symmetry

Article

Flow-Based IDS Features Enrichment for ICMPv6-DDoS Attacks Detection

Omar E. Elejla [1], Mohammed Anbar [2,*], Shady Hamouda [3], Bahari Belaton [4], Taief Alaa Al-Amiedy [2] and Iznan H. Hasbullah [2]

[1] Department of Computer Science, Al-Aqsa University, Gaza 4051, Palestine
[2] National Advanced IPv6 (NAv6) Centre, Universiti Sains Malaysia, USM, Penang 11800, Malaysia
[3] Department of Business Information Technology, Liwa College of Technology, Abu Dhabi 51133, United Arab Emirates
[4] School of Computer Sciences, Universiti Sains Malaysia, USM, Penang 11800, Malaysia
* Correspondence: anbar@usm.my; Tel.: +60-4-653-4633

Abstract: Internet Protocol version 6 (IPv6) and its core protocol, Internet Control Message Protocol version 6 (ICMPv6), need to be secured from attacks, such as Denial of Service (DoS) and Distributed DoS (DDoS), in order to be reliable for deployment. Several Intrusion Detection Systems (IDSs) have been built and proposed to detect ICMPv6-based DoS and DDoS attacks. However, these IDSs suffer from several drawbacks, such as the inability to detect novel attacks and a low detection accuracy due to their reliance on packet-based traffic representation. Furthermore, the existing IDSs that rely on flow-based traffic representation use simple heuristics features that do not contribute to detecting ICMPv6-based DoS and DDoS attacks. This paper proposes a flow-based IDS by enriching the existing features with a set of new features to improve the detection accuracy. The flow consists of packets with similar attributes (i.e., packets with the same source and destination IP address) and features that can differentiate between normal and malicious traffic behavior, such as the source IP address's symmetry and the whole flow's symmetry. The experimental results reveal that the enriched features significantly improved the IDS's detection accuracy by 16.02% and that the false positive rate decreased by 19.17% compared with state-of-the-art IDSs.

Keywords: IPv6; ICMPv6; DDoS; flow-based representation; intrusion detection; intrusion detection system

check for updates

Citation: Elejla, O.E.; Anbar, M.; Hamouda, S.; Belaton, B.; Al-Amiedy, T.A.; Hasbullah, I.H. Flow-Based IDS Features Enrichment for ICMPv6-DDoS Attacks Detection. *Symmetry* **2022**, *14*, 2556. https://doi.org/10.3390/sym14122556

Academic Editor: Marcin Michalak

Received: 10 November 2022
Accepted: 1 December 2022
Published: 3 December 2022

1. Introduction

Internet Protocol version 6 (IPv6) is designed to replace Internet Protocol version 4 (IPv4) to overcome the problem of IPv4 address exhaustion. IPv6 improves many aspects of IPv4, such as security, address auto-configuration, router discovery, successful transmission notification, and mobility. IPv6 depends on Internet Control Message Protocol version 6 (ICMPv6) for core communication features, such as address resolution and neighbor discovery. Therefore, ICMPv6 is an essential and irreplaceable protocol for IPv6-enabled devices to communicate in IPv6 networks [1].

ICMPv6 is a network layer protocol in the Open Systems Interconnection (OSI) model designed to allow link-local devices to communicate using IPv6. ICMPv6 has two types of communication messages: informational sharing messages and error-reporting messages [2,3]. ICMPv6 uses informational messages to perform tests, diagnoses, and other central functions, whereas error-reporting messages are used for reporting errors in IPv6 packet deliveries. Unlike ICMPv4, which can be blocked at the network default gateway to mitigate ICMPv4-based attacks, ICMPv6 is an essential part of IPv6 that cannot be disabled or blocked from the network to prevent ICMPv6-based attacks [4]. Therefore, attackers utilize such communication media (ICMPv6) to perform attacks due to their mandatory existence.

However, ICMPv6 has many limitations that expose the IPv6 protocol to various security threats. Several studies have shown that ICMPv6 is vulnerable to several types of attacks. One of the most severe security threats based on ICMPv6 is Denial of Service (DoS) and Distributed DoS (DDoS) attacks [5]. Such attacks aim to overwhelm the targeted victim's machine by bombarding it with many packets that consume its resources and bandwidth, stopping it from working [6]. ICMPv6 attacks were first discovered in 2011, as reported in the Arbor Networks Security Report [7].

The problem of IPv4 Intrusion Detection Systems (IDSs) being unable to detect IPv6 DDoS attacks is due to the structural differences between the two protocols. Many researchers studied the problem of ICMPv6-DDoS attacks [8–11]. These researchers followed two approaches to detecting ICMPv6-DDoS attacks: signature and anomaly detection. IPv6 signature-based IDSs are limited to previously known attacks, with their signatures in the record. Therefore, they have a low accuracy in detecting new IPv6 attacks with unknown signatures, as mentioned in Section 3.1. On the other hand, IPv6 anomaly-based IDSs have several drawbacks: (i) they use packet-based traffic representation, which is irrelevant to detect such attacks, (ii) they depend on an irrelevant set of features to build the training and testing datasets, as shown in [9,12], and (iii) they suffer from a low detection accuracy with a high rate of false alarms in detecting attacks, as shown in Section 3.2.

This paper extends the work of Elejla et al. [8] on a flow-based IDS due to their relevant features and the dataset's availability. This paper has three main contributions: (i) an enriched set of flow-based features to improve the accuracy of detecting ICMPv6-DDoS attacks with a low false alarm rate; (ii) an ensemble features reduction mechanism to select the best set of features that contribute to the detection of ICMPv6-DDoS attacks; and (iii) extensive evaluation using several machine learning algorithms to prove the significant impact of the enriched set of flow-based features on the machine learning algorithm's detection accuracy.

The remaining parts of this paper are organized as follows: Section 2 presents a background of IPv6, ICMPv6 protocols, and the types of network traffic representations used by IDS. Section 3 reviews the state-of-art works of literature. Section 4 presents the architecture of the flow-based IDS proposed by this paper to detect ICMPv6-DDoS attacks. Section 5 explains the flow dataset and the evaluation metrics used to evaluate the proposed IDS. Additionally, it discusses the feature reduction process's results and the proposed IDS' results in detecting the attacks. Section 6 presents the conclusions and findings of this paper.

2. Background

This section provides a background of IPv6 and ICMPv6 with their security issues and the existing IDSs' traffic representation with their advantages and drawbacks.

2.1. IPv6 and ICMPv6 Protocols

The IPv4 protocol address consists of 32 bits representing around 4 billion unique IP addresses. However, the number of IPv6-enabled devices is expected to reach 125 billion in 2023, far exceeding the ability of the IPv4 protocol to represent using its limited address space [13]. Therefore, in 1998, the Internet Engineering Task Force (IETF) designed the IPv6 protocol, also known as IPng, with 128 bits of address space, to replace the IPv4 protocol [13]. IPv6 protocol has an address pool of 3.410^{38} addresses that can cater to current and future needs.

IPv6 is considered more secure than the IPv4 protocol due to the built-in security mechanisms, such as end-to-end security features and an IP Security (IPSec) protocol [14]. However, IPv6 nodes and networks have been targeted by attackers using a few discovered weaknesses and vulnerabilities. For example, attackers have been successful in performing DoS and DDoS attacks against IPv6 targets [7,9] using penetration tools such as The Hacker's Choice (THC) [15].

One of the main differences between IPv6 and IPv4 is its high dependency on the ICMPv6 protocol. ICMPv4 is an optional protocol in IPv4 networks, but the ICMPv6 is compulsory for IPv6 networks to work correctly [14]. Some of the services in IPv6 that use ICMPv6 messages are similar to those provided by ICMPv4 messages for the IPv4 protocol. However, there are also newly introduced services served by ICMPv6 messages in the IPv6 protocol, such as host address auto-configuration and the Neighbor Discovery Protocol (NDP) [4]. In addition, many similar IPv6 functions to IPv4 only use ICMPv6 messages but require separate and specific protocols in IPv4, such as the address resolution protocol and Internet group management protocol. Consequently, the critical roles of the ICMPv6 protocol make it an essential and irreplaceable part of IPv6 networks [5].

The ICMPv6 protocol is responsible for testing, diagnosing, and other critical functions using ICMPv6 informational messages. In addition, it is responsible for generating responses and reporting delivery errors of IPv6 messages using ICMPv6 error messages [16]. However, malicious users regularly misuse ICMPv6 messages to perform various attacks on IPv6 nodes and networks, such as DoS and DDoS attacks [14]. DoS and DDoS attacks are common in IPv6 networks, performed by overwhelming the victim device with numerous packets until it consumes its resources. These attacks aim to limit the availability of the victim's devices and prevent them from serving legitimate users.

The ICMPv6 protocol dictates that all IPv6 nodes must deal with and respond to any received ICMPv6 message. For example, in the case of the ICMPv6 Neighbor Solicitation (NS) message, once received, the receiving node must respond with the ICMPv6 Neighbor Advertisement (NA) message without verification. Therefore, attackers misused this feature to perform DoS or DDoS attacks by bombarding numerous NS messages to the targeted victim's device to consume its resources and subsequently disrupt its services. Moreover, the actual implementation of the IPv6 protocol in different networks shows that it is vulnerable to several new IPv6 attacks that target the newly introduced IPv6 functions. In addition, IPv6 networks are also vulnerable to several existing IPv4 attacks, such as DDoS using ping request packets [16].

The ICMP protocol has a simple design and low security features, resulting in being exploited by malicious users in order to perform various attacks on network nodes or the network itself. To avoid the threats or exploitation of the ICMPv4 protocol, administrators can block and deny all ICMPv4 messages in the network. However, this action is impossible with ICMPv6 due to the high dependency of the IPv6 protocol on its message for correct functioning. ICMPv6 messages must be allowed in IPv6 networks to work appropriately, unlike IPv4 networks [17]. Consequently, to avoid the threats of ICMPv6, deploying IPv6 IDS in an IPv6 network can help to analyze IPv6 traffic in order to detect ICMPv6 attacks.

2.2. Existing IDSs' Representations

The IDSs' traffic representations are categorized into two classes: packet-based and flow-based. The packet-based representation traditionally captures the whole payload and headers of all of the packets. An alternative way to represent network traffic is called flow-based representation. Flow-based representation combines a set of packets that have common characteristics and stores them in one record. The IDSs' traffic representations have direct impacts on the detection ability and efficiency of the IDS.

2.2.1. Packet-Based Representation

IDSs that use packet-based traffic representation capture all packets passing through a particular edge point of the network where the network traffic passes. The monitoring device where the IDS is installed must capture the packets without filtration or loss, and then store the captured packets with their complete information, including the information of the OSI model and the time of receiving them [9]. The stored packets' details enable IDSs to detect attacks using deep inspection.

However, the main disadvantage of packet-based traffic representation is the size of the generated traffic data required by the IDS for analysis and inspection. Winter et al. [9]

showed that capturing the network traffic of a 1GBaseX network generated six gigabytes of data per minute. As a result, IDSs that use packet-based traffic representation are typically complex and slow due to the massive number of packets that need to be captured, processed, and analyzed. In addition, packet-based traffic representation contains the complete packet details, including private or sensitive data, preventing researchers from sharing their datasets with others. Although anonymization can hide these details, it demands additional efforts from the IDS and increases the possibility of errors.

2.2.2. Flow-Based Representation

IDSs that use flow-based traffic representation combine packets with the same characteristics in one record. The targeted attacks for detection determine the flows' definition and the characteristics of the flows' format. IPv4 flow-based IDSs usually form the flows based on IPv4 addresses, port numbers, and the used protocol [9]. However, a flow-based IDS needs to pre-format the flows before applying its detection mechanism, adding extra computation in preprocessing the traffic. In addition, flow-based IDSs cannot detect attacks that depend on the packets' details that are unavailable while building the flows.

Flow-based IDSs have several advantages. First, they generate smaller-sized traffic data than packet-based representation due to the inclusion of fewer details. For example, Sperotto et al. [18] showed that a flow-based representation could represent the transfer of a gigabyte-sized file using a single flow, which is equivalent to only 0.1% of the packet-based representation. Second, a flow-based IDS is lightweight since it uses compact-sized traffic data in inspection and analysis. Lastly, it allows for the sharing of the dataset since the sensitive information in the packets, such as IPv6 addresses, has been removed from the flow traffic.

In conclusion, each of the traffic representations has advantages and disadvantages. In addition, both are good for detecting specific attack types. For instance, flow-based representation is more suited to detecting attacks, such as DoS and DDoS. Table 1 summarizes the two IDS representation techniques with their advantages and disadvantages.

Table 1. Summary of advantages and disadvantages of IDSs' traffic representations.

Representation	Advantage	Disadvantage
Packet-based representation	Contains the details of the whole packets. Immediate availability of the traffic for IDS without preprocessing.	Every packet needs to be inspected. Has the problem of exposing sensitive details. Unable to detect attacks that use encrypted payload
Flow-based representation	Allows for detection of attacks that use encrypted payload Overcomes the problem of sensitive details Allows for building fast IDS	Availability of fewer packet details Flows need to be constructed before IDS works Inability to detect attacks that use encrypted payload

As shown in Table 1, the use of packet-based traffic representation enables IDSs to detect attacks by inspecting packet details. However, this representation generates a huge amount of traffic data; therefore, it is impractical for today's high-speed networks. Furthermore, IDSs that use packet-based traffic representation are considered heavyweight IDSs that are slow in analyzing these large amounts of traffic. In contrast, the flow-based traffic representation is preferable for detecting attacks such as DDoS, where it forms the traffic in a format more relevant to the IDS. DDoS attacks are performed by sending numerous amounts of packets to the victim device in a short time interval. This attack traffic can be easily represented in a flow-based representation format in a detectable and recognizable way by flow-based IDSs [18].

3. Literature Review

Table 1 shows that the packet-based traffic representation allows for the building of an IDS that detects attacks based on packet details. However, it generates a considerable amount of traffic data, making it impractical for today's high-speed networks. Furthermore, the amount of traffic data makes analyzing them cumbersome and time-consuming, resulting in a heavyweight IDS. In contrast, the flow-based traffic representation is preferred for detecting attacks such as DDoS since the traffic format is better suited for representing traffic data in high-speed networks. Since DDoS attacks involve massive amounts of packets sent toward a victim within a small-time interval, flow-based representation can easily represent the attack traffic in a detectable and recognizable way for flow-based IDSs [16].

3.1. Signature-Based IDSs

Signature-based IDSs depend on predefined signatures for attacks to be reported once they match/find any of them. Several signature-based IDSs have been proposed to detect IPv4 attacks. Some of these IDSs have been improved to support the detection of IPv6 DDoS attacks, including ICMPv6-DDoS attacks. Other IDSs have been proposed to detect IPv6 DDoS attacks from their first release, including attacks that target ICMPv6. IDSs that support IPv6 DDoS attacks are studied and presented in this section to evaluate and judge them.

Snort [19] is a free and open-source IDS that was initially developed for IPv4 attack detection in 2008 by Martin Roesch. It performs traffic analyses and logging to detect malicious packets. In addition, it allows for writing plain-text rules language in order to determine the packets that need to be collected. Snort started to support IPv6 attack detection from version 2.8 in 2007. It deals with IPv6 packets in the same manner as IPv4 packets: by replacing IPv4 fields with its similar IPv6 fields, such as Time to Live (TTL) being swapped with a hop limit. Snort fails to classify the performed attacks to IPv6 or IPv4 attacks because it cannot differentiate IPv6 from IPv4 packets [17]. In addition, it is considered weak at detecting attacks that use a connectionless protocol such as ICMPv6 and IPv6 autoconfiguration [17].

Vern Paxson developed Zeek [20] (previously known as Bro) in C language in 1999. Zeek has supported several IPv6 features, such as IPv6 reassembly, fragmentation, and tunnel decapsulation, since Bro version 0.8 [21]. Zeek has a customized scripting language, which allows users to write their policies. Although it was initially designed for IPv4, it supports writing specified rules for IPv6, including IPv6 DDoS attacks. However, an experiment in [22] showed that Zeek failed to detect 8% of IPv6 attacks, including DDoS attacks. Even though it is suitable for large-scale networks with a good flexibility [23], Zeek requires a massive amount of resources, resulting in a slow response to attacks [24].

Suricata [25] is a signature-based IDS developed by the Open Information Security Foundation (OISF) in 2009. It was mainly designed to replace Snort IDS and supported IPv6 from its first version. It was based on the same design and rules as Snort; however, it benefits from modern multicore processor architectures by supporting parallelism to be the fastest signature-based IDS [26]. Atlasis et al. [27] experimented on Suricata to check its ability to detect ICMPv6-based attacks. Unfortunately, they obtained disappointing results when they failed to detect any attacks. Moreover, it does not handle IPv6 traffic well compared to IPv4 in stream reassembly. Another experiment by Atlasis et al. [28] showed that Suricata could be evaded in several scenarios involving packet fragmentation and padding with more than six octets. In addition, in terms of memory usage, Suricata consumes the most memory compared to other IDSs.

The existing IPv6 signature-based IDSs suffer from several limitations. Signature-based IDSs rely on attack patterns to detect an attack; therefore, it has trouble recognizing unknown “zero-day” attacks because their patterns are not available in the database [29,30]. The lack of signatures is either because the attacks have not been detected yet—and, therefore, the signature is unknown—or because other IDSs detect the attack, but the attack signature has not been propagated. Moreover, signature-based IDSs can be easily misled due to their dependency on fixed attack behaviors to recognize attacks, making them unable to detect self-modifying behavioral attacks. These limitations are why researchers turn to anomaly-based IDSs.

3.2. Anomaly-Based IDSs

Anomaly-based IDSs look for a set of behaviors that describe attacks and other behaviors that characterize normal traffic. Several anomaly-based IDSs have been proposed to detect IPv6 DDoS attacks. Characterizing the behaviors of normal and attack traffic is made either by defining rules that describe the allowed and denied behaviors or by using machine learning algorithms to auto-learn the behaviors of normal and malicious traffic based on training traffic datasets [2].

3.2.1. Rules Anomaly-Based IDS

These IDSs depend on predefined rules written by their developers based on their knowledge of the attacks. The rules define the expected behaviors of malicious traffic in the network. Such IDSs can detect "zero-day" attacks as long as their attack behaviors fall within the defined rules. This section presents a few IDSs that use this technique to detect IPv6 attacks.

The first rule anomaly-based IDS was proposed by Barbhuiya et al. [31] to detect Man-in-the-Middle (MiTM) and DDoS attacks that use NS and NA packets. The proposed IDS is an active IDS that probes the host devices to reply with specified packets to be compared with the defined rules [2]. The probe packets are sent once a new packet with different IP-MAC addresses is received. The responses of probe packets are stored in six tables that require reserving part of the device's memory. Moreover, the probe packets induce an extra amount of traffic (responses) in the network, which might help to further consume the resources of the network, especially when a DDoS attack is present [32]. Additionally, the proposed IDS can only detect MiTM and DDoS attacks from spoofed MAC or IP addresses but not from real MAC-IP addresses [7].

Bansal et al. [32] improved the IDS version of Barbhuiya et al. [31] by reducing the number of used tables from six to four tables and using IPv6 MLD packets instead of NS and NA packets. However, it still depends on probe packets and looking for spoofed MAC-IP addresses to check whether the address is genuine or spoofed. Therefore, it still inherits the drawbacks of the old IDS, which are the consumption of the device's resources and failure to detect attacks from real MAC-IP addresses.

A genetic algorithm was used in [33] to detect attacks that exploit NDP vulnerabilities. The algorithm was trained and validated on the CERNET2 backbone network at Tsinghua University in China. The achieved results were 85% with a low error rate of around 2%. However, during the training phase, the network was void of malicious traffic, and malicious traffic was injected from a separate dataset, resulting in a non-consistent and biased dataset that might falsify the IDS in real implementation. Unfortunately, the authors did not detail the feature selection and ranking process in their paper.

3.2.2. Machine Learning Anomaly-Based IDS

Machine learning has proven its efficiency in several areas, including classifying traffic to detect malicious packets. Machine learning anomaly-based IDS is developed by training a machine learning algorithm on a training dataset to learn the behaviors of malicious traffic. These algorithms have various capabilities to learn the behaviors that depend on the algorithm's nature. However, they have the ability to auto-learn the behaviors, an ease of use, and a low-cost deployment. Therefore, several researchers have adapted machine learning algorithms to solve the problem of ICMPv6-DDoS attacks.

Apriori algorithm was used by Lai et al. [34] to detect IPv6 attacks based on six different features extracted from the packets. The extracted features are the IPv6 address, the port number, the protocol, the TCP flag, and the service. Some of these features are irrelevant to the attacks, such as the IP address and port number [8]. A small-sized dataset (5000 records) was collected from a four-PC network that was used to validate the IDS. Moreover, the IDS achieved an unreliable and low accuracy in detecting the attacks.

Zulkiflee et al. [35] used a Support Vector Machine (SVM) to detect IPv6 attacks. The algorithm was applied to a large dataset consisting of 250,008 records. A PSO algorithm was used to select the most relevant features. After features selection, the SVM was applied to a training dataset consisting of five features extracted from the traffic: the source IPv6 address, source and destination port numbers, time interval, and protocol. The achieved average detection accuracy of the ICMPv6 RA attack was 99.95%. However, the research only focused on attacks that use ICMPv6 RA packets and ignore other ICMPv6-based attacks.

Redhwan et al. [36] utilized the Back-Propagation Neural Network (BPNN) algorithm to detect IPv6 DDoS attacks. The algorithm was applied to a self-generated dataset collected from six network devices at the National Advanced IPv6 Center (NAv6) in Universiti Sains Malaysia (USM). The generated dataset was represented by 10 features: IPv6 source addresses, next header, ICMPv6 type, ICMPv6 code, ICMPv6 payload, traffic class, flow label, hop limit, payload length, and reserve bit. The testing results achieved a high detection accuracy of 98.3% with a 0.26 Root Mean Square Error (RMSE). However, the malicious packets in the used dataset were from one type of IPv6 packets, which was ECHO requests packets. In addition, the tiny size of the used dataset (2000 ECHO request packets) does not accurately reflect the normal behaviors of the included attacks [12].

A number of classifiers (decision tree, random forest, naive Bayes, MLP, and Bayesian) were applied by Alsadhan et al. [10] to detect NDP DDoS attacks. The researcher reformatted the traffic to flows instead of packets. Flow representation was used due to its advantages over packet representation, as mentioned in [10]. The best detection accuracy was achieved by the decision tree, the random forest (84%), which is not a reliable detection accuracy. Moreover, NDP has only five types of ICMPv6 packets, where other ICMPv6 packets are not considered within the used dataset. Moreover, the dataset was built based on 12 features without applying features ranking to select the most contributing features.

Anbar et al. [17] proposed a technique that exploits an Information Gain Ratio (IGR) and Principal Component Analysis (PCA) for feature selection and an SVM-based predictor model for detecting Router Advertisement (RA) flooding attacks. The proposed technique was evaluated using a realistic dataset, achieving an excellent detection accuracy of 98.55% and a low False Positive Rate (FPR) of 3.3%. Therefore, based on the results, the proposed technique is efficient in detecting RA flooding attacks.

Elejla et al. [4] proposed a deep-learning-based approach to detect ICMPv6 flooding DDoS attacks in IPv6 networks by introducing an ensemble feature selection technique that utilizes chi-square and IGR methods to select significant features for attack detection with a high accuracy. In addition, a Long Short-Term Memory (LSTM) was employed to train the detection model on the selected features. The proposed approach was evaluated using a synthetic dataset for the False-Positive Rate (FPR), detection accuracy, F-measure, recall, and precision, achieving 0.55%, 98.41%, 98.39%, 97.3%, and 99.4%, respectively.

Hammoodi et al. [5] proposed a deep-learning-based approach to detect RA flooding DDoS attacks. The authors utilized two ranking algorithms to select the most significant features that reflect the influence of the whole feature set. Thereafter, the authors employed the recurrent neural network as a classifier. The proposed approach showed promising results in terms of adopting deep learning to detect RA-flooding-based DDoS attacks, where the experimental results showed the effectiveness of the RNN by achieving a high detection accuracy of 99.6% with a low false detection rate of 0.3%.

The first research that utilized flow-based traffic representation in detecting DDoS attacks was conducted by Elejla et al. [8]. The flow was defined as a set of packets sent within a specific time interval with the same source and destination IPv6 addresses. A dataset built using 11 different basic features was used to train and test a number of classifiers. The used classifiers were a decision tree, SVM, naïve Bayes, K-Nearest Neighbor (KNN), random forest, and neural network. In addition to the cross-validation testing technique, the supplied dataset test testing technique was used to avoid the cross-validation technique's drawback (mentioned in [9,37]). The decision tree and random forest algorithms achieved the best detection ability among other classifiers, with a detection accuracy of 85%

and a false-positive rate of 17%. The results cannot be considered as reliable for the real implementation of the proposed IDS.

Table 2 summarizes the existing IDSs for ICMPv6-DDoS attacks with their description and drawbacks.

Table 2. Summary of the existing IDSs for ICMPv6-DDoS attacks.

IDS	Description	Drawbacks
Snort [19]	Open-source IDS Uses the same policies for IPv6 and IPv6 protocols	Limited to its database of signatures Unable to detect self-modifying or zero-day attacks Evadable using extension header Inaccurate in detecting DDoS attacks
Zeek [20]	Open-source IDS Uses the same policies for IPv6 and IPv6 protocols Allows users to write their own rules	Limited to its database of signatures Unable to detect zero-day and self-modifying attacks Requires a huge amount of resources Slow in analyzing traffic
Suricata [25]	Open-source IDS Uses the same policies for IPv6 and IPv6 protocols Supports multithreading	Limited to its database of signatures Unable to detect zero-day and self-modifying attacks Evadable using fragmentation or padding Consumes machine memory Inaccurate in detecting DDoS attacks
Barbhuiya et al. [31]	Detects NS and NA address spoofing Uses 6 tables Sends probe packets	Consumes network resources. Limited to attacks of NS and NA packets. Changing NIC card IP address is not allowed Unable to detect attacks from genuine IP address
Bansal et al. [32]	Improved IDS of Barbhuiya et al., [31] Detects NS and NA address spoofing Uses 4 tables Send probe packets	Consumes network resources. Limited to NS and NA attacks. · NIC card changing IP address is not allowed Unable to detect attacks from genuine IP address
Li et al. [33]	Detects NDP protocol attacks Uses fuzzy logic Low false rate (2%)	Limited to NDP attacks. Uses non-consistent traffic dataset Few details are given Unreliable detection accuracy (85%)
Lai et al. [34]	Detects IPv6 DDoS attacks. Uses Apriori algorithm. Uses 6 packets features.	Unreliable detection accuracy (72.2%). Uses a small dataset for testing (5000 records). Depends on irrelevant features such as IPv6 address.
Zulkiflee et al. [35]	Detects IPv6 DDoS attacks. Uses SVM classifier. High detection accuracy (99.95%). Uses 5 packets features.	Detects RA DoS only. Few datasets and experiments details are given. Depends on irrelevant features such as IPv6 address
Redhwan et al. [36]	Detects ICMPv6 DDoS attack. Uses BBNN classifier. High detection accuracy (98.3%) Uses 10 features of packets traffic	Small testing dataset. Limited to DDoS attacks of ICMPv6 ECHO request. Limited attacks' scenarios. Depends on irrelevant features such as IPv6 address
Alsadhan et al. [10]	Detects NDP DDoS attacks Uses flow representation of traffic Depends on 12 flow features.	Limited to NDP DDoS attacks. Unreliable detection accuracy (84%) No features ranking was used.
Anbar et al. [17]	Detects RA flooding attacks using IG and PCA for feature selection and SVM as a classifier.	Detects RA DoS only and relies on packets representation for detection.
Elejla et al. [4]	Detects ICMPv6 flooding DDoS attacks using ensemble feature selection mechanism and LSTM.	Lacks significant flow based features that contribute to the ICMPv6 DoS/DDoS attacks detection.
Hammoodi et al. [5]	Detects RA flooding attacks using ensemble feature selection mechanism and RNN.	Detects RA DoS only and relies on packets representation for detection.
Elejla et al. [8]	Detects ICMPv6 DDoS attacks Uses flow representation of traffic Depends on 11 flow features.	Unreliable detection accuracy (85%) No features ranking is used.

Each of the proposed IDSs that aims to detect IPv6 DDoS attacks has its limitations and drawbacks, as shown in Table 2. Flow representation has proven to be a promising technique for detecting such attacks. However, the existing flow-based IDSs must be further improved to detect ICMPv6 attacks accurately. Therefore, this paper aims to extend the flow-based IDS proposed by Elejla et al. [8] by improving its detection accuracy regarding the ICMPv6-DDoS attacks. We chose this particular IDS because it is the first flow-based

IDS with comprehensive features to detect ICMPv6 DDoS attacks with a well-documented systematic methodology and sufficient dataset size.

The proposed approach has three main advantages compared to the existing approaches listed in Table 2. The proposed approach (i) relies on flow-based traffic representation, which is the most suited for today's high-speed networks; (ii) preserves user privacy since the flows are void of personally identifiable information, such as source and destinations IP addresses, unlike the existing packet-based approaches that rely on the source and destinations IP addresses for attack detection; and (iii) introduces enriched flow-based features that significantly contribute to the detection of ICMPv6-DDoS attacks.

4. Proposed Flow-Based IDS

The proposed flow-based IDS consists of five interconnected stages that are responsible for achieving the research objectives. Each stage consists of several steps. Generally, the IDS works by passively capturing the network traffic and then performing a further analysis to detect DDoS attacks that exploit ICMPv6 messages. This IDS aims to detect ICMPv6-DDoS attacks on IPv6-link local networks. Figure 1 illustrates the architecture of the proposed flow-based IDS.

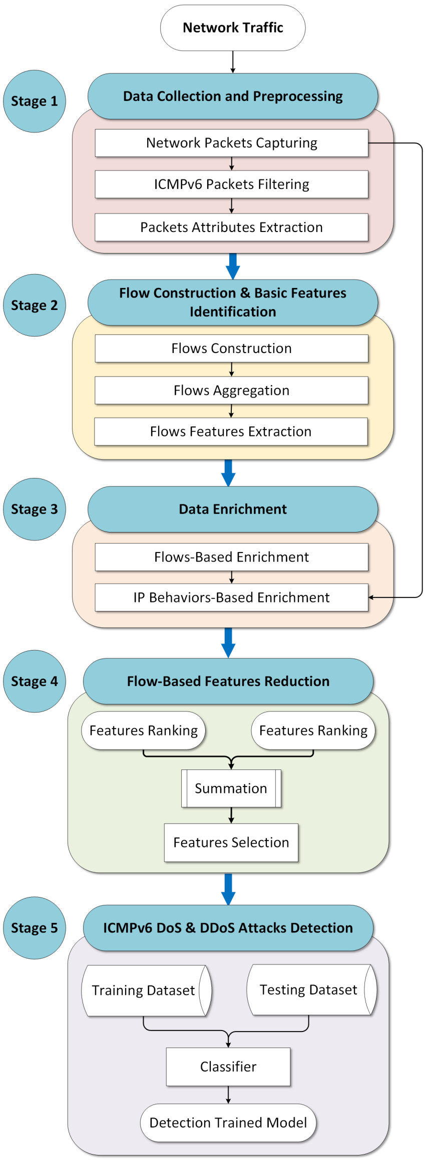


Figure 1. The proposed flow-based IDS architecture.

4.1. Data Collection and Preprocessing Stage

The proposed IDS starts with capturing and preprocessing the network traffic. This stage captures the network traffic and organizes it into a suitable presentation, as well as makes it free of noise to be ready as an input for the rest of the proposed IDS. The stage output is a set of packet attributes needed to construct flows and extract features. In this stage, the traffic with complete packet information is captured and collected for further processing. The collected packets are filtered using Wireshark filtering commands (i.e., ip.version==6 and ipv6.nxt==58) to exclude any IPv4 or non-ICMPv6 packets, which are out of the research scope. Figure 2 depicts the sequential steps for the design and implementation of stage one.

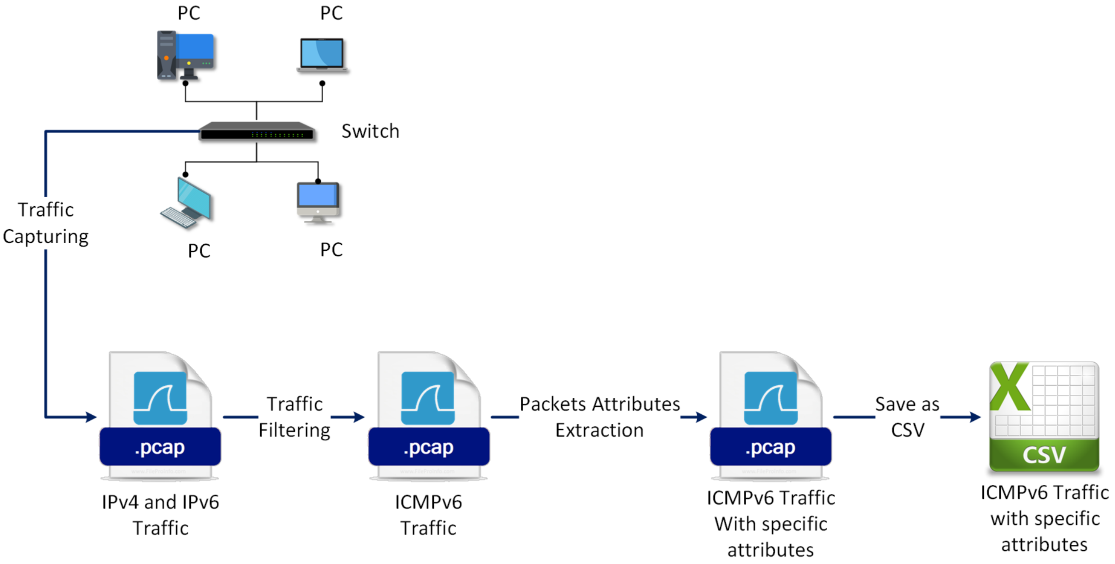


Figure 2. The sequential steps for the design and implementation of stage one.

After the traffic is filtered using the Wireshark tool, it is exported as a CSV file for further processing by the subsequent stages, as shown in Figure 2. This stage consists of three steps as follows:

4.1.1. Network Packet Capturing Step

This step captures all ingress and egress traffic in a row packet format without losing any information. The Network Interface Card (NIC) of the monitoring device is put into "Promiscuous" mode to receive all packets in the network regardless of their source and destination IP addresses. In addition, the switch interface attached to the monitoring device is configured as "Mirroring mode". Mirroring mode allows the device interface to capture all incoming or outgoing packets that pass through the network. Let us assume that the captured traffic in this step is T_{input}, representing a series of N packets P. Each P has a number of attributes x. Thus, the packet can be represented in terms of attributes as $P(x_1, x_2, x_3, \ldots, x_A)$. T_{input} is sent to the next step for further processing. The mathematical notation of T_{input} is shown in Equation (1).

$$T_{input} = P_1(x_1, x_2, x_3, \ldots, x_A); P_2(x_1, x_2, x_3, \ldots, x_A); \ldots; P_n(x_1, x_2, x_3, \ldots, x_A) \tag{1}$$

where: P_{i} is a captured packet.

N is the length of the input traffic (number of packets).

x_{i} is a packet attribute.

x_1 and x_2 are packet version and next header attributes, respectively.

4.1.2. ICMPv6 Packet Filtering Step

The packet filtering step is critical due to the processing complexity that unnecessary packets can add. Moreover, including such packets might confuse or complicate any advanced processes and lead to low detection accuracy. Since the proposed IDS aims to secure the IPv6 protocol, IPv4 messages are filtered out and dropped from the traffic. Moreover, to extend the filtering process, all messages other than ICMPv6 are ignored from the traffic. This step filters the captured traffic to include only IPv6 ICMPv6 packets, which have a frame version value equal to 6, and a next header value equal to 58).

By considering the previous mathematical notation, the output from this step is a new series $\mathrm{T_{output}}$ with n number of packets, where n <= N. Therefore, $\mathrm{T_{output}}$ can be defined in Equation (2):

$$T_{output} = \left\{ \begin{array}{c} P_1(x_1, x_2, x_3, \ldots, x_A); P_2(x_1, x_2, x_3, \ldots, x_A); \ldots; P_n(x_1, x_2, x_3, \ldots, x_A) \\ \text{if } x_1 = 6 \text{ and } x_2 = 58 \end{array} \right\} \tag{2}$$

4.1.3. Packets Attribute Extraction Step

This step extracts the required packet's header fields for feature extraction or flow construction. A vector of IPv6's header fields identified and extracted from the input traffic, T_{output}, is used as the input for the next stage. These attributes are identified based on their contribution to the flow construction or the next stage's basic flow feature extraction process. Unused attributes are filtered out and ignored. The first extracted packet feature is the packet-receiving time, which is necessary for calculating the flow duration time and other features. Additional features extracted include the IPv6 source and destination addresses and ICMPv6 type due to their importance in the flow construction. Flow label, packet length, next header, hop limit, payload length, and traffic class features are extracted to extract basic flow features. The output of this step, τ_{output}, has the same packet number n of T_{output} with a smaller number of attributes. T_{output} packets have A number of attributes, whereas τ_{output} packets will have a number of attributes, where $a \ < A$. Equation (3) defines the output of this step.

$$\tau_{output} = \left\{ \begin{array}{c} P_1(x_1, x_2, x_3, \ldots, x_A); P_2(x_1, x_2, x_3, \ldots, x_A); \ldots; P_n(x_1, x_2, x_3, \ldots, x_A) \\ \text{if } x_1 = 6 \text{ and } x_2 = 58 \end{array} \right\} \tag{3}$$

4.2. Flow Construction and Basic Features Identification Stage

This stage converts the packets processed and prepared in the first stage to flows. In addition, it prepares the flows for building the flow-based datasets and applying enrichment techniques in the third stage. Moreover, it extracts flow features for attack detection in the fifth stage.

4.2.1. Flow Construction Step

This step represents the packets in a flow representation. The flow of ICMPv6 packets is defined as "Packets that have the same source and destination IPv6 address (IP_{src} and IP_{dst}) and the same ICMPv6 type (ICMPv6Type) sent within the same interval of time (T)" [8,9]. Therefore, the ICMPv6 flow can be stated as

$\mathrm{F_{ICMPv6}} = (\mathrm{IP_{src}}, \mathrm{IP_{dst}}, \mathrm{ICMPv6Type})_{\mathrm{T}}$

The output of this step is a new flow representation of traffic based on the defined flow. The new represented traffic should be smaller in size than the original representation, which is one of the good characteristics of flow representation. This compact representation of network traffic will contribute to increasing the analysis efficiency of the flows using the approach.

4.2.2. Flow Aggregation Step

Currently, F_{ICMPv6} flows have three key elements (attributes): the IPv6 source address, IPv6 destination address, and ICMPv6 type. This step adds to the F_{ICMPv6} potential features by extracting attributes from the packets received from stage one. These potential features provide each flow of new information, strengthening its representativeness in detecting the attacks. The identification of new flow attributes, which will be used to extract the features, is based on practical experiments and a domain understanding of the DDoS attacks and the ICMPv6 protocol as mentioned in [8]. The selected flow attributes are grouped based on selection justification to contribute to detecting ICMPv6-DDoS attacks. The extracted attributes are the number of transferred packets (PacketsNumber), number of transferred bytes (TransferredBytes), flow duration, transferred bytes ratio (Ratio), and standard deviations of length (Length_STD), flow label (FlowLabel_STD), hop limit (HopLimit_STD), traffic class (TrafficClass_STD), next header (NextHeader_STD), and payload length (PayloadLength_STD). Calculations and reasons behind extracting these attributes are explained in detail in [8,16].

4.2.3. Flow Features Extraction Step

The flow features extraction is the final step in the flow construction and basic features identification stage. The step aims to identify and extract a number of basic features for ICMPv6-DDoS attack detection. The ICMPv6Type flow attribute, already included in the flow construction step, is one of the extracted features in this step. This feature is helpful for distinguishing between ICMPv6-DDoS attack types, as well as due to their expected contribution to detecting flows of ICMPv6-DDoS attacks. IPv6 source and destination addresses are supposed to be two distinct patterns that identify unique flows but have a problem distinguishing between each flow in the existing IDSs, as criticized in [16]. Therefore, this step excluded or ignored them from the flow features.

The final output of the stage is basic features derived from the flows. Each has a different meaning and indication in the normal flow and flows of ICMPv6-DDoS attacks. The 11 ICMPv6 basic flow features are the ICMPv6 type, packets number, transferred bytes, flow duration, bytes ratio, standard deviation of flow labels, lengths, traffic classes, hop limits, and payload length, as shown in [16].

4.3. Data Enrichment Stage

The required information for DDoS attack detection does not necessarily have to come from packets exchanged between two particular nodes (attacker and victim). DDoS attacks can also be accurately identified using more general information about the flows. Therefore, this stage aims to add extra features to the flow's traffic to enrich them with general information linking the flows to the behaviors of the flows' IP source addresses. We believe that reasonably chosen features can significantly enrich the flows, improving the accuracy in detecting ICMPv6-DDoS attacks. Figure 3 illustrates the proposed data enrichment stage.

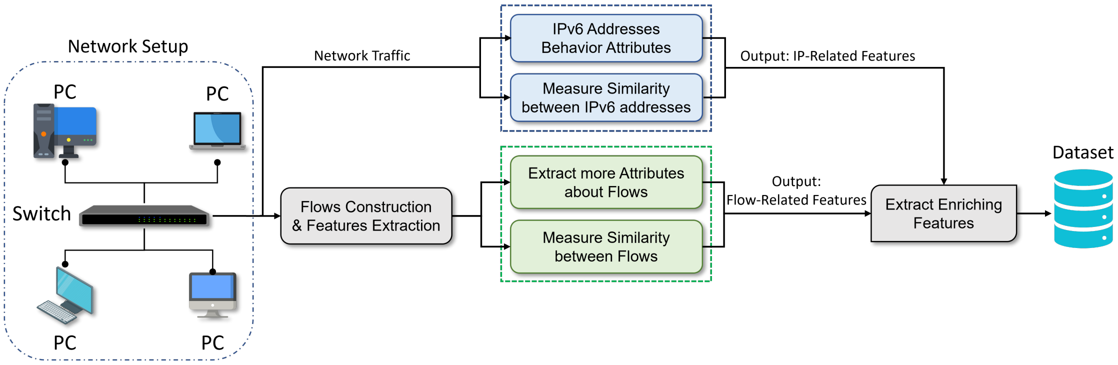

Network Setup
PC
PC
Switch
PC
PC
Network Traffic
IPv6 Addresses Behavior Attributes
Measure Similarity between IPv6 addresses
Output: IP-Related Features
Flows Construction & Features Extraction
Extract more Attributes about Flows
Measure Similarity between Flows
Output: Flow-Related Features
Extract Enriching Features
Dataset

Figure 3. The proposed flow-based IDS architecture.

The enrichment stage consists of two steps to collect metadata about IP sources addresses and the network. The first step is the flow-based enrichment step, which concerns the input flow traffic to collect and calculate more attributes from the traffic. The second step is the IP behavior-based enrichment step, which aims to collect metadata about the IP source address behaviors. The two steps are applied to benefit from their extracted features and enrich the flow's traffic with more valuable features. It's assumed that these features can improve the detection accuracy of the classifiers. The features are combined with the basic flow features extracted in the previous stages.

4.3.1. Flow Enrichment Step

Representing the traffic in a flow format provides a broader picture of packets with the same IP addresses and ICMPv6 type. However, DDoS attacks can be performed by sending packets from different spoofed IP sources and destination addresses or using different types of ICMPv6 messages. Detecting these attacks requires having a broader view of the traffic. Therefore, this step aims to enrich the flows with more features for attack detection. These features provide each flow with additional information, such as (i) the number of previous flows targeting the same destination IPv6 address (Flows_Same_IPdst) and (ii) the number of previous flows using the same ICMPv6 type (Flows_Same_ICMPv6Type). These features provide information about attacks sent from the same IPsrc or ICMPv6type regardless of other fields' values.

DDoS attack packets have many similarities in several attributes because attackers typically auto-generate attack traffic using fixed default parameters. Therefore, we can find flow similarity by determining the similarity between the flow and its previous flows based on their entire flow values. Flow similarity (Flow_Similarity) is the third feature added as another enriching feature. This feature helps to recognize attacks with similar fields' values other than IPsrc or ICMPv6type.

4.3.2. IP Behavior-Based Enrichment Steps

The characteristics and behaviors of the IPsrc addresses can be a strong indicator of attacks' existence or absence. Many researchers [4,5,17] use IPsrc address characteristics or behaviors to detect different attack types, including DDoS. Therefore, enriching the flows with information about IPsrc address behaviors should increase the flow's representativeness. This step focuses on enriching the flows with information that helps to detect attacks from fake or spoofed addresses. Fake IPsrc addresses are used to avoid the traceback and identification of the original attacker. The main characteristic of these IP addresses is that they appear suddenly in the traffic since they only exist during attacks. Based on this fact, this step determines the first-seen and last-seen times for each IP source address. We posit that DDoS attacks' IP source addresses are usually randomly generated fake IPsrc, which have a recent last-seen time that is almost the same as the first-seen time. An enriching feature that indicates how recently the IP source is seen in the network is named IP source first-seen (IPsrc_First_Seen). Its value, calculated by measuring the difference between the IPsrc time of first-seen and current time, represents the number of seconds that lapsed since the IP source address was first detected.

Attackers use fake IP addresses randomly generated by replacing the host part of the IPv6 address while the network prefix part remains the same. The generated fake IP addresses' network part typically remains the same but with a different host part, meaning that the generated IP source addresses look similar in most of their bit values. This step calculates the similarity between each flow's IP source address and the IP source addresses of the previous flows, which will be added as a new feature named IP source similarity (IPsrc_Similarity). This feature shows the similarity percentage between the IPsrc of each flow with the IPsrc of its earlier flows.

In summary, the final output of this stage is five new features that enrich the flows, which are Flows_Same_IPdst, Flows_Same_ICMPv6Type, Flow_Similarity, IPsrc_First_Seen, and IPsrc_Similarity. The total number of features of each flow is 16 features, comprising 11 basic features and 5 enriching features. Table 3 shows a summary of the 16 features.

Table 3. Summary of the 16 features.

Feature Name	Type	Description
ICMPv6Type	Basic	Type of ICMPv6 packet, e.g., NA
PacketsNumber	Basic	Number of packets that satisfy the definition of flow
TransferredBytes	Basic	Number of bytes within the flow
Duration	Basic	Time difference between the last and first packet in the flow
Ratio	Basic	Ratio of number of bytes in the flow with the duration
Length_STD	Basic	Standard deviation of packets' lengths
FlowLable_STD	Basic	Standard deviation of packets' flow labels
HopLimit_STD	Basic	Standard deviation of packets' hop limits
TraffiicClass_STD	Basic	Standard deviation of packets' traffic classes
NextHeader_STD	Basic	Standard deviation of packets' next headers
PayloadLength_STD	Basic	Standard deviation of packets' payload lengths
Flows_Same_IPdst	Enriching	Number of previous flows sent to the same IPdst
Flows_Same_ICMPv6Type	Enriching	Number of previous flows sent with the same ICMPv6 type
Flow_Similarity	Enriching	Similarity percentage between the previous flows
IPsrc_First_Seen	Enriching	Time duration of IPsrc appearing for the first time
IPsrc_Similarity	Enriching	Similarity percentage between IPsrc and previous IPsrcs

This step aims to form a set of enriching flow features extracted for each flow in the previous two steps. Five features were extracted in the previous two steps to improve the flow's traffic quality. This stage aims to prepare these features for the subsequent stages by combining them with the 11 basic flow features as one set of flow-enriching features in order to have the final set of features. However, these features might be redundant and produce a high dimensionality that could reduce the classifier's performance and detection accuracy. Therefore, these redundant features are reduced using the feature selection and ranking technique in the following stage.

4.4. Flow-Based Feature Reduction Stage

Although each feature of the 16 final flow features was chosen based on reasoning and reasonable justifications, some might be redundant or do not contribute significantly to the attack detection. Therefore, we must choose the most relevant features to avoid misclassification due to the non-contributing features. Moreover, the feature reduction aims to reduce the classifier's training times in building their models. We used hybrid feature ranking techniques to choose the best features that help to distinguish attack and non-attack (normal) flows. The stage's output is the selected features out of 16 features.

Two of the most common feature ranking algorithms were hybridized in order to have a combination of more than one opinion. Combining two ranking schemes that work in parallel allows us to have different points of view about the contributed features. We combined IGR and CHI-squared technique (CHI) ranking algorithms to select the most relevant features. The CHI ranking algorithm is a statistics feature selection method, and IGR measures the dependence between the feature and the class label. Such feature selection methods help to select a set of strong features that satisfy both selection aspects. In addition, the idea of hybridizing two feature selection algorithms has been employed in existing research to detect ICMPv6-based attacks, such as in [17], achieving an impressive accuracy in detecting ICMPv6-DDoS attacks.

Each ranking algorithm ranks the features based on their relevance to the class label. Therefore, the ranking algorithms give each feature a rank value between 1 to the number of features (16). Each feature gains rank values from IGR selection (R_{IGR}) and CHI selection (R_{CHI}). The feature with a high rank means that it is a significant feature and can contribute to the detection of DDoS attacks, whereas the feature with a low rank can be neglected.

To benefit from these ranks, a summation of the rank values was calculated to combine the two ranking results. Thus, the maximum value that can be achieved by a feature from the two algorithms is 32. Therefore, a feature's R_{IGR}+R_{CHI} value must be larger than 16 to be selected. These selected features were assumed to be the most relevant features that can differentiate between the flow of ICMPv6-DDoS attacks and normal flows. Therefore, the features were passed to the next stage, which is building a detection model based on them.

4.5. ICMPv6-DDoS Attack Detection Stage

The attack detection stage trains the classifiers to build detection models that can detect ICMPv6-DDoS attacks based on the structured flows. The flows were combined with the selected features to build the flow datasets. The classifiers were applied to the flow dataset to build detection models. The dataset included various scenarios of ICMPv6-DDoS attacks and normal traffic that allow for the classifiers to learn all of the potential possibilities. Therefore, this stage output was the trained models to be implemented online in IPv6 networks to detect any ICMPv6 DDoS attacks. The cross-validation testing technique was used, where the whole dataset was used to train and test the classifiers by splitting a part of it for training and the rest for testing by a specified ratio. In addition, the supplied set testing technique was used by training classifiers on the whole training dataset and testing the built model on another dataset purposely created for testing purposes. In other words, the testing dataset's flow traffic is entirely different from the training dataset's flow traffic.

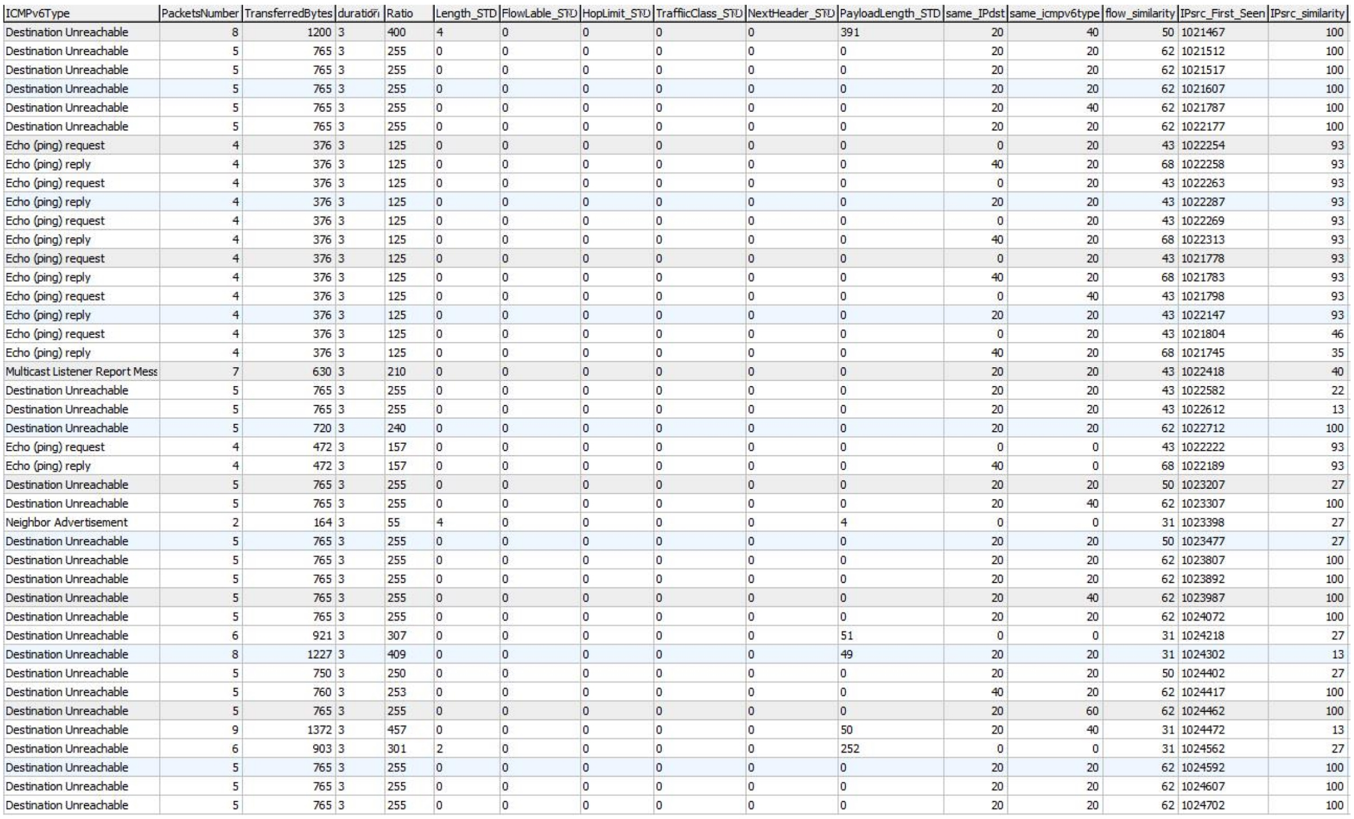
5. Analysis of Results and Discussions

This section presents the used flow datasets represented with the selected features (shown in Section 4.4). It also explains the metrics used to evaluate the proposed IDS, followed by the results of feature reduction and applying the classifiers to the datasets. Analyzing the results allows us to evaluate the efficiency of the proposed IDS.

5.1. Flow Dataset

The dataset used to evaluate the proposed IDS was obtained from [16]. The datasets were validated to make sure that the datasets were suitable for evaluating the IDS to detect all ICMPv6-DDoS attacks and that the given performance of the system was correct and expressive. Moreover, the availability of good datasets requirements was ensured in the dataset. Lastly, different classifiers were chosen and applied to the datasets to check their ability to achieve acceptable detection accuracies. Figure 4 shows a snapshot of the used flow-based dataset.

Five requirements of a good dataset proposed by Sperotto et al. [37] can be used to define a reliable dataset. The dataset must contain realistic traffic, diverse scenarios, complete and correct labels, a sufficient size, and representative features. The used dataset contained real-life IPv6 traffic from a university network comprising real IPv6-enabled nodes. It also included diverse attack and non-attack (normal) traffic scenarios with their responses included. We conducted three processes for dataset preparation: normalization, labeling, and balancing. The dataset was already correctly labeled as either "attack" or "normal" for malicious or normal packets. Balancing the dataset involves using the synthetic minority over-sampling technique (SMOTE) to increase traffic data by replicating fewer labeled records. Table 4 shows the characteristics of the used dataset.



ICMPv6Type	PacketsNumber	TransferredBytes	duration	Ratio	Length_STD	FlowLable_STD	HopLimit_STD	TrafficClass_STD	NextHeader_STD	PayloadLength_STD	same_IPdst	same_icmpv6type	flow_similarity	IPsrc_First_Seen	IPsrc_similarity
Destination Unreachable	8	1200	3	400	4	0	0	0	0	391	20	40	50	1021467	100
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	62	1021512	100
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	62	1021517	100
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	62	1021607	100
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	40	62	1021787	100
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	62	1022177	100
Echo (ping) request	4	376	3	125	0	0	0	0	0	0	0	20	43	1022254	93
Echo (ping) reply	4	376	3	125	0	0	0	0	0	0	40	20	68	1022258	93
Echo (ping) request	4	376	3	125	0	0	0	0	0	0	0	20	43	1022263	93
Echo (ping) reply	4	376	3	125	0	0	0	0	0	0	20	20	43	1022287	93
Echo (ping) request	4	376	3	125	0	0	0	0	0	0	0	20	43	1022269	93
Echo (ping) reply	4	376	3	125	0	0	0	0	0	0	40	20	68	1022313	93
Echo (ping) request	4	376	3	125	0	0	0	0	0	0	0	20	43	1021778	93
Echo (ping) reply	4	376	3	125	0	0	0	0	0	0	40	20	68	1021783	93
Echo (ping) request	4	376	3	125	0	0	0	0	0	0	0	40	43	1021798	93
Echo (ping) reply	4	376	3	125	0	0	0	0	0	0	20	20	43	1022147	93
Echo (ping) request	4	376	3	125	0	0	0	0	0	0	0	20	43	1021804	46
Echo (ping) reply	4	376	3	125	0	0	0	0	0	0	40	20	68	1021745	35
Multicast Listener Report Mess	7	630	3	210	0	0	0	0	0	0	20	20	43	1022418	40
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	43	1022582	22
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	43	1022612	13
Destination Unreachable	5	720	3	240	0	0	0	0	0	0	20	20	62	1022712	100
Echo (ping) request	4	472	3	157	0	0	0	0	0	0	0	0	43	1022222	93
Echo (ping) reply	4	472	3	157	0	0	0	0	0	0	40	0	68	1022189	93
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	50	1023207	27
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	40	62	1023307	100
Neighbor Advertisement	2	164	3	55	4	0	0	0	0	4	0	0	31	1023398	27
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	50	1023477	27
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	62	1023807	100
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	62	1023892	100
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	40	62	1023987	100
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	62	1024072	100
Destination Unreachable	6	921	3	307	0	0	0	0	0	51	0	0	31	1024218	27
Destination Unreachable	8	1227	3	409	0	0	0	0	0	49	20	20	31	1024302	13
Destination Unreachable	5	750	3	250	0	0	0	0	0	0	20	20	50	1024402	27
Destination Unreachable	5	760	3	253	0	0	0	0	0	0	40	20	62	1024417	100
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	60	62	1024462	100
Destination Unreachable	9	1372	3	457	0	0	0	0	0	50	20	40	31	1024472	13
Destination Unreachable	6	903	3	301	2	0	0	0	0	252	0	0	31	1024562	27
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	62	1024592	100
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	62	1024607	100
Destination Unreachable	5	765	3	255	0	0	0	0	0	0	20	20	62	1024702	100

Figure 4. Snapshot of the flow-based dataset using the final flow features.

Table 4. The characteristics of the used dataset.

Dataset	Total Packets	Total Flows	Attack Flow	Normal Flows
Flow-based Training Dataset	200,854	101,088 flows	49,187	51,901
Flow-based Testing Dataset	199,137	92,640 flows	42,084	50,556

Elejla et al. [16] ensured that their dataset fulfils a good dataset's requirements. The dataset size was sufficient, and the oversampling balancing technique was applied. In addition, the features were chosen based on reasonable justifications and explanations to ensure their representativeness.

Elejla et al. [16] applied a set of classifiers to classify the built flow dataset using two different testing approaches. Ten-fold cross-validation trained the classifiers on a portion of the dataset and then tested the trained models on the other portion. A supplied set test was another technique applied to train the classifiers on the dataset and then test the trained model on another dataset. This checking confirmed that the extracted features could differentiate between ICMPv6-DDoS and normal traffic. Elejla et al. [16] conducted the checking using two evaluation techniques based on detection accuracy and false positive rate evaluation metrics. The achieved detection accuracy ranged between 85.83% and 73.96% and the false positive rate ranged between 31% and 17%. These results proved that the dataset applies to ICMPv6-DDoS attack detection. Moreover, it proved the ability of flow representation and the extracted features in detecting ICMPv6-DDoS attacks. However, there is still room for further enhancement to improve the classifiers' ability to detect attacks.

5.2. Evaluation Metrics

IDS is evaluated on its ability to classify input datasets correctly. In this work, we evaluated the capability of the proposed flow-based IDS to detect ICMPv6-DDoS attacks using several common evaluation metrics in the literature, including the detection accuracy, precision, recall, F-measure, and false-positive rate [4]. Moreover, the evaluated metrics for this IDS were compared with the existing research to evaluate the proposed flow-based IDS improvement. The evaluation metrics are as follows:

Detection Accuracy

The detection accuracy metric reflects the IDS's ability to classify flows correctly from all existing flows. It describes the ability of the IDS to raise the alarm when an attack is detected in the network and remain silent for normal network traffic. The detection accuracy is calculated using Equation (4).

$$DetectionAccuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

The average detection accuracy among all applied classifiers is calculated using Equation (5).

$$AverageDetectionAccuracy = \frac{\sum_1^c DetectionAccuracy_i}{c} \tag{5}$$

False-Positive Rate

The false-positive rate reflects the inability to correctly classify normal flows from the total number of normal flows. It describes the weakness of the IDS in correctly classifying the normal flows as normal. The false-positive rate is calculated using Equation (6).

$$FalsePositiveRate = \frac{FP}{TN + FP} \tag{6}$$

The average false positive rate among the applied classifiers is calculated using Equation (7).

$$AverageFalsePositiveRate = \frac{\sum_1^c FalsePositiveRate_i}{c} \tag{7}$$

Where c is the number of classifiers.

Precision is the proportion of attacks correctly predicted vs. all samples predicted as attacks. The precision is calculated using Equation (8).

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

Recall or the detection rate is the proportion of all samples correctly classified as attacks vs. all attack samples. The recall is computed using Equation (9).

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

F1-Measure is the harmonic mean of precision and recall. In other words, it is a statistical technique involving precision and recall used for examining a system's accuracy. The F1-measure is calculated utilizing Equation (10).

$$F - Measure = \frac{2 \times precision \times recall}{precision + recall} \tag{10}$$

The description of *TP*, *TN*, *FN*, and *FP* is shown in Table 5.

Table 5. Terms of the evaluation equations.

Short Term	Description
TP (True Positive)	The percentage of attack flows classified as attack
TN (True Negative)	The percentage of normal flows classified as normal
FN (False Negative)	The percentage of attack flows classified as normal
FP (False Positive)	The percentage of normal flows classified as attack

5.3. Results

This section presents the result of the features reduction stage and the analysis of the test results of the proposed approach by applying different ML classifiers. Then, it discusses the impact of the selected features on the detection model.

5.3.1. Result of Flow-Based Features Reduction Stage

All of the features included in the set of final flow features do not necessarily contribute to the detection of the attacks. Therefore, the final flow features were filtered out to have a subset of the strongly related features. The features reduction stage (shown in Section 4.4) of the proposed approach aims to choose the most significant set of features. Moreover, feature reduction should help to improve the training times needed to build the detection model since the classifiers will depend on fewer features to build their models. IGR and the chi-squared algorithm were applied to the flow dataset to choose the best features. Each of the algorithms gives weight to indicate the strength of the relationship between each feature and the classification decision. Table 6 presents the results of the features ranking algorithms that were applied to the flow dataset constructed using the 16 final flow features set.

Table 6. Features reduction results.

Feature Name	Chi-Squared Rank	Information Gain Rank	Summation of Ranks
ICMPv6Type	11	11	22
PacketsNumber	9	8	17
TransferredBytes	10	10	20
Duration	7	7	14
Ratio	8	9	17
Length_STD	5	5	10
FlowLable_STD	3	4	7
HopLimit_STD	2	3	5
TraffiicClass_STD	1	2	3
NextHeader_STD	4	1	5
PayloadLength_STD	6	6	12
Flows_Same_IPdst	14	15	29
Flows_Same_ICMPv6Type	12	12	24
Flow_Similarity	13	14	27
IPsrc_First_Seen	16	16	32
IPsrc_Similarity	15	13	28

As can be seen from Table 6, each of the ranking algorithms assigned rank (according to their given weight) values to each feature, indicating the relation between the feature and the class value. Then, the features that achieved a summation of rank greater or equal to 16 were selected to evaluate the proposed IDS using several variant classifiers. Based on this condition, seven features failed to meet the condition as their rank summation was less than 16, and were therefore excluded. The excluded features were FlowLable_STD, Length_STD, TraffiicClass_STD, HopLimit_STD, NextHeader_STD, PayloadLength_STD and Duration. The selected features are ICMPv6Type, PacketsNumber, TransferredBytes, Ratio, Flows_Same_IPdst, Flows_Same_ICMPv6Type, Flow_Similarity, IPsrc_First_Seen, and IPsrc_Similarity.

5.3.2. Result of ICMPv6-DDoS Attacks Detection Stage

The selected features were fed to the classifiers to build several prediction models. The generated prediction models were evaluated using two evaluation approaches: the cross-validation test and supplied set test. The classifiers were chosen, as they are well-known classifiers, as well as provided by the WEKA tool to avoid reimplementing them again. The chosen classifiers were decision trees, SVMs, naïve Bayes, KNNs, random forest trees, and neural networks. The classifiers were chosen from different classification technique categories to ensure the efficiency of the proposed IDS.

The classifiers were used with their default parameters without any parameter modification. Table 7 presents the classifiers' detection accuracies after applying them to the flow dataset built with the selected features using the two evaluation techniques.

Table 7. The classifiers detection accuracy with cross-validation and supplied set.

Classifier	Cross-Validation Test	Supplied Set Test
Decision Trees	99.98%	99.96%
Support Vector Machines (SVMs)	98.70%	98.65%
Naïve Bayes	98.56%	98.53%
K-Nearest Neighbors (KNNs)	99.56%	99.98%
Random Forest Trees	99.99%	98.83%
Neural Networks	99.92%	99.91%

Using a selected set of flow features representing the flow-based dataset allows the classifiers to achieve high detection accuracies in detecting the attacks. As shown in Table 7, the classifiers have detection accuracies ranging from 98.53% to 99.99%. Moreover, the average detection accuracy, calculated using Equation (5), is 99.45% and 99.31% for the cross-validation test and supplied set test, respectively, with a total average of 99.38%. These high accuracies confirm the ability of the selected flow features to differentiate normal and attack flows. Moreover, the classifiers gave almost the same values in both approaches, indicating that the selected features are robust regardless of the testing approach. In other words, the selected features are descriptive and informative enough to allow the classifiers to achieve the same results even when trained and tested using a different dataset (supplied set test). The results in Table 7 confirm that the flow representation with the selected features is sufficient to build a trustworthy and reliable IDS to detect ICMPv6-DDoS attacks. Figure 5 compares the detection accuracies of the proposed IDS with the IDS by Elejla et al. [8].

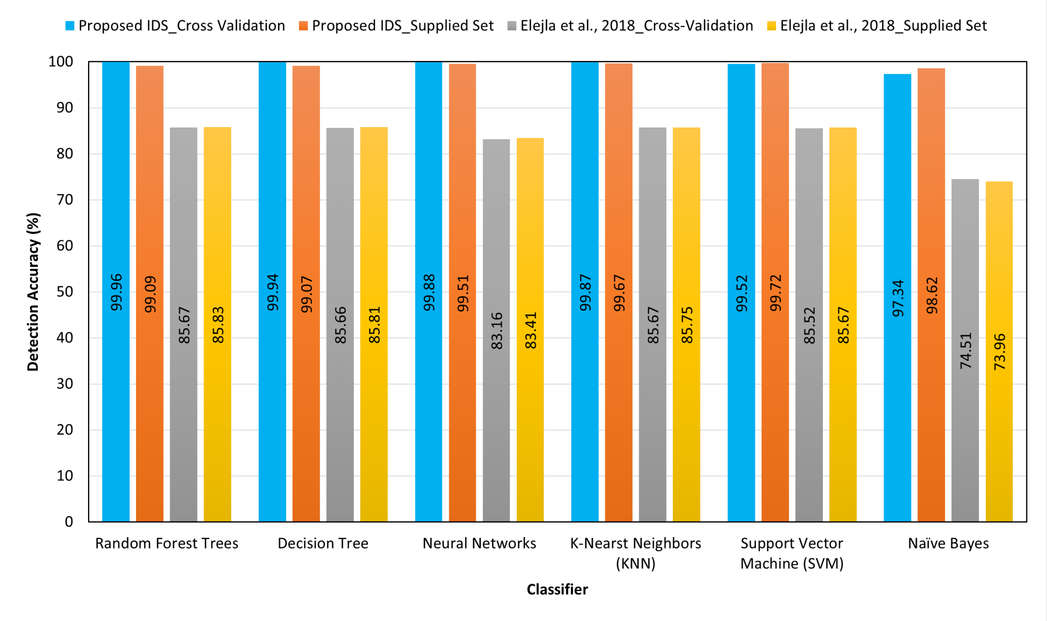


Figure 5. Comparison between detection accuracy of the proposed IDS and Elejla et al. [8].

Comparing the detection accuracy of the proposed IDS and IDS by Elejla et al. [8] showed that the enriched features considerably increase the average detection accuracy from 83.285% (using the basic flow features) to 99.414%, which is an enhancement of 16.02%. These enhancements will make any detection model built using the selected features more reliable and trustworthy.

Similarly, the false-positive rates were calculated for applying the classifiers to the dataset with the selected features. Table 8 shows the false-positive rate after applying the classifiers using two testing techniques to the flow datasets built with the selected flow features. The false-positive rates, which are close to zero, prove that we can build a detection model with a low false-positive rate based on classifiers that use the selected flow features. By calculating the average of these false-positive rates using Equation (7), they have a significantly small total average value of 0.585%, where the average for the cross-validation test equals 0.481% and, for the supplied set test, it equals 0.69%. Furthermore, the total average of false-positive rates for Elejla et al. [8] is 19.76%, where the average for the cross-validation test equals 19.71% and, for the supplied set test, it equals 19.81%. Therefore, the enriched features considerably decrease the false positive rate to 19.17%.

These results from both experiments showing low false-positive rates indicate that they are practical and that the built detection models are reliable and robust.

Table 8. The classifiers false positive rates with cross-validation and supplied set.

Classifier	Cross-Validation Test	Supplied Set Test
Decision Trees	0.02 %	0.04 %
Support Vector Machines (SVMs)	1.30 %	1.35 %
Naïve Bayes	1.44 %	1.47 %
K-Nearest Neighbors (KNNs)	0.04 %	0.02 %
Random Forest Trees	0.01 %	1.17 %
Neural Networks	0.08 %	0.09 %

Figure 6 depicts a comparison between the false-positive rates of the proposed IDS and the achieved ones in Elejla et al. [8] in order to show the gained enhancements. By comparing the classifiers' false positive rates using the two testing techniques in the cases of using the basic flow features by Elejla et al. [8] and the selected flow features to represent the datasets, it can be clearly seen that the false-positive rates drastically decreased compared to the previous results of the basic features (shown in [8]). The false positive rates are smaller than those achieved in [8] by 19.336%. The low false-positive rates confirm the enriching features' capability to provide the flow with information that is able to decrease the false-positive rates that might be generated from the classifiers. The evaluation of the proposed approach in terms of the precision, recall, and F-measure using the cross-validation and supplied test are shown in Figures 7 and 8, respectively.

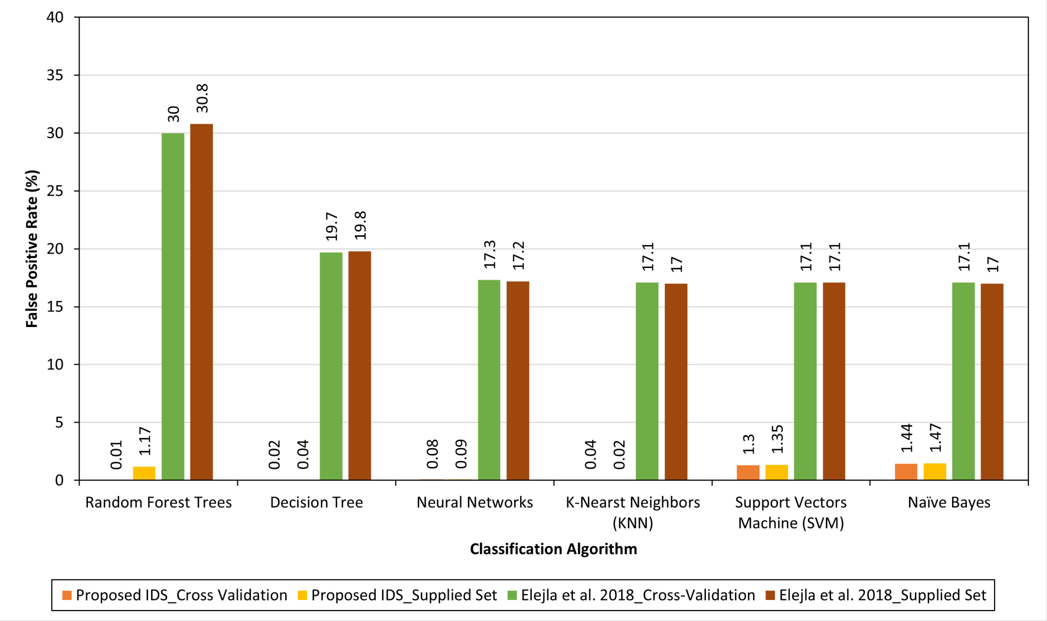


Figure 6. Comparison between false-positive rates of the proposed IDS and Elejla et al. [8].

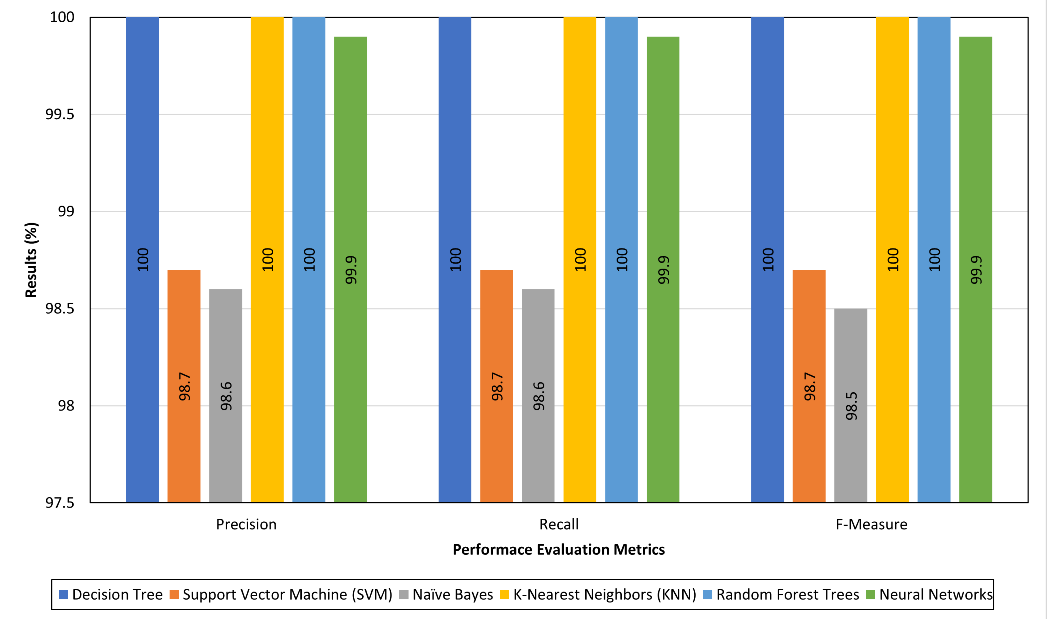


Figure 7. Results of the proposed IDS in terms of precision, recall, and F-measure for cross_validation scenario.

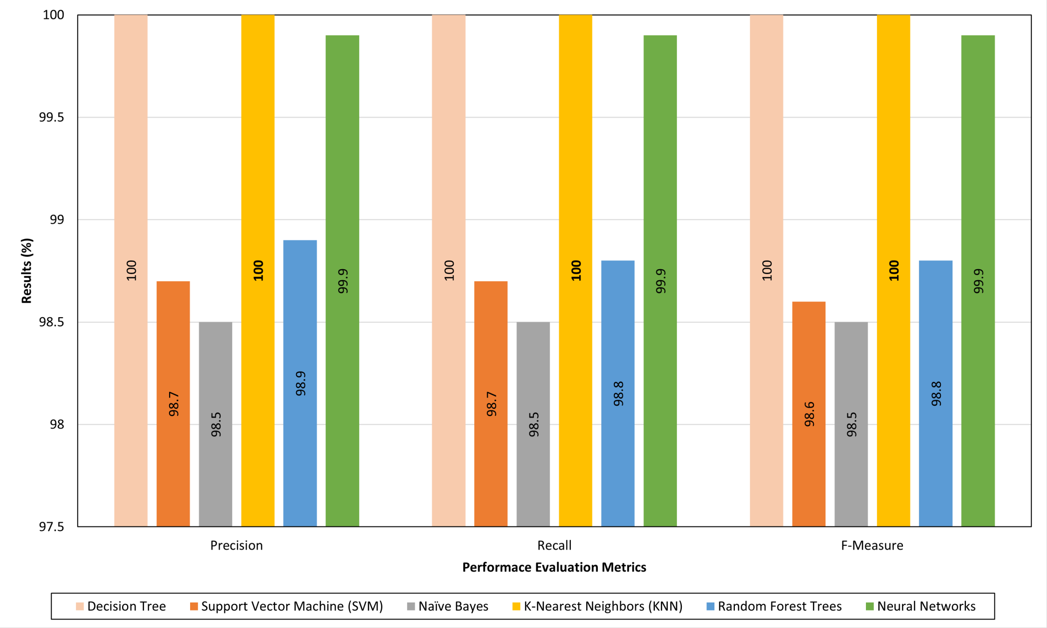


Figure 8. Results of the proposed IDS in terms of precision, recall, and F-measure for supplied_test scenario.

As shown in Figures 7 and 8, the total average precision equals 99.53% and 99.33% for the cross-validation test and supplied set test, respectively, with a total average of 99.38%. Meanwhile, the total average recalls are 99.53% and 99.31% for the cross-validation test and supplied set test, respectively, with a total average equalling 99.4%. Furthermore, the total average F-measure equals 99.30% and 99.40% for the cross-validation test and supplied set test, respectively, with the total average equal to 99.40%. The high precision percentage means that the proposed approach has a lower false-positive value. Meanwhile, the high recall percentage indicates that the proposed approach has a lower false-negative value.

5.4. Discussion

The improvement that occurs when enriching the basic features proves the efficiency of the added set of informative features. The classifiers' detection accuracies and the false-positive rates improved by 16.25% and 19.33%, respectively, after combining the enriched flow with the basic flow features. The improvement of adding the enriched features was further evaluated using the T-test, with the α value equal to 0.05. The T-test was applied to the 12 accuracy values of Elejla et al. [8] compared to the 12 accuracy values shown in Table 7. The p-value of the T-test between the accuracy of the selected and basic features of Elejla et al. [8] is 2.9986×10^{-10} The T-test result is smaller than α, showing a significant improvement in the classifiers' accuracies using the proposed enriched features compared to the basic features. These improvements indicate that enriching features add more information to the flows that help classifiers to achieve a better detection accuracy. Applying the similar T-test to the false positive results achieved a p-value of 2.2895×10^{-10}, which is smaller than the α value, confirming a significant improvement by adding the enriched features. The enriched flow features successfully help classifiers to detect attacks more accurately.

The robustness of the proposed flow-based IDS was evaluated using the supplied set test, where the test data do not exist in the training dataset. This mechanism allows the IDS to detect attacks in traffic seen for the first time and not in the training dataset. Thus, it simulates the online implementation of the IDS, where the trained model is set up to detect attacks from traffic that are different from the training traffic. The results achieved by the proposed flow-based IDS on the two testing techniques prove its efficiency in detecting attacks. This approach is limited to detecting ICMPv6-based DoS/DDoS attacks. However, other attacks that exploit the security vulnerabilities of ICMPv6, such as man-in-the-middle attacks (e.g., RA fake router attack), were ignored because they only involve the exchange of a few ICMPv6 messages and, thus, are not considered as DoS or DDoS attacks.

6. Conclusions

This paper proposed a solution to some IPv6 security problems related to the ICMPv6 protocol. The ICMPv6 protocol is susceptible to several attacks, such as DDoS attacks, which are the most popular among adversaries. Therefore, we proposed an IDS to detect DDoS attacks that exploit ICMPv6 messages by attempting to improve the accuracy of the previously proposed IDS to be more reliable with a better detection accuracy and a lower error rate. The proposed IDS extracts enriched features linking the flows to the behavior of the flows' IP source addresses. The combination of the enriched features and the basic flow features resulted in 16 flow features. The whole feature set was ranked to select the most contributing features in detecting ICMPv6-based DoS and DDoS attacks. The experimental results reveal that the enriched features significantly improved the IDS's detection accuracy by 16.02%, and that the false positive rate decreased by 19.17% compared with state-of-the-art IDSs. For future research directions, deep-learning algorithms can be used as classifiers to evaluate the impact of the newly proposed flow-based feature on detection accuracy. Furthermore, the applicability of the proposed enriched features in detecting DoS and DDoS attacks on other modern network architectures, such as Software-Defined Networking (SDN), can be explored. Lastly, the proposed approach can be enhanced to detect other ICMPv6-based attacks, such as man-in-the-middle.

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

Funding: This research is funded by Universiti Sains Malaysia (USM) via external grant (number 304/PNAV/650958/U154).

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. Elejla, O.E.; Belaton, B.; Anbar, M.; Smadi, I.M. A New Set of Features for Detecting Router Advertisement Flooding Attacks. In Proceedings of the 2017 Palestinian International Conference on Information and Communication Technology (PICICT), Gaza, Palestine, 8–9 May 2017; pp. 1–5. [CrossRef]
2. Bahashwan, A.A.; Anbar, M.; Hanshi, S.M. Overview of IPv6 Based DDoS and DoS Attacks Detection Mechanisms. In *Communications in Computer and Information Science*; Springer: Singapore, 2020; Volume 1132 CCIS, pp. 153–167. [CrossRef]
3. Conta, A.; Deering, S. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. RFC 4443. 2006. Available online: https://www.rfc-editor.org/info/rfc4443 (accessed on 14 September 2022). [CrossRef]
4. 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. [CrossRef]
5. Hammoodi, A.; Mohammed, H.; Taief, A.; Alamiedy, A. Deep learning approach for detecting router advertisement flooding-based DDoS attacks. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 1–15. [CrossRef]
6. Hoque, N.; Bhuyan, M.H.; Baishya, R.; Bhattacharyya, D.; Kalita, J. Network attacks: Taxonomy, tools and systems. *J. Netw. Comput. Appl.* **2014**, *40*, 307–324. [CrossRef]
7. 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]
8. Elejla, O.E.; Anbar, M.; Belaton, B.; Alijla, B.O. Flow-Based IDS for ICMPv6-Based DDoS Attacks Detection. *Arab. J. Sci. Eng.* **2018**, *43*, 7757–7775. [CrossRef]
9. Bahashwan, A.A.; Anbar, M.; Hasbullah, I.H.; Alashhab, Z.R.; Bin-Salem, A. Flow-Based Approach to Detect Abnormal Behavior in Neighbor Discovery Protocol (NDP). *IEEE Access* **2021**, *9*, 45512–45526. [CrossRef]

10. 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 International Conference on Developments in eSystems Engineering, DeSE, Cambridge, UK, 2–5 September 2018; pp. 134–140. [CrossRef]
11. Anbar, M.; Abdullah, R.; Saad, R.M.; Alomari, E.; Alsaleem, S. Review of security vulnerabilities in the IPv6 neighbor discovery protocol. In *Information Science and Applications (ICISA)*; Lecture Notes in Electrical Engineering; Springer: Singapore, 2016; Volume 376, pp. 603–612. [CrossRef]
12. Tayyab, M.; Belaton, B.; Anbar, M. ICMPv6-Based DoS and DDoS Attacks Detection Using Machine Learning Techniques, Open Challenges, and Blockchain Applicability: A Review. *IEEE Access* **2020**, *8*, 170529–170547. [CrossRef]
13. Heslop, B. By 2030, Each Person Will Own 15 Connected Devices—Here's What That Means for Your Business and Content. 2019. Available online: https://www.spiceworks.com/tech/iot/articles/by-2030-each-person-will-own-15-connected-devices-heres-what-that-means-for-your-business-and-content/ (accessed on 10 September 2022).
14. Anbar, M.; Abdullah, R.; Saad, R.M.; Hasbullah, I.H. Review of preventive security mechanisms for neighbour discovery protocol. *Adv. Sci. Lett.* **2017**, *23*, 11306–11310. [CrossRef]
15. Heuse, M. THC IPv6 Attack Tool kit. 2013. Available online: https://www.thc.org (accessed on 10 September 2022).
16. Elejla, O.E.; Anbar, M.; Belaton, B.; Hamouda, S. Labeled flow-based dataset of ICMPv6-based DDoS attacks. *Neural Comput. Appl.* **2019**, *31*, 3629–3646. [CrossRef]
17. 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]
18. Sperotto, A. Flow-Based Intrusion Detection. Ph.D. Thesis, University of Twente, Enschede, The Netherlands, 2010.
19. Roesch, M. Snort-Lightweight intrusion detection for networks. In Proceedings of the 13th Conference on Systems Administration (LISA 1999), Seattle, WA, USA, 7–12 November 1999; pp. 229–238.
20. Tiwari, A.; Saraswat, S.; Dixit, U.; Pandey, S. Refinements In Zeek Intrusion Detection System. In Proceedings of the 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 25–26 March 2022; Volume 1, pp. 974–979. [CrossRef]
21. Mo, T.P.; Wang, J.H. Design and Implementation of Intrusion Detection System. Diploma Thesis, Potsdam University, Brandenburg, Germany, 2011. [CrossRef]
22. Gehrke, K.A. The Unexplored Impact of Ipv6 on Intrusion Detection Systems. Master's Thesis, University of Phoenix, Phoenix, Arizona, 2012.
23. Gao, X.; Qiu, M.; Liu, M. Machine Learning Based Network Censorship. In Proceedings of the 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud), Washington, DC, USA, 26–28 June 2021. [CrossRef]
24. Bdair, A.H.; Abdullah, R.; Manickam, S.; Al-Ani, A.K. Brief of Intrusion Detection Systems in Detecting ICMPv6 Attacks. In *Computational Science and Technology*; Lecture Notes in Electrical Engineering; Springer: Singapore, 2020; Volume 603, pp. 199–213. [CrossRef]
25. OISF Foundation. Suricata: Intrusion Detection System. Available online: https://suricata.io/ (accessed on 14 September 2022).
26. Rietz, R.; Vogel, M.; Schuster, F.; König, H. Parallelization of network intrusion detection systems under attack conditions. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Singapore, 2014; Volume 8550 LNCS, pp. 172–191. [CrossRef]
27. Atlasis, A. Security Impacts of Abusing IPv6 Extension Headers. In Proceedings of the Black Hat Security Conference, Abu Dhabi, UAE, 3–6 December 2012; pp. 1–10.
28. Atlasis, A.; Rey, E. Evasion of High-End IPS Devices in the Age of IPv6. Technical Report, Black Hat. 2015. Available online: https://www.blackhat.com/docs/us-14/materials/us-14-Atlasis-Evasion-Of-HighEnd-IPS-Devices-In-The-Age-Of-IPv6.pdf (accessed on 13 September 2022).
29. Gascon, H.; Orfila, A.; Blasco, J. Analysis of update delays in signature-based network intrusion detection systems. *Comput. Secur.* **2011**, *30*, 613–624. [CrossRef]
30. Kabiri, P.; Ghorbani, A.A. Research on intrusion detection and response: A survey. *Int. J. Netw. Secur.* **2005**, *1*, 84–102.
31. Barbhuiya, F.A.; Biswas, S.; Nandi, S. Detection of neighbor solicitation and advertisement spoofing in IPv6 neighbor discovery protocol. In Proceedings of the 4th International Conference on Security of Information and Networks, Sydney, Australia, 14–19 November 2011; pp. 111–118. [CrossRef]
32. Bansal, G.; Kumar, N.; Nandi, S.; Biswas, S. Detection of NDP based attacks using MLD. In Proceedings of the Fifth International Conference on Security of Information and Networks-SIN '12, Jaipur, India, 25–27 October 2012; ACM Press: New York, NY, USA, 2012; pp. 163–167. [CrossRef]
33. Li, Y.; Li, Z.T.; Liu, S. A fuzzy anomaly detection algorithm for IPv6. In Proceedings of the 2006 2nd International Conference on Semantics Knowledge and Grid, SKG, Guilin, China, 1–3 November 2006; pp. 4–7. [CrossRef]
34. Wenke Lee.; Stolfo, S.; Mok, K. A data mining framework for building intrusion detection models. In Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344), Oakland, CA, USA, 9–12 May 1999; pp. 120–132. [CrossRef]
35. Zulkiflee, M.; Ahmad, M.; Sahib, S.; Ghani, M. A framework of features selection for ipv6 network attacks detection. *WSEAS Trans. Commun.* **2015**, *14*, 399–408.

36. Saad, R.M.A.; 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]
37. Sperotto, A.; Sadre, R.; van Vliet, F.; Pras, A. A Labeled Data Set for Flow-Based Intrusion Detection; In *International Workshop on IP Operations and Management*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2009; Volume 5843 LNCS, pp. 39–50. [CrossRef]