraded quality of service in the network connectivity. Although there are some attempts to use DL in IDS, they still rely on legacy datasets (e.g., old network traffic) to train the models. As a result, these IDS cannot detect new and emerging attack patterns, such as ICMPv6 flooding DDoS attacks. Additionally, IDS for IPv4 networks cannot detect IPv6 attacks due to protocol structure differences [9]. Therefore, this research proposes a deep-learning-based approach to detect ICMPv6 flooding DDoS attacks accurately and ensure IPv6 network service continuity.

This research paper has three contributions to the body of knowledge. First, we propose an ensemble method for selecting the most significant traffic features that contribute to detecting ICMPv6 DDoS flooding attacks. Second, we provide a comparative performance analysis of DL-based classifiers for IDS to identify the best DL model for detecting ICMPv6-based attacks. Third, we hybridize the best DL-based classifier with an ensemble feature selection based on the selected features for IDS. This work differs from other existing works, such as in [5,10–12], by emphasizing the selection of features that highly contribute to detecting ICMPv6 flooding DDoS attacks by proposing an ensemble feature selection mechanism. The feature selection is further improved by proposing a feature intersection based on a predefined threshold. The selected features are then used to train a solid deep learning model to accurately detect ICMPv6 DDoS flooding attacks.

The organization of the rest of this research paper is as follows: Section 2 provides an overview of IPv6, ICMPv6, ICMPv6 flooding DDoS attacks, and DL-based intrusion detection systems. Sections 3 and 4 discuss related works and the proposed approach, respectively. Section 5 outlines the experimental results and discussion. Finally, Section 6 concludes this paper and offers several potential future works.

2. Background

This section provides the background of the IPv6 protocol and ICMPv6 protocol. Additionally, this section underlines the role of DL-based IDS in detecting DDoS flooding attacks.

2.1. IPv6 Protocol Overview

The Internet Society statistics indicate that IPv6 deployment has increased globally in the seven years since its introduction. IPv6 deployment has increased dramatically at the network and service provider levels, and many new businesses rely on it. More than a quarter of all internet-connected users use IPv6 connectivity. According to Google statistics regarding IPv6 adoption, over 36.27% of users connected to Google services over the IPv6 protocol on 1 January 2022 [13]. Additionally, according to the Google report, 24 countries have more than 15% IPv6 traffic, with an increasing number of IPv6 attacks [14].

IPv6 provides many advantages in terms of connectivity, expandability, and security. IPv6 is also a perfect solution for the massive number of IP addresses needed by IoT devices. Additionally, IPv6 introduces new protocols. For example, the Neighbor Discovery Protocol (NDP) is insecure and allows intruders to attack IPv6 networks. In addition, IPv6 is an OSI-model-compliant network protocol layer. For this reason, the IPv6 architecture is different from the IPv4 architecture in many ways, including how packet headers are formatted, the address range (32 bits for IPv4 and 128 bits for IPv6), and other characteristics, such as autoconfiguration and mobility [5,15].

Furthermore, IPv6 requires the mandatory Internet Protocol Security (IPSec) protocol, whereas IPv4 does not. However, even those additional features make the IPv6 protocol vulnerable to attacks since some unique IPv6 characteristics are exploitable by attackers. For example, an adversary could exploit the IPv6 multicast address, such as FF02::2, to discover routers on the network by sending a fake packet, forcing all routers in the network to respond and expose their presence, providing the adversary with enough information to attack the exposed routers [5].

2.2. ICMPv6 Overview

IPv6 is fundamentally reliant on two essential protocols, ICMPv6 and NDP, and each one consists of different classes of messages. The ICMPv6 comprises two message types, error messages code (i.e., 1 to 127) and information messages code (i.e., 128 to 255). Meanwhile, the NDP protocol is an ICMPv6 subset that utilizes five information messages. Additionally, the IPv6 protocol includes ICMPv6 as an obligation protocol. It performs a variety of essential roles, such as allowing the IPv6 node to ensure the uniqueness of the assigned IPv6 address in a link-local network with the help of the duplicate address detection (DAD) procedure [5,16].

Meanwhile, the DAD process is an integral part of the autoconfiguration (SLAAC) operation that permits nodes in the IPv6 network to generate unique IPv6 addresses. In addition, ICMPv6 also supports other vital features, such as determining the PMTU (path maximum transmission unit) and address resolution. As mentioned in RFC 2463 [17], the ICMPv6 protocol is the essential protocol of IPv6. It is responsible for several critical operations, including being used for communication between the router and hosts in IPv6 networks.

Additionally, the ICMPv6 protocol provides feedback on any issues along the destination routing path. For example, if the packet is not found by the router, it cannot be sent to the destination. ICMPv6 message characteristics have primarily been utilized to indicate errors, such as reporting unavailable destination hosts. Unfortunately, these core features have been designed for mitigation rather than security protection. Subsequently, attackers utilize the lack of security protection to carry out DDoS flooding attacks by misusing the new features of ICMPv6 messages [5].

2.3. ICMPv6 Flooding DDoS Attack

The ICMPv6 flooding DDoS attack is one of the most disruptive security threats in IPv6 networks. An adversary resorts to these attacks to exhaust the victims' network bandwidth and network resources by flooding them with packets within a specific time interval. The arrival of massive traffic forces the victim nodes to process them as quickly as possible, which eventually overwhelms the victims and becomes unavailable. The adversary usually

uses spoofed ICMPv6 messages to evade detection by network administrators or intrusion detection systems. Additionally, attackers could also send massive ICMPv6 error messages against hosts and routers to propagate their functionality. Similarly, the adversary could achieve the same result with ICMPv6 informative messages, such as neighbor solicitation (NS), echo request and echo reply, and router advertisement (RA) messages [18].

The outcome of any DDoS attack depends on the utilized packet type and the intended victim. Some attacks impact the victim node's availability, while others affect the victim and the network. For instance, if the destination is the host and the packet is an NS packet sent from different addresses, then the victim host would typically respond to all packets with a neighbor advertisement (NA) packet, resulting in DoS attacks, also known as "storm NS", which consume its resources. Overall, attackers misuse the ICMPv6 messages to attempt DoS or DDoS attacks on the IPv6 network to cut off legitimate users from essential services. The router advertisement (RA) message, for example, is an ICMPv6 information message with a type field value of 134. It helps routers provide link-local nodes with network information, including the network prefix and network default gateway, and permits routers to announce their presence on an IPv6 network. However, attackers could exploit the RA messages to flood the network with spoofed messages, forcing legitimate users to constantly update their neighbors' cache tables until their resources are depleted [12,15].

2.4. Deep-Learning-Based IDS

Learning approaches are the most effective way to produce IDS models because of their capability to automate the process, design a viable detection system, and reduce the amount of work required from individuals [19]. For example, ML and DL are self-learning anomaly-based IDS (AIDS). Lately, many researchers are considering DL techniques more efficient than ML techniques, especially when involving a significant volume of data. For example, nowadays, it is not difficult to find DL techniques utilized by researchers in various fields, such as speech recognition, visual information processing, and autonomous driving [9,20]. AIDS is scalable in nature due to its ability to integrate new detection techniques, including, but not limited to, DL techniques, making it a powerful tool in detecting new and unknown attacks, such as ICMPv6 flooding DDoS attacks.

ML-based intrusion detection is an outdated method for detecting abnormal network behavior, and it has received many criticisms for low throughput and high false-positive rates. According to Hodo's [9] survey of intrusion detection methods, traditional ML-based detection techniques lag behind deep networks. The DL algorithm used to train the layers of hierarchical networks was based on how the human brain learns. Therefore, it is greedy and susceptible, similar to how the brain learns. Since the discovery of deep networks, DL principles have been used to develop other techniques [21]. Therefore, this research paper utilizes the three most commonly employed algorithms in IDS (recurrent neural networks (RNNs), long short-term memory networks (LSTM), and gated recurrent unit (GRU)-based DL techniques) for detecting ICMPv6 flooding DDoS attacks. The following sections discuss these algorithms in more detail.

2.4.1. Recurrent Neural Networks

Recurrent neural networks (RNNs) [22] are a type of artificial intelligence neural network that considers the instance of the current input and what they have perceived as input in the past, implying that they have a second memory input. An RNN's decision at time $t-1$ influences the decision made at time t. Therefore, the RNN receives input from two different sources (the present and recent past) that work together to determine the RNN's response to new data. The main distinction between RNNs and feed-forward neural networks is the feedback loop. Unfortunately, one of the RNN's flaws is the vanishing gradient problem that occurs when the gradient is minimal, preventing the weights from changing and stopping the neural network from further training.

2.4.2. Long Short-Time Memory (LSTM)

Sepp Hochreiter et al. proposed LSTM [23] in 1997, which avoids the RNN's vanishing gradient problem by quickly learning long-term dependencies. A normal RNN generates a new hidden state by using the previous hidden state as input and the current input state. The LSTM serves the same purpose, but it also accepts old cell states. Figure 1 depicts the memory cell in detail. Each modified LSTM cell typically contains three gates, input, forget, and output, calculated using the following equations:

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{1}$$

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{2}$$

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{3}$$

$$h_t = o_t \, o \, tanh\left(c_t\right) \tag{4}$$

$$c_t = f_t \, o \, c_{t-1} + i_t \, o \, \tilde{c}_t \tag{5}$$

$$\tilde{c}_t = tanh\left(w_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{6}$$

Equation (1) is the formula for the input gate, Equation (2) is the formula for the forget gate, and Equation (3) is the formula for the output gate. The *tanh* activation function limits the output between −0 and 1, which is replaceable with other activation functions. The input data and the previous moment's memory and output are multiplied by the three gates. The formula for memory is Equation (4), which results from multiplying the output data from the current output gate with the cell state after undergoing the *tanh* function; the memory represents the short-term memory resulting from the action of the output and the long term memory. The cell state represents the long-term memory and is calculated using Equation (5), which multiplies the previous moment's cell state through the forget gate by the candidate state. Equation (6) calculates the candidate state, which represents the information to be deposited in the cell state, resulting from the action of the current input data and the output data from the previous moment.

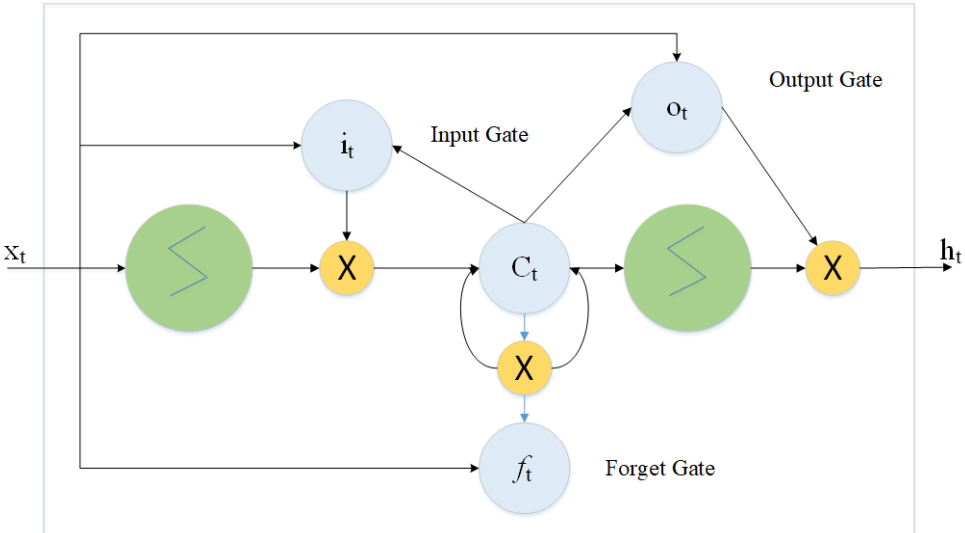

Figure 1. A cell of LSTM.

2.4.3. Gated Recurrent Unit

A gated recurrent unit (GRU) was proposed by Cho et al. [24] in 2014 and it synthesizes a single update gate by combining the forgetting gate and the input gate. It also changes the cell state and the hidden state, among other things. The final model is more straightforward than the standard LSTM model. Figure 2 depicts a cell of a GRU.

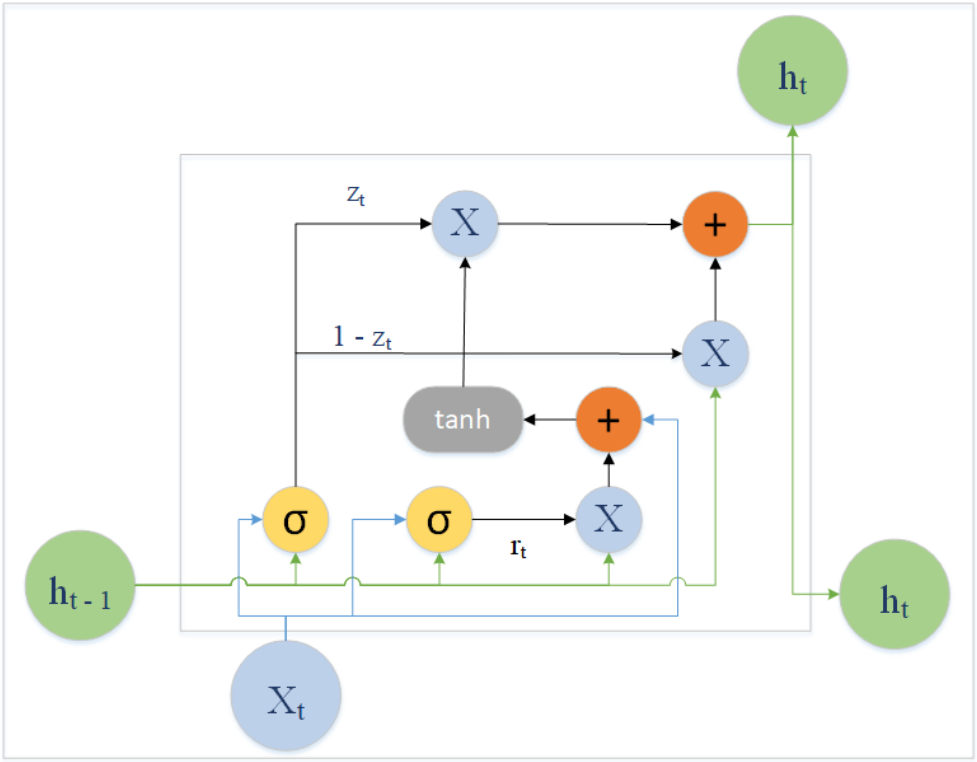

Figure 2. A cell of GRU.

GRU gates are calculated using the following equations:

$$z_t = \sigma\left(W_z\left[h_{t-1}, x_t\right]\right) \tag{7}$$

$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right]\right) \tag{8}$$

$$\tilde{h}_t = tanh(w[r_t\ o\ h_{t-1}, x_t) \tag{9}$$

$$h_t = (1 - z_t)\ o\ h_{t-1} + z_t\ o\ \tilde{h}_t \tag{10}$$

In principle, GRU is similar to LSTM, with an update gate calculated using Equation (7), a reset gate calculated using Equation (8), a memory calculated using Equation (9), and a candidate hidden layer calculated using Equation (10). The sigmoid function σ limits the output between 0 to 1, and the *tanh* function limits the output between -1 to 1. Wz, Wr, and W are the parameter matrix. Both the update and reset gates calculate the memory of the current input and the previous moment.

3. Related Works

Anomaly intrusion detection systems (AIDS) can detect DoS and DDoS attacks on IPv4 and IPv6 networks, employing various algorithms to achieve their stated goals. However, ML and DL algorithms are among the most prominent algorithms used in AIDS that achieve highly accurate detection of DoS and DDoS attacks. This section discusses the existing intrusion detection systems (IDS), including their limitations.

To detect ICMPv6 attacks, Elejla et al. [5] proposed a flow traffic representation and employed seven ML classifiers. The proposed system was tested and evaluated using an IPv6 dataset. However, experiment results show that it achieved a low detection accuracy for such attacks. Another flow-based IDS approach was proposed by Alsadhan et al. [10] based on ML techniques for detecting Neighbor Discovery Protocol (NDP) DDoS attacks. The method was tested and evaluated with IPv6 dataset. As a result, the proposed approach detected 99% NDP DDoS attacks and 91.17% replay attacks within the dataset. However, the proposed system is limited to detect NDP attacks. Meantime, Alharbi et al. [11] proposed an ML-based approach for detecting DDoS attacks in IPv6 networks. As a classi-

fication result, the approach achieved better detection results. However, no information about detection accuracy was provided, and it was evaluated using an IPv4 KDD Cup 99 dataset.

Moreover, Anbar et al. [25] developed another approach for detecting RA DDoS flooding attacks by employing two methods for feature selection: principal component analysis (PCA) and information gain ratio (IGR). In addition, they also utilized the SVM algorithm for the prediction model, then tested and evaluated the model using an IPv6 dataset, achieving a detection accuracy of 98.55% and a false-positive rate of 3.3%. However, this approach only detects router advertisement (RA) attacks. Next, an alternative approach was developed by Saad et al. [12] that employs the IGR and PCA to rank and select the relevant network traffic features before using the back-propagation neural network (BPNN) algorithm, achieving 98.3% detection accuracy. However, the proposed system only detects ICMPv6 echo request attacks. Additionally, Zulkiflee et al. [26] proposed an ML-based framework based to determine the relevant features to detect IPv6 attacks, particularly RA flooding attacks. As a result, the framework achieved a 99.95% average accuracy rate. However, it is limited to RA attacks.

In addition, Alsadhan et al. [3] proposed a machine learning system to detect NDP DDoS attacks in IPv6 network. They used classical machine learning algorithms. As a result, the decision tree (DT) and random forest (RF) algorithms achieve better results than other classifiers' algorithms. However, the proposed system performance is low. Finally, Salih et al. [27] proposed a security model to detect and classify covert channels in IPv6 networks by employing the naïve Bayesian classifier (NBC). As a result, the proposed model achieves good detection accuracy and a good true positive rate. However, the proposed approach's performance needs to be improved.

Table 1 lists several ML-based approaches designed to detect specific forms of ICMPv6 flooding DDoS attacks. For example, Anbar et al. [25] proposed an approach that detects RA DDoS flooding attacks, while Saad et al. [12] proposed a detection approach for echo request DDoS attacks. In addition, Elejla et al. [5] proposed a flow-based IDS that detects ICMPv6 flooding DDoS attacks but only achieves low performance. Alsadhan et al. [10] proposed a flow-based system limited to detecting NDP attacks and replay attacks. Alharbi et al. [11] proposed an approach for detecting DDoS attacks in IPv6 networks. However, the proposed approach was evaluated using an IPv4 dataset even though IPv6 is totally different from IPv4. Additionally, Zulkiflee et al. [26] proposed a framework to determine significant IPv6 features for detecting RA DDoS flooding attacks. Alsadhan et al. [3] proposed an NDP detection system. However, the proposed system achieves low performance. Another security model by Salih et al. [27] detects covert network channels in IPv6. However, this model also achieves low performance. Overall, based on Table 1, none of the existing approaches investigated deep learning techniques. The following section underlines the proposed deep-learning-based approach.

Table 1. A qualitative comparison of existing techniques with their limitations.

Author(s)	Techniques		IPv6 Protocol	Dataset Type		Detection Accuracy	Limitations
	ML	DL		IPv6	IPv4		
Alsadhan et al. [3]	✓	✗	✓	✓	✗	84.5%	This approach achieves low performance.
Elejla et al. [5]	✓	✗	✓	✓	✗	85.83%	Achieves low detection accuracy.
Alsadhan et al. [10]	✓	✗	✓	✓	✗	99% and 91.17%	Limited to detect NDP attacks.
Alharbi et al. [11]	✓	✗	✓	✗	✓	-	No information about detection accuracy. It was also evaluated with an IPv4 dataset.
Saad et al. [12]	✓	✗	✓	✓	✗	98.3%	Designed to detect only ICMPv6 echo request attacks.
Anbar et al. [25]	✓	✗	✓	✓	✗	98.55%	Designed to detect only RA DDoS.
Zulkiflee et al. [26]	✓	✗	✓	✓	✗	99.95%	Only detects RA DDoS attacks.
Salih et al. [27]	✓	✗	✓	✓	✗	96.46%	Still, the performance needs to be improved.

(✓): Used; (✗): not used; (-): not available.

4. The Proposed Approach

The proposed approach detects ICMPv6 flooding DDoS attacks in IPv6 networks by combining IGR, chi-square, and LSTM algorithms. First, IGR and chi-square algorithms are employed to select features that significantly contribute to detecting ICMPv6 flooding DDoS attacks. Then, the features selected by IGR and chi-square are used to train a detection model by the LSTM algorithm. Finally, we choose the LSTM algorithm after comparing three DL algorithms: RNN and its variants (LSTM and GRU), as shown in Section 5.3. Figure 3 illustrates the three successive phases of the proposed approach, namely, data preprocessing, ensemble feature selection, and LSTM-based ICMPv6 flooding DDoS attacks detection. The following subsections go over these three phases in detail.

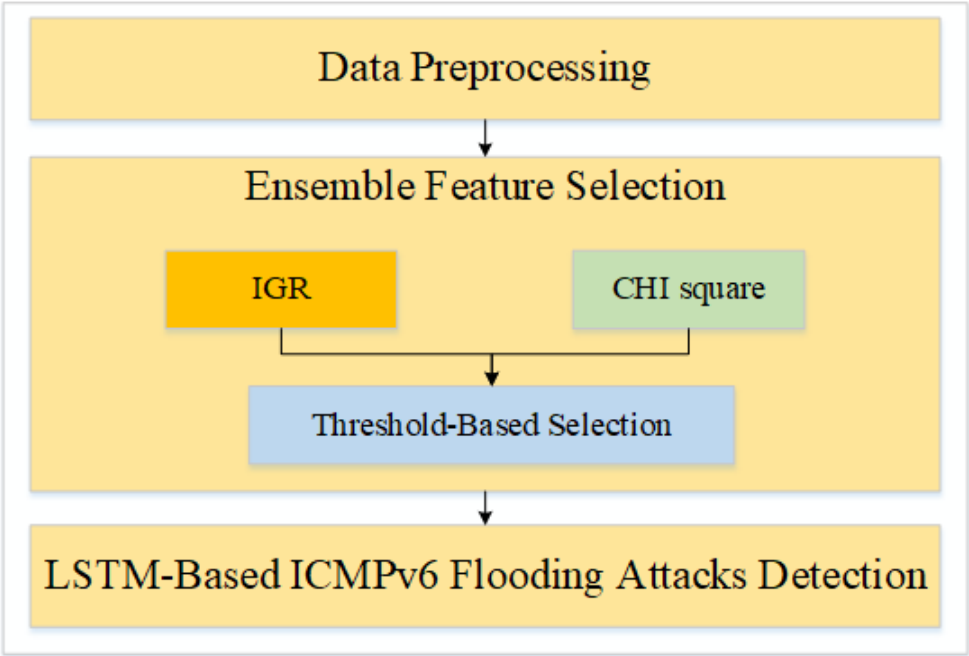

Figure 3. Phases of the proposed approach.

4.1. Data Prepossessing

Data preprocessing is the process of converting raw data into an understandable format. It is also a necessary step in data mining because we cannot work with raw data. The dataset published in [28] was used to evaluate the proposed approach's robustness in detecting ICMPv6-based attacks. Table 2 shows the description of the dataset. It is worth mentioning that there is a fundamental difference between IPv4 and IPv6 datasets due to the differences in the packet structure and features [25]. These features form the dataset used to evaluate detection mechanisms. An IPv4 dataset cannot be used to evaluate IPv6 detection mechanisms.

Table 2. Description of the dataset.

Total Number of Rows	175,305
Categorical Data	{normal, attack}
Number of Normal Packets	131,859
Number of Attack Packets	43,446
Number of Features	19
Features Names	Time, Source, Destination, Protocol, ICMPv6 type, MAC Src Addr, MAC Dst Addr, IPv6 next header, Length, Src Port, Dst Port, RA MTU, RA router lifetime, RA prefix, RA flag, NA flags, NS target, MLD flags, Class
Numerical Features	ICMPv6 Type, Src Port, Dst Port, RA MTU, RA router lifetime, RA prefix, RA flag, NA flags, NS target, MLD flags
Text Features	Time, Source, Destination, Protocol, MAC Src Addr, MAC Dst Addr, class

Before using ML algorithms, the data should be checked thoroughly by following these steps to ensure the dataset quality:

Data cleansing step cleanses the dataset by identifying incomplete, incorrect, inaccurate, or irrelevant parts of the data for replacement, modification, or deletion. All features with empty values in the dataset are replaced with 0.
Data transformation step changes the data by converting it from one format to another. For example, a unique numeric value replaces the text features in the dataset used. In other words, each feature's unique values are extracted and replaced with a unique number.
Data normalization step transforms data onto the unit sphere or translates data into the range [0, 1] (or any other range). Using Equation (11), a min–max normalization is applied to the feature vector:

$$x_i = \frac{x_i - min(x))}{max(x) - min(x)} \tag{11}$$

Data balancing step is a process to ensure the dataset has an equal distribution of classes, i.e., normal and attack. For example, Table 2 shows an unequal distribution of classes (43,446 normal vs. 131,859 attacks); therefore, we overcome this problem by employing the SMOTE [29] oversampling technique to bring the total number of normal classes to 131,859 samples.

4.2. Ensemble Feature Selection Technique

The ensemble feature selection process involves reducing the number of input variables while developing a predictive model. Reducing the number of input variables results in lower modeling computational costs and could improve the model's performance. Meanwhile, it aims to select the significant features that contribute to detecting ICMPv6-based attacks using two feature selection algorithms that distinguish between normal and attack traffic. These algorithms are information gain ratio (IGR) and the chi-square technique (CHI) [30]. IGR and CHI are among the two most effective algorithms in data analysis and are commonly used for dimensionality reduction. Employing several feature selection algorithms helps to provide a consensus on the set of features that significantly contribute to the detection of ICMPv6 attacks. In addition, the idea of ensemble feature selection algorithms has been employed in existing research to detect ICMPv6-based attacks, such as in [25]. It shows impressive results in the detection accuracy of ICMPv6 DDoS attacks.

In IGR and CHI, each feature in the dataset is assigned a score value. The following math notation is an example of how IGR and CHI work:

Let f be the basic features of the dataset where

$$f = f_1, f_2, f_3 \cdots f_n$$

where $n = 19$. Applying IGR on f will result in the following:

$$IGR\ (f') = (f_1, s_1), (f_2, s_2), (f3, s3) \cdots (f_n, s_n)$$

where s is the score value of each feature.

Meanwhile, applying CHI on f will result in the following:

$$CHI\ (f') = (f_1, s'_1), (f_2, s'_2), (f3, s'_3) \cdots (f_n, s'_n)$$

where s' is the score value of each feature.

The dataset features have been assigned a score value after applying IGR and CHI. The larger the feature's score value, the more important it is. By contrast, features with a small weight value reflect low importance. Then, the scored features resulting from IGR and CHI are selected based on two different thresholds (*th1* and *th2*), where *th1* is the threshold for IGR (f'), and *th2* is CHI (f'). Finally, the values of *th1* and *th2* are selected experimentally based on the observation of *th1* and *th2*'s impact on detection model performance.

The resulting set of features after applying *th1* is

$$IGR(f') = f_1, f_2, f_3 \cdots f_m$$

Meanwhile, the resulting set of features after applying *th2* is

$$CHI(f') = f_1, f_2, f_3 \cdots f_{m'}$$

where m and $m' <= n$.

The final feature set f' are the features that are selected by $IGR(f')$ and $CHI(f')$, denoted as

$$f' = (IGR(f') \bigcap CHI(f')$$

In a nutshell, the ensemble feature selection technique can be denoted using math notations as follows:

$$A = (x_1, y_1) \quad and \quad B = (x_2, y_2)$$

where A is $IGR(f')$, B is $CHI(f')$, y_1 is the rank value for IGR, and y_2 is the score value for CHI.

$$f' = \left(A \bigcap B\right) \leftrightarrow x_1 \epsilon A \wedge x_2 \epsilon B \wedge x_1 = x_2, \quad where \quad y_1 \geq th1 \wedge y_2 \geq th2$$

ICMPv6-Based Attack Detection

This phase trains the LSTM to build detection models that detect ICMPv6 flooding DDoS attacks based on the selected features (f'). The target function $f(x) = y_i$, where y_i = {normal, attack} is proposed as the attack detection process. This phase employs the features chosen during the ensemble feature selection phase. The training model is the result of this phase. Once deployed online, it will detect ICMPv6 flooding DDoS attacks. It is worth noting that the proposed approach is not limited to detecting ICMPv6 flooding DDoS attacks, but could also be applied to other types of datasets for detecting different attacks, or even datasets from different fields or areas.

5. Experimental Results

This section explains the metrics used to evaluate the proposed approach and provides insight and discussion of the experimental results.

5.1. Evaluation Metrics

This section describes the evaluation metrics used to assess the performance of DL-based IDS. Table 3 shows the evaluation metrics based on various attributes of the confusion matrix.

Table 3. The attributes of a confusion matrix.

	Predicted Class	
Actual Class	Attack True Positive False Positive	Normal False Negative True Negative

Where

- True positive (TP) indicates the instances where the classifier correctly classifies an attack.
- False negative (FN) indicates the instances where the classifier wrongly classifies an attack as normal.
- False positive (FP) indicates the instances where the classifier wrongly classifies a normal instance as an attack.
- True negative (TN) indicates the instances where the classifier correctly classifies normal instances.

In addition, researchers also use several other evaluation metrics in their studies, such as precision, recall, false alarm rate, true negative rate, accuracy, and F-measure.

- Precision is the proportion of attacks correctly predicted vs. all samples predicted as attacks.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

- Recall, or detection rate, is the proportion of all samples correctly classified as attacks vs. all attack samples.

$$\text{Recall} = \text{DetectionRate} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

- False alarm rate is also known as the false positive rate and is defined as the ratio of incorrectly predicted attack samples vs. all normal samples.

$$\text{False Alarm Rate} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{14}$$

- True negative rate is defined as the proportion of correctly classified normal samples vs. all normal samples.

$$\text{True Negative Rate} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{15}$$

- Accuracy, or detection accuracy, is the proportion of instances correctly classified vs. the total number of instances. However, it is only a useful performance metric when a balanced dataset is used.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{16}$$

- F1-measure is the harmonic mean of precision and recall. In other words, it is a statistical technique involving precision and recall for examining a system's accuracy.

$$\text{F1 Measure} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \quad (17)$$

The metrics used in this research are commonly used to evaluate the performance of IDS, such as in [31]. The proposed approach's evaluation involves calculating all those evaluation metrics.

5.2. The Result of the Feature Selection Phase

Training the detection model using qualified features will positively impact the detection model. However, the used dataset consists of non-significant features that degrade the classifier's performance. These features are the time, source IP, destination IP, protocol, MAC source address, MAC destination address, source port, and destination port. For the time feature, the classifiers (i.e., RNN and LSTM) consider the time interval feature during the training process to indicate attacks. For example, the packets received within these time intervals are flagged as attacks; otherwise, they are normal. As a result, in the case of online detection, attack packets that arrive later will be classified as normal because their times are outside of the attack's training time intervals. Other features, such as source and destination IP, protocol, and MAC source and destination addresses, have the same effects as time values because they can identify attack packets during training but fail to detect them during testing. For example, if a source IP address does not exist in the training model, it will be classified as an attack.

Furthermore, several existing IDS [12,26] use port numbers as a feature to differentiate attack and normal ICMPv6 packets in the detection model construction. However, the port number is not used by ICMPv6 since it is a network layer protocol. According to the OSI model, port numbers are only applicable for transport layer protocols. Therefore, the inclusion of source and destination ports as features is useless and will result in low detection accuracy. Other than the source and destination ports, the time, source and destination IPs, the protocol, and MAC source and destination addresses features are also eliminated from the dataset. As a result, our dataset comprises 10 features: ICMPv6 type, IPv6 next header, length, RA MTU, RA router lifetime, RA prefix, RA flag, NA flags, NS target, and MLD flag, which are then fed into the IGR and chi-square functions. Table 4 displays the IGR and chi-square scores for each feature.

Table 4. Scores of each feature using IGR and chi-square functions.

No.	Feature	Feature Score (IGR)	Feature Score (Chi-Square)
1	ICMPv6 type	4.06843988	1.17426781
2	IPv6 next header	1.51979191	3.72058524
3	Length	4.46315199	11.2899829
4	RA MTU	0.027046904	0.992166989
5	RA router lifetime	0.139719657	2.1899984
6	RA prefix	0.026007147	0.478028763
7	RA flag	0.118992257	2.1899984
8	NA flags	0.712575719	2.50574105
9	NS target	1.6334663	13.8955392
10	MLD flags	0.001439028	0.000998617

Let $th1 = 0.1$ and $th2 = 1$; if feature score (IGR) $\geq th1$ or feature score (chi-square) $\geq th2$, the feature is selected. Otherwise, the feature will be negated. The value of thresholds is determined experimentally and is based on the observation of datasets. Three features, RA MTU, RA prefix, and MLD flags, are dropped as their scores are lower than the predefined thresholds. Table 5 shows the selected features by IGR and chi-square after applying the thresholds.

Table 5. Selected features by IGR and chi-square.

No.	Feature	IGR ($th1 = 0.1$)	Chi-Square ($th2 = 1$)
1	ICMPv6 type	4.06843988	1.17426781
2	IPv6 next header	1.51979191	3.72058524
3	Length	4.46315199	11.2899829
4	RA router lifetime	0.139719657	2.1899984
5	RA flag	0.118992257	2.1899984
6	NA flags	0.712575719	2.50574105
7	NS target	1.6334663	13.8955392

Based on Table 5, $IGR\ (f') = \{ICMPv6\ \ type,\ IPv6\ next\ header,\ Length,\ NAflags,\ NS\ target,\ RA\ flag,\ RA\ router\ lifetime\}$ and $CHI\ (f') = \{ICMPv6\ Type, IPv6\ nextheader,\ Length,\ NA\ flags,\ NS\ target,\ RA\ flag, RA\ router\ lifetime\}$. We can notice that both IGR and chi-square selected the same features, which reflect the importance of the selected features and their contribution to the detection of ICMPv6 attack detection. As a result, the final feature set (f') is identified based on the proposed ensemble feature selection technique (refer to Section 4.2), which is equal to $ICMPv6\ type, IPv6\ next\ header, Length, NA, flags, NStarget, RA\ flag, RA\ router\ lifetime$. (f') features are used as input to train the DL model.

5.3. Deep Learning Models Setup

This section provides the details of the learning model setup and architecture. Three DL models, RNN, LSTM, and GRU, were trained based on selected features (f'). The RNN, LSTM, and GRU architectures are shown in Tables 6–8, respectively. The time batch size and epochs are 500 and 50, respectively. We use the *tanh* activation function for RNN, LSTM, and GRU. Meanwhile, we use softmax for the output layer and Adam for the optimizer, with a learning rate equal to 0.01.

Table 6. Architecture RNN.

Layer (Type)	Output Shape	Param #
simple_rnn_4 (SimpleRNN)	(None, 32)	1088.0
dropout_22 (Dropout)	(None, 32)	0
batch_normalization_11 (Batch Normalization)	(None, 32)	128.0
dropout_23 (Dropout)	(None, 32)	0
flatten_11 (Flatten)	(None, 32)	0
dense_11 (Dense)	(None, 31)	66.0
Total params: 1282.0 Trainable params: 1218.0 Non-trainable params: 64.0		

Table 7. Architecture LSTM.

Layer (Type)	Output Shape	Param #
lstm_4 (LSTM)	(None, 32)	4352.0
dropout_24 (Dropout)	(None, 32)	0
batch_normalization_12 (Batch Normalization)	(None, 32)	128.0
dropout_25 (Dropout)	(None, 32)	0
flatten_12 (Flatten)	(None, 32)	0
dense_12 (Dense)	(None, 2)	66.0
Total params: 4546.0 Trainable params: 4482.0 Non-trainable params: 64.0		

Table 8. Architecture GRU.

Layer (Type)	Output Shape	Param #
gru_3 (GRU)	(None, 32)	3360.0
dropout_26 (Dropout)	(None, 32)	0
batch_normalization_13 (Batch Normalization)	(None, 32)	128.0
dropout_27 (Dropout)	(None, 32)	0
flatten_13 (Flatten)	(None, 32)	0
dense_13 (Dense)	(None, 2)	66.0
Total params: 3554.0 Trainable params: 3490.0 Non-trainable params: 64.0		

5.4. Results and Discussion

This phase generates three training models through RNN, LSTM, and GRU using the architectures listed in Tables 6–8, respectively. These training models are generated using (f') features. RNN-, LSTM-, and GRU-based approaches require long training and inference time; therefore, we trained the model offline to overcome the issue. Then, the generated detection model (RNN, LSTM, and GRU prediction model) was deployed and operated online to detect ICMPv6 flooding DDoS attacks. We must retrain the model if significant changes occur to the environment, such as adding new servers or services.

Table 9 shows the confusion matrix of RNN, LSTM, and GRU. Based on Table 9, TNR, FNR, FPR, accuracy, and F-measure are calculated using Equations (12)–(17), respectively, as shown in Table 10.

Table 9. Confusion matrix of RNN, LSTM, and GRU.

Actual Class	Attack	Normal
Confusion matrix of RNN		
Attack	25,668	619
Normal	3412	23,045
Confusion matrix of LSTM		
Attack	25,598	689
Normal	146	26,311
Confusion matrix of GRU		
Attack	25,631	656
Normal	234	26,223

Table 10. Evaluation performance of RNN, LSTM, and GRU.

Model	TNR (%)	FNR (%)	FPR (%)	Accuracy (%)	F- Measure (%)
RNN	87.1	2.3	12.8	92.3	92.71
LSTM	99.4	2.62	0.551	98.41	98.39
GRU	99.11	2.49	0.884	98.31	98.29

As shown in Table 10, RNN, LSTM, and GRU can detect ICMP6 flooding DDoS attacks using the (f') features with a detection accuracy of 92.3%, 98.41%, and 98.3%, respectively, and with FPR of 12.8%, 0.551%, and 0.884%, respectively. We can observe that LSTM outperforms RNN and GRU in the accuracy, FPR, and F-measure metrics. Meanwhile, RNN, LSTM, and GRU reported almost identical FNR results, but RNN achieved the worst FPR. Figure 4 depicts the result of recall and precision of RNN, LSTM, and GRU.

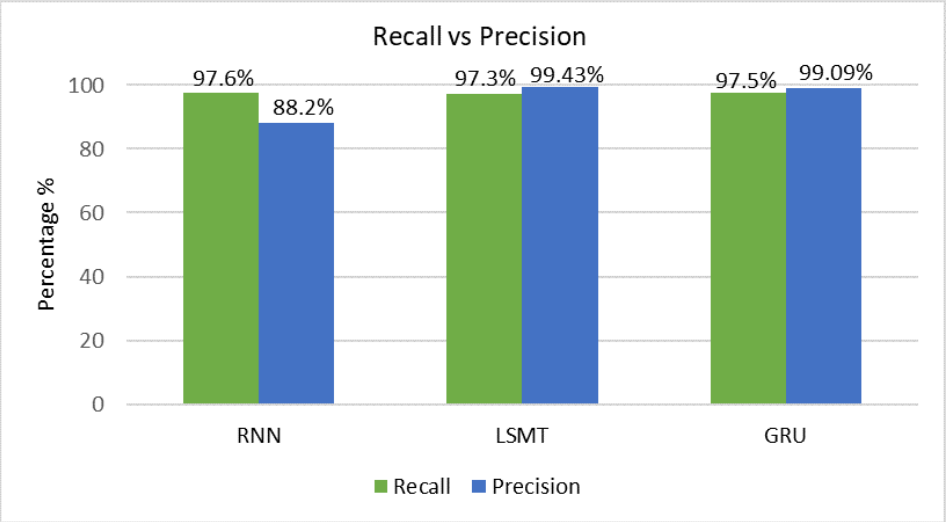

Figure 4. Recall and precision of RNN, LSTM, and GRU.

Figure 4 shows that LSTM outperforms RNN in terms of precision (99.43%) and performs nearly as well as GRU in precision and recall. The high precision percentage means it has a lower false positive value. Meanwhile, the high recall percentage of LSTM indicates a lower false negative value. Figure 5 depicts the AUC (area under the curve)–ROC (receiver operating characteristics) curves of RNN, LSTM, and GRU.

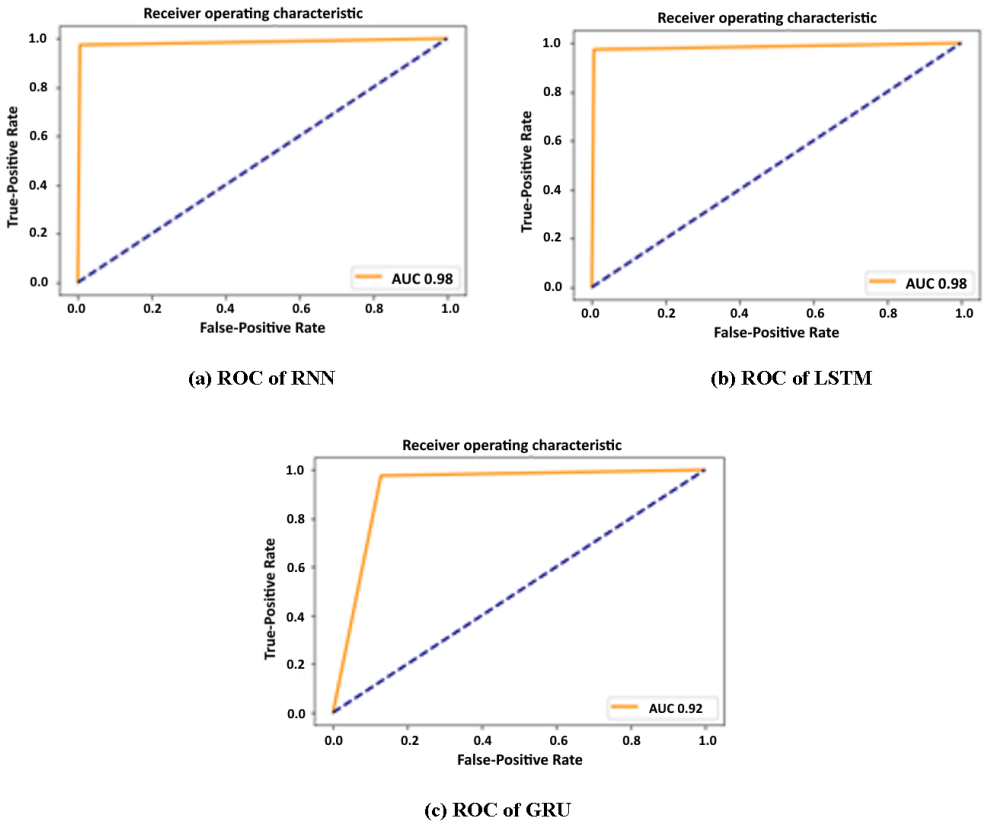

Figure 5. ACU–ROC curves of RNN, LSTM, and GRU.

The AUC–ROC curve is a performance metric for classification problems with varying threshold settings. AUC represents the degree or measure of separability, while ROC is a probability curve. It indicates how well the model can distinguish between classes. The greater the AUC, the better the model predicts 0 classes as 0 and 1 classes as 1. AUC

values range from 0 to 1, and the closer the AUC is to 1, the better the model is, which indicates a good measure of separability. Conversely, the closer the AUC curve is to 0, the poorer the model is. As shown in Figure 5b,c, LSTM and GRU have the same AUC value of 0.98. As the AUC value approaches 1, both classifiers excel at differentiating between normal and attack traffic, implying a high detection accuracy.

Meanwhile, RNN has a lower AUC value (0.92) than LSTM and GRU due to RNN having a slightly higher FPR (12.8%), which negatively impacts the calculated AUC value. Overall, the LSTM classifier outperforms the GRU and RNN classifiers in evaluation performance, consistent with the fact that LSTM and GRU were designed to overcome RNN's vanishing gradient problem. As a result, LSTM is being used in research as a classifier to build a training model capable of detecting ICMPv6 flooding DDoS attacks with high detection accuracy and a low FPR.

Lastly, the detection time was measured for all detection models to measure their performance in real-time network environments. The experiment was run thrice on a physical machine to ensure reliable results. The average detection time was calculated as tabulated in Table 11. The average detection time of RNN, LSTM, and GRU are 5.530, 8.337, and 8.110 s, respectively, on a Windows 10 workstation with an Intel Core i5-3570 3.40 GHz CPU and 16 GB RAM. The detection time can be reduced further by employing a high-end server.

Table 11. Detection Time of RNN, LSTM, and GRU.

Run #	RNN (Second)	LSTM (Second)	GRU (Second)
Run 1	5.566	8.311	8.440
Run 2	5.421	8.270	7.860
Run 3	5.603	8.431	8.029
Average Detection Time	5.530	8.337	8.110

6. Conclusions

The proposed approach employs two feature selection algorithms (IGR and chi-square) to select a set of features that helps to detect ICMPv6 DDoS flooding attacks. The feature selection is further improved by proposing a feature intersection based on a predefined threshold. Having the proposed approach deployed at the network gateway will enable it to monitor the inbound and outbound network traffic and extract the final feature set (f') that highly contributes to detecting ICMPv6 DDoS flooding attacks against vital servers, such as web and email servers, in the victim networks. f' is then utilized for training three DL models (RNN, GRU, and LSTM). The three-DL detection model was evaluated using a benchmark dataset, and the results reveal that the LSTM algorithm outperformed RNN and GRU in terms of TNR, FNR, FPR, accuracy, F-measure, recall, and precision. The future work of this study includes improving the detection accuracy using bio-inspired algorithms as feature selection techniques, which are very efficient in feature selection and classification. Additionally, bio-inspired methods could potentially improve the LSTM algorithms' hyperparameters. In addition, the applicability of applying the proposed approach over other datasets for detecting different attacks, such as those based on NDP, could be explored. Another promising future work includes exploring the possibility of applying the proposed approach in other research domains or areas than network security.

Author Contributions: Conceptualization, M.A. and O.E.E.; methodology, M.A. and O.E.E.; software, M.A.; writing—original draft, M.A., O.E.E. and A.A.B.; writing—review and editing, M.A., I.H.H., A.A.B., S.H., S.F. and O.E.E.; supervision, M.A., S.H. and S.F.; project administration, S.H., S.F. and M.A.; funding acquisition, S.H., S.F., and M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by research grant funded by Liwa College Of Technology (LCT), Abu Dhabi, UAE. Project No: IRG-BIT-005-2021 (USM External Grant Number: 304/PNAV/6501263/L136).

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. Radhakrishnan, R.; Jamil, M.; Mehfuz, S.; Moinuddin, M. Security issues in IPv6. In Proceedings of the International Conference on Networking and Services (ICNS'07), Athens, Greece, 19–25 June 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 110–110.
2. Caicedo, C.E.; Joshi, J.B.; Tuladhar, S.R. IPv6 security challenges. *Computer* **2009**, *42*, 36–42. [CrossRef]
3. Alsadhan, A.A.; Hussain, A.; Alani, M.M. Detecting NDP distributed denial of service attacks using machine learning algorithm based on flow-based representation. In Proceedings of the 2018 11th International Conference on Developments in eSystems Engineering (DeSE), Cambridge, UK, 2–5 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 134–140.
4. Shiranzaei, A.; Khan, R.Z. IPv6 security issues—A systematic review. *Next-Gener. Netw.* **2018**, *638*, 41–49. [CrossRef]
5. Elejla, O.E.; Belaton, B.; Anbar, M.; Alnajjar, A. Intrusion detection systems of ICMPv6-based DDoS attacks. *Neural Comput. Appl.* **2018**, *30*, 45–56. [CrossRef]
6. Zekri, M.; El Kafhali, S.; Aboutabit, N.; Saadi, Y. DDoS attack detection using machine learning techniques in cloud computing environments. In Proceedings of the 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), Rabat, Morocco, 24–26 October 2017; pp. 1–7.
7. Ahmad, Z.; Shahid Khan, A.; Wai Shiang, C.; Abdullah, J.; Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4150. [CrossRef]
8. Liu, Y.; Liu, S.; Zhao, X. *Intrusion Detection Algorithm Based on Convolutional Neural Network*; DDEStech Transactions on Engineering and Technology Research: Lancaster, PA, USA, 2018.
9. Hodo, E.; Bellekens, X.; Hamilton, A.; Tachtatzis, C.; Atkinson, R. Shallow and deep networks intrusion detection system: A taxonomy and survey. *arXiv* **2017**, arXiv:1701.02145.
10. Alsadhan, A.; Hussain, A.; Liatsis, P.; Alani, M.; Tawfik, H.; Kendrick, P.; Francis, H. Locally weighted classifiers for detection of neighbor discovery protocol distributed denial-of-service and replayed attacks. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3700. [CrossRef]
11. Alharbi, Y.; Alferaidi, A.; Yadav, K.; Dhiman, G.; Kautish, S. Denial-of-Service Attack Detection over IPv6 Network Based on KNN Algorithm. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 8000869. [CrossRef]
12. Saad, R.M.; Anbar, M.; Manickam, S.; Alomari, E. An intelligent icmpv6 ddos flooding-attack detection framework (v6iids) using back-propagation neural network. *IETE Tech. Rev.* **2016**, *33*, 244–255. [CrossRef]
13. Google. Statistics About IPv6 Connectivity Among Google Users. 2022. Available online: https://www.google.com/intl/en/ipv6/statistics.html?safe=active (accessed on 1 January 2022).
14. Aleesa, A.; Zaidan, B.; Zaidan, A.; Sahar, N.M. Review of intrusion detection systems based on deep learning techniques: Coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. *Neural Comput. Appl.* **2020**, *32*, 9827–9858. [CrossRef]
15. Bahashwan, A.A.; Anbar, M.; Hanshi, S.M. Overview of IPv6 based DDoS and DoS attacks detection mechanisms. In *International Conference on Advances in Cyber Security*; Springer: Singapore, 2019; pp. 153–167.
16. Ahmed, A.S.; Hassan, R.; Othman, N.E. Secure neighbor discovery (SeND): Attacks and challenges. In Proceedings of the 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI), Langkawi, Malaysia, 25–27 November 2017; pp. 1–6.
17. Conta, A.; Deering, S.; Gupta, M. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (ipv6) Specification; RFC 4443, IETF; 2006. Available online: https://datatracker.ietf.org/doc/rfc4443/bibtex/ (accessed on 27 April 2022).
18. Ahmed, A.S.A.M.S.; Hassan, R.; Othman, N.E. IPv6 neighbor discovery protocol specifications, threats and countermeasures: A survey. *IEEE Access* **2017**, *5*, 18187–18210. [CrossRef]
19. Mishra, P.; Varadharajan, V.; Tupakula, U.; Pilli, E.S. A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 686–728. [CrossRef]
20. Majeed, P.G.; Kumar, S. Genetic algorithms in intrusion detection systems: A survey. *Int. J. Innov. Appl. Stud.* **2014**, *5*, 233.
21. Liu, X.; Xie, L.; Wang, Y.; Zou, J.; Xiong, J.; Ying, Z.; Vasilakos, A.V. Privacy and security issues in deep learning: A survey. *IEEE Access* **2020**, *9*, 4566–4593. [CrossRef]
22. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]
23. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
24. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
25. Anbar, M.; Abdullah, R.; Al-Tamimi, B.N.; Hussain, A. A machine learning approach to detect router advertisement flooding attacks in next-generation IPv6 networks. *Cogn. Comput.* **2018**, *10*, 201–214. [CrossRef]

26. Zulkiflee, M.; Haniza, N.; Shahrin, S.; Ghani, M. A framework of ipv6 network attack dataset construction by using testbed environment. *Int. Rev. Comput. Softw. (IRECOS)* **2014**, *9*, 1434–1441. [CrossRef]
27. Salih, A.; Ma, X.; Peytchev, E. New intelligent heuristic algorithm to mitigate security vulnerabilities in IPv6. *IJIS Int. J. Inf. Secur.* **2015**, *4*, 2382–2619.
28. Elejla, O. A Reference Dataset for ICMPv6 Flooding Attacks* Omar E. Elejla, "Bahari Belaton," Mohammed Anbar and "Ahmad Alnajjar" School of Computer Science, Universiti Sains Malaysia, Penang, Malaysia "National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Penang, Malaysia. *J. Eng. Appl. Sci.* **2016**, *100*, 476–481.
29. Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]
30. Thaseen, I.; Kumar, C. Intrusion detection model using fusion of chi-square feature selection and multi class SVM. *J. King Saud Univ. Comput. Inf. Sci.* **2017**, *29*, 462–472.
31. Sahoo, K.S.; Puthal, D. SDN-assisted DDoS defense framework for the internet of multimedia things. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2020**, *16*, 1–18. [CrossRef]