Cogn Comput (2018) 10:201–214
https://doi.org/10.1007/s12559-017-9519-8

A Machine Learning Approach to Detect Router Advertisement Flooding Attacks in Next-Generation IPv6 Networks

Mohammed Anbar[1] · Rosni Abdullah[1] · Bassam Naji Al-Tamimi[2] · Amir Hussain[3]

Received: 17 February 2017 / Accepted: 10 October 2017 / Published online: 23 October 2017

Abstract Router advertisement (RA) flooding attack aims to exhaust all node resources, such as CPU and memory, attached to routers on the same link. A biologically inspired machine learning-based approach is proposed in this study to detect RA flooding attacks. The proposed technique exploits information gain ratio (IGR) and principal component analysis (PCA) for feature selection and a support vector machine (SVM)-based predictor model, which can also detect input traffic anomaly. A real benchmark dataset obtained from National Advanced IPv6 Center of Excellence laboratory is used to evaluate the proposed technique. The evaluation process is conducted with two experiments. The first experiment investigates the effect of IGR and PCA feature selection methods to identify the most contributed features for the SVM training model. The second experiment evaluates the capability of SVM to detect RA flooding attacks. The results show that the proposed technique demonstrates excellent detection accuracy and is thus an effective choice for detecting RA flooding attacks. The main contribution of this study is identification of a set of new features that are related to RA flooding attack by utilizing IGR and PCA algorithms. The proposed technique in this paper can effectively detect the presence of RA flooding attack in IPv6 network.

Keywords RA flooding attack · Network security · IGR · PCA · SVM · IPv6 security

✉ Mohammed Anbar
anbar@nav6.usm.my

Rosni Abdullah
rosni@nav6.usm.my

Bassam Naji Al-Tamimi
Mr.altamimi@gmail.com

Amir Hussain
ahu@cs.stir.ac.uk

[1] National Advanced IPv6 Center of Excellence (NAv6), Universiti Sains Malaysia, 11800, Gelugor, Penang, Malaysia

[2] College of Computer Science and Engineering, Taibah University, Al-Madinah Al-Munawarah, Kingdom of Saudi Arabia

[3] Institute of Computing Science and Mathematics, School of Natural Sciences, University of Stirling, Stirling, Scotland, UK

Introduction

Internet Protocol version 6 (IPv6) is the next-generation Internet protocol expected to replace the current IPv4 protocol. IPv6 presents a number of improvements and simplifications compared with IPv4. The primary difference between the two is that IPv6 uses 128 bit addresses, whereas IPv4 uses 32 bit ones. Furthermore, IPv6 ships with a new protocol called neighbor discovery protocol (NDP) that introduces new security vulnerabilities, which allow attackers to easily attack IPv6 networks [1, 2]. The security community has built new tools and experimented with new IPv6 security methods to perform IPv6 network penetration testing; one of these tools is The Hackers' Choice, which is commonly called the THC-IPv6 toolkit [3]. IPv6 possesses new security vulnerabilities in addition to those inherent in current IPv4 networks. Most security vulnerabilities are presented in the Neighbor Discovery Protocol [4].

NDP in IPv6 is similar to Address Resolution Protocol (ARP) in IPv4 [5]. The main purposes of both protocols are to locate the media access control (MAC) address of

the destination host in the same local network and identify the MAC address of the router to communicate with a node in an external network. Consequently, actual exchange of messages can occur between the two nodes. In IPv6, NDP uses ICMPv6 type field values ranging from 133 to 137 to achieve its purpose [6]. Table 1 presents the NDP-related ICMPv6 type field values and their use. NDP is a stateless protocol that lacks authentication of its messages by default. This lack of authenticity and statelessness result in many possible attacks, such as router advertisement [7].

Unlike the IPv4 network, IPv6 provides minimal options to detect NDP-based attacks because the IPv6 protocol is relatively new and has gradually become common. Security researchers have conducted extensive studies and have designed important tools for IPv6

Several successful machine learning techniques, namely, principal component analysis (PCA), information gain ratio (IGR), and support vector machine (SVM), have been applied to detect IPv4-based attacks [8–11]. However, existing machine learning techniques applied in IPv4 networks cannot be utilized to detect RA flooding attacks since these methods cannot locate and inspect ICMPv6, given that the structure of IPv6 is different from that of IPv4. As a result, RA flooding attacks can bypass these methods. In this study, a new technique is developed to detect the presence of RA flooding attacks in IPv6 networks. This technique is a combination of IGR, PCA, and SVM. IGR and PCA are feature reduction techniques used to select a set of new features that has significant contributions in detecting RA flooding attacks, and SVM utilizes the results of IGR and PCA to train a prediction model. Consequently, the anomaly of input traffic can be detected with this model.

The rest of the paper is organized as follows. The "Background" section presents a review of NDP based attacks and machine learning techniques. The "Related Work" section reviews the previous literatures related to RA flooding attacks. The "Proposed Technique" section describes the proposed technique to detect RA flooding attacks. The "Experiment Evaluation" section presents an evaluation of the proposed technique, and the "Conclusion and Future Work" section provides the conclusions and possible future research directions.

Background

The problem is ICMPv6 flooding attacks (RA flooding) in IPv6 networks, and the methods are IGR, PCA, and SVM

ICMPv6 Flooding Attacks (RA Flooding)

IPv6 routers use NDP to discover one another's presence and determine link-layer addresses and prefix information. A receiving node does not validate router advertisements. Thus, any node that receives a fake RA updates its communication parameters blindly based on the RA. A malicious node can propagate bogus address prefix information to reroute legitimate traffic and prevent the victim from accessing the desired network [12, 13]. Flooding a local network with a different network prefix and having hosts and routers update the network parameters with information based on the announced prefix would consume all available CPU and memory resources, thus rendering the system unusable and unresponsive.

Given that the IPv6 protocol is enabled by default in most modern operating systems, all nodes in the network are affected in their default configuration. For example, in Windows, a personal firewall or a similar security product is not protected against an RA flooding attack.

An RA message is sent to the FF02::1 multi-cast group so that all hosts on the same link receive the announced fake prefix. In turn, these hosts configure their default gateway based on the fake announced prefix. A flag in IPv6 router advertisements determines the default router preference. First, by default, the legitimate router sends out RA messages with the router preference flag set to "medium."

Fake RA messages commonly set the preference flag to "high," thus forcing the hosts to use it as their default gateway. Changing the preference flag value is a normal procedure. However, signature-based techniques cannot differentiate between legitimate and non-legitimate RA messages.

The attacker sends a large number of RA messages to all hosts on the same link. Consequently, the nodes' resources are consumed because these nodes continue to generate a new IPv6 address for each announced prefix.

IGR

IGR [14] depends on the entropy algorithm [15] that measures the disorder in a system and the information gain that measures the decrease in entropy achieved in the classification based on a particular feature.The purpose of using IGR is to identify important and effective features to detect an RA flooding attack. IGR assigns the important and effective features with a large weight value. In contrast, the less important features are assigned with small weight value. In the equations below, Ex is the set of all training data examples, $value(x, f)$ with $x \in Ex$ defines the value of a specific example x. Hence,

$$IGR(Ex, f) = \frac{Gain(Ex, f)}{SplitInfo(Ex, f)} \tag{1}$$

$$Gain(Ex, f) = Entropy(Ex) - \sum_{v \in value(f)} \frac{|Ex_v|}{|Ex|} \times Entropy(Ex_v),$$

Table 1 NDP related ICMPv6 type

ICMPv6 packet type	Description
Router Solicitation(RS)-Type 133	Message sent by host to request a router to send a router advertisement.
Router Advertisement(RA)-Type 134	Routers advertise their presence together with various links and Internet parameters either periodically or in response to an RS message.
Neighbor Solicitation(NS)-Type 135	Neighbor solicitations are used by nodes to determine the link layer address of a neighbor or to verify that a neighbor is still reach- able via a cached link layer address. .
Neighbor Advertisement(NA)-Type 136	NA message sent by nodes to respond to an NS message.
Redirect-Type 137	Routers may inform hosts of a better first hop router for a destination.

$$Ex_v = \left\{ x \in \frac{E_x}{value(x, f)} \right\} = v$$

The entropy function is the Shannon's entropy that is defined as

$$Entropy\,(Ex) = -\sum P_i \log_2 (P_i) \tag{2}$$

where P_i is the probability of class i.

The split information value, $SplitInfo(Ex, f)$, that represents the potential information generated by splitting the training data set Ex into v partitions, corresponding to v outcomes on attribute f is defined as

$$SplitInfo(Ex, f) = -\sum_{i=1}^{v} \frac{|Ex_i|}{|Ex|} \times \log_2 \frac{|Ex_i|}{|Ex|} \tag{3}$$

where Ex_i is the set of all training examples in partition i. The attribute with the maximum gain ratio is s split attribute.

PCA

PCA is a well-developed method to reduce dimensionality and multivariate analysis. Examples of its applications include data compression, image processing, visualization, exploratory data analysis, pattern recognition, and time series prediction [16–18].

PCA is a mathematical method that transforms a number of possibly correlated variables into a new set of uncorrelated variables called principal components. The first principal component has the highest variability in the dataset. In many datasets, the first several principal components have the highest contribution to the variance in the original dataset. Therefore, the rest can be disregarded with minimal loss of the information value during the dimension reduction process [19].

PCA is popular because of its three important properties. First, it is an optimal (in terms of mean squared error) linear scheme for compressing a set of high-dimensional vectors into a set of low-dimensional vectors and then reconstructing the original set. Second, the model parameters can be computed directly from the data, similar to the example of digitalizing the sample covariance matrix. Third, compression and decompression are easy to perform with given model parameters. They require only matrix multiplication.

Given a set of observations $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in}), \forall i \in (1, 2, \ldots, m)$ where each observation is represented by a vector of length n, the data set is represented by a matrix $\boldsymbol{X}_{n\times m}$

$$X_{\mathbf{n},\mathbf{m}} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \tag{4}$$

The mean is defined by the expected value, as presented in Eq. (5).

$$m = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{5}$$

The covariance matrix is defined as

$$\sum = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - m) \times (x_i - m)^t \tag{6}$$

The covariance matrix is one of the most important mathematical concepts in data analysis. If the data in the new coordinate system are presented by y, then linear transformation G of the original coordinates should be determined, as presented in Eq. 7.

$$y = Gx = D^t \times x \tag{7}$$

Replacing G with D^t would make any further comparison of principal components with other transformation methods much simpler. The covariance matrix in the y space is defined by Eq. (8):

$$\sum y = D^t \times \sum x \times D \tag{8}$$

where $\sum x$ is the covariance of the data in x space. Since $\sum y$ needs to be diagonal, D can be recognized as the matrix of eigenvectors of $\sum x$, providing D is an orthogonal matrix.

$\sum y$ is a diagonal matrix shown in (9). Thus, $\sum y$ can be identified as the diagonal matrix of eigenvalues of $\sum x$.

$$\sum y = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \tag{9}$$

We let n be the dimensionality of the data.

The covariance matrix is used to calculate diagonal matrix y. y is sorted and rearranged in the form of n as $(\lambda_1 > \lambda_2 > \ldots > \lambda_n)$ so that the data exhibits the maximum variance in y_1, the next largest variance in y2, and so on (with minimum variance in y_n).

This feature extraction problem has been studied by many researchers. For example, Xu et al. [20] selected eight relative values as features that are independent of network flow. Zargar et al. [21] proposed and investigated the identification of effective network features to probe attack detection using the PCA method to determine an optimal feature set (e.g., principal component analysis, factor analysis, projection pursuit, and independent component analysis). In the current work, PCA was employed because of its ideal mean-square error and linear dimension reduction technique [18].

SVM

SVM is a powerful machine learning approach that has been employed, both in its original and variant forms, in a range of challenging real-world applications [22]. The simplest SVM model addresses the binary classification problem that is separated by a hyperplane defined by a number of support vectors. Support vectors are subsets of training data used to define the boundary between two different classes, namely, RA attack and normal. In situations where SVM cannot separate two classes, the input data are mapped into high-dimensional feature spaces using a kernel function. In a high-dimensional space, creating a hyperplane that allows linear separation is possible; this hyperplane corresponds to a curved surface in low-dimensional input space. The kernel function plays an important role in SVM [10, 18].

In practice, various kernel functions, such as linear, polynomial, or radial basis function (RBF), can be utilized. Figure 1 shows the basic idea behind the use of kernel functions by SVMs. In Fig. 1, the original objects (left side of the schematic) are mapped, that is, rearranged, by using a set of mathematical functions known as kernels. The process of rearranging these objects is known as mapping (transformation). In this new setting, the mapped objects (right side of the schematic) are linearly separable. Instead of constructing the complex curve (left schematic), an optimal line is fixed to separate GREEN and RED objects.

We suppose that N training data points (x_1, y_1), (x_2, y_2), (x_3, y_3), $\ldots$, (x_n, y_n), exist.

$xi \in Rd \quad and \quad y_i \in \{+1, 1\}$. We consider a hyperplane defined by (w, b), where w is a weight vector and b is a bias.

If the training data are linearly separable, then pair $(w, b) \in R^n \times \mathbf{R}$ exists, such that

$$w^T x + b \geqslant 1 \qquad \forall x \in A \tag{10}$$

$$w^T x + b \leq 1 \qquad \forall x \in B \tag{11}$$

with the decision function given by

$$f(x)_{w,b} = \sin(w^T x + b) \tag{12}$$

Where w is the weight vector and b is the bias. The inequality constraints, Eqs. (11) and (12) can be combined to obtain

$$y(w^T x + b) \geqslant 1 \qquad \forall \qquad x \in A \cup B \tag{13}$$

The maximal margin classifier optimizes this condition by separating the data with the maximal margin hyperplane. The learning problem is thus formulated as minimize $\frac{1}{2}\|w\|^2$ subject to the constraints of linear separability. The optimization is a quadratic programming (QP) problem:

$$\begin{aligned} &min_{w,b}\phi(w) = \tfrac{1}{2}\|w\|^2 \\ &s.t. \qquad y(w^T x + b) \geqslant 1 \end{aligned} \tag{14}$$

$$K(X_i, X_j) = \left\{ \begin{array}{ll} X_i.X_j & linear \\ (\gamma X_i.X_j + c)^d & Polynominal \\ \exp(-\gamma|X_i - X_j|^2) & RBF \\ \tanh(\gamma X_i.X_j + c) & Sigmoid \end{array} \right\} \tag{15}$$

where $K(X_i, X_j) = \phi(X_i) \cdot (X_j)$, that is, the kernel function, represents a dot product of input data points mapped into the higher dimensional feature space by transformation ϕ. where γ is an adjustable parameter of certain kernel functions.

Related Work

Similar to IPv4 headers, IPv6 headers possess no security mechanisms. IPsec is available in IPv6. Administrators rely on the IPsec protocol suite for security. The same security risks for Man in the Middle (MitM) attacks in Internet Key Exchange (IKE) in IPv4 are also present in IPv6 [23].

An IPsec authentication header (AH) can be used with NDP (NS/NA) messages to enhance security and verify that messages contain proper and accurate information. Security associations (SAs) can be created only through IKE. However, IKE requires a functional IP stack to function, which results in a bootstrapping problem. SA can only be configured manually, which is a tedious or impractical task

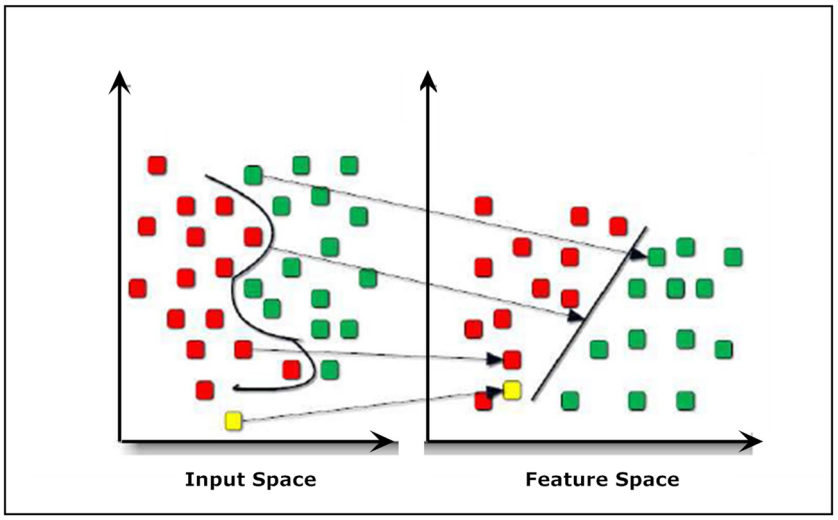


Fig. 1 Mapping input space to feature space

because of the volume. Even when SAs are established, verifying the ownership of dynamically generated IP addresses is impossible [23].

Mechanism to secure NDP with a cryptographic method that is independent of IPsec (the original and inherent method of securing IPv6 communications) is called Secure Neighbor Discovery(SEND). SEND was introduced to prevent address theft by proposing cryptographically generated addresses (CGA) and protecting data with an RSA signature [24]. Numerous researchers [25] reported that the use of SEND is complex and may causea DoS attack on the target node because of the high complexity and computation involved in encryption and decryption processes, which negatively affect the node resources.

The NDP Monitor (NDPMon) is a diagnostic software application used by IPv6 network administrators to monitor ICMPv6 packets. NDPMon monitors the local network for anomalies in the function of nodes using NDP messages, especially during stateless address autoconfiguration. When an NDP message is flagged, NDPMon notifies the administrator by writing to the system log or by sending an email report. NDPMon may also execute a user-defined script [26].

IPv6 RA Monitoring Daemon [27] monitors ND traffic to detect possible attacks when discrepancies exist between the information advertised in ND packets and the information stored on a local database. A key challenge in this detection mechanism is the introduction of IPv6 fragmentation. Concealing an attack by fragmenting packets into multiple fragments is simple. This condition may limit or even eliminate the effectiveness of the aforementioned detection mechanism. The main drawbacks of these passive schemes are lack of dynamism, scalability, false alarms, and violation of protocol stack.

Barbhuiya et al. [1] adapted the basic idea of active IDS [28] and applied it to NDP-related attacks. The change in the new MAC-IP pair can be detected by sending an NS request packet to the target host to inquire if the MAC-IP pair is genuine or spoofed. If the MAC-IP pair is genuine, the host will respond with a corresponding NA reply packet. If the MAC-IP pair is spoofed, no reply (if the IP address is nonexistent) or multiple replies (from the genuine host and attacker) may be generated. The proposed mechanism focuses on detecting two types of NDP-related attacks, namely, NS and NA spoofing

Saad, et al. [29] proposed rules to detect the abnormal behavior of ICMPv6. The back-propagation neural network (BPNN) algorithm was employed to verify the abnormal behaviors detected by the proposed rules; these abnormal behaviors are a result of the presence of an ICMPv6-based DDoS attack. The experimental result showed that the proposed technique can detect ICMPv6-based DDoS attacks with a detection accuracy of 98.3%. However, the proposed technique focuses only on detecting ICMPv6 echo request flooding attacks. In addition, the proposed technique uses features that are not effective in detecting ICMPv6 flooding attacks when operating online. For example, time and source IP address features and training data are collected until time t, whereas the classifier operates online to detect the attack after t. This scenario leads to misclassification of the incoming network traffic. Furthermore, the classifier uses the source IPs as indicators during model training. Therefore, any IP that does not exist within the range of IPs in the training model are incorrectly classified

RA guard technique was proposed and explained in RFC 6105 [30] in 2011 to handle rogue RAs generated maliciously or unintentionally by unauthorized or improperly configured routers or devices. RA guard is installed in layer-2 switches to analyze RA messages and to filter out RAs that are sent by unauthorized devices based on certain criteria. It compares its criteria with selected information extracted

from IPv6 frames, such as MAC source address, physical port, IPv6 source, and prefix list. The decision of the RA Guard-enabled switch to discard or forward the RA message will be based on the comparison result. The determination can be based either on layer-2 configurations with configured access rules or on dynamically learning procedures of RA senders during a learning period.

Thus, RA Guard prevents DoS attacks that are based on RA messages only and this is one of its drawbacks. Moreover, it has more shortcomings as explained in [31, 32] that prevented widespread adoption of RA guard. Those limitations are as follows: (1) RA guard does not provide any security protection for devices connecting wirelessly such as WiFi devices. (2) RA guard does not offer protection for the egress direction of traffic since it only monitors ingress traffic in switch ports. (3) RA guard cannot be used on trunk ports with merge mode of ports. (4) Common and low-end switches do not typically support RA guard functionalities; therefore, additional cost is required to replace existing switches with newer and more expensive hardwares.

RFC7113 [31] was created in 2014 to describe two evasion techniques that affect some implementation of RA Guard. It also updates the original RA Guard specification, RFC6105 to eliminate the evasion vectors in those implementation. RA guard filtration can be circumvented using a combination of fragmented packets and extension header with the RA message. The use of fragmentation with extension header makes it impossible for layer-2 device to identify RA messages since reassembly of fragmented packet is not performed at layer-2.

SEND-based RA Guard was proposed to complement SEND in securing IPv6 local network in the environment where SEND might not be suitable or fully supported by all device in the network. An RA Guard-enabled switch acts as a node-in-the-middle, where upon capturing an RA packet at its interface, uses information from SEND to authenticate packet by verifying the CGA and RSA signature of the sender [30]. However, due to the complexity of router authorization mechanisms in SEND, AlSa'deh recommended alternative method of using Cryptographically Generated Address (CGA) with RA-Guard to prevent address theft and detect fake RAs [25].

Proposed Technique

The proposed technique aims to detect RA flooding attacks in an IPv6 link local network. An RA message appears in the network in the form of ICMPv6 (type 134). A router in an IPv6 network periodically sends an RA message that contains network prefix lifetime and configuration type to all nodes on the same link. The IPv6 node can induce the router to send an RA message by sending the RS to the FF02::2 multi-cast group. Once the IPv6 nodes configured their routing table and IPv6 address based on the RA and implanted a default gateway, a receiving node does not validate the RA. Thus, any node that receives a fake RA updates its communication parameters blindly based on the received RA. Figure 2 shows the three consecutive phases of the proposed technique, namely, filtering, feature reduction, and detection. These three phases are discussed thoroughly in the following subsections.

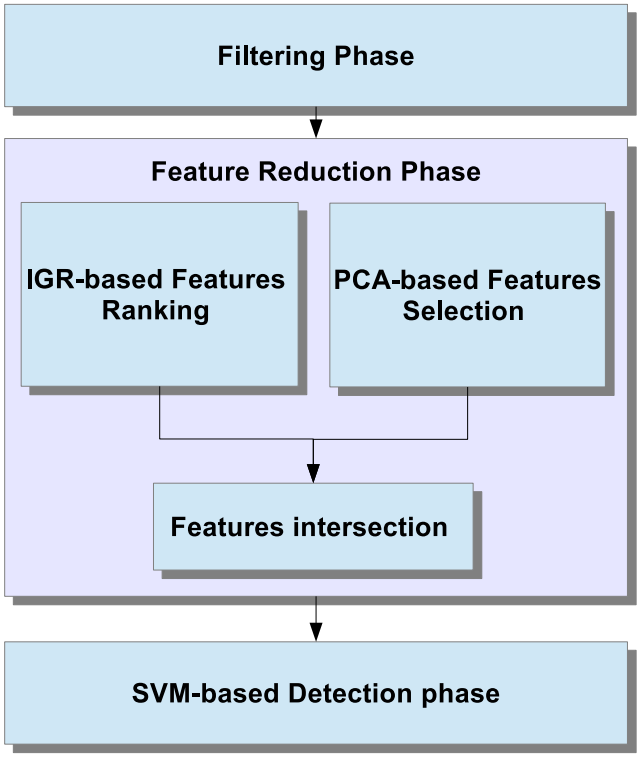


Fig. 2 The architecture of the proposed technique

Filtering Phase

This phase deals with the voluminous amount of IPv6 network traffic. This traffic contains different types of protocols that do not contribute to the detection of RA flooding attacks. Thus, the incoming traffic is filtered into ICMP type 134 (RA). The filtering phase consists of one log table, which is used to log the ICMP type 134 features, such as time, source IP, destination IP,protocol, and packet size. The usefulness and importance of the filtered features are verified in the next phase. Figure 3 shows the flowchart for the filtering phase.

We suppose that series of packets $\xi = (x_i, x_{i+1}, \ldots x_k)$ exist, where i refers to the number of incoming packets. For each packet $x_i = (f_j, f_{j+1}, \ldots, f_N)$

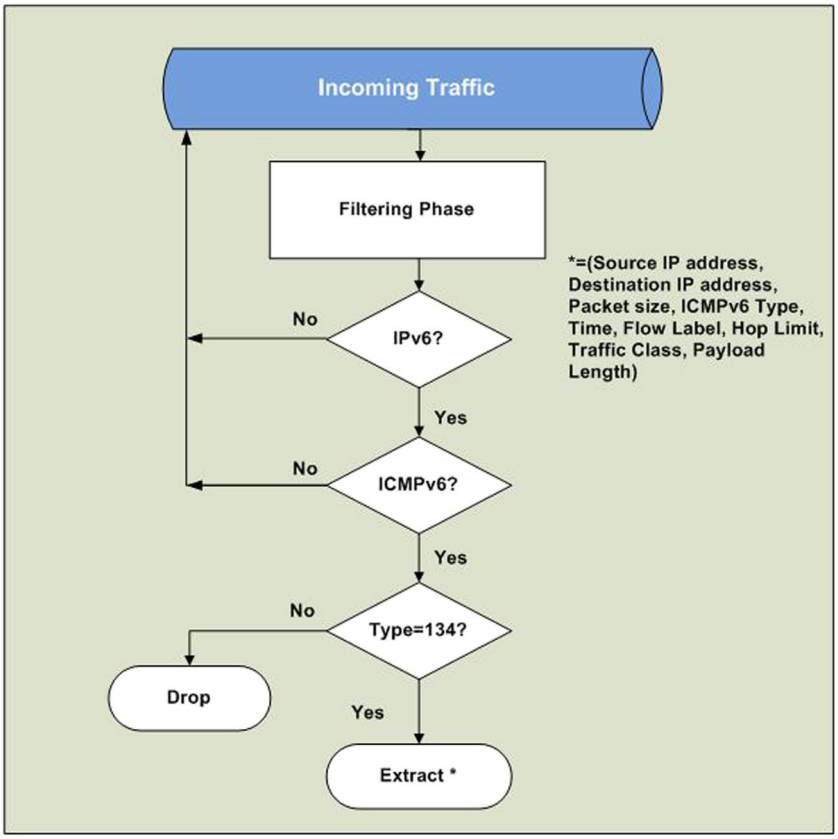


Fig. 3 The flowchart of filtering phase

where $j = 1, 2, \ldots, N$ refers to the packet features number for each ξ out of N features. The important features, m, that contribute to detecting the RA flooding attack is $m < N$. In the filtering phase, the extracted features m are logged into log table V, as shown in Algorithm 1.

Algorithm 1 Filtering algorithm

ρ ={Source IP address, Destination IP address, Packet size, ICMPv6 Type, Time, Flow Label, Hop Limit, Traffic Class, Payload Length}
for $i = 1 \quad to \quad k$ **do**
  for $j = 1 \quad to \quad N$ **do**
    if $f_j \in \rho$ **then**
      add f_j to **V**
    end if
  end for
end for

Features Reduction Phase

In the filtering phase, m features for ICMP type 134 (RA) were saved into the log table. The feature reduction phase selects the most contributed m features that reflect all the filtered features. The main advantage of the feature reduction phase is that it increases the detection accuracy for RA flooding attacks. The feature reduction phase consists of three sub-steps: IGR-based feature ranking, PCA-based feature extraction, and feature intersection.

IGR-Based Feature Ranking

Features ranking is achieved by the new proposed technique to identify important and effective features to detect an RA flooding attack. One of the most effective algorithms used in feature ranking is IGR.

In IGR-based feature ranking, each filtered feature in log table $x_i(f_j)$, where $\forall i \in (1, 2, \ldots, k)$ and $\forall j \in (1, 2, \ldots, m)$ is assigned a weight value. $IGR(x_i, f_j)$, based on Eq. (1) in "IGR".

The features f_j with a large weight value is highly important. By contrast, features with small weight value reflect low importance. For example, the feature with a zero weight value is disregarded from the extracted feature list. Therefore, the extracted feature list is $\delta_1 = \{f_1, f_2, \ldots, f_r\}$. The output list is then used as an input for features intersection step.

PCA-Based Feature Selection

In this step, PCA is adopted to reduce the dimensionality of the data and to select only the significant features. PCA is used to select an independent set of features. The resultant

features will be used as input for next step, if a set of features are selected only by IGR or PCA then these features will not be considered as significant features. Therefore, feature intersection step is proposed to make sure that selected features are selected by IGR and PCA.

PCA is a popular method of feature selection [33] and is widely used in dimensionality reduction for data analysis and compression. PCA transforms a relatively large number of variables into a small number of uncorrelated variables. This transformation is implemented by fixed orthogonal linear combinations of the original variables with the largest variance. In feature extraction, a subset of relevant features is extracted from the total number of features in the dataset and used to build the RA detection phase (discussed in the "SVM Detection Phase" section). The extracted feature list is $\delta_2 = \{f_1, f_2, \ldots, f_{r1}\}$, where $r1 \leq m$ is the output of this phase.

Features Intersection Step

Improving the prediction performance provides fast and cost-effective predictors and improved understanding of the underlying process that generates the data. RA flooding attack detection accuracy increases by selecting the correct and most contributed features relevant to the RA flooding attack. Feature intersection selects the features that exist in the output of both feature ranking step δ_1 ("IGR-Based Feature Ranking") and feature selection step δ_2 ("PCA-Based Feature Selection") and excludes the rest of the features. The intersection of the result provided by several selection and ranking techniques may lead to a highly accurate feature list. The features intersection phase is defined as follows:

We assume two different sets δ_1 and δ_2. $\delta_3 = \delta_1 \bigcap \delta_2$, where δ_3 is the result of intersection, δ_1 represents the set of features obtained from the IGR-based feature ranking step, and δ_2 represents the set of features obtained from the PCA-based feature selection step. The output of this phase is the features that exist in feature ranking and feature selection steps.

SVM Detection Phase

To detect the presence of an RA flooding attack in an IPv6 network, supervised learning should be considered in the measurement, in which the observation of the target function is known. The attack detection process is suggested to be the target function $f(x) = y_i$, where $y_i = \{normal, attack\}$. SVM is adopted as a training algorithm. The process of separating genuine from malicious ones is based on the training model that contains the genuine packets only. The network administrator trains the target network

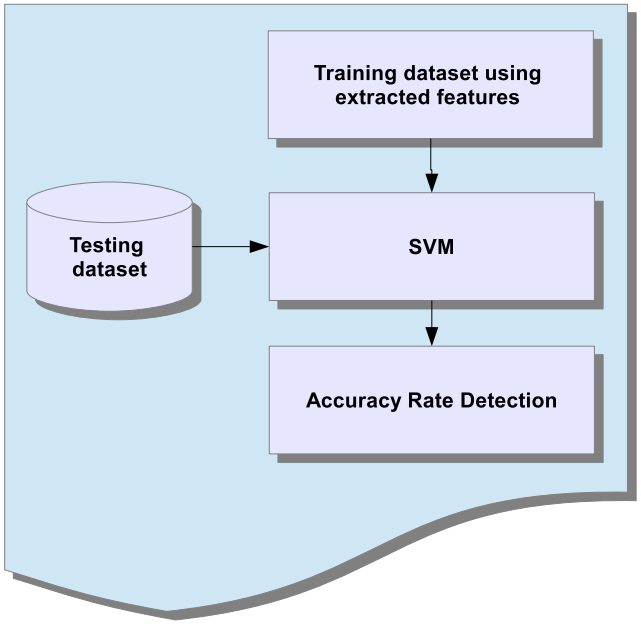


Fig. 4 The process of SVM-based detection phase

in an offline manner to come up with a solid detection model. Then, SVM operates online based on the trained model to detect the RA flooding attack. SVM aims to identify the optimal hyperplane that separates classes of y_i by using the RBF kernel function presented in Eq. (15). RBF is the most popular choice of kernel type used in SVM because of its localized and fixed responses across the entire range of the real x-axis [34, 35]. This phase uses the selected features from the feature reduction phase. These features consist of both attack and normal traffic to be used for SVM training. The output of this phase is the training model. This model is subsequently utilized to detect the RA flooding attack from any input traffic. Figure 4 shows the main process of SVM.

As shown in Fig. 4, SVM uses a subset of the input dataset as the training model. The number of features for the trained dataset is equal to the number of features of δ_3, and the number of features for the testing dataset may or may not be equal to the number of features of δ_3. Having less or more features for training dataset will lead to negative impact (such as low accuracy detection).

The main contribution of applying IGR, PCA, and SVM techniques in IPv6 networks is to reveal their capabilities in detecting new security issues (RA flooding attacks in this study) that exist in IPv6 networks; while the difference between applying these techniques in IPv4 and IPv6 networks is the selection of the nominated features that highly contribute to detecting RA flooding attacks. Given that each attack has related features that help in the detection, these features would most probably be selected in

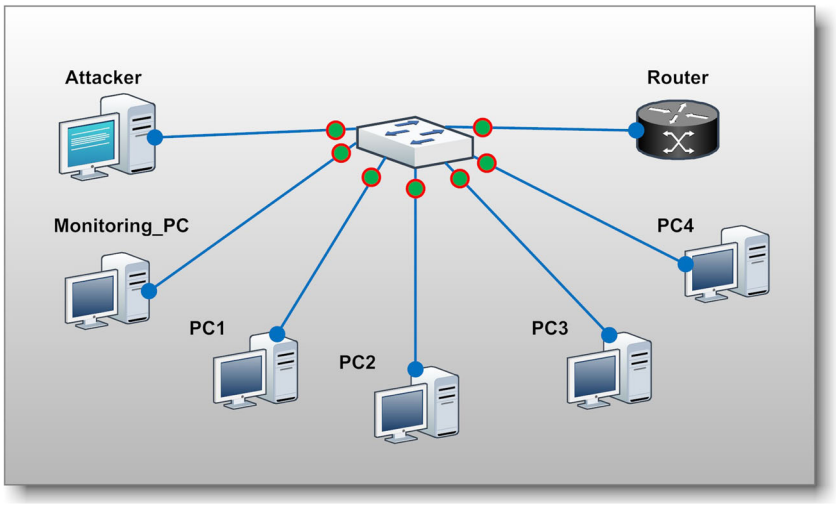


Fig. 5 Test bed topology

the proposed technique. Therefore, the benefit of the proposed technique is to select a set of significant features that lead to accurate detection of the presence of RA flooding attack.

At this stage, the proposed technique focuses on the detection accuracy for RA flooding rather than memory and CPU complexities. CPU and memory complexities have a linear correlation with the number of packets captured and analyzed. To prevent CPU and memory saturation, a high-end server architecture can be employed to perform packet capturing and analysis while keeping the processing time low. Using the feature reduction phase leads to an improved SVM training model that can accurately detect RA flooding attacks from input traffic.

Experiment Evaluation

This section evaluates the robustness of the proposed technique in detecting the presence of RA flooding attacks. A real dataset is used to evaluate the proposed technique. The characteristic of the real dataset is provided in the "Dataset Definition" section. Experiments are conducted to mature the output of this phase by using the simulated dataset presented in the "Feature Reduction Phase" section because the main contribution of this study is the process of selecting the reduction features in phase 2. The accuracy of detecting RA flooding attacks from input traffic data is experimentally measured using the simulated dataset in the "Detection Phase" section.

Dataset Definition

The proposed technique is evaluated with a real dataset with a total of 199,138 different time-stamped row packets of both RA attack and normal traffic. Normal traffic is obtained from the National Advanced IPv6 Center of Excellence (NAv6) [36], and the RA attacks are obtained from an isolated testbed to avoid the propagation of the attacks to the real network. A GNS3 simulator is used to design the topology of the testbed (Fig. 5), which consists of six IPv6-enabled hosts as listed in Table 2. The THC-IPv6 toolkit is employed to launch an RA flooding attack. The generated traffic consists of fake RA messages (generated by the THC

Table 2 Hosts in the designed topology

PC name	Usage	OS type
PC1	Normal user	Windows 7
PC2	Normal user	Windows XP
PC3	Normal user	Ubuntu
PC4	Normal user	Windows 7
Attacker	Utilizes THC-IPv6 tool to trigger an RA flooding attack	Backtrack 5 R3
Monitoring PC	Sniffs all the packets that bypass the network.	Windows XP

Table 3 Machine specifications

CPU	Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.67GHz
Memory	5.00 GB
Operating system	Windows 7 (64 bit)

toolkit) and legitimate RA messages (generated by the legitimate router, as shown in Fig. 5). The difference between the generated and detected attack numbers is that the generated represents the entire traffic that bypassed the designed topology while the detected attack represents the traffic that have fake RA message. The entire traffic that bypassed the network is captured with the Wireshark sniffer tool [37]. Therefore, the captured traffic from the testbed along with real traffic from NAv6 are merged to be the input for the proposed technique. Table 3 shows the specifications of the machine used for training and detection.

The dataset attributes that represent the features of δ_3 values are normalized. The conversion of non-integer features (Traffic Class, ICMPv6 Type, Hop Limit, Flow Label features) is done by extracting the distinct values for each feature and assign each distinct value a unique integer number The remaining Payload Length size attribute value is unchanged.

Feature Reduction Phase

The initial set of features, m, consists of nine features based on Algorithm 1. The selection of the most significant features among these nine features is crucial. Relying on simple heuristics to select the significant features may lead to inappropriate feature selection, which reduces the detection accuracy for RA flooding attack. In addition, using the entire set of features without any reduction is resource consuming. Thus, IGR and PCA are employed independently on the dataset, which includes m features to reduce the high-dimensional data vectors. Therefore, detection is handled in a low-dimensional space with high efficiency and minimal use of system resources. The features that exist in the output of both PCA and IGR are considered the significant features that contribute to proper detection of RA flooding attacks. The effectiveness of the selected features is demonstrated in the "Detection Phase" section by using a real dataset.

The experiment for PCA and IGR was conducted using RapidMiner [38] with their default parameters setting. The features in Table 4 are the experimental results of extracting features using IGR. Table 5 shows the experimental results of extracting features using PCA.

$\delta_1 = \{Length, Source\ IP, Traffic\ Class, ICMPv6\ Type, Hop\ Limit, Flow\ Time, Payload\ Length, IP\ Destination\}$, where δ_1 represents features that ranked by IGR.

Label, $\delta_2 = \{Traffic\ Class, ICMPv6\ Type, Hop\ Limit, Flow\ Label, Payload\ Length\}$, where δ_2 represents features that ranked by PCA.

$\delta_3 = \delta_1 \bigcap \delta_2$, where δ_3 represents the result of features reduction phase.

$\delta_3 = \{Traffic\ Class, ICMPv6\ Type, Hop\ Limit, Flow\ Label, Payload\ Length\}$.

As a result, features $\in \delta_3$ have a high impact in detecting RA flooding, whereas the remaining features are negligible because the weight is equal to 0, as shown in Table 4, or because of the standard deviation that is equal to 0, as shown in Table 5.

Detection Phase

In this phase, the training model is created through SVM. The training model is divided into training dataset1 and

Table 4 Output of IGR

Attribute	Weight
Length	0.854
Source IP	0.787
Traffic class	0.586
ICMPv6 type	0.493
Flow label	0.455
Hop limit	0.455
Payload length	0.455
Time	0.346
Destination IP	0.257

Table 5 Output of PCA

Attribute	Weight
Flow label	0.435
Hop limit	0.434
Payload length	0.427
Traffic class	0.425
ICMPv6 type	0.361
Time	0.0
Destination IP	− 0.138
Source IP	− 0.167
Length	− 0.282

Table 6 Details of the datasets

Dataset	No. of row packets
Training Dataset1	199138
Training Dataset2	199138

training dataset2. Training dataset1 is generated with δ_3 features, and training dataset2 is generated using the original set of features m. The purpose of the diversity in training datasets is to test the robustness of the technique in detecting the presence of RA flooding attacks in IPv6 networks and to study the effect of selected features on the detection process. Table 6 shows the details of the testing datasets. Figure 6 depicts the evaluation of the proposed technique in terms of the accuracy and false positive rate while Fig. 7 depicts the evaluation of the proposed technique in term of the precision and recall rate.

The observed effect of training dataset size on the false positive rate (FPR) is slightly different. The mean of FPR increases slightly and the variance decreases as the training dataset size increases. This result indicates that in the case of FPR, small training sets can be used to produce low FPRs. The false negative rate (FNR) and FPR statistics demonstrate that using large training sets entails minimal benefits [39]. The output of this phase is utilized to measure the accuracy, false positive, precision, and recall of the proposed technique. Accuracy is a criterion used to measure IDS performance. The number of false alarms the system produces and the percentage of detection and failure are correctly declared. According to [40], a system that has 80% accuracy may be a system that properly classifies 80 cases out of the 100 existing classes. The standard formula to calculate the accuracy of detecting RA flooding attacks is shown in Eq. (16), and the calculation of FPR is shown in Eq. (17).

The calculation of precision is shown in Eq. (18). Equation (19) is used to calculate the Recall. Table 7 presents a description of each metric in the accuracy equation.

$$Accuracy = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) * 100 \tag{16}$$

$$FPR = \left(\frac{FP}{TN + FP}\right) * 100 \tag{17}$$

PRECISION (P) is the proportion of attack cases that are correctly predicted relative to the predicted size of the attack class as calculated using the following equation

$$PRECISION = \left(\frac{TP}{TP + FP}\right) * 100 \tag{18}$$

RECALL (R) is the proportion of correctly predicted attack cases to the actual size of the attack class as calculated using the following equation

$$RECALL = \left(\frac{TP}{TP + FN}\right) * 100 \tag{19}$$

As shown in Fig. 6, the proposed technique can detect the presence of an RA flooding attack by using the features δ_3 selected in the feature reduction phase. The detection accuracy is 98.55%, and the FPR is 3.3%. Accuracy decreases to 94.93% with an FPR of 4.2% when the set of original features m is used. Therefore, the high accuracy of the proposed technique is attributed to the proposed feature reduction methods. Such methods aim to select the most important features that contribute to the detection of the RA flooding attack.

As shown in Fig. 7, the proposed technique with the selected features δ_3 has a higher precision percentage (99.1%) compared to proposed techniques with the original set of features, which means that it has lower false-positive

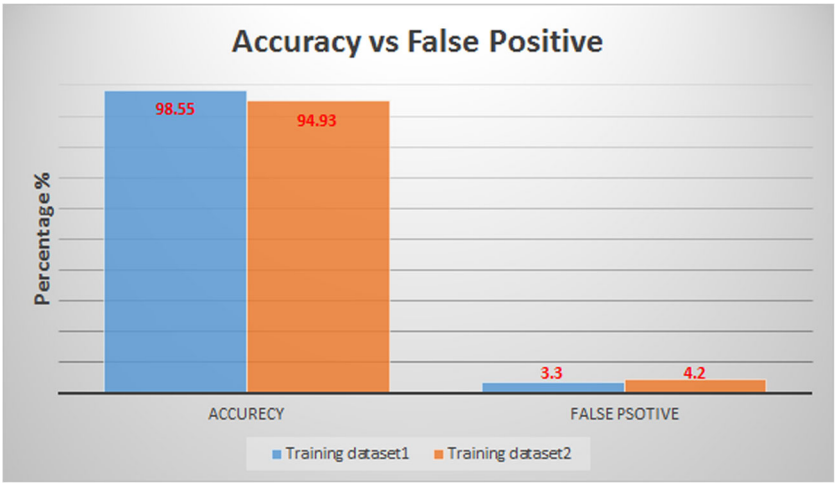
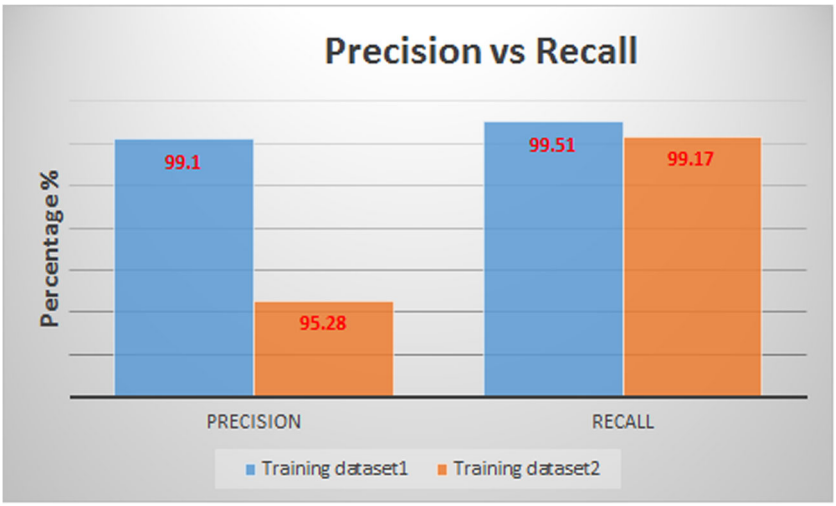


Fig. 6 Accuracy vs false positive

Fig. 7 Precision vs recall



value. Meanwhile, the proposed technique with the selected features δ_3 has a higher recall percentage (99.51%) compared to proposed techniques with the original set of features, which means that it has lower false-negative value. The robustness of SVM as a training model is achieved, as revealed by the highly accurate results in detecting the RA flooding attack. Ignoring one of the selected features would negatively affect the accuracy of the proposed technique.

The biologically inspired algorithms have been utilized to enhance the performance of machine learning algorithms [41, 42]. The classification performance obtained by SVM is influenced by the choice of proper values for their free parameters. Biologically inspired algorithms can be used as optimization techniques to select the proper values of the free parameters of SVM.

A better improvement for SVM classification can be achieved by utilizing Cognitive computation algorithm such as Cognitive Diagnosis Algorithm that was proposed in [18] or utilizing bio-inspired algorithms such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for features selection process.

Utilizing an efficient features selection techniques to select the significant features that are related to RA flooding attack will have a positive impact on the detection accuracy. As a result, it will assist the network administrators to take the right action to contain the presence of RA flooding attacks in the IPv6 network.

Conclusion and Future Work

A machine-learning-based technique to detect RA flooding attacks was proposed. The proposed technique consists of three phases. The filtering phase filters the related RA flooding attack protocols and features. The feature intersection phase, which is based on IGR and PCA, selects the best features that can highly contribute to detecting RA flooding attacks. The detection phase, which is based on SVM, creates a training model to detect the RA flooding attacks. Three different simulated datasets were generated to evaluate the effectiveness of the proposed technique in detecting RA flooding attacks. In addition, the effect of the feature reduction phase on the accuracy of detecting RA flooding attacks was evaluated. The results showed that the proposed technique is sufficiently accurate to detect RA flooding attacks. The feature reduction phase exerts a positive effect and contributes significantly to the detection of RA flooding attacks.

Future work can focus on increasing the accuracy of the proposed technique by using another well-trained model generated by more efficient training algorithms. In addition, different Dimensionality reduction algorithms such as Large-margin Weakly Supervised Dimensionality Reduction (LWSDR) and Dimensionality Reduction with Subspace Structure Preservation (DRSSP) can be evaluated to assess their impact on the accuracy detection

Table 7 Description of accuracy metrics

Metric	Description
True positive (TP)	Number of samples correctly predicted as attack class
False positive (FP)	Number of samples incorrectly predicted as attack class
True negative (TN)	Number of samples correctly predicted as normal class
False negative (FN)	Number of samples incorrectly predicted as normal class.

of training algorithms. Furthermore, more biologically inspired machine learning algorithms such as Deep Learning [43], Genetic Algorithms and Particle Swarm Optimization algorithms [44] can also be utilized, as these have been proven to be highly efficient in features selection and classification. Moreover, the scope of our proposed approaches can be extended to the detection of other NDP-based attacks.

Acknowledgements The authors are grateful to the anonymous reviewers for their constructive comments and suggestions, which greatly helped improve the quality of the paper. Professor A. Hussain is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant no. EP/M026981/1.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Barbhuiya FA, Bansal G, Kumar N, Biswas S, Nandi S. Detection of neighbor discovery protocol based attacks in ipv6 network. Netw Sci. 2013;2(3-4):91–113.
2. Goel JN, Mehtre B. Stack overflow based defense for ipv6 router advertisement flooding (dos) attack. In: Proceedings of 3rd international conference on advanced computing, networking and informatics. New Delhi: Springer; 2016. p. 299–308.
3. Caicedo CE, Joshi JB, Tuladhar SR. Ipv6 security challenges. Computer. 2009;42(2):36–42.
4. Narten T, Simpson WA, Nordmark E, Soliman H. Neighbor discovery for ip version 6 (ipv6), Tech. Rep. 2461, 2007, obsoleted by RFC 4861, upyeard by RFC 4311. [Online]. Available: http://www.ietf.org/rfc/rfc2461.txt.
5. Finlayson R, Mann T, Mogul J, Theimer M. A reverse address resolution protocol, Tech. Rep., 1984, rFC-903, JUN. [Online]. Available: http://www.ietf.org/rfc/rfc903.txt.
6. Hendriks L, Sperotto A, Pras A. Characterizing the ipv6 security landscape by large-scale measurements. In: IFIP international conference on autonomous infrastructure, management and security. Cham: Springer; 2015. p. 145–149.
7. Barbhuiya FA, 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. New York: ACM; 2011. p. 111–118.
8. Xu X, Wang X. An adaptive network intrusion detection method based on pca and support vector machines. In: Advanced data mining and applications. Berlin: Springer; 2005. p. 696–703.
9. De la Hoz E, De La Hoz E, Ortiz A, Ortega J, Prieto B. Pca filtering and probabilistic som for network intrusion detection. Neurocomputing. 2015;164:71–81.
10. Bamakan SMH, Wang H, Yingjie T, Shi Y. An effective intrusion detection framework based on mclp/svm optimized by time-varying chaos particle swarm optimization. Neurocomputing. 2016;199:90–102.
11. Shyu M-L, Chen S-C, Sarinnapakorn K, Chang L. A novel anomaly detection scheme based on principal component classifier. In: 3rd IEEE international conference on data mining; 2003. p. 353–365.
12. Yang X, Ma T, Shi Y. Typical dos/ddos threats under ipv6. In: International multi-conference on computing in the global information technology. Guadeloupe: IEEE; 2007. p. 55–55.
13. Anbar M, Abdullah R, Saad RMA, Alomari E, Alsaleem S. Review of security vulnerabilities in the IPv6 neighbor discovery protocol. Singapore: Springer Singapore, 2016, pp. 603–612. [Online]. Available: https://doi.org/10.1007/978-981-10-0557-2_59.
14. Hota H, Shrivas AK. Decision tree techniques applied on nsl-kdd data and its comparison with various feature selection techniques. In: Advanced computing, networking and informatics. Cham: Springer; 2014. p. 205–211.
15. Viertiö-Oja H, Maja V, Särkelä M, Talja P, Tenkanen N, Tolvanen-Laakso H, Paloheimo M, Vakkuri A, Yli-Hankala A, Meriläinen P. Description of the entropy algorithm as applied in the yearx-ohmeda entropy module. Acta Anaesthesiol Scand. 2004;48(2):154–61.
16. Lv JC, Yi Z, Li Y. Non-divergence of stochastic discrete time algorithms for pca neural networks. IEEE transactions on neural networks and learning systems. 2015;26(2):394–9.
17. Liu G, Yi Z, Yang S. A hierarchical intrusion detection model based on the pca neural networks. Neurocomputing. 2007;70(7):1561–8.
18. Yang J, Gong L, Tang Y, Yan J, He H, Zhang L, Li G. An improved svm-based cognitive diagnosis algorithm for operation states of distribution grid. Cogn Comput. 2015;7(5):582–93.
19. Wang W, Battiti R. Identifying intrusions in computer networks based on principal component analysis, Tech. Rep DIT-05-084. 2005.
20. Xu T, He D, Luo Y. Ddos attack detection based on rlt features. In: 2007 international conference on, computational intelligence and security; 2007. p. 697–701.
21. Zargar G, Kabiri P. Identification of effective network features for probing attack detection. In: NDT '09. First international conference on networked digital technologies, 2009. Ostrava: IEEE; 2009. p. 392–397.
22. Tanveer M. Robust and sparse linear programming twin support vector machines. Cogn Comput. 2015;7(1):137–49. [Online]. Available: https://doi.org/10.1007/s12559-014-9278-8.
23. Al-Shaer E. Modeling and verification of firewall and ipsec policies using binary decision diagrams. In: Automated firewall analytics. Cham: Springer International Publishing; 2014. p. 25–48.
24. Arkko J, Kempf J, Zill B, Nikander P. SEcure Neighbor Discovery (SEND), RFC 3971 (Proposed Standard), Tech. Rep. 3971, Mar. 2005, upyeard by RFCs 6494, 6495, 6980. [Online]. Available: http://www.ietf.org/rfc/rfc3971.txt.
25. AlSa'deh A, Meinel C. Secure neighbor discovery: review, challenges, perspectives, and recommendations. IEEE Secur Priv. 2012;10(4):26–34.
26. Beck F, Cholez T, Festor O, Chrisment I. Monitoring the neighbor discovery protocol. In: ICCGI, 2007. international multi-conference on computing in the global information technology, 2007; 2007. p. 57–57.
27. Chown T, Venaas S. Rogue ipv6 router advertisement problem statement, Tech. Rep., 2011, rFC-6104, Feb. [Online]. Available: https://tools.ietf.org/html/rfc6104.
28. Ramachandran V, Nandi S. Detecting arp spoofing: an active technique. In: International conference on information systems security. Berlin: Springer; 2005. p. 239–250.
29. Saad RM, Anbar M, Manickam S, Alomari E. An intelligent icmpv6 ddos flooding-attack detection framework (v6iids) using back-propagation neural network. IETE Tech Rev. 2015;33:244–55.

30. Levy-Abegnoli E, Van de Velde G, Popoviciu C, Mohacsi J. Ipv6 router advertisement guard, IETF, Tech. Rep., 2011, rFC-6105, Feb. [Online]. Available: https://tools.ietf.org/html/rfc6105.
31. Gont F. Implementation advice for ipv6 router advertisement guard (ra-guard), Internet Engineering Task Force (IETF), Tech. Rep., 2014, rFC-7113, Feb. [Online]. Available: https://tools.ietf.org/html/rfc7113.
32. Headquarters A. Ipv6 configuration guide, cisco ios release 12.4, Cisco, Tech. Rep., 2012. [Online]. Available: http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6/configuration/12-4t/ipv6-12-4t-book/ip6-eigrp.html.
33. Uğuz H. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowl-Based Syst. 2011;24(7):1024–32.
34. Sharma R, Pachori RB. Classification of epileptic seizures in eeg signals based on phase space representation of intrinsic mode functions. Expert Syst Appl. 2015;42(3):1106–17.
35. Lin S-l, Liu Z. Parameter selection in svm with rbf kernel function. J Zhengzhou Univ Technol. 2007;35(2):1–4.
36. NAv6. National advanced ipv6 centre, http://www.nav6.usm.my, 2016 online; accessed 1 OCT. 2016.
37. Narayanan HT et al. Seamless decoding of normal and oid compressed snmp pdus-an enhancement to wireshark. Procedia Eng. 2012;38:1479–86.
38. Naik A, Samant L. Correlation review of classification algorithm using data mining tool: weka, rapidminer, tanagra, orange and knime. Procedia Comput Sci. 2016;85:662–8.
39. Livadas C, Walsh R, Lapsley D, Strayer WT. Using machine learning techniques to identify botnet traffic. In: IEEE conference on local computer networks, Proceedings 2006 31st. Piscataway: IEEE; 2006. p. 967–974.
40. Elhamahmy M, Elmahdy HN, Saroit IA. A new approach for evaluating intrusion detection system. International Journal of Artificial Intelligent Systems and Machine Learning. 2010;11:2.
41. Gepperth A, Karaoguz C. A bio-inspired incremental learning architecture for applied perceptual problems. Cogn Comput. 2016;8(5):924–34. https://doi.org/10.1007/s12559-016-9389-5.
42. Javed SG, Majid A, Ali S, Kausar N. A bio-inspired parallel-framework based multi-gene genetic programming approach to denoise biomedical images. Cogn Comput. 2016;8(4):776–93. [Online]. Available: https://doi.org/10.1007/s12559-016-9416-6.
43. Wen G, Hou Z, Li H, Li D, Jiang L, Xun E. Ensemble of deep neural networks with probability-based fusion for facial expression recognition, Cogn Comput. 2017. [Online]. Available: https://doi.org/10.1007/s12559-017-9472-6.
44. Siddique N, Adeli H. Nature-inspired chemical reaction optimisation algorithms, Cogn Comput. 2017. [Online]. Available: https://doi.org/10.1007/s12559-017-9485-1.