A Neuro-Fuzzy System to Detect IPv6 Router Alert Option DoS Packets

Shubair Abdullah
Instructional and Learning Technology, Sultan Qaboos University, Oman

Abstract: *Detecting the denial of service attacks that solely target the router is a maximum security imperative in deploying IPv6 networks. The state-of-the-art Denial of Service detection methods aim at leveraging the advantages of flow statistical features and machine learning techniques. However, the detection performance is highly affected by the quality of the feature selector and the reliability of datasets of IPv6 flow information. This paper proposes a new neuro-fuzzy inference system to tackle the problem of classifying the packets in IPv6 networks in crucial situation of small-supervised training dataset. The proposed system is capable of classifying the IPv6 router alert option packets into denial of service and normal by utilizing the neuro-fuzzy strengths to boost the classification accuracy. A mathematical analysis from the fuzzy sets theory perspective is provided to express performance benefit of the proposed system. An empirical performance test is conducted on comprehensive dataset of IPv6 packets produced in a supervised environment. The result shows that the proposed system overcomes robustly some state-of-the-art systems.*

Keywords: *DoS attacks, IPv6 router alert option, Neuro-Fuzzy, IPv6 network security.*

Received February 23, 2017; accepted July 8, 2018
https://doi.org/10.34028/iajit/17/1/3

1. Introduction

The main reason behind considering IPv6 as the network protocol of the future is that the amount of potentially allocated IPv4 addresses is insufficient. IPv6 extremely increases the address space from 32-bit to 128-bit. Such enhancement gives an address for every Internet-capable device on the planet [24]. Despite the Internet is not far away from IPv4 exhaustion and because of lingering security concerns, percentage of users that access Internet services over IPv6 is still far from ambition, it is barely reached 13.7% [8].

As the deployment of IPv6 proceeds, security issues concurrently come up. Several research have been published to address the open issues of IPv6 protocol vulnerabilities [4, 9]. Some vulnerabilities discovered to date can be fixed through software patches, for instance, tunnel incorrect configuration that allows external traffic to flow through it. Other vulnerabilities originate from the protocol itself. This kind of vulnerabilities, such as the extension header, which is the focus in this paper, could result temporary services loss or Denial of Service attacks (DoS) conditions. The extension headers are where all the options from IPv4 packet header were placed in [12]. The first extension header in a packet is Hop-by-Hop option header. One of the options defined within the Hop-by-Hop option header is the Router Alert option. The router alert option is used to inform routers that the IPv6 packet contents require additional processing. This option could be used for Multicast Listener Discovery (MLD) and the Resource Reservation Protocol (RSVP) [23].

This option can be exploited to cause DoS attack on the routers [27]. Two examples of attacks mentioned here. Firstly, a router is overwhelmed with a huge number of IPv6 packets with router alert option defined from one or multiple sources. Secondly, a router moves the IPv6 packets containing the Router Alert option to the slow path, which might result in consuming a significant share of the router's slow path. This is in turn will affect other applications operating in the slow path and cause DoS attack. The existing DoS classification methods over IPv6 networks need for more research particularly in this crucial type of DoS attacks where the routers are the target.

Robust IPv6 router DoS attacks detection is a big challenge given the difficulty of distinguishing the packets of router DoS attacks from normal packets. Moreover, the lack of public dataset of IPv6 packets makes it harder to train and test newly developed systems. These observations are the motivation of this research, which is an attempt to tackle the problem of classification of IPv6 packets utilizing the strength of neuro-fuzzy methodologies and using small-supervised training dataset. The contributions of this paper are:

1. Proposing a method to simulate normal and DoS attack packets over IPv6 network.
2. Developing a system model to detect IPv6 packets generated by a DoS attack launcher.

3. Providing mathematical analysis from the fuzzy sets theory perspective to express performance benefit of the developed system.

Three types of experiments were performed: comparing the system with some of the state-of-the-art systems, measuring the impact of tuning parameters on the classification accuracy, and investigating the system as a regression problem.

The paper is organized as follows. Section 2 reviews the literature. The IPv6 packets detection system is presented in section 3, which is followed by the implementation in section 4. Section 5 presents the experimental results and some discussions are provided in section 6. Finally, the paper is concluded in section 7.

2. Literature Review

A DoS is malicious attack aims to make network resources unavailable for legitimate applications. Hackers launch DoS attacks by utilizing the vulnerabilities of the IPv6 protocols using the same methods as in IPv4 [23]. In addition to classical DoS such as TCP SYN flood and incomplete Hyper Text Markup Protocol (HTTP) requests [25], the IPv6 protocol is vulnerable to other types of DoS attacks, such as ICMPv6 and DHCPv6 attacks [14, 27]. Various DoS attacks detection methods have been proposed in the literature over the past decade [1, 14, 26]. Most of the detection methods presented employ the ICMPv6 protocol messages, which is used by the network management processes such as Router Advertisement (RA), Neighbor Solicitation (NS), and Duplicate Address Detection (DAD) to accomplish their tasks [7].

Due to lack of security considerations in the deployment vulnerabilities of IPv6 protocol, the ICMPv6 messages flooding attacks have witnessed a steady increase. The flooding attacks are not stand-alone attacks. They require a kind of malicious actions to be performed. Some examples of these malicious actions are IP spoofing, Man In The Middle (MITM), and network reconnaissance. In order to make it difficult for the hackers to launch the flooding attacks, all kind of malicious actions should be detected. Based on this fact, a number of countermeasures and solution mechanisms are developed to mitigate malicious actions that could lead to DoS attacks.

Administrators prefer the IP Security (IPSec) mechanism in defending the IP spoofing. IPSec is defined as mandatory feature in IPv6. It has enhanced the original IP protocol through providing authenticity, integrity, confidentiality and access control to each IP packet. IPSec uses two protocols: Authentication Header (AH) and Encapsulating Security Payload (ESP), and provides cryptographically based security services for traffic at the IP layer [21]. Despite its effectiveness, IPSec not performs efficiently if the spoofing attack targets large number of nodes that may be distributed in different time zones. As a result, the traffic generated by the spoofed nodes cannot be differentiated from the legitimate traffic [19]. Moreover, some research has installed and tested the IPSec in IPv6 network and concluded that it cannot detect DoS attacks [28]. The mechanism of Secure Neighbor Discovery (SEND) is another option that could be applied to prevent pre DoS attacks malicious actions. SEND mechanism uses Cryptographically Generated Addresses (CGAs), a digital signature, and an X.509 certification to protect the Neighbor Discovery Protocol (NDP), which is a protocol used to perform several critical tasks, such as discovering nodes on the same link, determining link-layer address, detecting duplicate address, and finding routers. NDP is prone to malicious actions as it assumes that all nodes on the link trust each other. Malicious users could impersonate legitimate nodes by falsifying NDP messages to generate DoS attacks. Although SEND applies some security solutions, it is still unfavorable tool to prevent pre DoS attacks malicious actions. Some research has concluded that SEND mechanism is not supported by some operating systems and it consumes too much processing time and bandwidth [5, 6].

The third mechanism to investigate is the Router Advertisement (RA) guard. The RA guard mechanism is intended to handle threats of RA messages that are sent by routers to advertise their presence in IPv6 network segments periodically or in response to router solicitation messages. Some examples of RA messages threats are IP spoofing and MITM. This mechanism is installed in layer-2 switches to filter the IPv6 frames based on predefined filters. It extracts some information from the IPv6 frames, such as IPv6 address and physical source address, and decides to pass or discard them based on the filters. The RA guard mechanism suffers many implementation problems. It does not provide security protection for WiFi devices and it monitors only the ingress IPv6 frames [14].

The severity degree of malicious actions has showed that this research area demands for new detection mechanisms. One of possible choices is the IPv6 Intrusion Detection Systems (IDS). The presence IDSs are based on the IPv4 features and they require some development to become functional in IPv6 networks. In addition, more research groups are required to investigate the detection of malicious actions especially those related to neighbor discovery, router discovery, auto-configuration IPv6 features [20] as well as the extension header of IPv6 packets.

3. IPv6 Packet Classification

This section presents the details of the proposed router alert DoS attack detection system. First, the system model is presented. Second, it presents a method to collect and label a dataset of normal and DoS attack packets for the purpose of training and testing the system. Then, an explanation of the process of developing a neuro-fuzzy expert system to classify the IPv6 packets and detect the router alert DoS attack in IPv6 networks.

3.1. System Model

Figure 1 shows a system model that is able to detect unknown IPv6 packets generated by a DoS attack launcher. The proposed system combines artificial neural network and fuzzy logic to improve IPv6 packets classification accuracy [15]. Before the training stage, a number of normal IPv6 packets and DoS attack packets are simulated and labeled as normal packets and suspicious packets to constitute a supervised training dataset for router alert DoS attack detection.

A method of simulating normal and DoS attack packets over IPv6 network is proposed to generate any number of semi-realistic normal and attack situation to significantly reduce the amount of packets, which are related to known situation but inaccurately labeled in the training stage.

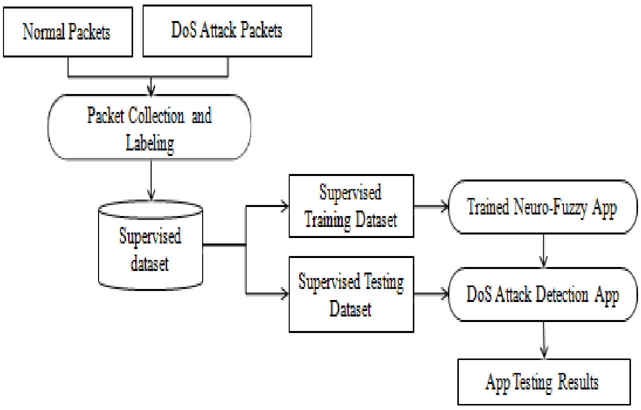

Figure 1. System model.

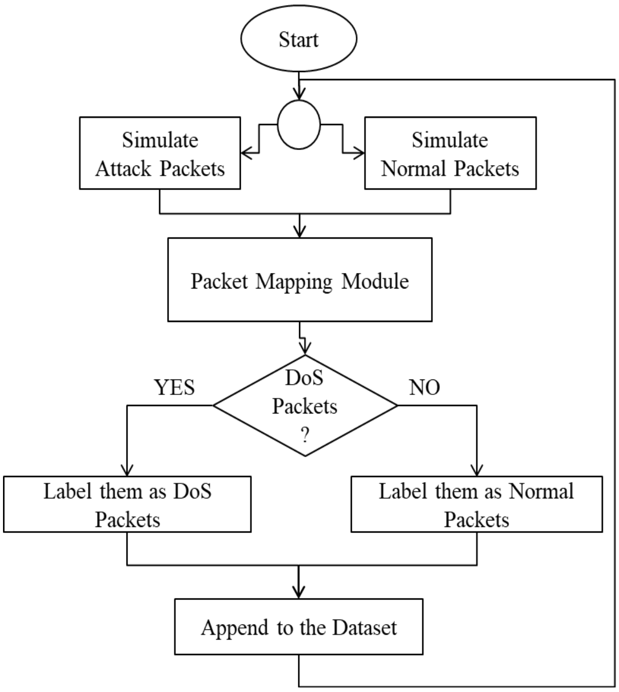

Figure 2. Packet simulation method.

3.2. Normal and DoS Packets Simulation

Since producing IPv6 normal packets is difficult, a new simulation method is used throughout the research. The simulation process creates a dataset of normal and DoS attack packets. It focuses on properties that are extracted by inspecting the Hop-by-Hop Option Header of the packets, i.e., the Router Alert Options in addition to the standard properties, such as timestamp, source IPv6, source port, and destination port. The dataset will be used to train the classifier and produce a trained model. The simulation method can generate any number of normal router alert options, which allows producing better model generalization.

The simulation method consists of three stages. The first stage generates the synthetic packets. The second stage involves labeling of the packet, which is done by the packet mapping module. At the third stage, the packets are appended to the dataset. Figure 2 depicts the simulation method.

Normal packets and DoS packets are constructed with scapy, which is powerful interactive packet management software [10]. The Packet Mapping Module tags the packets as “normal” or “DoS attack” manually. Figure 3 shows scapy commands used for packet generation on Ubuntu. Firstly, the packets that contain a Hop-by-Hop Options header with a router alert option are created through “packetRouterAlert” command. Then the packets are flooded to the destination port of 80 HTTP through “srflood()” command. These two commands are used create normal packet since the “srflood()” command does not flood the network with huge number of packets per second to overwhelm a router. To create the DoS attack packets with router alert options, the two commands has been repeated rapidly and simultaneously executed on more than one node.

```
>>> packetRouterAlert =
IPv6(dst="2001:db8:aaaa:a:3db5:ec68:7a98:11cb
") /
IPv6ExtHdrHopByHop(options=RouterAlert(val
ue=0)) / TCP(sport=RandShort(),dport=80)
>>> srflood(packetRouterAlert)
```

Figure 3. Scapy instructions: Router Alert Option packets flooded to 2001:db8:aaaa:a:3db5:ec68:7a98:11cb.

Two famous simulation software systems are employed to build the virtual topologies of routers and virtual hosts, the Graphical Network Simulator (GNS3) software and the Oracle VM virtual box. To make the environment closer to reality, the following configurations are made:

- Two network segments with one Ethernet switch in each segment.
- One Cisco 7200 series router ISO has been used to simulate the attacked router.
- The network nodes are allocated as follows: one node to function as a Windows 2008 server, one node to function as a Windows 7 client PC, and two nodes are configured with Linux Ubuntu 10.10 desktop OS.
- The IPv6 addresses are assigned dynamically using stateless address assignment method.

3.3. Neuro-Fuzzy Inference System Design

Due to its uncertainty nature, the problem of classifying IPv6 packets requires a fuzzy system whose objective is approximate reasoning rather than exact solution. There is a need of a neuro-fuzzy system that produces truth-value ranges in degree between zero and one and leaves the final verdict to the networking security professionals. Instead of describing a set of IPv6 packets as absolute zero or one, fuzzy functions could be employed to explain the set of packets as a matter of degree. The zero shows normal packets and one expresses absolutely DoS attack packets, and any value within the range indicates the degree of DoS attack. This formula is indeed close to human intuition.

In this paper, the packets are represented using two statistical properties:

1. *NUM*: the number of packets with router alert options and
2. *AVRG*: the average of the timestamps of the packets with router alert options. The proposed neuro-fuzzy inference system will take the two properties as input and tries to infer a value between zero and one through a mechanism that encompasses IF-THEN rules and fuzzy logical operations.

The process of designing a neuro-fuzzy inference involves the following steps [10]:

1. *Definition of linguistic variables*: there are two linguistic variables: number of packets with router alert options "NUM" and average of timestamps of these packets "AVRG". Table 1 shows the linguistic variables and their ranges. To simplify the notation of values of each variable, the values of NUM variable are divided by 10^9 and the values of AVRG are multiplied by 10^3.

Table 1. Linguistic variables and their ranges.

Linguistic Variable	Linguistic Values	Notation	Input Ranges	Adjusted Ranges
NUM	Normal	N	from 0 to ~25×10^9	from 0 to 2.5 (division by 10^9)
	Suspicious	S	from ~15×10^9 to ~40×10^9	from 1.5 to 4 (division by 10^9)
AVRG	Normal	N	from 0 to ~1.75E-03	from 0 to 1.75 (multiplication by 10^3)
	Suspicious	S	form ~1.25E-03 to ~3.00E-03	from 1.25 to 3 (multiplication by 10^3)

2. *Determination of fuzzy sets*: A fuzzy set is an extension of classical set. It does not have a crisp that clearly defines its boundary. Its boundary is defined by membership function that decides the elements membership degree from zero to one. The Trapezoidal membership function provides an adequate representation of expert knowledge and significantly simplifies the process of computation. Figure 4 shows the Trapezoidal membership function of fuzzy sets of NUM and AVRG variables.

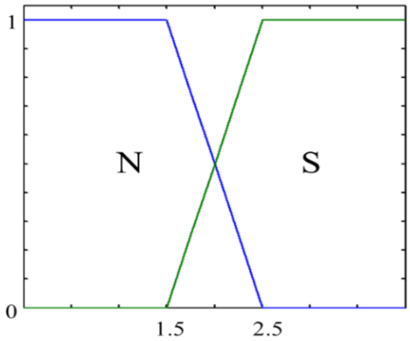

Figure 4. Trapezoidal membership function.

3. *Construction of fuzzy rules*: two fuzzy linguistic variables are defined with two fuzzy sets for each:

$$NUM = \begin{cases} Sifnumberofpacketsissuspicious \\ Nifnumberofpacketsisnormal, \end{cases}$$

$$ACRG = \begin{cases} Sifaverageoftimestampsofpacketsissuspicious \\ Nifaverageoftimestampsofpacketsisnormal. \end{cases}$$

There are four possible values for the system output "y". The value of "y" describes the case in two fuzzy sets: either N: Normal case or A: DoS attack case. In order to map each input fuzzy set into an output fuzzy set and construct the fuzzy rules, this research uses the Fuzzy Associative Memory (FAM) which is the matrix form of representing a system's fuzzy rules, as in Figure 5.

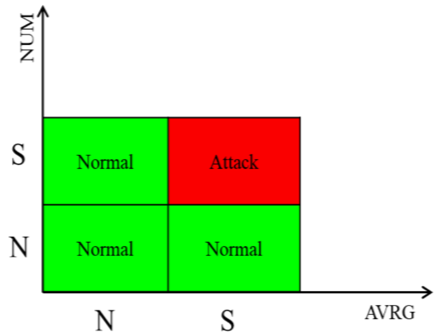

Figure 5. The square FAM representation.

The fuzzy rules are:

$$IF\ x_1 is\ S\ and\ x_2\ is\ S\ THEN\ y\ IS\ A,$$
$$IF\ x_1 is\ S\ and\ x_2\ is\ N\ THEN\ y\ IS\ N,$$
$$IF\ x_1 is\ N\ and\ x_2\ is\ N\ THEN\ y\ IS\ N,$$
$$IF\ x_1 is\ N\ and\ x_2\ is\ S\ THEN\ y\ IS\ N,$$

Where x_1: NUM, x_2: AVRG, and y: attack (*A*) or normal (*N*)

4. *Determination of rules aggregation and defuzzification processes*: the rule aggregation involves combining the fuzzy sets that represent the rules' output into a single fuzzy set. The fuzzy rule aggregation process occurs only once for each output. Prior to the last process, the defuzzification process, which takes the aggregate output fuzzy set and produces a crisp number that represents the degree of seriousness of the packets in terms of DoS attacks.

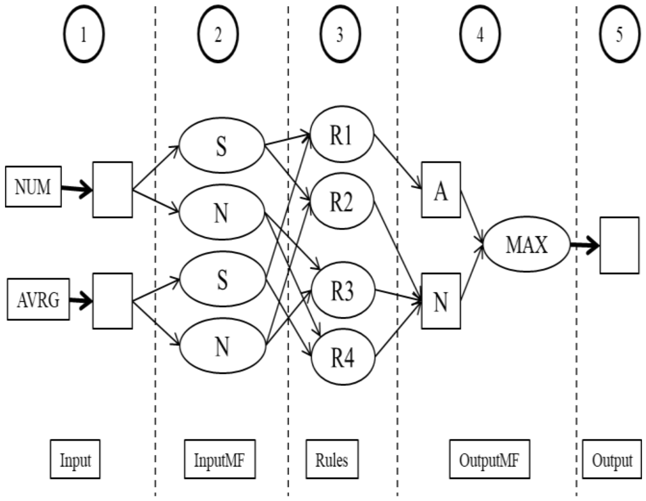

Figure 6. Architecture of neuro-fuzzy inference system.

4. The Implementation

Tied with the previous section, this section presents the implementation of a two-input one-output neuro-fuzzy inference system for router alert option DoS attack detection. Suppose that *S* is a supervised training dataset for DoS router alert detection system with 3-tuple elements. *S* is divided into two subsets: S_N to contain normal packets and S_D to contain the DoS packets:

$$S_N \subset S \wedge S_D \subset S \leftarrow\rightarrow S \equiv S_N\ \cup\ S_D. \qquad (1)$$

The S set can be represented by the set builder notation:

$$S = \{x | x \in S_N \vee x \in S_D\}. \qquad (2)$$

Where x is 3-tuple, i.e., (a,b,c), a=NUM, b=AVRG, and *c* is the class, $c = 0\ if\ x \in S_N$and $c = 1\ if\ x \in S_D$

In a normal situation over the network (no DoS attacks), any group of router packets that are captured within a certain period of time and featured by NUM and AVRG is considered as normal, that is:

Let $\mathrm{X_N} = \{(\mathrm{NUM, AVRG}) | (\mathrm{NUM, AVRG}) \in \mathrm{S_N}\}$ is a set of router alert packets. Then, the normal situation can be represented by:

In a normal situation over the network (no DoS attacks), any group of router packets that are captured within a certain period of time and featured by NUM and AVRG is considered as normal, that is:

$$\forall\, X_N (X_N\ \subset\ S_N). \qquad (3)$$

If the universe of discourse is the normal situations only, the truth-value is true. However, in case of adding the DoS attack situations to the universe of discourse, the truth value will be false since that X_D, set of DoS attack packets:

$$X_D \not\subset S_N\ \wedge\ X_D \subset S_D. \qquad (4)$$

Here in this case X_D is called counterexample for (1) since it turns its truth-value into false. The ultimate goal of the proposed system is counterexamples detection that represent router alert option DoS attacks. All of the examples will be represented with degree of membership between zero and one. Figure 6 shows the architecture of the proposed system, which is a multilayer feed-forward network. Each neuron in a particular layer receives input, performs a particular function, and transmits its output to neuron in the next layer. Every single neuron is either an adaptive neuron or fixed neuron. The adaptive neurons (represented by squares) have input parameters while the fixed neurons (represented by ovals) have no parameters. The system uses a method of fuzzy "IF-THEN" inference rules called Mamdani. A typical Mamdani rule is as follows:

$$IF\ X_1\ is\ A\ and\ X_2\ is\ B\ THEN\ y\ is\ C.$$

Where X_1 and X_2: input variables, *A* and *B*: input fuzzy sets, *y*: output variable, and *C*: output fuzzy set.

Each layer of the system involves a number of neurons and performs a specific task:

1. *Input layer*: the neurons is this layer only pass input data (NUM and AVRG) to layer two.

2. *Fuzzification layer*: neurons of this layer compute the membership degrees of inputs to produce fuzzy sets. They apply Trapezoidal membership function that is a function of a vector v depends on four scalar parameters a, b, c, and d:

$$trapmf(v; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right). \quad (5)$$

Where the parameters a and d locate the "feet" of the trapezoid and b and c locate the "shoulders".

3. *Rules layer*: every single neuron in this layer represents one Mamdani fuzzy rule. It receives inputs from layer 2, the fuzzy sets, applies the rule, and produces a fuzzy value from the consequent part.
4. *Defuzzification layer*: the inference mechanism followed in the system is First Aggregate Then Infer (FATI). The neurons in the layer are divided into two sub layers.

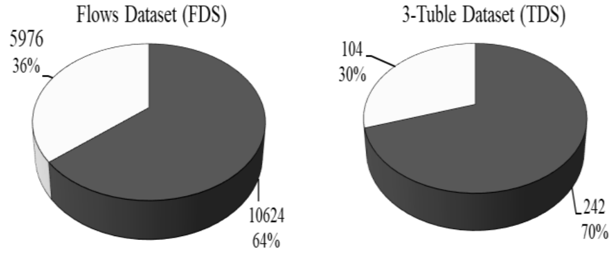

Figure 7. Datasets used in system evaluation (grey part represent normal flows and dotted part represents DoS attack flows).

The neurons in the first sub layer aggregate the output (fuzzy sets) for each rule into a single fuzzy set using a fuzzy aggregation operator, the maximum, which is one of the most aggregation operators. The second sub layer performs the defuzzification task that defines the way of extraction of the final crisp value from the aggregated output fuzzy sets. The centroid defuzzification method is used, which is a popular defuzzification method that returns the center of area under the trapezoidal curve.

5. *Output layer*: this layer outputs the final crisp value.

5. The Experiments

The strength of the proposed approach is the creation of synthetic dataset of router alert options normal packets and/or DoS attack packets, which is, basically, a method to generate dataset for training and testing new developed approaches. The synthetic dataset is used to compare the proposed system against three of well-known systems. In all experiments, the simulated supervised Flow Packets Dataset (FPD) is used. The FPD dataset consists of 16.6k packets, 67% of them are normal packets and 33% of them are DoS attack packets. To extract the NUM and AVRG, the packets are split into time slices based on the timestamps. Initially, a parameter is established to determine the frequency at which the feature extraction process that is responsible of computing NUM and AVRG is fired. This parameter is expressed in ms and is called Time Period (TMP). The default value of TMP is 150 ms. According to TMP settings and the number of packets in FPD dataset, the experiments have extracted 346 elements, 70% of them are labeled as normal packets and the remainder 30% are labeled as DoS attack packets. These extracted elements represent the training/testing dataset (TSD). Figure 7 shows the details of FPD and TSD datasets. Each element in TSD is expressed in 3-tuple feature: NUM, AVRG, and label. Table 2 provides definitions of features.

Table 2. Extracted statistical features.

Feature	Description	Value
NUM	Number of packets with router alert options.	Integer number
AVRG	Average of the timestamps of the packets with router alert options	Real number
Label	Element label	0 or 1

5.1. Comparison the Proposed System

The first test aimed at comparing the performance of the system with three famous systems: Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) [18], Adaptive Neuro-Fuzzy Inference System (ANFIS) [17], and kNN-based Evolving Neuro-Fuzzy Inference System (kENFIS) [2]. As the proposed system produces a truth-value between zero and one to indicate the seriousness of packets in terms of DoS attacks, the continuous output values are separated by deciding on cutoff. One cutoff has been decided in the experiments: 0.60, where the normal packet values are less than 0.60 and the router alert options DoS attack packet values are greater than or equal to 0.60. Two common metrics have been used in the comparison:

1. *Accuracy*: the accuracy of a classifier on dataset computed by:

$$Accuracy = \frac{\#of\ correctly\ classified\ packets}{\#of\ packet\ samples} \quad (6)$$

2. *F-measure*: the performance of the proposed system per class (normal and attack) computed by:

$$f - measure = \frac{2 \times precision \times recall}{precision + recall} \quad (7)$$

Where precision is the percentage of correctly classified packets over all predicted packets in and recall is the percentage of correctly classified packets over all ground truth packets in a class.

The process began by training the system using 70% of the TDS and testing it using 30% of TDS. To avoid bias of classification results, the process involved dividing each of training set and testing set into two parts: 30% DoS attack packets and 70%

normal packets. Ultimately, two sets training and testing sets are produced, as shown in Table 3. The classification accuracy is calculated according to Equation (1). To calculate the precision and the recall, four factors: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) are observed for each system. Table 4 shows the results of all metrics for the proposed system and the competing systems. Figure 8 compares the classification accuracy of the four systems. The results show that the accuracy of the proposed system is slightly higher than ANFIS and DENFIS, and is one degree lower than kENFIS. A balance between the precision and the recall is needed for binary classification. F-measure metric conveys this balance. The results of f-measure calculations are shown in Figure 9.

Table 3. Training and testing sets.

Set	Normal packets	DoS packets	Total
TDS	242	104	346
Training set	170	72	242
Testing set	73	31	104

Table 4. Testing results.

	Proposed System	ANFIS	DENFIS	kENFIS
TP	26	26	22	23
FP	18	21	16	14
TN	55	52	57	59
FN	5	5	9	8
Precision	0.59	0.55	0.58	0.62
Recall	0.84	0.84	0.71	0.74
Training samples	242	242	242	242
Testing samples	104	104	104	104

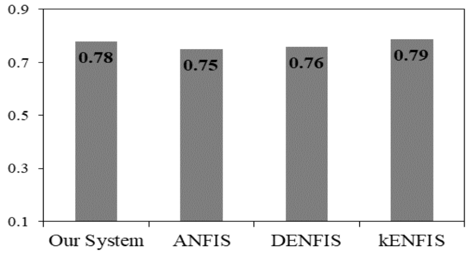

Figure 8. Overall accuracy.

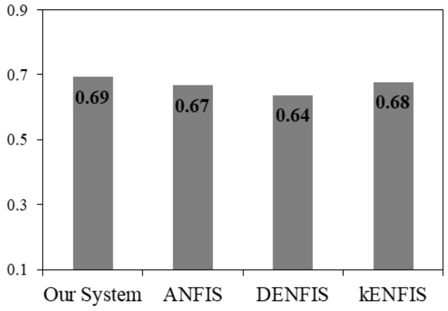

Figure 9. F-Measure results.

5.2. Impact of Parameters

This test has been conducted to figure out the effect of some parameters related to the training process and to the system optimization. Two parameters are used,

1. The training dataset used (training purity).
2. The TMP.

The training purity that introduced by [29], is used to quantify the effect of training information from purity point of view. It is calculated as follows:

$$Training\ Purity = \frac{\#\ of\ correctly\ identified\ packets}{\#\ of\ training\ packets} \quad (8)$$

The training purity is computed according to the classes (the normal and the DoS). Figures 10 and 11 report the training purity for the normal class and the DoS class respectively versus different number of training packets.

Three experiments have been conducted to investigate the impact of TMP on the classification accuracy. Three values are set for the TMP parameter: 150 ms, 300 ms, and 450 ms, and the number of packets is fixed to 16.6k. Table 5 shows information about experiment conducted and Table 6 shows the results of the experiments in terms of precision and recall for all TMP sets.

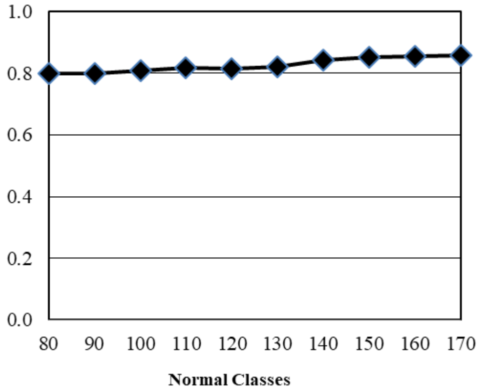

Figure 10. Training purity for normal classes.

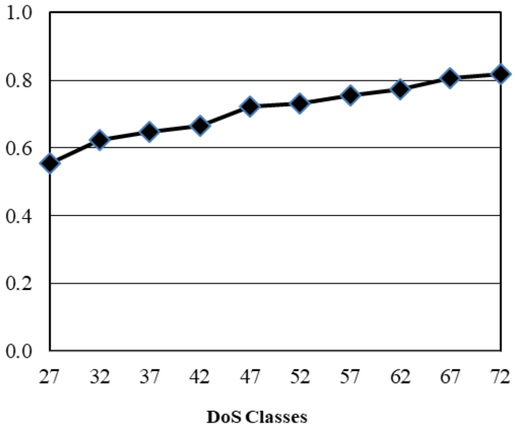

Figure 11. Training purity for DoS classes.

Table 5. Datasets used to investigate the impact of TMP.

TMP	packets used	extracted	training and testing
150 ms	16.6k	346	242 & 104
300 ms	16.6k	194	136 & 58
450 ms	16.6k	109	76 & 33

Table 6. Results of impact of TMP parameter investigation.

TMP:	150 ms	300 ms	450 ms
TP:	26	13	5
FP:	18	8	5
TN:	55	31	19
FN:	5	6	4
Precision:	0.59	0.62	0.50
Recall:	0.84	0.68	0.56
Accuracy:	78%	76%	73%
Training samples:	242	136	76
Testing samples:	104	58	33
Total:	346	194	109

5.3. The Regression Issue

The decision taken based on output of the proposed neuro-fuzzy inference system is an issue of discretionary. There is a necessity to decide numerically whether the in hand packets represent normal or DoS attack. The output value of the system ranges from zero and one, where zero represents normal packets and one represents DoS packets. The classification problem in this case is considered as regression problem. Consequently, the system is considered as excellent when the difference between the produced value and the observed value is equal to zero [3, 13]. This experiment is devoted to test the accuracy as a regression issue. Two metrics were used:

1. The Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Z_i - \acute{Z}_i)^2} \qquad (9)$$

2. The Non-Dimensional Error Index (NDEI):

$$NDEI = \frac{RMSE}{stdev(Z)} \qquad (10)$$

Where *n* is the total number of samples, Z_i is the observed output, $\acute{Z}_i$ the proposed system output, and *stdev(Z)* is the standard deviation of the desired output.

A set of 300 samples from TSD are used, 100 DoS attack samples and 200 normal samples. A different procedure has been followed in this experiment to assure lower bias and lower variance. Initially, the samples are split into training part and testing part using the technique of 10-fold cross-validation [16]. The proposed system is trained 10 times, at each time there were nine folds (270 samples) for training and one fold (30 samples) for testing. Further step has been taken, the stratification [22], which aims to create folds in a way that they contain the same proportion classes. Figures 12 and 13 depict the RMSE and NDEI calculation results.

6. Discussion

The proposed system outperforms DENFIS and kENFIS and neutralizes ANFIS when the DoS packets are presented. As shown in Table 4 TP row, out of 31 samples, the proposed system and ANFIS correctly classified 26 samples while kENFIS and DENFIS correctly classified 23 and 22 samples respectively. In terms of detecting normal packets (TN row), the proposed system is outperformed by DENFIS and kENFIS. Generally, the overall percentages are close to each other. However, the lower false detection rates and higher true detection rate guarantee that the proposed system competes well. Figure 14 shows the percentages of packets classification for the four systems. The synthetic dataset of router options packets is used in the experiments due to the difficulties of providing such datasets. As shown in Figure 9, the proposed system can achieve excellent training purity for each class, which reflects effectiveness of the simulated training dataset used. The results show also that the training purity rise when more training packets are available.

More attention has been paid on the correlation between the training purity and the number of training packets to gain a clear picture about which class is affected more by the number of training packets. The Pearson coefficient is calculated for the number of training packets (independent variables) and the training purity (dependent variables). The calculation results were as follows: for the normal packet class, the Pearson value is 0.98, and for the DoS class, the Pearson value is 0.97. The results indicate a strong positive correlation between the number of packets and the training purity. Based on that, a public comprehensive dataset that provides a ground for testing IPv6 router alert option DoS detection system is highly recommended.

The proposed system behaves like network monitor. It provides functions to collect periodically packets sent, functions to aggregate all packets and split them into time slices, and functions to extract the NUM and AVRG traffic features. The recent DoS attacks in the current high-speed Internet are synchronized rapidly from multiple sources against a sole target. Their speed, ranges between 10 and 60 Gbps per second [11], requires detecting DoS attacks in early stages before they severely harm routers. According to the system' structure, the key factor that influences the detection promptness is TMP, the interval between feature extraction operations. Intuitively, execution of the feature extraction at very low frequencies yields to collect large number of packets and consequently increases the chances of getting accurate results. However, such situation could increase the computational processes, which cause slow detection system since the feature extraction process always involves mathematical operations. To tune the system perfectly, three different values of TMP have been investigated: 150 ms, 300 ms, and 450 ms, all values gave high-speed classification rates. The best accuracy, as depicted in Table 6, is obtained by 150. Therefore, the 150 ms setup is recommended for the proposed system.

Despite the excellent RMSE and NDEI results over the 10-fold cross validation, the experiment involved also calculation of the overall accuracy of the 10- fold cross validation. The overall accuracy of cross-validation is calculated as follows:

$$CVA = \frac{1}{n}\sum_{i=1}^{n} A_i \quad (11)$$

Where CVA is the cross-validation accuracy, n is the number of folds, and A_i is the accuracy of each fold.

The calculation results of CVA for the RMSE and NDEI measures are 0.3035 and 0.6330 respectively. Since small values of RMSE and NDEI reflect good performance of classification systems, the efficiency of the proposed system in detection of router alert option DoS attacks in IPv6 networks could be concluded from the results.

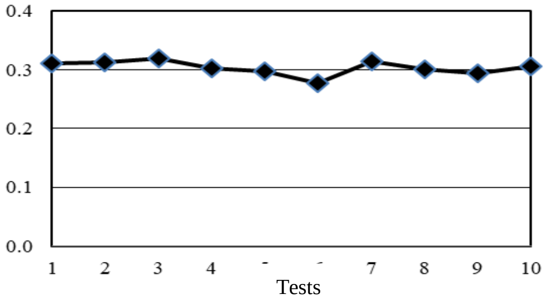

Figure 12. RMSE calculation.

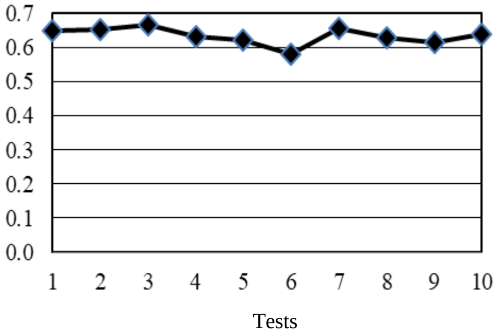

Figure 13. NDEI calculations.

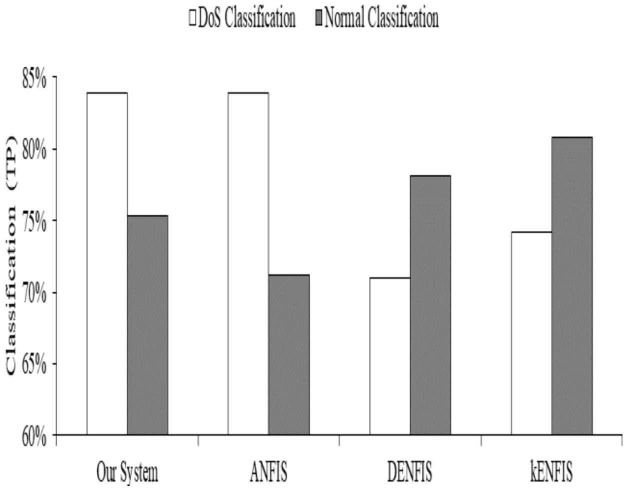

Figure 14. Percentages of classification.

7. Conclusions

This paper proposed neuro-fuzzy interference system to classify the IPv6 router alert options packets into normal and DoS attacks based. The proposed system is expressed by a mathematical analysis from the fuzzy sets theory perspective. The classification is done based on two traffic properties, NUM: the number of packets with router alert options, and AVRG: the average of the timestamps of the packets with router alert options. The NUM and AVRG properties are provided to the system that encompasses IF-THEN rules and fuzzy logic operations. After processing, the system explains the packets as a degree between zero and one, zero shows normal packets and one expresses absolutely DoS attack. The system has been verified to work effectively in crucial situation of small-supervised training dataset. Three types of experiments were conducted: comparing the system against some of state-of-the-art systems, measuring impact of tuning parameters on the classification accuracy, and investigating the system as a regression problem. The results showed that the system could effectively classify the router alert options packets into normal and DoS packets.

References

[1] Abdulla S., "Survey of Security Issues in IPv4 to IPv6 Tunnel Transition Mechanisms," *International Journal of Information Security*, vol. 12, no. 2, pp. 83-102, 2017.

[2] Abdulla S., Ramadass S., and Altyebb A., "kENFIS: kNN-based Evolving Neuro-Fuzzy Inference System for Computer Worms Detection," *Journal of Intelligent and Fuzzy Systems*, vol. 26, no. 4, pp. 1893-1908, 2014.

[3] Abdulla S. and Altyebb A., "Intelligent Approach for Android Malware Detection," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 8, pp. 2964-2983, 2015.

[4] Ahmed A., Hassan R., and Othman N., "Security Threats for Ipv6 Transition Strategies: A Review," *in Preceedings of 4th International Conference on Engineering Technology and Technopreneuship*, Kuala Lumpur, pp. 83-88, 2014.

[5] AlSa'deh A. and Meinel C., "Secure Neighbor Discovery: Review, Challenges, Perspectives, and Recommendations," *IEEE Security and Privacy*, vol. 10, no. 4, pp. 26-34, 2012.

[6] An G., Kim K., Jang J., and Jeon Y., "Analysis of SEND Protocol Through Implementation and Simulation," *in Preceedings of International Conference on Convergence Information Technology*, Gyeongju, pp. 670-676, 2007.

[7] Arjuman N. and Manickam S., “A Review on Icmpv6 Vulnerabilities and its Mitigation Techniques: Classification and Art,” *in Preceedings of International Conference on Computer, Communications, and Control Technology*, Kuching, pp. 323-327, 2015.

[8] Basnet R. and Sung A., “Classifying Phishing Emails Using Confidence-Weighted Linear Classifiers,” *in Preceedings of in International Conference on Information Security and Artificial Intelligence*, Chengdu, pp. 108-112, 2010.

[9] Bilski T., “From IPv4 to IPv6-Data Security in the Transition Phase,” *in Preceedings of The 7th International Conference on Networking and Services*, Venice, pp. 66-72, 2011.

[10] Biondi P., Scapy. Available: http://www.secdev.org/projects/scapy/, Last Visited, 2018.

[11] Caudle R., How to Minimize the Impact from DDoS attacks. Available: https://gcn.com/articles/2015/07/27/ddos-attack-mitigation.aspx, Last Visited, 2018.

[12] Deering S. and Hinden R., “RFC 2460: Internet Protocol,” Internet Engineering Task Force (IETF) Request for Comment, 1998.

[13] Edinson P. and Muthuraj L., “Performance Analysis of FCM based ANFIS and ELMAN Neural Network in Software Effort Estimation,” *The International Arab Journal of Information Technology*, vol. 15, no. 1, pp. 94-102, 2018.

[14] Elejla O., Anbar M., and Belaton B., “ICMPv6-Based DoS and DDoS Attacks and Defense Mechanisms: Review,” *IETE Technical Review*, vol. 14, no. 4, pp. 390-407, 2016.

[15] Fullér R., *Introduction to Neuro-Fuzzy Systems*, Springer Science and Business Media, 2013.

[16] Fushiki T., “Estimation of Prediction Error by K-fold Cross-Validation,” *Statistics and Computing*, vol. 21, no. 2, pp. 137-146, 2011.

[17] Jang J., “ANFIS: Adaptive-Network-Based Fuzzy Inference System,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685, 1993.

[18] Kasabov N. and Song Q., “DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and its Application for Time-Series Prediction,” *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144-154, 2002.

[19] Kent S. and Atkinson R., “RFC 2401: Security Architecture for the Internet Protocol,” Internet Engineering Task Force (IETF), 1998.

[20] Liu Z. and Lai Y., “A Data Mining Framework for Building Intrusion Detection Models Based on IPv6,” *in Preceedings of in International Conference on Information Security*, Seoul, pp. 608-618, 2009.

[21] Modares H., Moravejosharieh A., Keshavarz H., and Salleh R., “Protection of Binding Update Message in Mobile IPv6,” *in Preceedings of 6th UKSim/AMSS European Symposium on Computer Modeling and Simulation*, Valetta, pp. 444-447, 2012.

[22] Olson D. and Delen D., *Advanced Data Mining Techniques*: Springer Science and Business Media, 2008.

[23] Partridge C. and Jackson A., “RFC: 2711: IPv6 Router Alert Option,” Internet Engineering Task Force (IETF) Request for Comment, 2070-1721, 1999.

[24] Szigeti S. and Risztics P., “Will IPv6 bring Better Security?,” *in Proceedings of The 30th IEEE EUROMICRO Conference*, Rennes, pp. 532-537, 2004.

[25] Tripathi N. and Mehtre B., “DoS and DDoS Attacks: Impact, Analysis and Countermeasures,” *in Preceedings of in National Conference on Advances in Computing, Networking and Security*, India, 2013.

[26] Ullrich J., Krombholz K., Hobel H., Dabrowski A., and Weippl E., “IPv6 Security: Attacks and Countermeasures in a Nutshell,” *in Proceedings of the 8th USENIX Conference on Offensive Technologies*, San Diego, pp. 5-5, 2014.

[27] Weber J., “IPv6 Security Test Laboratory,” Master Thesis, Ruhr-University Bochum, Germany, 2013.

[28] Yang X., Ma T., and Shi Y., “Typical dos/ddos Threats under ipv6,” *in Preceedings of International Multi-Conference on Computing in the Global Information Technology*, Guadeloupe City, pp. 55-55, 2007.

[29] Zhang J., Chen C., Xiang Y., Zhou W., and Vasilakos A., “An Effective Network Traffic Classification Method with Unknown Flow Detection,” *IEEE Transactions on Network and Service Management*, vol. 10, pp. 133-147, 2013.

Shubair Abdulla received his BSc degree in computer science from Basra University in 1994. He received his MSc and PhD degrees in computer science from University Sains Malaysia (USM) in 2007 and 2014 respectively. Currently, he is working at Sultan Qaboos University, Oman, Muscat. His research interests include data mining, network security, and fuzzy inference systems.