

DETECTING DISTRIBUTED DENIAL OF SERVICE ATTACKS IN IPV6 BY USING ARTIFICIAL INTELLIGENCE TECHNIQUES

Abeer Abdullah Al-Sadhan

A thesis submitted in partial fulfilment of the requirements of Liverpool
John Moores University for the degree of Doctor of Philosophy

November 2019

Table of Contents

CHAPTER 1	7
1.1 Research Problem	9
1.2 Research Novelty	10
1.3 Research aim	10
1.4 Research objectives	10
1.5 Thesis organisation	11
CHAPTER 2	13
LITERATURE REVIEW	13
2.1 Introduction	13
2.2 Ipv4 And Ipv6 Overview	14
2.2.1 Importance of IPv6	17
2.2.2 Comparison between IPv4 and IPv6.....	19
2.3 IPv6 Security Issues	21
2.4 Neighbour Discovery Protocol Messages	24
2.5 NDP Security Vulnerabilities	27
2.6 NDP Common Attack	28
2.6.1 Neighbour Discovery Protocol Denial of Service Attack.....	28
2.6.1.1 Duplicate Address Detection.....	28
2.6.1.2 Smurf Attack	29
2.6.1.3 DDoS Attack	29
2.6.1.4 Replay Attack.....	31
2.7 Intrusion Detection and Prevention System	31
2.8 Classification of IDSs	33
2.8.1 Host and Network Based IDS	33
2.8.1.1 Packet-based NIDSs and or flow (stream)-based NIDSs.....	34
2.8.2 Artificial Intelligence IDS.....	36
2.8.3 Stateful Protocol Anomaly Detection	38
2.9 Summary	38
CHAPTER 3	39
3.1 Introduction	39
3.2 Machine Learning Types	42
3.2.1 Supervised Learning	42
3.2.2 Unsupervised Learning	44
3.3 Machine Learning Technique	45
3.3.1 Decision Tree	46
3.3.2 Naïve Bayes	47
3.3.3 Bayesian Networks (BNs).....	48
3.4 Local Weighted Learning Technique	49

3.4.1	Memory-Based Locally Weighted Learning	51
3.4.2	Incremental Locally Weighted Learning	51
3.5	Machine Learning in Security.....	51
3.6	Comparative Analysis of Related Existing Machine Learning Methods in IDS53	
3.7	Summary	57
CHAPTER 4.....	59
4.1	Introduction	59
4.2	The Proposed model.....	59
4.2.1	Proposed Framework	61
4.2.2	Data Collection and Pre-processing stage.....	62
4.2.2.1	Data Collection algorithms:	64
4.2.3	Flow Construction and Features extraction	66
4.2.4	ICMPv6Type Analysis.....	74
4.3	Summary	Error! Bookmark not defined.
CHAPTER 5.....	77
5.1	Introduction	77
5.2	Locally Weighted Learning Model Results and Discussion	77
5.3	Performance Evaluation Metrics.....	78
5.3.1	Confusion matrix	79
5.3.2	Sensitivity analysis (Replayed and DDoS Attack)	80
5.4	Results and Discussion of Base Classifier Models on the Original Dataset	80
5.5	Feature Selection	82
5.6	Results and Discussion of Locally Weighted Learning Models on Reduced Dataset.....	84
5.6.1	Dataset Description.....	85
5.6.2	Model Development using Cross-Validation Technique.....	85
5.6.3	Evaluation of Locally Weighted Learning Models.....	87
5.6.3.1	Evaluation of LWL-BN's Model	87
5.6.3.2	Evaluation of LWL-DT's Model	89
5.6.3.3	Evaluation of LWL-NB's Model	91
5.6.4	Summative Evaluation of the developed Locally Weighted Models on the Reduced Dataset.	93
5.7	Evaluation by Comparison with Current Research Studies	96
5.8	Summary	98
CHAPTER 6.....	99
6.1	Conclusion.....	99
6.1.1	Comparison with other researches.....	103
6.1.2	Research Methodology.....	103
6.1.3	The Novelty.....	104
6.2	Contributions.....	102
6.3	Limitation and Future works.....	103
References.....	105

List of Tables

Table 2.1: Important Features of IPv6

Table 2.2: Table of Comparison between IPv4 and IPv6

Table 2.3: NDP messages, Function and ICMPv6 number

Table 2.4: comparison between Flow-based NIDSs and Packet-based NIDSs

Table 3.1: Categorisation of Supervised Learning Techniques

Table 3.2: Categorization and sample of Unsupervised Learning technique

Table 4.1: Examples of attacks commands in the virtual network.

Table 4.2: The file in Database and the traffic (packets) label.

Table 4.3: Example of the streams after constructing them with features.

Table 4.4: Details of Original Features

Table 4.5: The accuracy result with different classifiers

Table 4.6: Information Gain Attribute Evaluator and Ranker Search Method Feature Ranking

Table 4.7: ICMPv6 type distribution with respect to class label.

Table 5.1: Confusion Matrix for this study

Table 5.2: Result of Base Classifiers on the Original Dataset.

Table 5.3: Details of Original Features

Table 5.4: Information Gain Attribute Evaluator and Ranker Search Method Feature Ranking

Table 5.5: Performance Evaluation of LWL-BN's model.

Table 5.6: Confusion Matrix of LWL-BN's Model.

Table 5.7: Sensitivity, True and False Positive values for LWL-BN

Table 5.8: Performance Evaluation of LWL-DT's model.

Table 5.9: Confusion Matrix of LWL-DT's Model

Table 5.10: Sensitivity, True and False Positive values for LWL-BN

Table 5.11: Performance Evaluation of LWL-NB's model.

Table 5.12: Confusion Matrix of LWL-NB's Model

Table 5.13: Sensitivity, True and False Positive values for LWL-NB

Table 5.14: Comparative analysis of each model's overall accuracies.

Table 5.15: Comparative accuracies per attack type.

List of Figures

- Figure 2.1: Availability of IPv6 connectivity around the world
- Figure 2.2: Google IPv6 access over eight years retrieved January 2019.
- Figure 2.3: IPv6 and IPv4 Header Format
- Figure 2.4: Neighbour Discovery Message Structure
- Figure 2.5: Architecture of DDoS Attacks.
- Figure 2.6: IDSs categorized based on their detection mechanism
- Figure 3.1: A typical decision tree structure for mammal classification
- Figure 3.2: Pictorial representation of LWL method.
- Figure 4.1: Flowchart of the proposed model.
- Figure 4.2: Dataset: Virtual Test bed Network Architecture.
- Figure 4.3: Flowchart of ICMPv6 packets filtration
- Figure 4.4: ICMPv6 Type Data Distribution.
- Figure 4.5: ICMPv6Type data distribution with respect to the class label.
- Figure 4.6 (a): ICMPv6 distribution (Higher Axis Unit)
- Figure 4.6 (b): ICMPv6 distribution (Lower Axis Unit)
- Figure 5.1: Normal and attacks label distribution.
- Figure 5.2: Research model.
- Figure 5.3: Comparative Accuracies of Models per Class Label.

ABSTRACT

The fast growth of the Internet usage has caused problem on Internet protocol address space. To solve this problem, Internet Protocol version 6 (IPv6) was created to expand the availability of address spaces. An important part of the IPv6 suites is the Neighbour Discovery Protocol (NDP), which is geared towards substitution of Address Resolution Protocol in router discovery, and function redirection in Internet Protocol version 4 (IPv4). NDP includes the routing function which is determines which route a data packet will follow to arrive at its intended destination, and the address function which assigns unique addresses to each and every device connected to a network for identification purposes. NDP messages are broadly categorized into five types and each message type carries out distinct tasks, these messages are: Router Solicitation (RS), Neighbour Solicitation (NS), Router Advertisement (RA), Neighbour Advertisement (NA), and Redirect. NDP security vulnerabilities is openness of a network of computer, wherein there exists a lack of trust among users. Typically, huge numbers of NDP messages can be used to flood a network, resulting in disconnecting of the connected devices. Due to the limitations of existing defence mechanisms, NDP are still prone to network-based attacks and these vulnerabilities must be considered while creating an IPv6 network.

In this thesis, we present a novel detection method for DDoS and Replayed attacks that are launched using NDP in IPv6. This detection method is a stream-based network representation, instead of packet-based representation. The proposed detection method makes use of Locally Weighted Learning machine learning techniques, with three different algorithms as its based learner Bayesian network, Decision tree and Naïve Bayes. LWL-Bayesian Network model achieved the highest detection rate of 96.48%. LWL-Naïve Bayes model is the next best model, with an accuracy rate of 96.024%, while the LWL-Decision Tree model had the lowest overall detection rate of 93%. Comparatively, all developed IDSs are capable of detecting DDoS and Replayed attacks based on NDP-based network traffic as well as the detection of anomalies. They all demonstrated strong predictive ability, however, the LWL-Bayesian Network model proved to have the best overall performance to develop a locally weighted IDS model among the three models. In short, the development of the detection models, has strong predictive capabilities, with high accuracy rate, does not overfit, has low computational costs, and uses less time for model development and attack detection.

CHAPTER 1

INTRODUCTION

A global usage of the IPv4 was recorded and as the number of users increases, the availability of IPv4 addresses nears extinction and thus become the main limitation of the IP version. Other limitations of IPv4 as reported by (Goralski, 2009) include the occurrence of fragmentation, inability of make use of IPv4 Type of Service as well as Time to Live fields as intended and also the IPv4 Options field is both limited in scope and seldom used. Hence, new method for data transmission in context of computer network were developed as well as new extension or alternatives were also proposed and developed among which Internet Protocol version 6 (“IPv6”) stands out. The advent of the IPv6 (Robert, 2017) importantly crushed the limited internet addresses availability of its predecessor – IPv4, by having the capability to uniquely provide for each person on earth and also by providing a bigger bit size for both source and destination addresses – 128bits (Budhathoki, n.d.). Also, in the case of redundancy of fields in a typical IPv4, IPv6 Options field among others is being improved and located between IPv6 header and the transport-layer header in a packet. Importantly for both IP version protocols is the security of the network they are used to facilitate. In this regard, it is essential that connected devices are well protected, secured and available for legitimate users – this is the concept of cyber security.

According to (Elijah, Abdullah, JhanJhi, Supramaniam, & O, 2019), cyber security covers the overall protection of the devices and network infrastructures for propagating digital communications as well as the data and information against cyber–attacks, potential leakage, worms, and or information theft. As computer network usage through numerous application have evolved in recent times, so equally are its security threats (David & Thomas, 2019). (Thiyagarajan P., 2019) described the amalgamation of both the digital and physical worlds as well as the non-displacement of physical crime but the occurrence of crime in both worlds. For example, criminals with records of crippling critical infrastructure such as water systems or power grids may remotely attack the digital system that now controls such infrastructure in the modern digital world. Also, previous bank robbers in the physical world may also conduct an attack on financial organizations for financial data and fund thefts. A major challenge faced by Cyber security is securing data and information i.e. providing integrity, availability and confidentiality (Mabayoje, Balogun, Ameen, & Adeyemo, 2016). As information technology

such as personal digital assistants (PDAs) and personal computer grows, every form of computer network is at risk of being attack by hackers using different types of schemes and or weapon such as worms, Trojans, rootkits, botnet attacks, social engineering platforms, spam, adware, virus and any other malicious means. Cyber security takes to at least abate or at best eliminate cyber threats (i.e. any malicious activity in the cyberspace) either horizontally (considering threat from the look of the attacker) or vertically (considering and evaluating threat from the outlook of the victims) (Dua & Du, 2016).

As the total number of security breaches are rapidly increasing in the world today, more hacking tools are available even for non-technical attackers in order to break a network security. Although cyber security does not only cover data security but the overall protection of whole functional information system, cyber security secures against potential and even intentional malicious threat and thus elaborately cover four main areas according to (Thiyagarajan P., 2019), namely: (1) Application security, (2) Information security (3) Disaster recovery and (4) network security. In this research work, the fourth category i.e. network security, is being considered. Even with the advent and initial adoption of IPv6 in our modern society, the security of both IPv4 and IPv6 networks is under perpetual threat of various types. IPv4 networks have been under siege of series threats as mentioned earlier and most of these IPv4 threats are now being extended into IPv6 while the newly developed and adopted IPv6 networks are also being exploited for new vulnerabilities by attackers. A well-known typical network security system is the Intrusion Detection Systems (IDSs) which mainly monitors networks' IP traffic and analyse incidents that may or may not threaten or violate the computer and network security policies that are put in place. An IDS serves by not just detecting attacks, intrusion or bugs (internally or externally) but it also makes prompt reporting to the network administrator for prompt response (Elijah et al., 2019). More so, other network security system exists in form of anti-virus, fire-wall, adware detection, cryptography etc. but they are becoming less effective, especially the firewall, due to the demand of users flexibility and the ability of hackers to bypass it (David & Thomas, 2019).

Thus, it becomes essential to also developed IDSs for IPv6 network as the effectiveness and efficiency of IDSs developed for IPv4 network is obviously recorded both in academia and in the industry. The development of systems heavily relies on the implementation and usage of machine learning algorithms to develop IDS models as seen through vast and major research works carried out. The application of machine learning technique in order to create IDS systems took the lead from the likes of expert system and other hand-coding mediums as the

implementation of machine learning techniques for detecting intrusion is able to conduct an advance analyses which uncover hidden insights inherent in datasets extracted from a real-time or simulate network traffic. Clearly, the focus of this research work is in the development of IDSs for a typical IPv6 network. However, there are various research work and implementation of IPv6 IDSs making it necessary to not just develop an IDSs but one with strong predictive power capable of detecting known extended IPv4 threats in IPv6 as well as an IDSs that can detect the newly exploited vulnerabilities of a typical IPv6 network. These facts paved the way for this critical research of new techniques, tools, methods and even algorithms that will be useful in developing the kind of IPv6 IDSs that swiftly detect popular, new and dangerous cyber threats as pertained to network security in the modern cyberspace.

1.1 Research Problem

IPv6 is vulnerable to Replayed and Distributed Denial of Service (DDoS) attacks that expose the security and availability of the Neighbour Discovery Protocol (NDP) messages. Before proposing new IDS to detect such attacks, several issues have to be addressed which represent the problem statement of this research. The problem statements of this research are as follows.

- High error rate and low accuracy have been achieved by the existing detection systems due to their dependency on the unsuitable representation of traffic (packets representation).
- The possible existing NDP DDoS and Replayed attacks have not been completely covered by any of the existing detection systems. Therefore, if these IDSs are able to detect a number of these attacks, they fail to detect others.
- The existing detection systems were built using a set of non-qualified features such as the capturing time, which leads to packets misclassification. This is because the detection model building process involves the classifier's consideration of the attack packets' time intervals as a feature that can be used to indicate an attack, which means the attacks falling outside the determined intervals are considered as secure (Elejla *et al.*, 2016).
- The absence of benchmark datasets in IPv6 that include all the possible NDP-DDoS and Replayed attacks to be used in evaluating any proposed system.

The study has been created a new dataset of Replayed and DDoS attacks in IPv6, this data will be created through a process of various attack simulations within a controlled network topology due to Ethical regulation concern of the Communication Act 2000, the Data Protection Act

1998 which disallows any live attack simulation or penetration testing on public networks. However, the dataset will be a standard benchmark data of IPv6 that offers the possibility for further and future security analysis and research.

1.2 Research Novelty

1. A new suitable representation of the NDP traffic based on streams format. This representation helps to detect the attacks by combining packets that shared characteristics in one stream.
2. A set of novel features that are able to represent NDP traffic based on the stream representation to detect Replayed and NDP-DDoS attacks. These features are used to represent NDP traffic for any classification technique. In addition, feature selection scheme able to choose the best-related based on ranking algorithm.
3. Create a dataset of simulated Replayed and NDP-DDoS attacks to be used in evaluating by other researchers to evaluate their security approaches for such attacks detection.
4. A detection model able to detect the targeted Replayed and DDoS attacks with high detection accuracy and low error rate built by applying machine learning algorithms.

1.3 Research aim

This research aims to detect Replayed and DDoS attacks of the NDP protocol due to the severity of the attacks and the importance of NDP protocol by using Machine-learning techniques. As the traditional IDS it has the ability to detect the known attacks but it will fail to detect the new attacks, so using the machine learning IDS is a strong method to solve this issue with high accuracy rate.

In addition, the development of an intrusion detection model that has strong predictive capabilities, does not overfit, has low computational costs, and uses less time for model development and attack detection.

1.4 Research objectives

1. In order to generate an NDP features dataset, a typical NDP dataset will be produced and used to guide the suggested model. An informative NDP features set will be defined in order to generate accurate characterisation of normal and malicious protocol behaviour. Furthermore, a benchmark standard will be identified to encourage further study through the collation of a dataset.

2. A model will be presented that focuses upon NDP anomalies, their classification and detection, with an emphasis on improvement in accuracy.
5. Utilise a number of supervised machine learning techniques in order to make comparisons of the strength of selected algorithms.
6. The proposed model is adapted and improved in order to make reductions in the detection rate of false negatives (FN) and increases detection rate of true positives (TP).
7. The efficiency of the suggested solution in securing NDP will be measured by making comparisons to the standard NDP detection model. The design of the suggested model will be verified and validated to improve accuracy, sensitivity and specificity of detection.

The proposed model consists of five phases which aim to detect Replayed and NDP-DDoS attacks. The first phase starts with capturing the traffic from the network. Moreover, it filters the non-IPv6 traffic, which is out of the research scope. The second phase extracts the required attributes to build the streams and extract the features from the filtered packets. The third phase builds the streams based on the $S6_{NDP}$ definition as well as extracting the features for each stream. The second phase deals with the packets, the third phase deals with streams. On phase 2, the packets are prepared to be ready to build the streams. phase 3, keeps the attributes that are needed for building the streams such as the IP, time and NDP type. The fourth phase aims to evaluate the selected features and excludes any unrelated features using the ranking algorithm. The last phase is to apply a machine learning classifier to the built streams' datasets with the selected features in order to build a detection model.

1.5 Thesis organisation

Chapter 2 Literature Review, the review will begin by introducing IPv6 and detailing its main functions. A detailed analysis will follow that outlines the main weaknesses and threats to IPv6, the ways in which IPv6 security and development have been driven and the techniques used to identify IPv6-NDP attack.

Chapter 3 will present a vivid introduction to machine learning, and its types (supervised and unsupervised) while revealing the categories for each type of machine learning. More so, the Locally Weighted Learning method was discussed as well as its types (i.e. the memory based and the incremental), the application of machine learning in developing Intrusion Detection Systems was reviewed.

Chapter 4 provide the design of the proposed model. An appropriate NDP feature set is defined. The NDP feature set is then used to accurately detect NDP attacks. Furthermore, the creation of NDP datasets and the tools needed for the creation and processing. In addition, apply the classification techniques on the dataset and get the result of the detection model.

Chapter 5 focuses on the efficacy of the suggested model, with regards to the accuracy of detection. In addition, a validation of the suggested model is undertaken by making accuracy and usefulness comparisons with other existing models.

In Chapter 6: A summary of the research which details the potential contributions of the model, its limitations and ideas for future research are presented in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter presents a vivid introduction into IPv6 and its. Also, a comparison between IPv4 and IPv6. More so, the importance and security issues in IPv6. The main protocol in IPv6 and its vulnerabilities and the common attacks. In addition, intrusion detection system and its types. Lastly, the stateful protocol anomaly detection.

Internet Protocol (IP) addresses are a vital component for the proper functioning of any given computer network. They facilitate the identification of devices (server, mobile device, computer, router etc.) connected to a network (Levin & Schmidt, 2014). Mainly, there are two functions primary to IP addresses, vis-à-vis, the routing function that determines which route a data packet will follow to arrive at its intended destination, and the address function which assigns unique addresses to each and every device connected to a network for identification purposes.

Typically, a communication system for linking various computers or hosts together is referred to as a Computer Network. This type of network is a made up of computers connected by communication lines through which they interact among themselves, either directly or through a computer within the network. However, a computer network does not only consist of connected computers, but also switches, routers, and servers that facilitate communication among the computers. A computer network as described by (Budhathoki, 2011), is useful for exchanging information among computers, acting as a communication tool for voice and video, implementing various applications and technologies such as distributed systems and databases etc., enabling optimal management and sharing of resources, and more importantly, it enables collaboration, communication, and engagement among people scattered across various societal institutions such as educational institutions, public and private organizations, national and international governmental bodies and agencies (Budhathoki, 2011).

A computer network can either be internet or intranet or extranet, Internet being a public network of networks, having no central administration, formed by interconnection of very large numbers of connected computer networks, making available shared resources, facilitating

collaboration among members of educational and research communities across the globe, and providing important applications such as Internet Relay Chat, Electronic Mail, World Wide Web (WWW), and File Transfer Protocol (FTP) among others. On the other hand, Intranet refers to any privately owned connection of computers that make use of web browsers (a front end for accessing local information from the local servers) and web servers (local resources for storage of company's data that exist in various forms), which are also Internet technologies, for information sharing and collaboration within an organisation. Intranet provides a variety of services within an organization while it is safely guarded from the public reach, which includes provisioning of product information to staff in sales and marketing departments, tutorials and technical support in various departments, among many functions. Lastly, Extranet refers to an Intranet-based network, extended to external users of another organization through the usage of secured access. User IDs and password, and other application level security are implemented on this type of network to ensure that the organisations' intranets are well secured (Budhathoki, 2011).

2.2 Ipv4 and Ipv6 Overview

Dating back to the existence of computer networks, there are two basic versions of IP address namely IP version 4 (IPv4) and IP version 6 (IPv6). The former, IPv4, was first released in 1978 and in 1981. This version of IP is widely deployed, providing an address field of 32 bits having over 4.3 billion unique internet addresses with approximately 3.7 billion useable host addresses (Shiranzaei and Khan, 2015).

Typically, an IPv4 packet header usually has the minimum length of 20 bytes and a maximum of 60 bytes (Hinden, 2017). IPv4 packets are made up of thirteen (13) fields including IPv4 source and IPv4 destination address which are 4-byte (32-bit). Discussing five of those fields, (i) Version: which indicated the type of the current IP address (typical contains 0x04 indicating version 4). (ii) Header Length: serves as the Internet header length (IHL) in 4byte (32bit) units referred to as "words", which comprises of padding and all other option fields required for aligning the header on a 32-bit boundary. (iii) Type of Service (TOS): include all parameters affecting how routers and other equipment are handling packets. It is widely known as Differentiated Services DS code point. (iv) Identification: provides the destination host the information for reassembling like-numbered fragments. It is a 16bit number set for each packet. This field is one among the many other fields peculiar for fragmentation. Lastly, (v) Protocol:

8 bits field containing the number that indicates which transport-layer protocol receives and processes the data content in the packet. Such numbers are 6 for TCP, 17 for UDP, and many others (Hinden, 2017).

This rather large pool of IPv4 addresses faced an exhaustion of space, with IANA exhausting its IPv4 free pool in February 2011, along with RIRs exhaustion of their IPv4 addresses that were previously unallocated, and the rapid expansion of Internet Service Providers (ISPs) and business networks, all leading to the quest to create a successor and also a transition phase (Chuangchunsong et al., 2014). The succeeding IP address version is known as IP version 6 (IPv6), developed in 1991 but was integrated only in 1997, added to Internet Corporation for Assigned Names and Numbers (ICANN) Domain Name Server (DNS) in 2004. IPv6 provides a unique 2¹²⁸ internet addresses formatted as eight 16-bit hexadecimal numbers fields, separated by the symbol “:” (Shiranzaei and Khan, 2015).

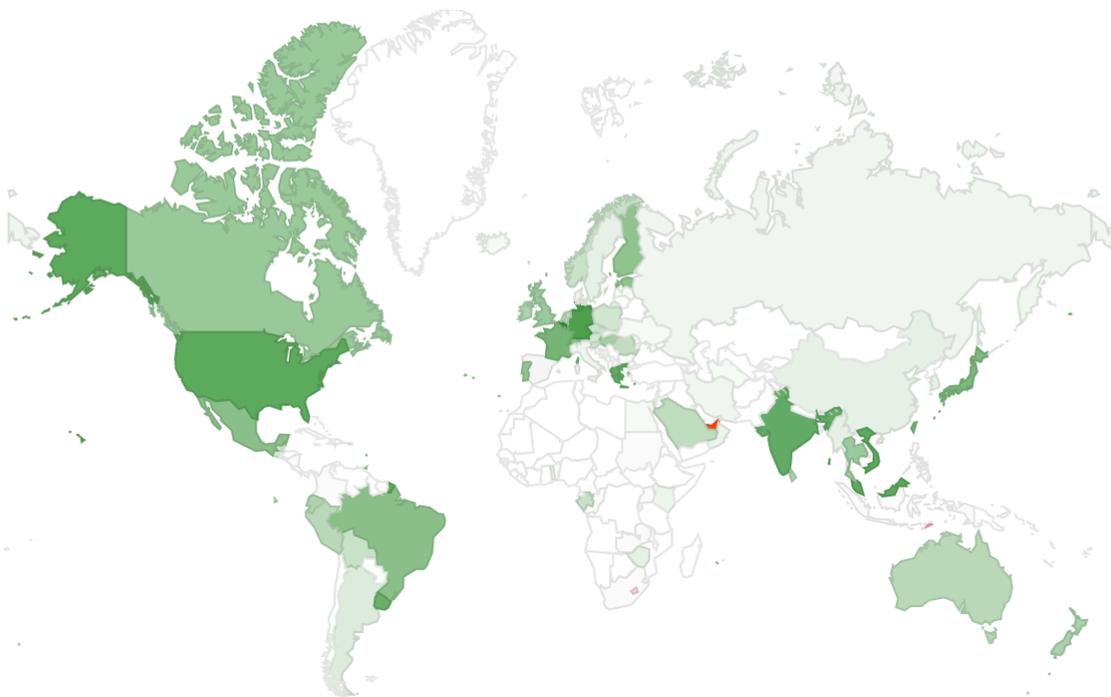
Many IPv4/IPv6 transition/migration mechanisms and tools have been witnessed enabling the connection between the fading IPv4 network and the succeeding IPv6 network, DS-lite, 4rd, and 4over6 networks. These transition mechanisms will continue until the complete change from IPv4 to IPv6 is executed (Chuangchunsong et al., 2014). Developing a transitional mechanism to IPv6 proved to be a long-standing solution against many measures such as the efficient distribution and or use of the existing IPv4 addresses, which are but temporary solutions (Levin and Schmidt, 2014).

More so, the depletion of IPv4 was dragged down following some strategies such as Network Address Translation (NAT), and Classless Inter-domain Routing (CIDR). NAT enables the use of a single IP address by multiple devices through the manipulation of the port number portion of the packet header for mapping incoming traffic to each device in the network, while CIDR makes use of the earlier unused and unavailable blocks of addresses by providing a variable-sized network prefix which allows the number of addresses included in allocated address blocks to be more flexible (Frankel and Green, 2008). One of the developed transitional mechanisms is tunnelling, which facilitates IPv6 connectivity over an IPv4 backbone network. More so, the same technique was further advanced to enable IPv4 connection on an IPv6 backbone network,

providing IPv6 hosts the ability to connect to an IPv4 destination (Chuangchunsong et al., 2014).

Considering the scale of adaptation of IPv6 and availability scale as shown in Figure 2.1, it is reasonable to conclude that the transition process from IPv4 to IPv6 will spark the interests of cybercriminals: the protocol still being in its early stages. A variety of security tests need to be conducted in an effort to reveal potential security threats; the more the tests conducted the greater the chance of bridging the security gap and guarding against future attacks. Furthermore, Figure 2.2 shows the latest Google IPv6 accesses over the world in the last eight years.

Per-Country IPv6 adoption



[World](#) | [Africa](#) | [Asia](#) | [Europe](#) | [Oceania](#) | [North America](#) | [Central America](#) | [Caribbean](#) | [South America](#)

The chart above shows the availability of IPv6 connectivity around the world.

-  Regions where IPv6 is more widely deployed (the darker the green, the greater the deployment) and users experience infrequent issues connecting to IPv6-enabled websites.
-  Regions where IPv6 is more widely deployed but users still experience significant reliability or latency issues connecting to IPv6-enabled websites.
-  Regions where IPv6 is not widely deployed and users experience significant reliability or latency issues connecting to IPv6-enabled websites.

Figure 2.1: Availability of IPv6 connectivity around the world (Google, 2019)

IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.

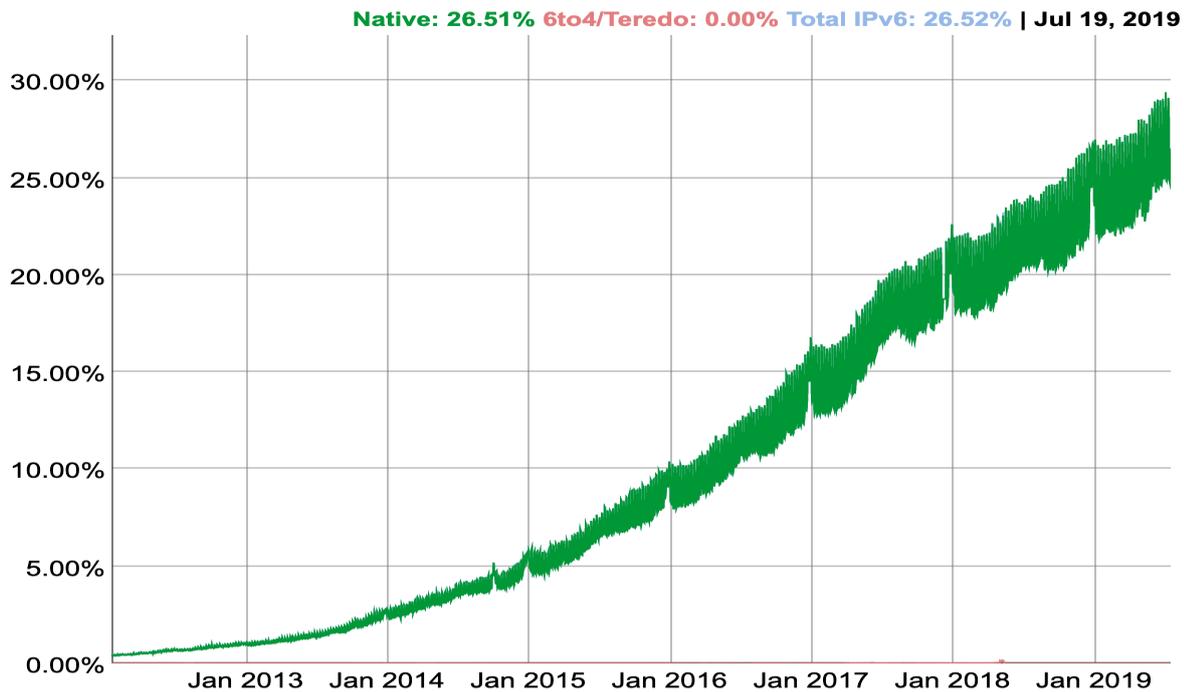


Figure 2.2: Google IPv6 access over eight years retrieved January 2019.

2.2.1 Importance of IPv6

IPv6 is a carefully developed technology, viable, providing a long-lasting solution for future years. Its development was born out of the need for a successor and thus prove a better technology than (IPv4) (Frankel and Green, 2008). IPv6 is a new general use version of Internet Protocol for global purposes. It was built in 1998 (Moravejosharieh et al., 2012). It is designed to overcome the main limitations of IPv4 including the lack of security and the exhaustion of IP address space (Electronic Design, 2012).

IPv6 makes use of a simpler header when compared with IPv4 and also a more efficient extension header mechanism, thus providing efficiency in routing. It also eliminates the threat of potential broadcast storms by having no broadcast medium and it requires no processing for checksums and provides flow labels for per-flow processing without opening the transport inner packet for identification of traffic flows (Budhathoki, 2011). More so, IPv6 ensures security and mobility by complying with IPSecs and mobile IP standards functionalities. The mobility standards are automatically provided in IPv6 unlike IPv4 and IPsec is a mandatory in all IPv6 networks – enabled on every node to ensure security (Budhathoki, 2011).

The Table 2.1 depicts, concisely, the importance of IPv6, highlighting its features.

Table 2.1: Important Features of IPv6 (Headquarters, 2011; Rafiee, Löwis, & Meinel, 2012)

S/N	Features	Highlights
1.	Bigger Address Space	Aggregation
		Renumbering
		Global flexibility and reachability
		Multihoming
2.	Transition Richness	Dual stack
		6to4 tunnels
		Translations
3.	Simple Header	Flow labels
		Efficient routing
		Extension headers
		Elimination of broadcasts
		Performance and forwarding rate scalability
4.	Security and Mobility	IPsec mandatory (or native)
		Mobile IP RFC-compliant
5.	Quality of Service	Resolve trunk bandwidth allocation
		Improved QoS management of video and audio app systems
6.	Auto configuration	Plug and Play mechanism
		End-to-end without NAT

IPv6 was developed to provide solutions to existing IPv4 problems and to solve foreseeable problems of IPv6 itself in the future, thereby making it a technology that will stick with humans for a very long period of time.

2.2.2 Comparison between IPv4 and IPv6

IPv4 and IPv6 are uniquely distinct by possessing different values for common features, which are tabulated in Figure 2.3 and Table 2.2. These feature values are the peculiar characteristic of each IP version and they give more insight about each version (Shiranzaei and Khan, 2015)(Li, Jimmel and Shima, 2010).

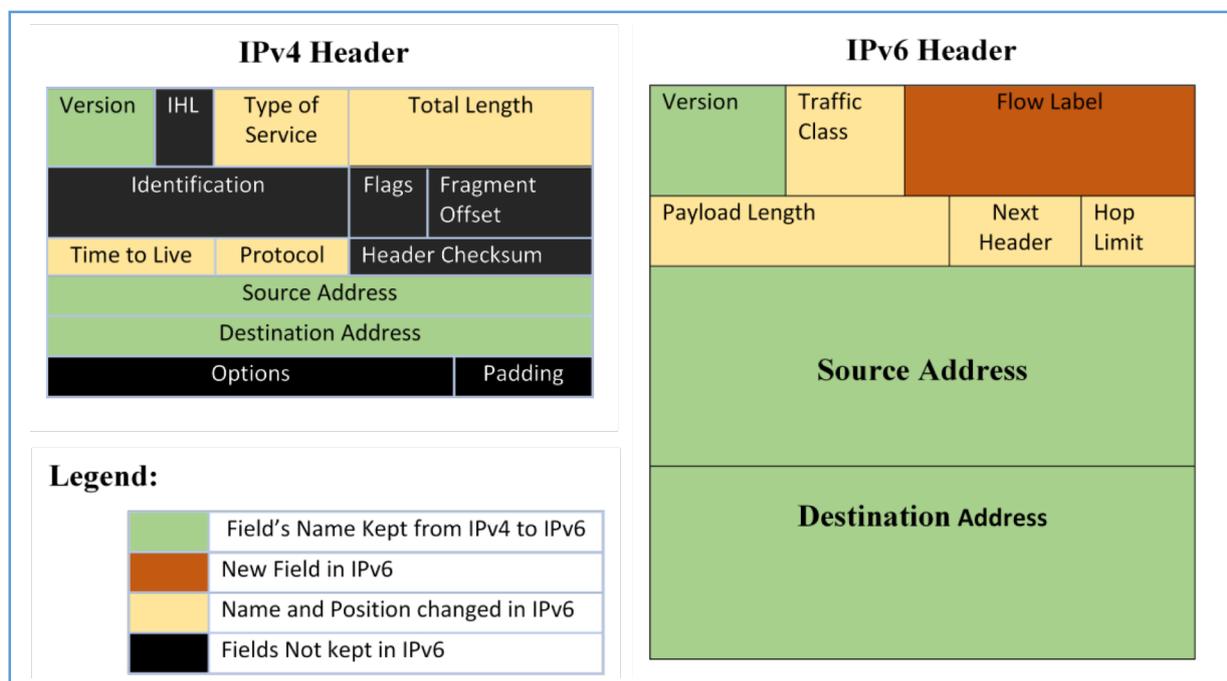


Figure 2.3: IPv6 and IPv4 Header Format.

In Table 2.2 below, a tabular comparison between both IP versions is being presented. Summarily, aside the obvious difference between both IP versions as being displayed in the Figure 2.3 above, both IPv4 and IPv6 possesses distinct features in the context of total number of unique address available, the addresses style and type, configuration, fragmentation, file transfer protocol (FTP), size of source and destination addresses, maximum packet size of the transmission unit, and the router discovery mechanism among others. More so, the applicability of network address translation, simple network management protocol, and IPSec differs for both IP versions.

Table 2.2: Table of Comparison between IPv4 and IPv6

SN	Category	IPv6	IPv4
1.	Year of Deployment	1998	1981
2	Address Length	16bytes (128bits)	4bytes (32-bits)
3.	Total Number of Unique Addresses	340,282,366,920,938,463,463,374,607,431,768,211,456	4,294,967,296
4.	Addresses Type	Anycast: one out of the predefined interface receives packets among others. Multicast: some set of interfaces receive packets.	Multicast: only specify interfaces receives packet. Broadcast: all interfaces receive packet. Unicast: only one interface receives packet
5.	Address Style	A set of eight (8) hexadecimal digit separated by colons.	Represented by four sets of decimal number, usually separated by dot .
6.	Configuration	“Plug & Play” autoconfiguration feature, existing as either stateless or stateful	The configuration of this IP address is usually carried out manually or automatically too using Dynamic Host Configuration Protocol (DHCP).
7.	File Transfer Protocol (FTP)	Does not support this protocol	Use this protocol for sending and receiving information within a network
8.	Fragmentation	This can be done only by the sender	Both sender’s and forwarding routers have the ability to fragment big packet
9.	Maximum Transmission Unit (MTU)	A packet can be supported maximally with this link for up to 1280byte.	The maximum packet size supported by a specific link is 576byte.
10.	Router Discovery	It uses the ICMPv6 Router Solicitation and Router Advertisement message for this purpose	It uses ICMP Router Discovery for defining default gateway router for reaching devices across various network
11.	Renumbering	It automatically renumbers IP addresses of networks when merged or extended.	Manual renumbering of two or more IP addresses when two or more networks are merged or extended
12.	Security	End-to-end backing for user authentication, data integrity and data encryption.	Make use of tunnelling between two networks

13.	IPSec Support	Uses some cryptographic protocols for securing key exchange and communication.	It is optional
14.	Mobility	Faster routing, handover and hierarchical mobility are executed using MIPv6	Uses Mobile IPv4. Mobile node address is re-established again upon change in location
15.	Header	It has 8 header fields.	It has 14 header fields.

2.3 IPv6 Security Issues

Currently, commerce and government deeply depend on the Internet, with the Internet being one of the most complex and largest human-engineered distributed systems widely deployed, it is safe to say that a threat to the Internet is a threat to human society. Since the establishment and deployment of the Internet, it has recorded various massive attacks targeting its fundamental protocols, process and infrastructure as well as various endpoint devices connected therein (Nazario and Kristoff, 2012). The Internet is made up of several but interdependent subsystems connecting endpoint devices. Usually, when users (hosts or organisations) or endpoint devices are threatened causing unavailability or integrity loss at some point, subsystems are utilised to contain and resolve the threat – this scope had mostly concerned the Internet security community and many mechanisms had been developed to curb or handle such situation. But threats to the system itself or trusted subsystems (equipment for moving data around and the protocols) are much more devastating with cascading results and this has recently been receiving attention from the community, which now engages in examining the vulnerabilities.

Network security refers to the protection of hardware and software data within a network system from malicious activities, threat, accidental damage and or intrusion, thereby enabling the continuous smooth running of the network service without interruption keeping its consistency, authenticity, usability, integrity and availability (Min, 2011). Network security incidents can lead one or more of the following: service failures, divulgements, Internet paralysis, and data damage, have been recorded over time which results in huge social harm and drastic economic losses. This is due to the crucial flaw of the network – openness, as it tends to have grown over the years as the number of network users increases. The main universal network protocol – TCP/IP, is being used on several types of endpoint devices across

several different platforms accessing different types of media across the globe, enabling global a security threat regardless of platform and location restrictions.

With the advent of IPv6 becoming a reality, as its deployment is growing faster as a result of the exhaustion of IPv4 block addresses, is it important to elucidate on the security protocols being developed for IPv6. There are several security protocols for IPv6 responsible for different features namely IP Security (IPSec), Privacy Addresses, and Secure Neighbour Discovery Protocol and Cryptographically Generated Addresses (Frankel and Green, 2008). IPSec, designed by IETF, is a suite of protocols that provides data protection through the signage and encryption of the data before transmission is carried out across public networks (Frankel and Krishnan, 2011). Fundamentally, the concept of security entails the authentication, privacy, integrity and availability of data and services – the first three are being provided by IPsec (Cooper, Gont & Thaler, 2016). IPSec works at the network layer providing security through the usage of the extension header feature possessed by IPv6, as it provides two main services namely Authentication Header (AH) and Encapsulated Security Protocol (ESP) (Frankel and Green, 2008)(Min, 2011). Data-origin authentication, confidentiality and integrity protection, access control, limited traffic-analysis protection and replay protection are provided by ESP while AH specially provides replay protection and data-origin authentication (Frankel and Green, 2008). IPSec also works with the TCP/UDP transport layer as well as the Internet layer, offering transparency to applications and users by securing most TCP/IP protocol suites. It offers protection against Denial-of-service attack, data pilfering, theft of user credentials, and data corruption within a private network (Security et al., 2018).

Privacy addresses is fundamental in defeating address tracking within an IPv6 network, as the generation of cryptographic hash for device static interface identifier and other values which are both used by the Duplicate Address Detection (DAD) protocol of IPv6 makes sure that the privacy address is unique within the local network (Frankel and Green, 2008). In other words, when transfers between devices using IPv6 address do not make use of tunnelling, the source and destination address as well as the occurrence of the session can be tracked by a snooper (a middle man) and even the mobility of the devices from one network to another can be tracked too by an eavesdropper, thereby becoming a dire privacy threat. Privacy address is implemented as the solution to this menace, as it makes use of a constantly changing pseudorandom number representing an interface identifier for each host in a network thereby preventing snoopers from detecting or tracking a given device (Cooper, Gont & Thaler, 2016).

Neighbour Discovery Protocol (NDP), is a peculiar protocol of IPv6 that enables a device to access information about its local IPv6 network for the verification of its unique auto configured IP address (Frankel and Green, 2008). NDP is a core IPv6 protocol; replacing the address resolution protocol, redirect function, and router discovery functions found in IPv4. It is a stateless protocol used by IPv6 nodes when joining an IPv6 network, without using DHCP (Ahmed, Hassan and Othman, 2017). NDP is well known to be susceptible to attack as it possesses a weak authentication process (Ahmed, Hassan and Othman, 2017), but in a controlled network of trusted hosts, NDP is noticeably secure (Frankel and Green, 2008). A Secured NDP (SEcure Neighbour Discovery SEND) makes the provision of preventing attackers from subverting Neighbour Discovery (ND) communication, and or compromising IPv6 devices from communicating on a public IPv6 network, through the usage of Cryptographically Generated Addresses (CGAs) – a specially generated pair of public and private key pairs for verifying the ownership of the sender's address (Frankel and Green, 2008). SEND is an extension of NDP providing three additional features addresses which are ownership proof, message protection and router authorization. Creating a SEND packet requires the attachment of four new options vis-à-vis RSA Signature, CGA, Nonce, and Timestamp. More so, two new ICMPv6 messages are used during the process of router authorization, these two messages are Certificate Path Solicitation (CPS) and Certificate Path Advertisement (CPA) (Ahmed, Hassan and Othman, 2017).

The security of the network has proved to become more difficult as IPv4 and IPv6 co-exist in today's modern network, as the old security issues still linger on (such as rogue device, packet flooding, application-layer attacks) while new security issues related to IPv6 are becoming known (Malik and Dutta, 2018). Despite the fact that IPv6 possesses some security protocols, it still has security issues owing to the fact that basic Internet protocol standards are made public providing the same information to network security experts as well as attackers (Cooper, Gont & Thaler, 2016). IPsec, being the most vital reinforcement in the security of IPv6, requires the deployment of an independent Public Key Infrastructures (PKIs) which is still really not in existence thereby hindering the comprehensive application of IPsec (Min, 2011). Such type of key management framework includes ISAKMP and IKE, which are responsible for facilitating true end-to-end secure communication of an IPv6 network. Thus, the usage of IPsec is currently limited to the establishment of Virtual Private Network (VPN) (Frankel and Green, 2008) and the proposed security of IPv6 via IPsec is still a mirage.

Furthermore, IPsec cannot on its own prevent all possible type of attacks faced by a network, as it cannot defend a network against attacks peculiar to other layers of the network other than the network layer. These attacks are, but not limited to: DoS attack, application layer attack, Sniffer, and Flood attack. It is also known that security flaws are discovered during maintenance stages of any system, and are most realized between application and software system, thus requiring the adoptive usage of other network security solutions such as VPN, firewall, network filtration, anti-virus gateway, loophole scan etc. (Min, 2011). Poor management is another security issue faced by the IPv6 network. There are no mature network management software and no mature network management devices for IPv6, which in turn leads to the non-existence of methods for large scale supervision and management of the IPv6 network and no methods for large scale performance analysis and network failure positioning (Min, 2011). Though IPv6 was designed with security in mind, it became expedient that the understanding of the vulnerabilities in its network infrastructures must be fully acquired and properly managed by network administrators in order to establish a secured network (Malik and Dutta, 2018).

Resolving most of the security issues facing the IP network is ameliorated through the implementation of and strict compliance with security policy, making sure that insecure protocols and technologies are not integrated into the network configuration as much as possible i.e., the use of the most secure and best available technology at all times, and regular security updates must be adhered to – in the form of system patches at the discovery of new exploits (Cooper, Gont & Thaler, 2016).

2.4 Neighbour Discovery Protocol Messages

It is necessary to indicate that NDP is a primary protocol in the IPv6 suite consisting of IPv6 stateless address auto configuration (SLAAC) and Neighbour Discovery for IPv6. It handles functionalities such as the identification of duplicate addresses, determining nodes of same link, router discovery, link-layer addresses discovery and many others. Through NDP, IPv6 offers host initialization – the process wherein a host sends a router solicitation message to routers in the same link to obtain network parameter as well as other hosts to obtain link-layer and IP address data of its neighbour for the avoidance of duplicate address, and address auto configuration – which enable an IPv6 host to produce its own IP address without the need of a server (Ahmed, Hassan and Othman, 2017).

NDP messages are operated within the ICMPv6 message structure and thus formatted in a special manner by the network administrator.

The structure of NDP message is shown in Figure 2.4.

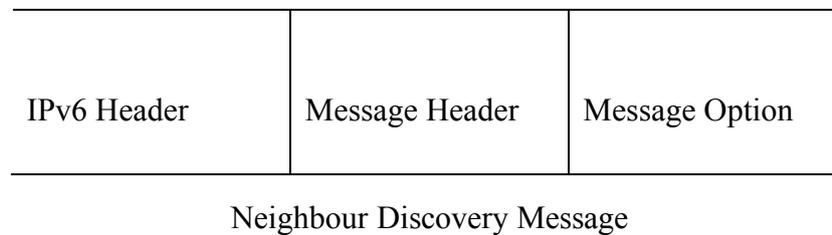


Figure 2.4: Neighbour Discovery Message Structure

NDP messages are broadly categorized into five types and each message type carries out distinct tasks, these messages are: Router Solicitation (RS), Neighbour Solicitation (NS), Router Advertisement (RA), Neighbour Advertisement (NA), and Redirect.

1. RS message allows nodes to explore IPv6 routers in a given subnet.
2. NS message is used for discovering the link-layer address of an on-link IPv6 node or for the confirmation of an already established link-layer address.
3. RA messages are sent by IPv6 routers pseudo-periodically, containing information (e.g. link MTU, specific routes, link prefixes, duration etc.) for hosts in response to RS messages.
4. NA messages are also a response to an NS message. which hold vital information for hosts.
5. Redirect messages are sent by a router only, usually as a unicast traffic, and processed by hosts only, they inform an originating host about the better first-hop address for a certain destination. Table 2.3 presents a tabular representation of NDP message type, its function, and each corresponding ICMPv6 number (Ahmed, Hassan and Othman, 2017).

Table 2.3: NDP messages, Function and ICMPv6 number

Message Name	Function	ICMPv6
Router Solicitation	Router Discovery	133
Neighbour Solicitation	Neighbour Discovery	135
Router Advertisement	Router Presence	134
Neighbour Advertisement	Neighbour Presence	136
Redirect	Better Next Hop	137

The exchange of messages by hosts and routers within an NDP are peculiar to certain functions as stated previously, Address Resolution, Redirect Function, Duplicate Address Detection (DAD), Router Discovery, Neighbour Unreachability Detection (NUD). The DAD process in an IPv6 network is carried out through sending NS messages by an IPv6 node, having set the NS message based “Target Address” field to IPv6 address for which duplication is being identified. If the sending node does not receive an NA message defending the use of the address it then initializes the address on the interface but once the node receives a multicast NA with the “Target Address” field, it immediately disables the usage of the IP address on its interface (Ahmed, Hassan and Othman, 2017).

NDP message contains some protocol options which perform some extract functions. First of the option protocols is “Source and Target Link-Layer Address Option” which signifies the link-layer address of the NDP message sender, and this option is available for all NDP messages except redirect and NA message. The second protocol option is the “Redirected Header Option”, which is always included in the Redirect message, used by router for specifying the IPv6 packet that sent a Redirect message. The third protocol option is the “Prefix Information Option”, contained in RA onward departure messages, used to specify information about address prefixes and address auto-configuration. The fourth protocol option is the “Route Information Option” mainly used for indicating individual routes affixed to a local routing table, and they are usually carried by RA messages. Lastly, there exist the “MTU Option” also

carried by RA messages and reportedly overriding IPv6 MTU, it indicates the IPv6 MTU of the link as used by network analysts when IPv6 proved not to be familiar with a link.

2.5 NDP Security Vulnerabilities

NDP security vulnerabilities are also related to the vulnerabilities suffered due to openness of a network of computer, wherein there exists a lack of trust among users. Typically, huge numbers of NDP messages can be used to flood a network, resulting in disconnecting and or freezing of the connected devices. Due to the limitations of existing defence mechanisms, NDP are still prone to network-based attacks and these vulnerabilities must be considered while creating an IPv6 network.

There are three types of vulnerabilities common to NDP, namely: Redirect, Denial-of-Service (DoS) and Flooding Denial-of-Service attack types (Ahmed, Hassan and Othman, 2017). Redirect attacks are executed by malicious nodes that are responsible for directing packets away, thereby rendering the packet untraceable from the last hop router and also direct other genuine receivers to alternative nodes. On the other hand, DoS seeks to disrupt the information flow within a network. DoS prevent proper flow of information between a victim and other hosts within a network or some specific addresses. NDP is vulnerable to Flooding Denial-of-Service. Flooding Denial-of-Service sends traffic from other hosts to the victim, this traffic is usually bogus, and renders the victim machine completely useless by exhausting its resources.

NDP facilitates autoconfiguration of node, i.e. the generation of IPv6 address automatically for each interface within a network, relieving the network administrator from the manual configuration of IP address which is a norm for an IPv4 network. This autoconfiguration can either be stateful or stateless autoconfiguration, NDP is used by a node to discover other nodes in the same link of a stateless autoconfiguration process. For an unsecured network, a malicious node can take advantage of NDP to easily misguide other nodes to follow a packet's instruction which may lead to a subversion attack and or generate and send a flood of ICMPv6 messages to a victim node or network segment thereby causing a decreased performance (Caicedo, Joshi and Tuladhar, 2009).

2.6 NDP Common Attack

NDP is one of the essential protocols of IPv6, operating in the link layer, and responsible for auto-configuration of a node's address, determining link-layer addresses of other nodes, detection of available routers and Domain Name System (DNS) servers among other responsibilities. Neighbour Discovery Protocols are susceptible to various forms of attack, as NDP defined ICMPv6 packets can be hijacked and carefully crafted fake responses can be sent from a malicious node resulting in malicious attacks (Anbar, Abdullah and Saad, 2016), three of which will be discussed in detail in subsection 2.6.1. NDP attacks or threats can be broadly categorised into three forms with respect to routing process. Some attacks are router-independent; some are unrelated to routing data while other attacks are carried out via remote manipulation. Some attacks are non-routing-based including "*Neighbour Solicitation/Advertisement Spoofing*", "*Duplicate Address Detection Dos Attack*" etc., other attacks categorised as routing-based threats include "*Bogus On-Link Prefix*", "*Parameter Spoofing*", "*Spoofed Redirect Message*" etc. (Ahmed, Hassan and Othman, 2017).

2.6.1 Neighbour Discovery Protocol Denial of Service Attack

This form of Denial of Service attack is quite different from the usual DoS, though the attack aim is to deprive legitimate users of an organization's network resources. In real time, DoS attack cost businesses their reputation through fostering negative publicity or entirely shutting-down the business. DoS attacks that are peculiar to NDP are:

2.6.1.1 Duplicate Address Detection

Usually, the generation of address by an interface is carried out by the node sending an RS message to all the routers within a link using a multicast address for the purpose of securing network prefix value, after which DAD is used by the node to check for the uniqueness of its newly generated address. Through DAD, the node sends an NS packet containing the generated IP address, to all existing nodes on the link with the purpose of getting feedback as to whether the newly generated IP address is being used, if the NS message got no reply, the node assumes the address is unique and makes use of it (Anbar, Abdullah and Saad, 2016).

Thus, a DoS attack is carried out on DAD by an attacker who falsely responds with an NA packet claiming that the newly generated IP address exists. This false response with an NA packet is repeated for every newly generated IP address until the new node gives up on initialization of its interface (Anbar, Abdullah and Saad, 2016).

2.6.1.2 Smurf Attack

A typical Smurf attack keeps its victim busy in responding to a deluge of incoming requests. This is also possible in an IPv6 network, a machine is first compromised, then the victim machine is used as the source for sending spoofed ICMP echo request packets to a multicast group. After receiving such request, other hosts respond to the request by sending packets capable of slowing down the target computer and in extreme cases, the target computer is made unavailable to the work (Anbar, Abdullah and Saad, 2016).

2.6.1.3 DDoS Attack

A typical infrastructure of IPv6 is the Domain Name System (DNS) which possesses some vulnerabilities which when exploited can be used to execute an attack such as amplification and reflection attack – a popular form of distributed denial of service (DDoS) attack (Zekri et al., 2017). DDoS attacks have been executed on network of computers and devices in recent times, such as the Oct 21st, 2016 attack on east coast USA and also the one recorded as the largest and most sophisticated attack which had American banks, including Wells Fargo, JP Morgan etc., as its victims (He, Zhang and Lee, 2017). DDoS attacks often have different intentions and launch methods that could be used to distinguish the attack and also how to prevent or detect it.

DDoS attacks are subtle as they exhibit no obvious features which can be used to identify such packets as malicious. The tools used in carrying out this form of attack are very easily obtainable, (in fact a compromised machine used to send packets to a victim might not be aware of it being compromised) which makes the attack quite frequent. Maintaining a simple structure of many-to-one feature, DDoS attacks are complex and they do have tremendous impact. Also, making use of IP spoofing, attackers remain hidden while executing the attacks.

The intention of DDoS attack is to displace legitimate users from accessing specific network resources. It is triggered either by sending malformed packets to the victim or the attacker tries to exhaust the network resources (bandwidth, router processing capacity etc.), which is known as network-level flooding attacks, thereby disrupting legitimate users' connectivity. The attack can be an application-level flooding attack which targets and exhausts the server resources (e.g. memory, CPU, and database/disk bandwidth etc.) (Barbhuiya, Biswas and Nandi, 2011).

More so, the attacker's incentives play a role in distinguishing this attack. Various incentives include intellectual challenge (often carried out by younger hackers), cyber warfare (most politically rooted), economic gain, or ideological belief.

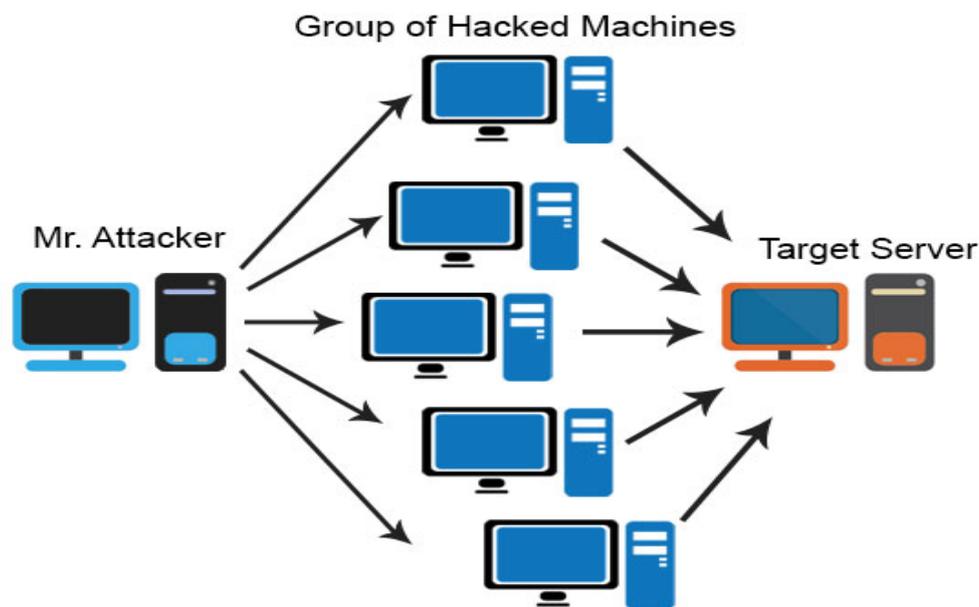


Figure 2.5: Architecture of DDoS Attacks (Educba, 2019)

In a DDoS attack as illustrated in Figure 2.5, the attacker creates master machines to control the slave machines to be used for executing the attack, once the compromised machine grows significantly; the attacker launches its attack on the victim by sending exhausting packets to network or server resources, using spoofed IP addresses to cause loss of service.

DDoS attacks are carried out on IPv6 networks, mostly by using the ICMPv6 in quite a number of methods, including excessive sending of ICMPv6 packets to a victim, sending error messages that cause a drop in active sessions of established communication, and invalidation of legitimate addresses or disabling of interfaces if an infiltration occurs onto a link's maintenance messages (Saad et al, 2013)

To perform a DDoS attack using ICMP, attackers make use of the ICMP_ECHO_REQUEST packet by sending it, using the broadcast IP address, to this victim's network. The machines on the victim's network reply to this request with the ICMP_ECHO-REPLY packets, and then the attacker may inundate the victim's network using an intermediary network, and consequently

saturates the victim's network bandwidth, which leads to the unavailability of the network to its legitimate users (Mahjabin et al, 2017).

Deploying a Network IDS (NIDS) at the default gateway of IPv6 network tends to curb DDoS attacks (Elejla et al, 2018) as it captures packets passing through the network, and analyses them for the purpose of detecting DDoS attacks. The method of representation of the network traffic determines the performance of NIDS. This network traffic can either be represented as flow-based or packet-based.

2.6.1.4 Replay Attack

This kind of attack is executed by replaying either the neighbour discovery message or the router discovery message using malicious node, with the aim of securing network access. The attacker captures an NDP message between legitimate users (and can even modify the content of the message) and resends the message (or the modified version) in order to gain access to a network – this type of attack is referred to as “Replay Attack” (Anbar, Abdullah and Saad, 2016).

Through the usage of a sniffer program, a third party can capture packets traveling over the network. The captured command or correct message can be used to gain access to a secure computer or execute commands that are normally unreadable and encrypted, thereby making the communication over the network unsecured. The captured command or message is not required to be deciphered before being used by the attacker.

2.7 Intrusion Detection and Prevention System

The intent of the Internet is for end devices to be intelligently integrated together and built up in a more complex fashion than the core network devices which are solely responsible for the connectivity between the end devices. However, the existence of firewalls, middleware and several other interfering technologies/devices led to the virtual disappearance of a true end-to-end communication. Though IPsec was designed for end-to-end security of communication, Intrusion Detection/Protection System (IDS/IPS) and firewalls are required for passage of encrypted traffic within a network without engaging security analysis or filtering (Frankel and Green, 2008).

Intrusion is any type of malicious activity that tries to deny the security aspects of a computer system. It is defined as any set of actions that attempts to compromise the integrity, confidentiality or availability of any resource (Mitchell and Vora, 2013). Intrusion detection is a process of gathering intrusion related information occurring in the process of monitoring the events and inspecting them for signs of malicious acts (Maharaj, 2014). Intrusion Detection ID is the process of monitoring and analysing the events occurring in a computer system in order to detect malicious activities taking place through the network. ID is an area growing in significance as more and more sensitive data are stored and processed in networked systems (Mitchell and Vora, 2013). Intrusion detection is the process of identifying and responding to malicious activity targeted at computing and networking sources (Hemant, Sarkhedi and Vaghamshi, 2013). As stated by (Maharaj, 2014), ‘the primary goal of intrusion detection is to model usual application behaviour, so that we can recognize attacks by their peculiar effects without raising too many false alarms’. More so, intrusion detection goal is to detect violations in the security of information system, though its approach to security is passive as it only monitors systems and raises alarms in scenarios where violation of security is detected (Reddy, Reddy and Rajulu, 2011).

Intrusion Detection System is a software program that identifies and reports malicious programs or packets or attacks or threats attempting to gain access to a network system while an Intrusion Prevention System detects intrusion and takes preventive measures (Vinitha, 2013). Both systems are used for providing defence-in-depth to network security framework as information security is under a deluge of real threats emanating from several network attacks (Hamed, Ernst and Kremer, 2018). Since network-based computer systems are vital to our modern society IDS/IPS provides three essential security functions (Hemant, Sarkhedi and Vaghamshi, 2013):

- 1) Data confidentiality: Information that is being transferred through the network should be accessible only to those that have been properly authorized.
- 2) Data integrity: Information messages should maintain their integrity from the moment they are transmitted to the moment they are received. No corruption or data loss is accepted either from the random events or malicious activity.
- 3) Data availability: The network or a system resource ensures that it is accessible and usable upon demand by an authorized system user.

Intrusion Detection system is a combination of hardware and software that detects intrusions in the network. IDS monitors all the events in the network by gathering and analysing information from various areas within the network. It identifies possible security breaches, which include attacks from within and outside the organization and hence can detect the signs of intrusions (Mitchell and Vora, 2013). On the other hand, Intrusion Prevention System can be referred to as the active form of IDS that responds to the malicious program or threat or attack as soon as it is being detected.

2.8 Classification of IDSs

IDS have become a standard component in security of infrastructures as they allow network administrators to detect policy violations (Reddy, Reddy and Rajulu, 2011). As an information security infrastructure, IDS exists in various forms and is often classified into different types based on location, scope of operation and method of detection.

2.8.1 Host and Network Based IDS

A host-based or network-based IDS, is a form of classifying IDS by the location. An NIDS is referred to as network-based, when it monitors network traffic and thus make use of the contents of the original packet of the network for analysing network protocols, transport and application in order to identify suspicious activity.

On the other hand, a host-based HIDS simply monitors a machine and audits the tracking data from the host's operating system. It runs on a particular machine and detects any malicious packet intruding the system.

For simplification, a host-based HIDS runs on the host machine and audits tracking data of the host in order to detect and report malicious attacks while a network-based NIDS runs on a computer network comprising of more than one host, monitors the whole network and scans through the network to detect and report malicious activities (Hemant, Sarkhedhi and Vaghamshi, 2013).

IDS can be Host-Based (HIDS) or Network-Based (NIDS) depending on its location for sourcing data (i.e. capturing packets). Furthermore, IDS can either be Signature-based (SIDS) or Anomaly-based (AIDS) depending on its detection mechanism (Elejla et al, 2018).

As shown in Figure 2.6. Anomaly-based IDS can further be sub-categorized as Rule-based IDS, and Artificial-Intelligence Based IDS and Stateful Protocol Analysis which enables IDS to detect and trace the protocol states providing an important knowledge for understanding and responding to an attack.

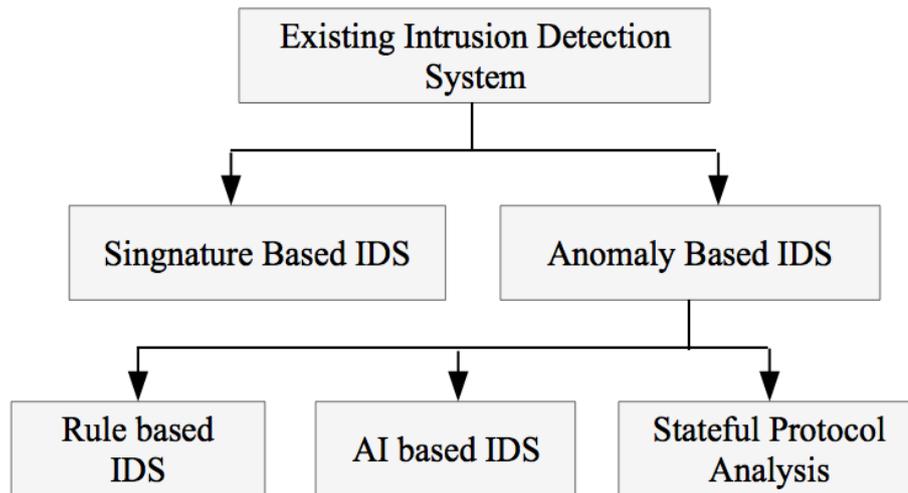


Figure 2.6: IDSs categorized based on their detection mechanism

2.8.1.1 Packet-based NIDSs and or flow (stream)-based NIDSs

Furthermore, there are two methods based on the source of data to be analysed in NIDS. Packet-based NIDSs and flow (stream)-based NIDSs are two variants of the NIDS that have been designed to analyse varying data sources. The former must examine the entire payload content in addition to headers, while the latter tends to avoid examining every packet travelling through a network link, thereby focuses on the aggregated data of associated packets of network traffic in the form of flow. Consequently, flow-based NIDSs are not required to analyse the same large quantity of data as packet-based NIDSs (Alaidaros, 2011).

The differences between Flow-based NIDSs and Packet-based NIDSs are shown in Table 2.4 (Sperotto, 2011) (cisco, 2011).

Table 2.4: comparison between Flow-based NIDSs and Packet-based NIDSs

Differences	Flow-based NIDSs	Packet-based NIDSs
Layer	Flow records incorporate aggregated data up to the transport layer (i.e. layer 4 in OSI).	Packets incorporate comprehensive payload and headers up to the application layer (i.e. layer 7 in OSI). Consequently, these are regarded as having greater adaptability when applying intrusion detection patterns.
Privacy	Fewer privacy concerns result because a considerable proportion of the possibly private connection information remains within the transmission network at all times.	Packet-based NIDSs are provided with the complete payload information for each packet, with several of these incorporating confidential information.
Size	A comparatively reduced quantity of data to analyse since an element of the processing is delegated to the probe device. Consequently, efficient use of resources is a typical feature.	When a hardware pre-filter is not used, all packets must be processed. Consequently, efficient use of resources is often not the case.
Encryption	Encrypted payload has no impact on the degree to which these NIDSs are operable.	Signature matching cannot be conducted for almost all encrypted payload cases, thereby negatively impacting detection capability.

Summarily, a flow-based NIDS is typically developed using data sourced and aggregated up to the fourth OSI (transport) layer while its packet-based NIDS counterpart is developed using data extracted up to the seventh OSI (application) layer architecture. Thus, packet-based NIDS strength lies in the fact that it is built upon a more comprehensive payload information than flow-based NIDS. However, this strength leads to its inability to conduct a signature matching for almost all encrypted payload cases whereas encrypted payload has no effect in limiting the ability of a typical flow-based NIDS while detecting attacks.

2.8.2 Artificial Intelligence IDS

Over the years, network systems have recorded a deluge of cyber-attacks, which are not limited to infection of (networked or connected) computers and or devices by viruses, information theft and leakage, misuse of information and even unauthorized access to classified information. Various security mechanisms have been designed, developed and deployed to eliminate or at least drastically ameliorate this menace, these mechanisms include anti-viruses, firewalls, security expert systems (containing thousands of line of code). Though these mechanisms are effective, attackers are known to circumvent the operations of these mechanisms and successfully gain access to the network, and thus the development of IDS is also a security tool primarily for detecting malicious activities on a host and within a network as a whole.

However, with the inception of data mining – a branch of Artificial intelligence, the development of IDS has taken on a new outlook. Data Mining refers to the process of extracting hidden, previously unknown and useful information from large databases (Mitchell and Vora, 2013). It was stated by (Mitchell and Vora, 2013) that, ‘It is a convenient way of extracting patterns and focuses on issues relating to their feasibility, utility, efficiency and scalability.’ Essentially, data mining techniques can be used to build up Intrusion Detection Systems in real time environments, as the techniques of data mining can be used in several ways (Jaiganesh, Mangayarkarasi and Sumathi, 2013), which include: “*Data summarization*” that summarizes evolutionary data with statistics; “*Visualization*” that renders the summarization into graphical data; “*Clustering*” that deals with the collection of peculiar objects into various clusters; “*Association*” that correlates the existence of a set of items with another range of values for another set of variables; “*Classification*” performs mainly the prediction of a hierarchy of class from an existing set of events or transactions; “*Forecasting/Prediction*” reveals how certain attributes within the data will behave in the future; and lastly “*Sequential Patterns*” deals with discovery and investigation of a sequence of actions or events.

With the possible techniques of data mining, the goals of data mining are widely categorized into:

- 1) Prediction: discovers the relationship between independent variables and relationship between dependent and independent variables. Data mining shows how particular attributes within the data will behave in future.
- 2) Identification: data patterns are used to identify the existence of an item, an event, or an activity or some new patterns of entity behaviour.

- 3) Classification: data mining can be used to separate the data so that different classes or categories can be recognized based on a combination of parameters.
- 4) Optimization: data mining can be used to optimize the use of limited resources such as time, space, money or materials and to maximize output variables under a given set of constraints.

Data mining, filled with techniques of great potential, is used for developing IDS as it becomes advantageous. Of its many advantages, a few are highlighted as follows:

- 1) It is extremely difficult to program an Intrusion Detection and Prevention System using ordinary programming languages that require the explication and formalization of knowledge.
- 2) The adaptive and dynamic nature of machine-learning makes it a suitable solution for this situation.
- 3) The environment of an IDS and its classification task highly depend on personal preferences. What may seem to be an incident in one environment may be normal in other environments. This way, the ability of computers to learn enables them to know someone's "personal" (or organizational) preferences, and improve the performance of the IDS, for this particular environment.

Developing IDS by adopting the data mining technique, makes use of the classification process of the technique. Since classification is the mapping of data items into one of several predefined categories having learned rigorously the peculiar characteristics of the predefined categories from sets of data items. An ideal application of the classification technique of data mining to the development of IDS will collect sufficient "normal" and "abnormal" audit data for a user host and then apply a classification algorithm to learn a model that can predict new unseen audit data as belonging to either normal or abnormal class. In a case of developing a network-based IDS, the data collected will include "normal" and "attack" traffic – for a binary classification problem, however, the "attack" traffic can be further broken down and renamed to specify the particular type of attack, e.g. "Replay", "DDoS", resulting in a multi-classification problem.

An artificial intelligence IDS, is an example of IDS developed to use the advanced data mining technique. There exist several classification algorithms broadly categorized into: Decision tree based, rule-based, memory-based reasoning, Neural Network, Naïve Bayes and Bayesian Belief Networks and Support Vector Machines (Karim and Rahman, 2013).

2.8.3 Stateful Protocol Anomaly Detection

An IDS using the classification (supervised learning) technique of data mining, executes its function by outputting the class to which a data item belongs having learned about its features and the predefined classes. Typically, the traffic in a network of devices is meant to be secure having emanated from legitimate users whose system activity within the network is referred to as normal. When the profile and activities of legitimate users of a network are tracked, known and monitored, any form of discrepancies to the identified pattern of a user is referred to as anomaly. In this case, an IDS performs a stateful protocol anomaly detection by checking the normal system activity such as network bandwidth, ports, protocols and device connection, if there be any abnormal activity in the system or the network, it informs the administrator. Classification techniques (i.e. classification algorithms) are widely used for anomaly detection and it is often a binary classification problem (Mitchell and Vora, 2013).

2.9 Summary

The deployment of IPv6 network, currently co-existing with the IPv4, ushers in an era of modern network of devices propagating secure communication. More so, the new IPv6 network possesses its own vulnerabilities as well as the existing IPv4 network had its own vulnerabilities which saw the development of several defence mechanisms. However, cyber-attacks are on the high side, discovering and taking advantage of new vulnerabilities in IPv6 and posing new forms of attacks and threat, while attacks of IPv4 are still on going.

One of the core protocols on IPv6 – NDP, which eases connectivity of IPv6 nodes in an IPv6 network, is susceptible to attack. NDP can be manipulated to execute malicious activities within the network as described in this chapter. Intrusion detection and Prevention System had been developed for detecting malicious activities in a host machine as well as intrusion or anomaly in a network. Peculiar to NDP are DoS, Replay and DDoS attack types, and these attacks though carried out in a new way as compared to the same attack in an IPv4 network, IDS are capable of detecting and preventing such attacks.

On this note, NDP, its message type, format and its vulnerabilities in an IPv6 network, attacks common to NDP, IDS / IPS were discussed. IDS have been investigated to combat various forms of attacks in IPv6 by various researchers. The research on IDS needs to be continued as attackers are consistently finding new vulnerabilities and creating dynamic attacks. Interestingly most IDS framework are publicly available for both security experts to deploy and attackers to find a loophole.

CHAPTER 3

MACHINE LEARNING

3.1 Introduction

This chapter presents a detailed introduction into machine learning, and its types (supervised and unsupervised) while revealing the categories for each type of machine learning. Also, the selected machine learning technique for this research work is revealed and discussed i.e. Decision Tree, Naïve Bayes and Bayesian Network. More so, the Locally Weighted Learning method is discussed as well as its types (i.e. the memory based and the incremental), the application of machine learning in developing Intrusion Detection Systems is reviewed and lastly a concise summary is presented.

The development of software with the ability to learn from experience or history is the most important aim of Machine Learning (ML). Machines are now empowered to learn from experience of a specific application and thus are capable of making guided future actions based on the learning obtained from experience. According to Holzinger, (2017), ML involves understanding intelligence for the design, development and implementation of algorithms capable of learning from data, thereby gaining useful knowledge from experience, and also improving learning behaviour over time. MLs are strong mathematical models which are also based on Statistical Learning Theory (SLT), as SLT had so far contributed immensely in the evolution of better learning algorithms (Dutta, Subramaniam and Sanjeevikumar, 2018). SLT is a large framework that studies vital question of learning and inference, as well as extracting knowledge, decision making, construction of formal models from data, and making predictions, drawing from functional analysis and statistics (Motzev, 2018).

ML offers practical solutions to a lot of problems in our society ranging across several domains, applications and daily life use-cases (Anon, 2017) such as cyber security, autonomous driving, speech recognition, recommender systems, smart factory, smart health etc. The success of ML and the enormous usefulness that ML has achieved in our modern society is tied to the power and applicability of *statistical learning theory*, acceptance of the concept of *probable information in an uncertain world* by engineers, and the visible and convincing success of *deep learning* (Besold *et al.*, 2017). Primitively, the ML challenge is to discover relevant knowledge in data that are of usually arbitrarily high dimensional space and inaccessible to a human.

However, sense making, decision making under certainty, and in context understanding make up the grand challenges of ML. Solving these challenges is the factor behind the broadened international effort, cross-domain, supporting collaborative, inter- and trans- disciplinary work of experts from seven distinct sections ranging from data pre-processing to data visualisation: for the mapping of discovered knowledge from high dimensional spaces into lower dimensions, thereby rendering the discovered knowledge useful, usable and accessible to end users (Holzinger, 2016).

Since ML deals with usable intelligence which can be characterised as (i) learning from historical data, (ii) extracting knowledge, (iii) generalisation – predicting/guessing (iv) eliminating the problem of dimensionality, and (v) disentangling underlying explanatory factors of the used data – i.e., making sense of the contextual data, thus the need for integrated machine learning approach (Holzinger, 2019). Furthermore, data protection, privacy, safety, social implications, user acceptance and security are also considered when executing an integrated machine learning approach. In addition, the rapid growing deluge of data, artificial intelligence and statistical learning make hidden intelligence visible and accessible to humans, based on the enormous influence of probabilistic inference that enables inferences on what was previously unknown having learned from data and subsequently produce actionable insights or predictions capable of supporting major decision making processes (Shiranzai and Khan, 2015).

The predictive power of Gaussian Process (GP) – the generalization of the normal probability distribution, in successfully dealing with stochastic processes in time is yet a huge success factor that had made uncertainty and probabilistic reasoning an elaborate and successful field of study (Rasmussen, 1996). GP, developed by and named after Carl Friedrich Gauss, is often used as a prior probability distribution over functions being applied on high-dimensional data (which is vital in our modern-day science). Furthermore, the development of *probabilistic programming* – a concept of programming that enables the taking of runtime generated values through random sampling procedures, unlike its traditional programming counterpart provided much important practical usefulness in the view of uncertainty and probabilistic reasoning, such as, the ability to randomly draw values from probability distributions and to condition the variables of those values in a program via observations (Mansinghka *et al.*, 2018). This combinatorial advent of Bayesian probabilistic reasoning, Gaussian Process, as well as probabilistic programming had led to a massive breakthrough in different applications of human endeavours, as quite a lot of daily real-world problems are being transformed into

probability distributions and harnessed by probabilistic programs which make out probabilistic inferences.

ML was born out of the ideas of developing algorithms or frameworks that can be used to automatically learn from (historical) data in order to gain knowledge or extract meaningful information from experience and the algorithms are capable of improving their respective learning behaviour over time (Holzinger, 2017). Originally known as “*the artificial generation of knowledge from experience*”, ML began its experiments with games aimed at replacing humans and their manual method in the analysis of data. Just as humans, ML learns from data, but automatically, for the purpose of making informed predictions and decisions using available resources. ML has grown in the last two decades, though it’s always been a field of overlapping interest between computer science and cognitive science, and has seen several achievements in various fields of study such as cyber security, finance, astronomy, natural language understanding and inference (Wang *et al.*, 2012). ML had erupted into more than classroom or tertiary institutions research lab scale into a wide industrial scale where large corporations invest into diverse research projects involving ML use-case applications. This eruption had led to integration and large-scale fusion of industries and academia, as huge business potentials of the future are being harnessed. Large corporations such as Microsoft, Amazon, Google are giants in this stride, just as small-scale organisations are actively using ML to solve business problems as relevant to the modern day.

All ML algorithms’ performance is totally dependent on the data and its representation. How data is being represented is key to the learning and understanding process of all ML. Thus, the design of data pre-processing, data transformation and mapping activities are thoroughly handled and carried out in order to support an effective machine learning development process and tentatively yield a robust machine-learning model. ML must have a contextual understanding and be able to discriminate between relevant and irrelevant features during the process of feature engineering.

The community of ML experts all around the world contributes immensely to the development of algorithmic systems capable of automatic learning from data without human interference. This is known as *automatic machine learning (aML)*, it becomes more effective with the availability of “Big Data” (Hutter, Kotthoff and Vanschoren, 2019). Big data refers to large or deluge data collected by satellites, high throughput machines, telescopes, smart phones, sensor networks etc. Though this form of ML is known to face challenges such as small data, rare events, hard problems, complex data problems etc., on the other hand, there exist *interactive*

machine learning (iML) which integrates humans into its functional process and strengthens itself by making use of human cognitive abilities to solve hard problems, complex data problems or rare events (Holzinger, 2016).

3.2 Machine Learning Types

Machine learning deals with designing algorithms or methods that enable learning in a computer, learning in this context does not necessarily involve consciousness but discovering statistical regularities or other patterns in a given data set. Thereby giving the machine the ability to approach a problem in a manner similar to humans and also, through this enablement, the machine can discover difficult insights previously hidden to humans either due to high dimensionality of the data or the volume of the data (Chao, 2011).

Machine learning algorithms can be categorised into supervised and unsupervised learning types.

3.2.1 Supervised Learning

Supervised learning simply refers to the kind of learning whereby machine learning generates a function that can reasonably map sets of observations to some predefined class, i.e. the dataset containing rows of observations having some attributes as columns, is being labelled by humans (domain expert) and fed into a machine learning algorithm for it to learn from (Chao, 2011) (Sathya and Abraham, 2013). Then, in most cases, unlabelled new instances are fed to the developed model for classification in order to measure the learning power of the model.

Machine learning that falls under the supervised learning category tries to find the relationships between the attributes and the labels or classes of a given set, which is the properties and knowledge that are learnable from the labelled dataset. Supervised learning can be achieved in two ways depending on the dataset (Chao, 2011), if the attribute vector x is corresponding to a label $y \in L, L = \{l_1, l_2, \dots, l_c\}$, then the learning problem is referred to as classification. But when the attribute vector x is corresponding to a real value $y \in \mathcal{R}$, then the learning problem is called a regression problem. The embodiment of knowledge developed from supervised learning (either classification or regression) is mostly used for recognition or prediction.

All supervised learning algorithms are usually categorized based on their approach to solving classification and or regression problems. In this study, we shall adopt the categorization of

supervised learning algorithms into four different types of models, which are linear models, parametric models, non-parametric models and non-metric models. Table 3.1 displays a tabular view of the categories and the categorisation of supervised learning algorithms respectively.

Table 3.1: Categorisation of Supervised Learning Techniques

Category	Classifiers
Linear Model	Linear regresses, Rigid regression Logistic regression Support vector regression Support vector machine Multi-layered Perceptron
Parametric Model	Probabilistic graphical model Naïve Bayes Gaussian discriminant analysis Hidden Markov model
Non-parametric Model	K-nearest neighbours Kernel density estimation Kernel regression, Local regression
Non-metric Model	Decision tree Classification and regression tree

As depicted in Table 3.1, linear models consist of algorithms that can be used for regression and classification. Also, the parametric model represents models that are built on well-defined probabilistic distribution offering the discarding of the training set and reservation of parameters for testing and prediction. Non-parametric models are built on probability, their function based on the assumption that similar features vectors have similar labels. Lastly, the

non-metric model takes no account of the similarity of feature vectors (they are without any natural notion of similarity metric), which is always the case of discrete features. In some studies, there exists a fifth category referred to as *Mixed Method*, which makes use of aggregation of base classifiers' results, e.g. Bagging, Adaboost, Random forest etc.

3.2.2 Unsupervised Learning

Unsupervised learning is simply the form of learning in which a machine learning algorithm uses an unlabelled dataset, i.e. the dataset used as training input is not annotated by humans (domain experts) before the process of learning is being carried out (Ayodele, no date). Thus, the form of learning achieved is different from its supervised learning counterpart. Due to the form of learning achieved through unsupervised learning, it is used for finding association among features, clustering, dimensionality reduction and even probability density estimation. Finding and learning these properties can be executed simultaneously using unsupervised learning algorithms and the results may be further used for a supervised learning process.

Clustering is one of the most used functions of unsupervised learning algorithms, it focuses on separating some set of samples into several distinct groups based on some form of peculiar similarities they share or some distance measures values (Guerra *et al.*, 2010). This is fundamentally achieved by minimizing the intra-group distance while on the other hand, maximizing the inter-group distance. Clustering is essentially used for discovering the structure underlying within a given sample, a vital application for business and medical purposes. For example, the separation of patients into groups based on some peculiar attributes and tentatively designing specific methods for treatment for each discovered group is one application of a clustering technique in medicine, another example is the partitioning of customers of a particular business and treating each group's business needs which will ensure effective and efficient customer service. More so, for each cited case, the samples contained in each distinct discovered group can be labelled accordingly and further used to develop supervised learning models.

Another use of unsupervised learning is probability estimation, very functional in estimating the appearing probability of each sample vector through some of the supervised learning models (i.e. non-parametric models, or semi-parametric models or parametric models). This enables the discovery of the underlying mechanism for generating samples and the learning probability distribution can be further used for the detection of special or rare events as well as

outlier detection. Furthermore, dimensionality reduction is another feature of unsupervised learning. Given some instances, dimensionality reduction is mainly used to learn the relationship between all features and discovers the latent factors that control the feature values of a sample. In other words, dimensionality reduction is the process of discovering redundancy and irrelevancy among features of a given dataset, i.e. of two or more features that are highly correlated; the best will be kept while the other is discarded. Or in the case of correlation with the label, features providing insignificant information with respect to the labels are removed. Thus, a compact representation of each sample is being derived, which is more informative for subsequent application. Of the many advantages of dimensionality reduction, the reduction of computation complexity and prevention of possible over-fitting of data during the learning process are the essence of this process achieved by the unsupervised learning techniques.

Table 3.2: Categorization and sample of Unsupervised Learning techniques

Category	Techniques
Clustering	K-means clustering Expectation Maximization (EM)
Dimensionality reduction	Factor Analysis Principal Component Analysis (PCA)
Density Estimation	Graphical Models Gaussian mixture model (GMM)

Table 3.2 pictorially depicts the categories of unsupervised learning methods and some sample algorithms for each category (Chao, 2011) (Bhosale and Ade, 2014).

3.3 Machine Learning Technique

As discussed previously, there are several machine learning techniques, but for this study three machine learning algorithms were extensively discussed and implemented, namely Decision Tree (DT), Naïve Bayes (NB) and Bayesian Networks (BN). The selected machine learning techniques were chosen as each one of them possesses a distinct method of learning thereby

providing heterogeneity to the implementation of the locally weighted learning method as used in this research work. The subsections below discuss each selected machine learning method

3.3.1 Decision Tree

Decision trees are a way of representing a series of rules constructed greedily in a top-down manner, beginning from the root node to the leaf, that lead to a class or value (Jain, 2012). A decision tree is a tree structure consisting of internal and external nodes connected by branches. An internal node is a decision-making unit that evaluates a decision function to determine which child node to visit next. The external node, on the other hand, has no child nodes and is associated with a label or value that characterizes the given data and leads to its visit. Essentially, there are three basic elements of a decision tree, viz (i) an edge or a branch that represents one of all the possible values of an attribute, i.e. one of the test attribute outcomes, (ii) a decision node that represents the test attributes, and (iii) a leaf that represents the class to which an object belonged to (it is also known as answer node). Many decision tree construction algorithms involve a two-step process. For the first process, a very large decision tree is grown. Through the second step process, reduction of the large-sized tree and overcoming data overfitting is achieved by pruning the given tree. The pruned decision tree that is used for classification purposes is called the classification tree.

An important feature of the decision tree algorithm is the attribute selection, which is the basis for constructing the trees. Attribute selection enables the decision tree algorithm to construct its tree by creating branches via splitting on important attributes based on some criterion score, which can either be gain ratio or information gain (entropy) etc. (Li, 2010; Bhosale and Ade, 2014). Using one of these criteria, decision tree algorithms are able to find the best attribute and make use of it as the root node and other test nodes. The discriminative power of each attribute over classes is computed using one of these methods and then used to build a tree, which is then pruned for generalization of new instances.

It is noteworthy to highlight the fact that the decision tree requires no domain knowledge in the construction of its tree, it handles high dimensional data, the resulting tree is easy to understand and interpret, and also it can handle both discrete and categorical data (Vinitha, 2013). On the other hand, it is weakened by the fact that it can only work with categorical output attribute, it is also known to be unstable and the trees developed from numerical datasets can be complex in some cases. Concisely, a decision tree is similar to a flow-chart tree structure but with internal nodes that represent test attributes, branches as outcomes of a test and leaf nodes as

classes and it is a widely used machine learning algorithm (Hamed, Ernst and Kremer, 2018). Figure 3.1 below pictorially depicts a decision tree structure.

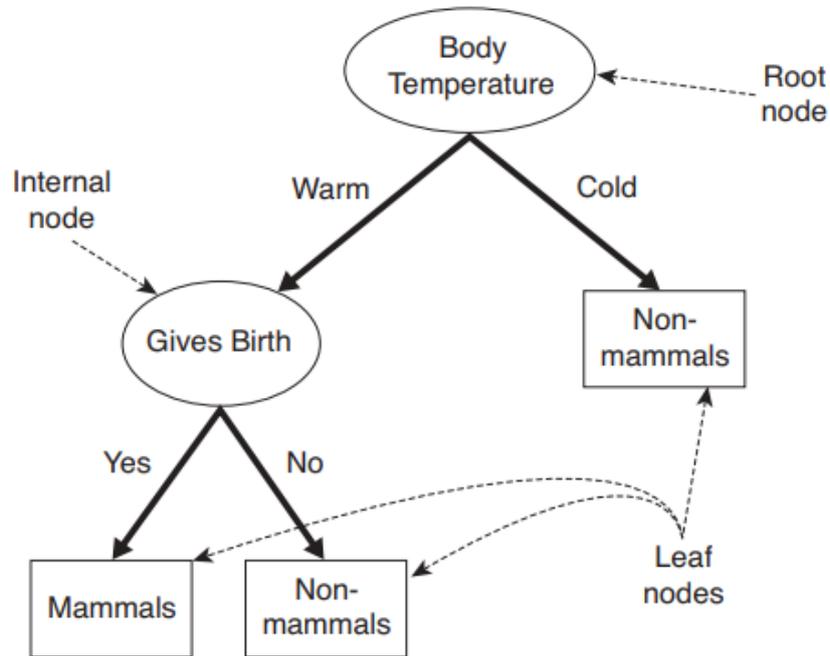


Figure 3.1: A typical decision tree structure for mammal classification (Tan et al, 2006).

3.3.2 Naïve Bayes

Naïve Bayes, which simply called Bayes, is a method for developing models that are able to solve supervised classification problems based on probabilistic graphic models (Jain and Rana, 2016). Among the benefits of using Naïve Bayes is its ease to construct, not requiring complicated iterative parameter estimation schemes, easy interpretation of its constructed rules (which is important for unskilled users of classifier technology), and it often outputs excellent and robust models in various domain applications, given its simple rule construction (Chai, Hn and Cheiu, 2002).

The naïve Bayes method of classification is one of the oldest approaches to solving classification problems, even with a simple structure; it outputs robust models, which are surprisingly effective in most cases.

3.3.3 Bayesian Networks (BNs)

Belonging to the family of probabilistic graphical models (GMs), Bayesian Networks (BNs) are graphical structures used in the representation of knowledge about an uncertain domain (Hodo *et al.*, 2017). Also, referred to as *belief networks*, BNs graphical representation entails the node as a random variable with the edges connecting the nodes representing the probabilistic dependencies among the corresponding random variables. Making use of computational and statistical methods, BNs are able to estimate the conditional dependencies within the graph. BNs algorithmic principles are deeply rooted in probability theory, statistics, graph theory and computer science (Faltin and Kenett, 2007). GMs that consist of undirected edges are referred to as Markov networks or Markov random fields, which for any two distinct nodes provide a simple definition of independence based on the concept of a Markov blanket. However, BN is different from Markov network as it possesses a Directed Acyclic Graph (DAG) structure which is intuitively understandable though mathematically rigorous. BNs' power lies within their effectiveness in the computation and representation of the Joint Probability Distribution (JPD) over a set of random variables.

Structurally, a DAG consist of two sets: (i) a set of directed edges – which is visually depicted as arrows between nodes, represents the direct dependences among the nodes, and (ii) set of nodes (vertices) – representing random variables and are visually depicted as labelled circles (the labels serve as the variable name) (Faltin and Kenett, 2007).

More so, the structure of BNs depicts the conditional independence enjoyed by each variable, as nodes are independent of their non-descendants on the graph given the state of their parents (Jain and Rana, 2016). The advantage of this conditional independence is its ability to reduce the number of parameters required to characterize the JPD of the variable and in turn create an efficient way of computing the posterior probabilities given the evidence. With respect to the DAG structure of BNs – mostly referred to as the *qualitative* part of the model, the *quantitative* parameters of BNs are similar to Markovian property, wherein the Conditional Probability Distribution (CPD) at each node depends only on its parents. The conditional probability for discrete random variables is usually represented by a table, itemising the local probability that a child node takes on each of the feasible values – for each combination of values of its parents. These conditional probability tables (CPTs) make it easy to uniquely determine the joint distribution of a collection of variables.

The evaluation of all possible inference queries can be executed for a given BN with specified JPD in a factored form, through the process of marginalization – the process of summing out

over “irrelevant” variables. There exist two main types of inference support for a node, (i) predictive support – which is based on all evidence nodes connected to a particular node through its parent nodes (this is referred to as top-down reasoning), (ii) diagnostic support – also referred to as bottom-up reasoning, is based on all evidence nodes connected to a particular node through its children nodes. More so, BNs is widely applied to several human daily use-cases such as cyber security, signal processing, bioinformatics, weather forecasting, natural language processing and even cellular networks (Balogun *et al.*, 2017). Its node can also be used to represent various hypotheses, latent variables and beliefs, thereby producing an intuitively appealing structure for representing both probabilistic and causal semantics, as it combines prior knowledge found in observed data and also use for learning causal relationships in the case of missing data.

3.4 Local Weighted Learning Technique

Local Weighted Learning (LWL) is a method of approximating function in a different way from other methods of function approximation (Englert, 2012). Every regression and or function approximation is centred on discovering the underlying relationships that exist between the input and the output variables (Stalph, 2014). Considering a supervised learning problem, the training dataset consists of input values that are associated to one corresponding output value, it is the function approximation goal to create a model that possesses the capability to predict the true output value for some given rows of corresponding input values (Stalph, 2014). This predicted output value is the closet value to the corresponding true value of the original function. Most commonly, the function approximation is globally created by making use of the complete training data to develop a model. Most machine learning algorithms, such as DT, BNs, Support Vector Machine (SVM), Linear and Logistic Regression (LR), create a model or function approximation globally. However, this method does not suit or work well in all cases, for example, when there exists a scenario in which no parameter value can produce a good fit or sufficiently good approximation. Also, some machine learning techniques are challenged with high computation complexities while developing models on big data thus leading to computational costs that are on the high side, this typically weakens global function approximation. In addition, a situation requiring many predictions in a short time, as well as incremental extension of a model, reveals the disadvantages of global function.

LWL as a technique was developed as a means to resolve the notable shortcomings of the global function approximation as highlighted in the previous paragraph, thereby providing a

local approximation function. LWL offers non-parametric methods as well as local function for the prediction using only a subset of the data (Stalph, 2014). It produces a local function approximation developed around the current point of interest, i.e. instead of using the original dataset as the whole function space for building a global model, LWL simply creates a local model based on the neighbouring data of the query point (Englert, 2012). As such, the data points become a weighting factor, as those close to the query point receive higher weight than those far away, which expresses the influence of the data point for the prediction. It is referred to as lazy learning due to the fact that no processing of training data is carried out except when a query point requires an answer. One of the other merits of LWL is its ability to yield very accurate function approximation even with new training points (Atkeson, Moore and Schaal, 1996).

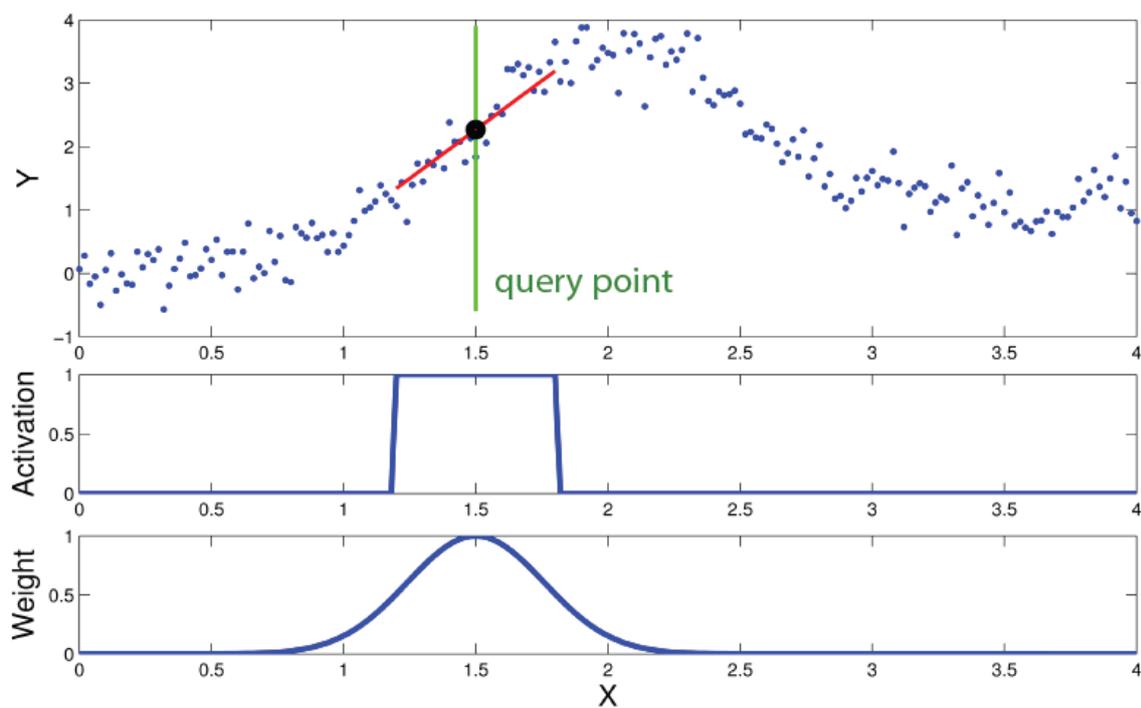


Figure 3.2: Pictorial representation of LWL method (Englert, 2012).

Figure 3.2 illustrates an example of LWL, the uppermost box contains the given dataset represented by blue dots - (x, y) data points, a green line representing the query point (i.e. the current instance to compute its local function using its neighbours), a red line representing the local linear model and a black dot representing the prediction. The middle graphical box, labelled ‘Activation’ reveals the activation site of the model. And lastly, the bottom graphical box – labelled ‘Weight’, shows the corresponding weighting kernel (the receptive field).

3.4.1 Memory-Based Locally Weighted Learning

A memory-based LWL build up function approximation locally having kept all training data in memory in order to compute prediction (Englert, 2012). Memory-based LWL results in models with high accuracy due to the local model and it has few opened parameters, however, it suffers from the fact that it cannot handle redundant input dimension due to its matrix inversion process for obtaining regression coefficient. Another weakness of memory-based LWL is the cost intensive computation for high dimensional data which tends to increase quadratically, thereby rendering the algorithm unsuitable for tasks requiring a lot of prediction in small time steps, as it is known for discarding each model after making predictions.

3.4.2 Incremental Locally Weighted Learning

The incremental LWL method on the other hand aims to solve the main problems faced by memory-based LWL, as it keeps each model and makes use of it in its future predictions. The incremental LWL method approximates non-linear functions by making use of combined multiple locally weighted linear models (Atkeson, Moore and Schaal, 1996). In addition, for incremental LWL to add up new data points, it either updates an existing model or in the case of no trustworthy model, it creates a new model, thereby making it unnecessary to save large training data in the memory. Incremental LWL implementation is well suited for training data with high dimensionality, and also in case of continuous data streams and redundant input dimensions. Also, its capability to combine high accuracy of prediction with low computational costs is yet another advantage of this type of LWL method. Lastly, incremental LWL is known to have high adaption over time which is essential when faced with systems that changes over time (Englert, 2012). Machine Learning in Security

With the digitalization of the world's today via interconnection of devices, ranging from mobile phones to computer, various meritorious features are existing in the modern society such as high – speed bank transaction, online shopping and even top secrets in governmental institutions. These meritorious features deals with large volume of data and information whose safety is of great concern (Thiyagarajan P., 2019). Hence, the existence of security mechanism is of great essence and they exist in various forms such as steganography, cryptography, expert systems, biometric authentication and many others. It is noteworthy to mention that cyber security is the term that refers to the collection of techniques and processes for protecting

networks, computers, data and programs from unauthorized access or attacks. By and large, cyber security is categorised into (i) application security, (ii) information security, (iii) disaster recovery, and (iv) network security.

The pivotal importance of all security measures is to provide three main utilities (Yavanoglu & Aydos, 2017) known as the famous three i.e. confidentiality, Integrity and availability. All the security measures functions in order to provide these highlighted utilities in their various capacities. However, due to the dynamic nature of attackers (either internal or external) some of the existing methods are being compromised or unable to adequately detect attacks.

Considering network security, the need for real time securing of data, information and devices, is some what impossible as it takes quite a number of security personnel to manually manage and monitor network traffics in real-time. Also, the detection of novel “zero day” attack is impossible for most security mechanisms such as expert systems and or signature-based security frameworks (Xin et al., 2018). Thus, the lookout for advanced security mechanisms to take care of this menace. In action to curb this menace, data mining, statistic, machine learning and several interdisciplinary capabilities are being involved to provide a panacea in the form of IDS.

Mainly, IDSs are part of a typical network security system alongside antivirus software and firewalls. IDS are developed to help in discovering, determining and the utmost identification of unauthorized system behaviour by analysis network traffic data (Buczak & Guven, 2016). The various form in which IDS exist can be categorised into three namely signature-based, mis-use based and hybrid categories.

The development of IDS is possible through the application and implementation of machine learning algorithms to develop IDS models using the extracted network traffics which contains both normal packets and (simulated) attacks (Dua & Du, 2016). Through the usage of data mining – the extraction of knowledge from a large volume of data, and machine learning – statistical learning and mathematical algorithms, strong patterns are being learned by machine learning models useful in the categorisation of new network instance(s).

Concisely, machine learning is useful in security as it facilitate the development of IDS through advanced analysis of network data via the implementation of mathematical algorithms and or statistical learning techniques to produced models that are capable of discovering previously

unknown relationships and patterns that are valid in a large network data in order to classify a new network data instance.

3.5 Comparative Analysis of Related Existing Machine Learning Methods in IDS

An academic review of the existing IDSs developed for detecting IPv6-based attack revealed that various methods have been applied to detect various forms of attack. This research work is focused on the detection of DDoS, Replayed, DoS attacks and other anomalies. Thus, related researches on the detection of IPv6-based attack will be reviewed in this section.

A rule based technique was developed by (Aleesa, Hassan and Kamal, 2016) to detect RA flooding (IPv6 DDoS) on a BioBizz web application. The technique was used with Principal Component Analysis (PCA) method for feature selection. The accuracy of the model was measured with respect to time period, and it detected the attack at 100% accuracy during time period 23 – 24 seconds.

This study (Liu and Lai, 2009) , proposed the usage of an a priori algorithm on a six packets-based features for detecting DDoS attack type in IPv6 packet records. However, the achieved accuracy was low – 72.2%, which can be attributed to the inclusion of features that are actually irrelevant in identifying DDoS attacks.

The work of (Zulkiflee *et al.*, 2015) focused on developing a framework for better feature selection with respect to IPv6 Network for the detection of cyber security attacks. The Particle Swarm Optimization (PSO) features evaluation technique was adopted for finding the best features from the original dataset, after which the subset was fed as input into the Support Vector Machine classification algorithm resulting in an accuracy of 99.95%.

The study of Salih, Ma and Peytchev, (2015) showcased the hybridisation of c4.5 and information gain technique for feature selection and the usage of Naïve Bayes for the classification of covert channels in IPv6. It shows an accuracy of 96.46%.

Using Back-propagation Neural Network, (Saad *et al.*, 2016) carried out a research in detecting ICMPv6 DDoS flooding attack using real datasets from Nav6 laboratory. The developed model had an accuracy of 98.3%.

The research conducted by (He, Zhang and Lee, 2017), the classification of virtually generated 9 hours long network packages containing attack and normal packets in a cloud infrastructure by seven different classifiers (Logistic Regression, Support Vector Machine (linear, Radial Basis Function, and poly – kernel variations), Decision Trees, Naïve Bayes, Random Forest,

K-means and Expectation Maximization) respectively. Though the data is not a multivariate normal distribution, Random Forest achieved an accuracy of 94.6% on a single host server while on a multiple host server; SVM (linear) had 99.73% accuracy.

Several machine learning algorithms were applied to a new form of packet representation of IPv6 traffic in order to detect ICMPv6-based DDoS in the research work carried out by Elejla *et al.*, (2018). Usually, IPv6 traffic is represented in a packet-based form, but this research work provided a flow-based representation of the traffic and thus supplied it as input into seven different classifiers. Two datasets were generated and used for testing, the first test adopted 10-fold cross validation while the second test represented a real-world scenario. The test simulating the real-world scenario yielded the highest accuracy of 85.83% for the flow-based model while the packet-based models had the highest accuracy of 61.28%.

An implementation of deep learning to detect DDoS for multi-vector attack in an SDN environment was carried out by (Niyaz, Sun and Javaid, 2016) using a Stacked Auto Encoder (SAE) model, and it was able to detect DDoS packets at 95.65% accuracy and, it could differentiate between normal and attack packets with 99.82% accuracy, though it has limitations in its processing capabilities.

The study carried out by Ye *et al.*, (2018) presented an SVM based method for detecting DDoS attack type in a Software Defined Network. The framework achieved an overall accuracy of 95.25% on a 6-tuple characteristic simulated dataset. However, the model had a poor performance in detecting ICMP attack.

Aziz and Okamura, (2017) presented a FlowIDS approach for detecting anomalies on SMTP traffic flows using Decision Tree (DT) and Deep Learning (DL) algorithms. During simulation, the packet dropped up to 33% which prohibited the network bandwidth from being total crippled in the face of an attack. The experiment made use of two datasets and was conducted on a single site, enabling the detection and mitigation of SMTP flood attacks by mixing FlowIDS (DL and DT) and Suricata NIDS, SMTP flow attacks are detected early at the source attack sites and signatures are updated to other sites before the spreading of the attack.

Salih, Ma and Peytchev, (2017) developed a novel framework in the detection of covert channels attacks in IPv6. The work utilised hybridisation of fuzzy logic and genetic algorithm, as a novel method, to produce the best set of fuzzy rules that is sufficiently able to detect covert channels attack. The experiment was tested on the primarily generated dataset, having accuracy of 97.7% and also on 20% of the NSL-KDD'99 dataset with an accuracy of 93.59%.

In the research work of (Li *et al.*, 2018), deep learning was introduced into the detection of DDoS attack which saw to the development of a deep learning-based IDS in an Open Flow-based SDN. The deep learning algorithms used include Bidirectional Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN), and they were all used for classification of the ISCX2012 dataset. The developed model was used to implement a DDoS defender which was tested in real time, showcasing its ability to clean malicious packets in an SDN environment. Achieving a very high accuracy of 98% on the test dataset, this deep learning-based model emphasised its strength.

'DeepDefense' (Yuan, Li and Li, 2017), was developed for identifying DDoS attack using a deep learning algorithm. The research work made use of CNN, different variations of RNN (i.e. LSTM and Gated Recurrent Unit Neural Network (GRU)) and Random Forest (RF) algorithm to carry out its research. The study performed comparative analysis of variations of deep learning techniques and also with machine learning algorithm (Random Forest was selected). DeepDefense implemented four (4) deep learning models namely LSTM, CNNLSTM, GRU and 3LSTM whose results were compared among each other. The best deep learning model, 3LSTM, could identify DDoS attack with an accuracy of 98.410% and AUC score of 99.450%. Also, a performance comparison of the deep learning model against the machine learning model was carried out where the LSTM model achieved an accuracy score of 97.606% with an AUC score of 99.096% whereas the RF model achieved an accuracy score of 93.637% and an AUC value of 93.628

The model developed by Anbar *et al.*, (2018) was an enhanced Support vector machine model using information gain ratio (IGR) and PCA technique for feature selection. The model had three phases, the first for filtering of the voluminous traffic into ICMP type 134, then the second (feature reduction) phase wherein the features of the original datasets underwent a pre-selection stage of finding the best features for the purpose of detecting the type of attack. IGR and PCA techniques were used to find best features separately and their respective results were intersected. After which the third (detection) phase implemented the SVM algorithm for detecting RA attacks using the subset of the original dataset. Accuracy of 98.5% was achieved and the False Positive Rate (FPR) was at the value of 3.3%

The study of Potluri and Diedrich, (2016) established the usage of accelerated deep neural networks in building IDS. Using the popular NSL-KDD'99 dataset, DNN architecture was implemented for the purpose of detecting attacks using both parallel and serial computing and the time required for training the DNN models on both computing platforms was also compared. It generated 4 types of dataset using the NSL-KDD'99, having varying accuracies

for the datasets with the best accuracies spotted at 97.7% for detecting DoS attack in the 5-classed data (i.e. Normal, DoS, Probe, Remote to Local (R2L), User to Root (U2R)).

The research work carried out by Barbhuiya, Biswas and Nandi, (2011), was aimed at detecting spoofing attacks on IPv6, namely neighbour advertisement and solicitation. Their work saw to the development of six algorithms namely NS-Handler, NA-Handler, VERIFY_IP-MAC, RESPONSE_ANALYSER, UNSOLICITED_ADVT_HANDLER, and MiTM-Detector, which were implemented in C++ code, and also six datasets which are NST, NAT, PRB, AUTH, LOG and USAT– tables in MySQL. The research developed a software-based approach that required no additional hardware for detecting neighbour solicitation and neighbour advertisement spoofing as well as DoS and Man – in – the – middle attacks. However, the developed schemes cannot detect attacks such as duplicate address detection, malicious router attack and many others.

A fuzzy algorithm was implemented by (Li, Li and Liu, 2006), using a genetic algorithm to formulate the fuzzy rules for detecting anomalies in a typical IPv6 network of devices. Making use of CERNET2 dataset, the developed IDS was able to detect normal and anomaly traffic, as well as different types of attacks such as DoS, U2R, and Probe. Though not effective in detecting U2R, the developed model performed well when the parameter value is set to 0.5, achieving an accuracy above 85% and a lower value of false alarm rate.

Through the reviewed literature, it is evident that immense research is ongoing in the field of cybersecurity. More so, it is seen through the reviewed works that various implementations of machine and deep learning algorithms produced global approximation functions which are liable to the shortcomings highlighted in previous sections.

As such, the clear differences of this research work with all other closely related work is the fact that the research will be implanting the locally weighted method of learning for a machine learning algorithm. As opposed to the development of a global approximation model which were the usual developed models by most research works as seen through the review of literature, this research work will produce locally developed models that takes into consideration the locale of the query points for each prediction instance.

Although the research work of Elejla et al., (2018) implemented a flow-based representation of data for its detection of ICMPv6-based DDoS attack which was also implemented in our case, the dataset used in this research work is bespoke (i.e. newly extracted) for the purpose of this research work. Furthermore, Elejla et al., (2018) implemented seven machine learning algorithms (C4.5 decision tree, Support vector machine, Naïve Bayes, k-nearest neighbors

(KNN), Neural networks, Random Forest trees, and conjunctive rules) which were used to develop some global approximation functions which is quite opposite to the core of this research work.

More so, this research work is different from Anbar et al., (2018) research work that made use of IGR and PCA feature selection method alongside Support Vector machine algorithm for the development of its detection model as the author only implemented the IGR feature selection method on the newly extracted flow-based network traffic dataset while developing our bespoke DDoS and Replayed models using locally weighted Decision tree, Naïve Bayes and Bayesian Network algorithms.

3.6 Summary

Machine learning is now being fully utilized for the betterment of the society and to ease human endeavours. In the context of this research work, it is applied to network security – one of the core systems of the modern world. Networking enables the connection of devices over defined protocols and facilitates the communication and sharing of information across users via various types of end points. The security of this network, in order to ensure data and information availability, integrity and authenticity is of key interest to network security and as such various mechanism had been put in place, such as IPsec which is central to the security of the IPv6 network. More so, the security of IPv6 is not limited to IPsec, it also included other security mechanisms, among which IDS is chief.

Machine learning types were discussed, the learning strategies for each type were also discussed as well as the categories for each type were highlighted and examples were given in tabular form. Pictorial representations of facts as well as equations were given to further buttress explanations. The selected machine learning algorithms (BN, DT, and NB) and also LWL method were extensively discussed, their algorithms and formulas were also highlighted. The novelty of this research work lies in the introduction and implementation of the locally weighted learning method to produce local approximation function for detecting different attack types (DDoS, Replayed and anomaly detection) in a typical IPv6 network as defined in the cyber security space as opposed to the widely implemented global approximation function method.

In addition, the novelty of this research work is also grounded in dataset creation and preparation, as the dataset used for this study was newly created having simulated an IPv6 network and captured network traffic, transformed the captured packets into a flow-based representation, and lastly implemented a feature selection technique to select optimal features. The next chapter will provide the design and the implementation of the proposed solution.

CHAPTER 4

DESIGN AND IMPLEMENTATION OF THE PROPOSED MODEL

4.1 Introduction

An important feature of the Internet Protocol version 6 (IPv6) suites is the Neighbour Discovery Protocol (NDP), which is geared towards substitution of the Address Resolution Protocol in router discovery, and function redirection in Internet Protocol version 4. However, NDP is vulnerable to Replayed and Denial of Service (DoS) attacks. In this chapter, a novel detection method for Replayed and DDoS attacks is presented, launched using NDP in IPv6. The proposed system uses flow-based network representation, instead of packet-based. It exploits the advantages of Locally Weighted Learning techniques, with three different machine-learning models as its base learners.

4.2 The Proposed model

A novel method has been introduced in developing this model, which makes use of Locally Weighted Learning (LWL) machine learning technique. A locally weighted learning LWL method was adopted to build fast and efficient models using Bayesian Networks (BN), Decision Tree (DT), and Naïve Bayes (NB) algorithms as the base learners. According to Englert, (2012), LWL is the method that enables base algorithms to provide a local function for the prediction of current point of interest using a subset of the training data instead of a global function that makes use of the complete training data. It is often referred to as a lazy learning as it delays the processing of training data until a query point requires a prediction. LWL has advantages in the area of parameter values. In many cases, a global method has no parameter value that could provide a sufficiently good approximation. This is resolved in LWL methods that are typically non-parametric and execute current predictions based on a subset of data (Atkeson, 1996).

Furthermore, LWL method is most effective in the reduction of computation cost, as explained by Englert, (2012), since it utilises data points that are close to the query point so the prediction receives higher weight than those far away, and thus a local model is generated using the

neighbouring data points. Bayesian Networks, Decision Tree and Naïve Bayes algorithms were chosen based on their ability to produce interpretation and easy to understand models, each algorithm differs in the manner of learning and also, they are capable of producing a robust model. Each one of these machine-learning algorithms in conjunction with LWL was used to develop local function approximation.

4.2.1 Proposed Framework

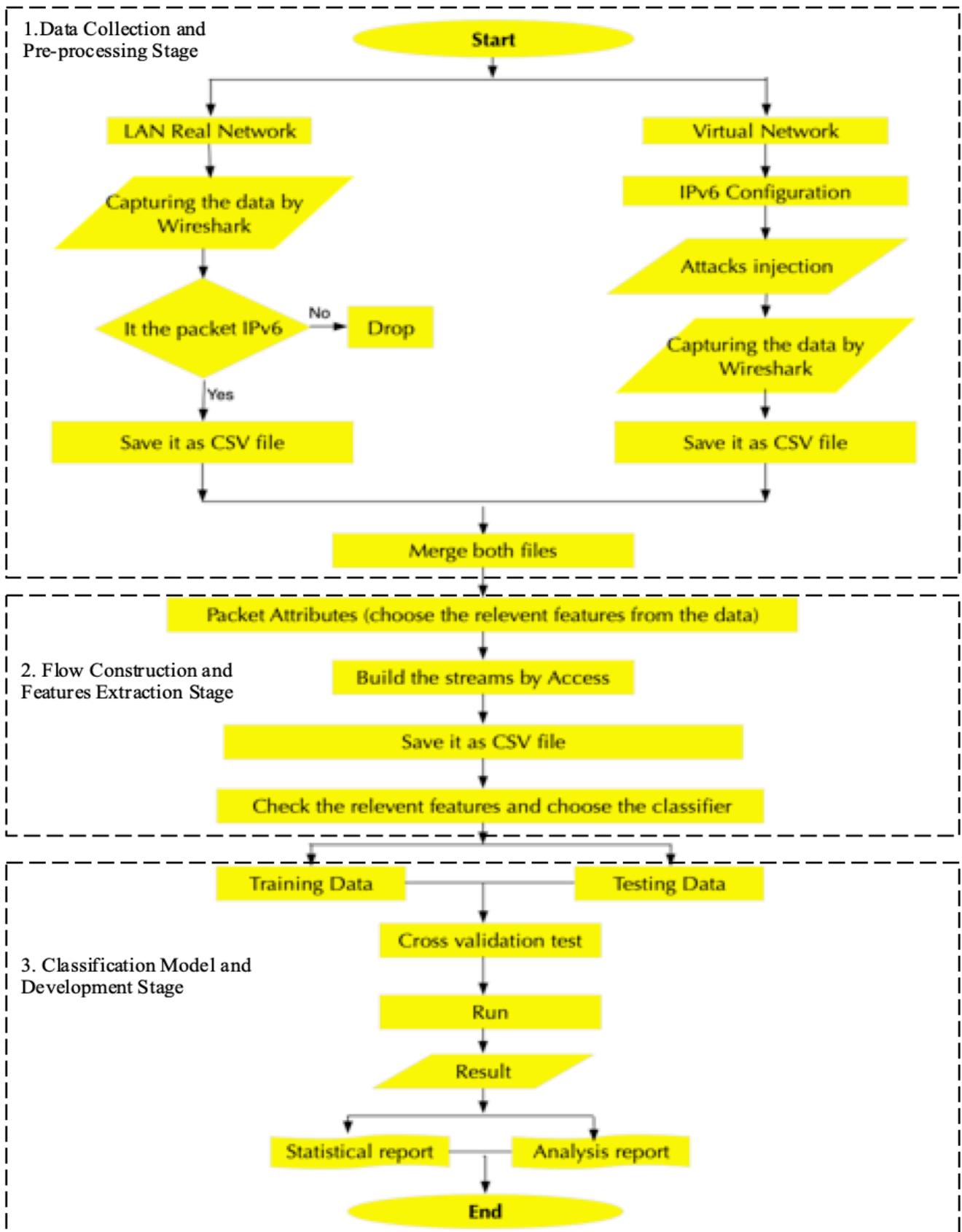


Figure 4.1: Flowchart of the proposed model.

As can be seen in Figure 4.1, the framework of the proposed IDS is structured into three main components: data collection and pre-processing flow construction and feature extraction, mining and classification. These components consist of several tasks that are completed in stages, and explained in the subsequent sub-sections.

4.2.2 Data Collection and Pre-processing stage

In this section, the programming languages as well as tools implemented in developing the proposed solution are presented.

In order to create our dataset, a real network has been used to capture the normal traffic using Wireshark tool (Wireshark, 2017). In addition, Wireshark is a network packet analyser that first seizes the packets that are passing through the network and then decodes these packets into formats that can be read by humans. Although network packet analyser tools were previously quite expensive, today, Wireshark helps in troubleshooting the networks, helps in communications protocol development, and analysing network traffic by offering a free as well as open-source packet analyser. It is possible to use Wireshark for decoding the most known protocols as well as work on protocols from layers two to seven of the Open Systems Interconnection (OSI) model.

Furthermore, a virtual test bed network has been created using Graphical Network Simulator-3 GNS3 (GNS3, 2017) and attached to the real network. An open-source program, Graphical Network Simulator 3 (GNS3) is a graphical network simulator that helps in various network devices being emulated. It also aids in creating a network simulation that has varied complexity levels rapidly and simply. Furthermore, GNS3 assists in simulating the systems without having to rely on physical hardware. Moreover, it also implements Dynamips for imitating real Cisco Internetwork Operating System (IOS) images, virtual box, and switches. Oracle virtual machine (VM) is open-source software that is free and offers a server virtualization software. Oracle VM tends to be utilised for creating multiple virtual servers from a physical server. All Oracle VM can be considered to be independent systems that have their own virtual CPUs, operating systems, storage, and network interfaces.

Different software tools have been used to build the virtual testbed. GNS3 version 1.3.2 is used to build the virtual Test bed network and connect it to the real network. Oracle virtual machine is used to install different Operating Systems ‘OSs’. Different OSs are installed for different

aims which either generate normal traffic (such as Windows XP SP2) or performed attack (such as Kali) or collecting the traffic (such as Windows 7 SP1). CISCO router and switch are used to connect the OSs together and attach the virtual network to the real network. THC toolkit and SI6 tools are used to perform the attacks. Wireshark is used to collect the traffic (normal and malicious).

The testbed consists of the following software and tools:

- Oracle VM virtual machine.
- Windows 7 SP1
- Windows XP SP2
- CISCO router IOS Images.
- Wireshark for capturing and filtering packets.
- Kali: THC-Toolkit version
- Ubuntu: SI6 Networks' IPv6 toolkit.

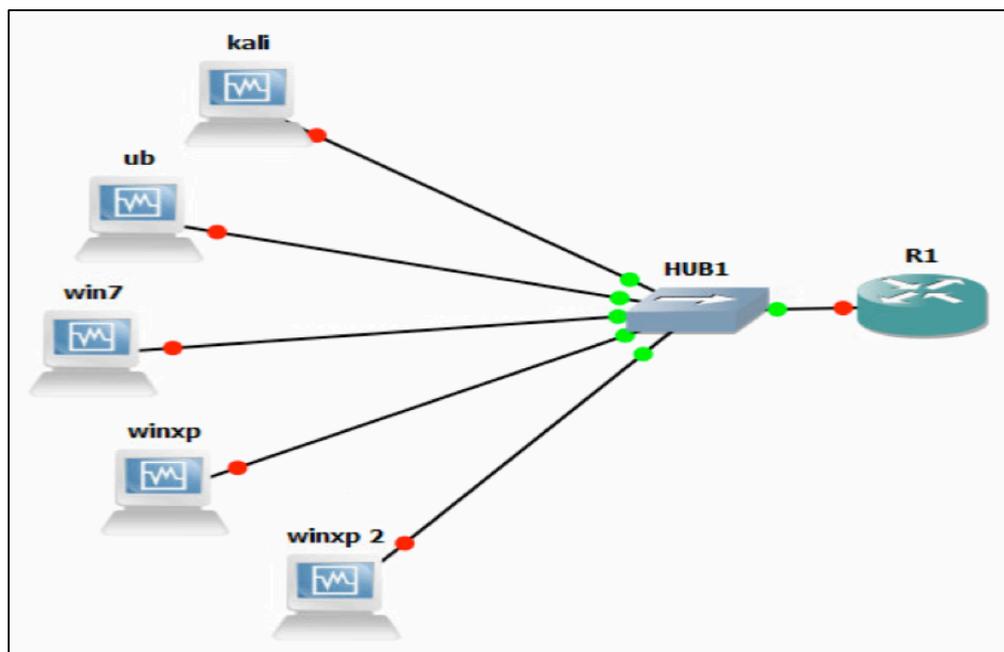


Figure 4.2: Dataset: Virtual Test bed Network Architecture.

The real network used and the virtual network consist of different OSs and network devices in order to have as much as possible realistic level of the datasets. Having this diversity in OSs aims to make sure the created dataset includes all the possible behaviours and scenarios of the traffic, which will lead to the creation of solid detection models. Moreover, the OSs' diversity

makes sure that the created detection model will be OS independent. As long as the detection model experiences a rich dataset, it is strong and able to classify all included scenarios.

Since IPv4 traffic will be filtered out, the proposed work will be limited to IPv6 networks. Moreover, as the dataset will contain only Replayed and DDoS attacks, the created detection model will be limited to these attacks. The strength of the built detection model depends on the included scenarios of attacks therefore; the created dataset will include all the available scenario of the existing attacking tool, which are THC, and SI6 tools. These tools are selected due to their common use to perform these attacks in real life.

4.2.2.1 Data Collection algorithms:

The next algorithms show the data collection stage from both real network LAN and the virtual network VN in order to create the dataset.

Let x represents a set of N number of packets P , and X_{IPV6} is the set of IPV6 packets. Algorithm 1 and 2 depict the data capturing process.

Algorithm 1: Data capturing from LAN.

Read Data:

Let X be a set of packets P

$$X = \{P_1, P_2, P_3, \dots, P_N\}$$

Let $P_N \in X_{IPV6}$ Where $X_{IPV6} \subset X$

Put $X_{IPV6} \rightarrow CSV_1$

where CSV_1 is a Comma Separated Value file for real data

Algorithm 2: Virtual data.

Let V represent a virtual network

Configure $V \Rightarrow V \in V_{IPV6}$

where V_{IPV6} is IPV6 virtual network

Let $t \in T$

where T represents a set of attacks,

$$V_{\text{Attack}} = \{V/V \in V_{\text{IPv6}} \mid t \in T\}$$

Put $V_{\text{Attack}} \rightarrow \text{CSV}_2$

where CSV_2 is a Comma Separated Value file for virtual data.

The virtual network consists of the various Operating Systems, switch and router. The router is configured using Cisco image with IPv6 enabled. The switch is a normal switch used to connect the PCs together and to the router. The other PCs have been used to generate traffic and simulate the real network. Moreover, Kali has been used to perform different types of DDoS attacks on the network using THC-toolkit. Similarly, Ubuntu PC has been used to perform DDoS attacks using the SI6 tool. Table 4.1 shows the performed attacks from the two PCs (Kali and Ubuntu) to the other PCs in the network.

Table 4.1: Examples of attacks commands in the virtual network.

#	Time	Command	Tool	Description
1	2017/05/25 10:39:35	Flood_advertise6 Eth0 <i>Fe80::4014:ae92:e379:d2d1</i>	THC- Toolkit	Send NA messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
2	2017/05/25 10:44:50	Flood_advertise6 Eth0	THC- Toolkit	Send NA messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
3	2017/05/25 10:47:24	Flood_solicit66 Eth0 <i>Fe80::4014:ae92:e379:d2d1</i>	THC- Toolkit	Send NS messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
4	2017/05/25 10:50:32	Flood_solicit66 Eth0 <i>FF02::1</i>	THC- Toolkit	Send NS messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
5	2017/05/25 10:56:04	Flood_router26 Eth0 <i>Fe80::4014:ae92:e379:d2d1</i>	THC- Toolkit	Send RA messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
6	2017/05/25 11:00:15	Flood_router26 Eth0 <i>FF02::1</i>	THC- Toolkit	Send RA messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
7	2017/05/25 15:29:33	Sudo RA6 -i enp0s3 -d <i>FF02::1</i> -F 2000	SI6	Send RA messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
8	2017/05/25 15:37:41	Sudo RA6 -i enp0s3 -d <i>Fe80::4014:ae92:e379:d2d1</i> -F 2000	SI6	Send NS messages from Fake IPv6 addresses to “WIN7 IPv6 address” (<i>Fe80::4014:ae92:e379:d2d1</i>)
9	2017/05/25 15:47:41	Sudo NS6 -i enp0s3 -d <i>FF02::1</i> -t <i>Fe80::a00:27ff:fe00:f725</i> -F 1000	SI6	Send NS messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)

10	2017/05/25 15:52:39	Sudo NS6 -i enp0s3 -d <i>Fe80::4014:ae92:e379:d2d1</i> <i>Fe80::a00:27ff:fe00:f725</i> -F 1000	-d -t	SI6	Send NS messages from Fake IPv6 addresses to “WIN7 IPv6 address” (<i>Fe80::4014:ae92:e379:d2d1</i>)
11	2017/05/30 21:50:02	Sudo NS6 -i enp0s3 -d <i>Fe80::a00:27ff:fe00:f725</i> -F 1200	<i>FF02::1</i> -t	SI6	Send NS messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
12	2017/05/30 21:54:34	Sudo RA6 -i enp0s3 -d <i>2000::4014:ae92:e379:d2d1</i> -F 2000	-d	SI6	Send NS messages from Fake IPv6 addresses to “WIN7 IPv6 address” (<i>2000::4014:ae92:e379:d2d1</i>)
13	2017/05/30 21:58:24	Sudo RA6 -i enp0s3 -d <i>FF02::1</i> -F 500	-d	SI6	Send RA messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
14	2017/05/30 22:21:13	Flood_router26 Eth0		THC- Toolkit	Send RA messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
15	2017/05/30 22:26:07	Flood_router26 <i>2000::4014:ae92:e379:d2d1</i>	Eth0	THC- Toolkit	Send RA messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
16	2017/05/30 22:31:17	Flood_solicitatie6 Eth0 <i>FF02::1</i>		THC- Toolkit	Send NS messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)
17	2017/05/30 22:34:46	Flood_solicitatie6 <i>2000::4014:ae92:e379:d2d1</i>	Eth0	THC- Toolkit	Send NS messages from Fake IPv6 addresses to “all nodes IPv6 address” (FF02::1)

4.2.3 Flow Construction and Features extraction

By combining the traffic collected from the real and virtual networks, a dataset of IPv6 traffic was developed in Comma Separated Value (CSV) format. The traffic packets were filtered to contain instances of NDP traffic as depicted in Figure 4.3, which are the target of this research. The packets were filtered based on the type of IP (i.e. either version 4 or 6), followed by the ICMPv6Type (i.e. if it is Neighbour Advertisement (NA), Neighbour Solicitation (NS), Router Advertisement (RA), or Router Solicitation (RS)). Next, the CSV file was imported in Database and traffic packets were labelled into normal or Replayed or DDoS, using prior information about attack injection.

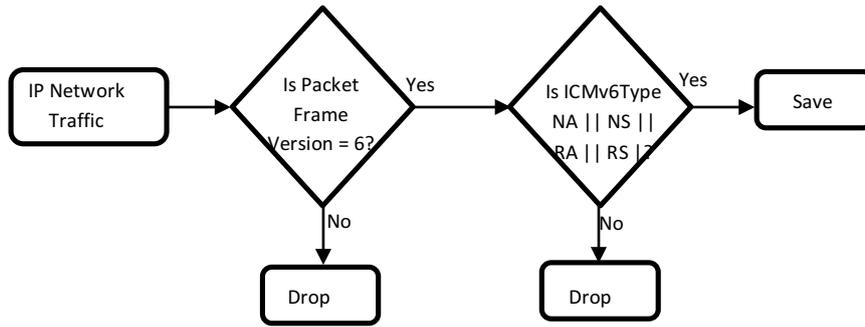


Figure 4.3: Flowchart of ICMPv6 packets filtration

After that this file has been opened in Database and this traffic has been labelled into normal or Replayed and DDoS packets based on the pre-known information about the included traffic (normal and attack) as shown in Table 4.2. Also it shows samples from the packets that will be used to construct streams and extract features.

Table 4.2: the file in Database and the traffic (packets) label.

No	Time	Source	Destination	ICMPv6 Type	Length	Traffic Class	Hop Limit	Flow Label	Next Header	Check Sum	Payload Length	Class
157500	25/05/2017 22:22:26	fe80::f59c:5238:a3fc:e1b7	ff02::1:ff1b:9400	Neighbor Solicitation	86	0x00	255	0x00000	ICMPv6	0x665a	32	Normal
157501	25/05/2017 22:22:26	fe80::ed0d:869:7d1b:9400	ff02::1:ffff:e1b7	Neighbor Solicitation	86	0x00	255	0x00000	ICMPv6	0x1e46	32	Normal
157502	25/05/2017 20:13:42	fe80::c601:4ff:fe28:0	ff02::1	Router Advertisement	118	0xe0	255	0x00000	ICMPv6	0x6c51	64	Normal
157504	25/05/2017 20:14:13	fe80::ff:fe00:6	ff02::2	Router Solicitation	70	0x00	255	0x00000	ICMPv6	0x7b22	16	Replayed
157506	25/05/2017 20:14:13	2000::c601:12ff:fec8:0	2000::ff:fe00:6	Neighbor Advertisement	78	0xe0	255	0x00000	ICMPv6	0xa912	24	DDoS
157507	25/05/2017 20:14:13	fe80::c601:12ff:fec8:0	fe80::ff:fe00:6	Neighbor Advertisement	78	0xe0	255	0x00000	ICMPv6	0x0d90	24	DDoS
157508	25/05/2017 20:14:13	fe80::c601:12ff:fec8:0	fe80::ff:fe00:6	Neighbor Advertisement	78	0xe0	255	0x00000	ICMPv6	0x0d90	24	Replayed

Classification technique aims to build a model by learning the behaviours from the given training dataset then test the model based on the testing dataset. The datasets must be represented using a set of qualifying features to allow the techniques to easily and effectively learn and model the behaviours (Bishop, 2006).

Generally, each classification problem has a set of defined features used for representing the problem's datasets. Based on these features, different research studies and experiments are conducted to enhance one or two paths; either selecting a subset of these features or improve the applied classification algorithms. For example, DARPA IPv4 dataset has defined a set of 42 different features to be used to represent the dataset for intrusion detection purpose. These features have been used in different research studies of IPv4 intrusion detection in the mentioned two paths (McHugh, 1998).

A major development of intrusion detection data sets existed in the Defence Advanced Research Project Agency (DARPA), US. This organization has funded numerous projects in the last few decades. DARPA sets out: to generate an intrusion-detection evaluation corpus to be shared by researchers, to evaluate IDS, to include a varied range of attacks and to measure both attack-detection rates and false-alarm rates. However, NDP-DDoS attacks are unable to use the existing features of IPv4 attacks (such as DARPA's features) due to IPv6 and IPv4 header differences. Moreover, the existing IDSs for IPv6 have used non-qualified features which have misclassification problem of the attacks. Therefore, there is a need to identify an initial set of features for these attacks to be the base for the attacks detection in this research and other researchers. Based on the literature to identify a new set of features, domain knowledge of the attacks is used such as in the DARPA dataset and several other researchers. To validate any set of the features, feature ranking techniques are applied and a subset of the most related features are selected. Based on the domain knowledge of the NDP Replayed and DDoS attacks, several features are specified to be added to the streams and representing the datasets which are IPv6Src: source address

1. IPv6Dst: destination address.
2. NDPTYPE feature is the first extracted feature to help the classifier to differentiate between the attacks types.
3. Packets Number feature: number of the sent packets within the stream.
4. Bytes Number feature: number of bytes sent from the source to the destination within the time interval.
5. SAT (Stream Active Time) feature: the active time of the stream between the source and destination. SAT is one of the most used and selected features in IPv4 datasets such as DARPA.
6. Bytes Ratio feature: the ratio of the sent bytes within the SAT time. It's calculated by dividing the number of sent bytes (Bytes Number) over the active time (SAT).

Several attacking tools (such as THC-toolkit, Si6, Scapy) have been proposed to tackle IPv6 network by different attacks includes NDP Replayed and DDoS attacks (Caicedo et al., 2008). These tools have successfully attacked IPv6 networks as shown in different research (SAAD, R., et al., 2014) and (Hauser, 2006). Attackers normally generate (send) their malicious traffic using these tools (by executing one command), which generate packets with almost the same header attributes. However, these header attributes are different when normal users are communicating, they send normal packets with different value of packets header attributes (such as packet length, traffic class, hop limit, etc.). Therefore, having features that model (represent) these cases will help to differentiate between NDP Replayed, DDoS malicious streams and normal streams. Based on this assumption, the rest of the features are chosen. The selected features are binary features, have two values: either 1 means the stream has different values of the packet header attribute or 0 means the stream has the same packet header attribute. The selected features are as follows:

7. Same_Length feature: 0 if all the stream's packets have the same length header attribute, 1 if at least one packet has different length (L_Diversity).
8. Same_Traffic_Class feature: 0 if all the stream's packets have the same traffic class header attribute, 1 if at least one packet has traffic class (TC_Diversity).
9. Same_hop_limit feature: 0 if all the stream's packets have the same hop limit header attribute, 1 if at least one packet has hop limit (HL_Diversity).
1. Same_Flow_Label feature: 0 if all the stream's packets have the same Flow Label header attribute, 1 if at least one packet has Flow Label (FL_Diversity).
2. Same_Next_Header feature: 0 if all the stream's packets have the same Next Header header attribute, 1 if at least one packet has Next Header (NH_Diversity).
3. Same_Checksum feature: 0 if all the stream's packets have the same Checksum header attribute, 1 if at least one packet has Checksum (CS_Diversity).
4. Same_Payload_Length feature: 0 if all the stream's packets have the same Payload Length header attribute, 1 if at least one packet has Payload Length (PL_Diversity).
5. SH: (source history): represents the history of the IPv6 source address. In other words, it shows how old the IPv6 address is in the network. It is calculated by subtracting the current time from the first seen time of the IPv6 address. This feature is used to detect fake IPv6 address, which will have 0 value since they just appeared in the network.

The stages of feature extraction, malicious streams labelling and development of IDS using machine-learning algorithms are presented in Algorithm 3.

Algorithm 3: Feature Extraction

Let $X \in \{CSV_1 + CSV_2\}$

Let F be a set of features

$\{ICMP_Type, Packet_Number, Byte_Number, Destination, SAT, Byte_Ratio\}$

Find $X_f \in F$

where X_f is a set of all extracted features from X.

Let $R = \{0, 1\}$ where 0= Normal, 1= Attack

$\forall r \in R, \text{ if } r = 0 \Rightarrow \text{destination} \in \{\text{destination}_x \text{ to } \text{destination}_y\}$

$\text{If } r = 1 \Rightarrow \text{destination} \in \{\text{destination}_{\text{fake}} = \text{destination}_x \text{ to } \text{destination}_y \ \& \ \text{Time}_{\text{fake}} = \text{Time}_x \text{ to } \text{Time}_y\}$

Let ML be a set of machine learning algorithms

Pass X_f to $ML_{\text{algorithm}} \Leftrightarrow$

$ML = 0, \text{ where } X_f \mid ML = 1 \text{ where } X_f$

The next phase has been applied to build the streams based on its definition and extract the features that are mentioned previously. This phase is applied using MYSQL queries applied on the traffic to convert the packets into streams with several features for each stream.

$$S6_{NDP} = (IPv6Dst, NDP \text{ type}, T)$$

The query, which is used in MySQL to build the stream, is as follows:

```
INSERT INTO LSFS
```

```
SELECT source
MIN( `ndp_packet_traffic_the attacks`.`Time` ) AS FirstSeen
MAX( `ndp_packet_traffic_the attacks`.`Time` ) AS LastSeen
FROM `ndp_packet_traffic_the attacks`
GROUP BY source;
SELECT No,
```

```

        ndp_packet_traffic_the_attacks`.Source, Destination, ICMPv6Type, COUNT(No) as PacketsNumber,
SUM( `length`) as BytesNumber, (max( `ndp_packet_traffic_the_attacks`.`Time`) - min( `ndp_packet_traffic_the
attacks`.`Time`)) as SAT
SUM( LENGTH) / (max( `ndp_packet_traffic_the_attacks`.`Time`) - min( `ndp_packet_traffic_the
attacks`.`Time`))) as BytesRatio
IF( MIN( `ndp_packet_traffic_the_attacks`.`length`) =MAX( `ndp_packet_traffic_the_attacks`.`length`), 0, 1) as
L_Diversity
IF( MIN( `ndp_packet_traffic_the_attacks`.TrafficClass ) =MAX( `ndp_packet_traffic_the
attacks`.TrafficClass ), 0, 1) as TC_Diversity
IF( MIN( `ndp_packet_traffic_the_attacks`.HopLimit ) =MAX( `ndp_packet_traffic_the_attacks`.HopLimit ), 0,
1) as HL_Diversity
IF( MIN( `ndp_packet_traffic_the_attacks`.FlowLabel ) =MAX( `ndp_packet_traffic_the_attacks`.FlowLabel ),
0, 1) as FL_Diversity
IF( MIN( `ndp_packet_traffic_the_attacks`.NextHeader ) =MAX( `ndp_packet_traffic_the_attacks`.NextHeader
), 0, 1) as NH_Diversity
IF( MIN( `ndp_packet_traffic_the_attacks`.checksum ) =MAX( `ndp_packet_traffic_the_attacks`.checksum ),
0, 1) as CS_Diversity
IF( MIN( `ndp_packet_traffic_the_attacks`.PayloadLength ) =MAX( `ndp_packet_traffic_the
attacks`.PayloadLength ), 0, 1) as PL_Diversity ,(min( `ndp_packet_traffic_the_attacks`.`Time`) -
lsfs.FirstSeen) as SH, Class, MClass.

FROM `ndp_packet_traffic_the_attacks`
INNER JOIN lsfs
ON lsfs.IPSrc = `ndp_packet_traffic_the_attacks`.Source
GROUP BY `ndp_packet_traffic_the_attacks`.Source, Destination, ICMPv6Type, Class, MClass,
`ndp_packet_traffic_the_attacks`.`Time` DIV 5
ORDER BY No ;

```

The dataset used for the model development contained 15 features excluding the MClass feature. The features include Source, Destination, ICMPv6Type, SAT, SH, ByteNumber, PacketsNumber, L_Diversity, TC_Diversity, and HL_Diversity, among others. After representing the traffic in the form of streams with the features for each stream, the traffic is use for the classifications. Table 4.3 shows example of the streams after constructing them with features.

Table 4.3: Example of the streams after constructing them with features.

Source	Destination	ICMPv6Type	Packets Number	Bytes Number	SAT	Packets Ratio	L_ Diversity	TC_ Diversity	HL_ Diversity	FL_ Diversity	NH_ Diversity	SH	CS_ Diversity	PL_ Diversity	Class
fe80::515e:f51b:609e:b916	ff02::1:ffe8:b027	Neighbor Solicitation	2	180	1	180	1	0	1	0	1	0	0	1	Normal
fe80::587:61ad:6be8:b027	ff02::2	Router Solicitation	2	132	2	66	1	0	0	0	0	0	1	1	Normal
fe80::515e:f51b:609e:b916	ff02::1:ffe8:b027	Neighbor Solicitation	2	180	1	180	1	0	1	0	1	0	0	1	Normal
fe80::3ac9:86ff:fe49:4b0e	ff02::2	Router Solicitation	2	132	4	53	1	0	0	0	0	1	1	1	Replayed
fe80::3ac9:86ff:fe49:4b0e	ff02::2	Router Solicitation	2	132	3	44	1	0	0	0	0	0	1	1	Normal
fe80::515e:f51b:609e:b916	ff02::1:ffe8:b027	Neighbor Solicitation	2	180	1	180	1	0	1	0	1	0	0	1	Normal
fe80::587:61ad:6be8:b027	ff02::2	Router Solicitation	2	132	1	132	1	0	0	0	0	0	1	1	Normal
fe80::515e:f51b:609e:b916	ff02::1:ffe8:b027	Neighbor Solicitation	2	180	1	180	1	0	1	0	1	0	0	1	Normal

fe80::587:61ad:6be8:b027	ff02::2	Router Solicitation	3	202	4	50.5	1	0	0	0	0	0	1	1	Normal
fe80::587:61ad:6be8:b027	ff02::2	Router Solicitation	2	132	4	33	1	0	0	0	0	0	1	1	Normal
fe80::515e:f51b:609e:b916	ff02::1:ffe8:b027	Neighbor Solicitation	2	180	1	190	1	0	1	0	1	1	0	1	DDoS
fe80::587:61ad:6be8:b027	ff02::2	Router Solicitation	2	132	1	132	1	0	0	0	0	0	1	1	Normal
fe80::d109:5f04:a0bd:e4a8	ff02::2	Router Solicitation	3	202	4	50.5	1	0	0	0	0	0	1	1	Normal
fe80::d109:5f04:a0bd:e4a8	ff02::2	Router Solicitation	2	132	4	33	1	0	0	0	0	0	1	1	Normal
fe80::587:61ad:6be8:b027	ff02::2	Router Solicitation	3	202	4	50.5	1	0	0	0	0	0	1	1	Normal
fe80::587:61ad:6be8:b027	ff02::2	Router Solicitation	2	132	4	40	1	0	0	0	0	0	1	1	Replayed
fe80::515e:f51b:609e:b916	ff02::1:ffe8:b027	Neighbor Solicitation	4	360	1	360	1	0	1	0	1	1	0	1	DDoS
fe80::515e:f51b:609e:b916	ff02::1:ffe8:b027	Neighbor Solicitation	2	180	0	220	1	0	1	0	1	1	0	1	DDoS

The traffic file in this stage is ready for the classification step; also it is loaded to train with the various machine learning models.

4.2.4 ICMPv6Type Analysis

This section discusses the ICMPv6Type and its effectiveness on the IPv6 network flow. Discussing this feature provides meaningful insights into the behaviour of attacks and thus aids the classifier to distinguish among the attack types and it further helps to interpret the result of the classification process. All NDP attacks are assumed to be from one NDP type except the Replayed attack type.

The generated dataset used consisted of five different types of ICMPv6, namely: NS, RS, NA, RA and RM. The NS consists of 64,232 instances, the RS consists of 17,697 instances, NA consists of 23,915 instances, RA consists of 17,566 instances and lastly, the RM type consists of 26 instances as depicted in Figure 4.4 and represented Table 4.5.

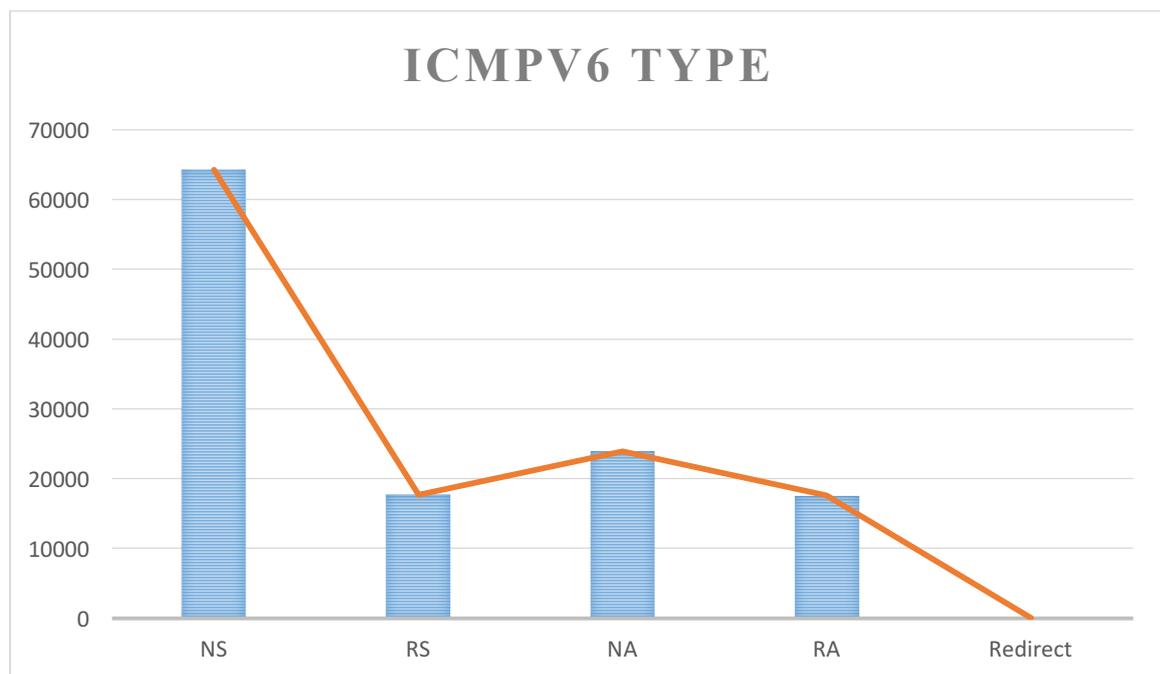


Figure 4.4: ICMPv6 Type Data Distribution.

As seen in Figure 4.4, the RM of ICMPv6 is too scanty as compared with other types. For the purpose of providing information to the classification algorithms, the distribution of the ICMPv6Type with respect to the class label is shown in Figure 4.5.

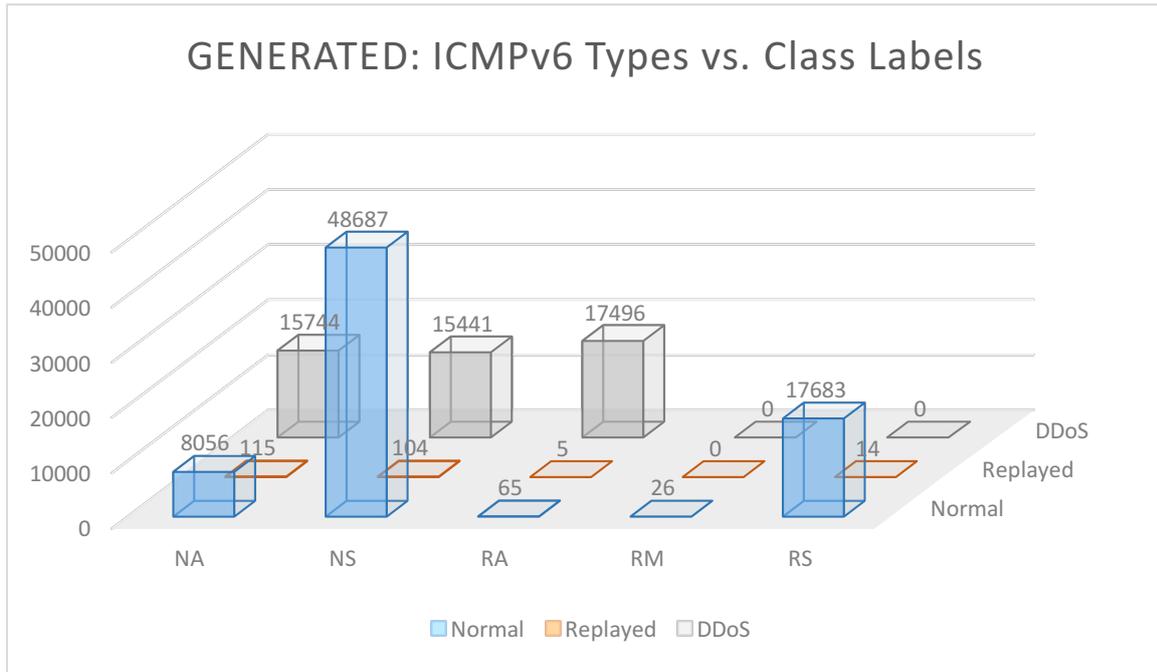


Figure 4.5: ICMPv6Type data distribution with respect to the class label.

From Figure 4.5, and with respect to Table 4.6, it is seen that all Redirect ICMPv6 types are present in normal network traffic and they are completely absent in both DDoS and Replayed attack types. More so, the RS ICMPv6 type is majorly present in normal network traffic, scarcely present in the Replayed attack, and completely absent in DDoS. Both NS and NA ICMPv6 type are densely present in the normal and DDoS network traffic, and significantly present in the Replayed attack type. Classifiers use this information in order to achieve a higher detection rate for each class label.

Further analysis into the details depicted in Figure 4.5, is to know the actual number of instances belonging to each category. Thus, Table 4.7 presents the actual number of ICMPv6 types belonging to each class label and Figures 4.6 (a) and 4.6 (b) represent this table pictorially.

Table 4.7: ICMPv6 type distribution with respect to class label.

Class Label	NA	NS	RA	RS	Redirect	Total
Normal	8056	48687	65	17583	26	74,517
Replayed	115	104	5	14	0	238
DDoS	15744	15441	17496	0	0	48,681

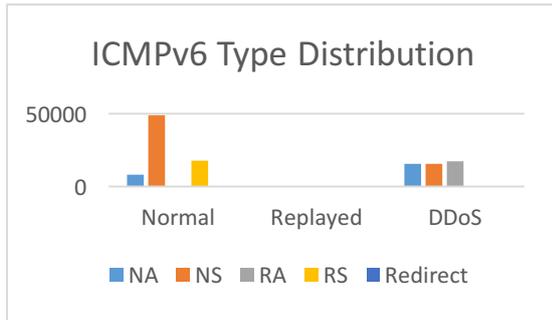


Figure 4.6 (a): ICMPv6 distribution (Higher Axis Unit)

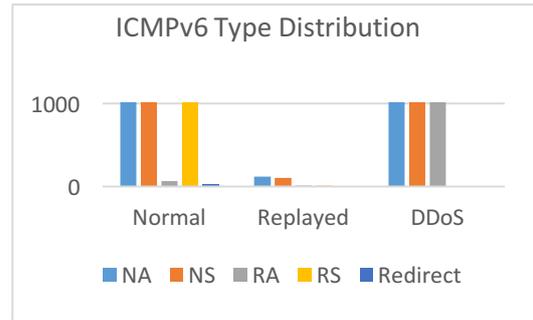


Figure 4.6 (b): ICMPv6 distribution (Lower Axis Unit)

4.3 Summary

Research into Intrusion Detection Systems for detecting DDoS attacks in ICMPv6 is in its infancy compared with IPv4. Machine learning algorithms have been widely applied in developing such kinds of IDS. This chapter presents IDS for detecting DDoS and Replayed attacks, as well as identifying Normal packets in NDP type network traffic by making use of Locally Weighted Learning techniques for classification. The proposed model consists of five phases aiming to detect the NDP (Replayed and DDoS attacks). The first phase starts with capturing the traffic from the network. Moreover, it filters the non-IPv6 traffic, which is out of the research scope (IPv4). The second phase extracts the required attributes to build the streams and extract the features from the filtered packets. The third phase builds the streams based on the Stream of IPv6 NDP ($S_{6\text{NDP}}$) definition as well as it extracts the features for each stream. The fourth phase aims to evaluate the selected features and excludes any unrelated features using the ranking algorithm. The last phase is to apply a machine learning classifier to the built streams' datasets with the selected features in order to build a detection model.

CHAPTER 5

EVALUATION AND ANALYSIS

5.1 Introduction

This chapter focuses on the experimental results, the method of evaluating the experiments, which implies evaluating the effectiveness and efficiency of the proposed framework. With respect to previous chapters, the detections of NDP DDoS and Replayed forms of attack are essential in this study, and as such, the results of the experiments will be discussed accordingly. In the establishment that the proposed framework achieves its aim, the experimental results were analysed using various standard performance metrics and thus compared with some closely related works to determine the overall performance and effectiveness of the developed framework.

This chapter is organized as follows: locally weighted learning model results and discussion are detailed in Section 5.2, performance evaluation metrics are discussed in Section 5.3, evaluating the classification models is discussed in Section 5.4, evaluation by comparison with related current researches is discussed in Section 5.5 and lastly a summary and discussion is presented in Section 5.6.

5.2 Locally Weighted Learning Model Results and Discussion

IDS models for NDPv6 attacks are developed with the data serving as input for locally weighted learning of all classifiers as a base learner. The classifiers used as base learners are as follows: Bayesian Network (BN), Decision Tree (DT), and Naïve Bayes (NB). As discussed in previous chapters, these selected base learners possess distinct learning methods (and are quite popularly used in developing IDS) which in turn provides heterogeneity to the implementation of the locally weighted learning method as used in this work. More so, the base learners provide simple, robust, and understandable models unlike black-boxed models produced by some other machine learning algorithm which are not quite interpretable in human's knowledge.

The dataset used in this research was network traffic represented in a flow-based method. As previously mentioned, the main priorities in the research are low computation cost, less time

taken, and avoidance of overfitting while aiming to build a robust model with high predictive power. Following the antecedent of Elejla et al., (2018) research work that firstly produced novel flow-based features for representing packet, the need for creating a new flow-based network data for this research work is grounded upon this research's data being bespoke and also the increment of available flow-based dataset for research purposes by others researchers so as to create an environment where comparative analysis and evaluated can be made using several flow-based IPv6 datasets.

5.3 Performance Evaluation Metrics

The measurement of the effectiveness of a classification model after categorising instances to their various predefined labels is referred to as performance evaluation. In light of this, the performances of the classification models developed were all measured using the following metrics (David & Thomas 2019; Elejla et al. 2016; Mabayoje et al. 2016).

- i- Detection Rate: this metric is often referred to as accuracy. In this study, it is the overall rate at which a model can detect all three class labels (i.e. Normal, Replayed, and DDoS). It is the percentage of test instances that were correctly classified. It is calculated as shown in Equation 5.1.

$$Detection\ Rate = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (Equation\ 5.1)$$

- ii- True Positive (TP): in this study, true positive refers to the rate at which actual instance of a particular label is classified as that label.

$$TP = \frac{TP}{TP+FN} \quad (Equation\ 5.2)$$

- iii- False Positive (FP): it is the value of incorrectly classified for a specific class label, i.e. the value of an instance categorised as another class label for a given dataset.

$$FP = \frac{FP}{FP+TN} \quad (Equation\ 5.3)$$

- iv- True Negative (TN): it is the value of attack instances that were classified as an attack.

$$TN = \frac{TN}{TN+FP} \quad (Equation\ 5.4)$$

-v- False Negative (FN): it is the value of normal instances that were classified as an attack.

$$FN = \frac{FN}{FN+TP} \quad (\text{Equation 5.5})$$

-vi- Kappa Statistic: Kappa is a chance-corrected measure of agreement between the classifications and the true classes. It is calculated by taking the agreement expected by chance away from the observed agreement and dividing by the maximum possible agreement. A kappa-value greater than 0 means that the classifier is doing better than chance.

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \quad (\text{Equation 5.6})$$

-vii- F-Measure: is a measure of a test's accuracy and is defined as the weighted harmonic mean of the precision and recall of the test. The best value will be at 1 and worst at 0 value.

$$F - \text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (\text{Equation 5.7})$$

-viii- Time Taken: this is the record of the number of seconds taken by the classification algorithm to build its model. The time was measured and recorded in *seconds*.

Using these metrics, the performances of the models developed using the selected classifiers as discussed in previous chapters will be benchmarked.

5.3.1 Confusion matrix

This is a tabular presentation of the classification results. It measures the performance of a classification model on a set of test data for which the true values are previously known (Niyaz et al. 2016). Though it is most suitable for binary classification, it is used in this study to depict the distribution of predicted outcomes.

Thus, since the dataset in this study has three class labels, the confusion matrix was adopted and re-defined to suit this study as depicted in Table 5.1.

Table 5.1: Confusion Matrix for this study

Classes	Normal	Replayed	DDoS
Normal	TP	FN	FN
Replayed	FP	TN	FN
DDoS	FP	FP	TN

5.3.2 Sensitivity analysis (Replayed and DDoS Attack)

Focusing on the analysis of the attack labels of this study, the dataset consists of the DDoS and Replayed attack types in its class attribute. These attack labels represent this kind of attack on a typical IPv6 network.

Sensitivity is the fraction of relevant instances that have been retrieved over the total number of relevant instances (David & Thomas 2019). It is calculated as seen in Equation 5.2:

Using Equation 5.2, the sensitivity value of each model per attack type is calculated, recorded and discussed. In addition, the TP and FP values for each attack type (Replayed and DDoS attacks) were also computed.

5.4 Results and Discussion of Base Classifier Models on the Original Dataset

The models developed in this section were all developed using the same 10-fold cross validation model development techniques, the machine learning algorithms were implemented having selected them from different ML algorithm family for the purpose of heterogeneity. The same of heterogeneity was to ascertain which base classifier will be selected for the locally weighted learning method implementation. More so, the dataset used was the original data with all sixteen features. This was carried out in order to investigate the importance of feature selection technique and to thereby compare the result of the developed method using the original dataset and the reduced dataset.

The Table 5.2, the Naïve Bayes algorithm produced the least performing model with an overall accuracy of 63.5% accuracy while the Bayesian algorithm produced the better performing model of 83.5% accuracy. Random Forest algorithm is an ensemble of decision tree algorithm,

it produced a model that had a tied performance with the Decision Tree algorithm itself – accuracy of 84.3% which is the best accuracy produced by the base classifiers. The poor performance of Naïve Bayes is tied to the fact that it treats features independently (Salih, Ma, & Peytchev, 2015) and determines the occurrence of an event based on the conditional occurrence of another event (Bhosale & Ade, 2014) which is not the case given the dataset used in this research work as most of the features are not independent of each other. The Bayesian Network algorithm on the other hand was able to produce its high performing model because it considers the relationship among the variables given in the data (Blum, Hopcroft, & Kannan, 2018) while modelling and tentatively making prediction.

Table 5.2: Result of Base Classifiers on the Original Dataset

Algorithm	Accuracy	Type	TP Rate	FP Rate
Decision Tree Algorithm	84.3	Normal	0,93	0.33
		Attack	0.66	0.06
Naive Bayes Algorithm	63.5	Normal	0.46	0.03
		Attack	0.96	0.53
MLP Algorithm	75.3	Normal	0.75	0.23
		Attack	0.46	0.04
Random Forest Algorithm	84.3	Normal	0,93	0.33
		Attack	0.66	0.06
Bayesian Algorithm	83.5	Normal	0.84	0.18
		Attack	0.81	0.15

In essence, all models fitted by the various algorithms requires improvement as they are below 90% accuracy and some model had high false alarm rate. Therefore, three base learners were selected for improving using this research model vis-a-vis the better performing model (i.e. Bayesian) and least performing model (i.e. Naïve Bayes) as well as the Decision Tree model

(as it performed as good as the ensemble Random Forest model) in order to see if significant improvement can be made on the least performing base classifier and the best performing base classifier. And finally, if improvement can be made on the decision tree model that tied with an ensemble method (i.e. Random Forest model). By so doing, the essentiality of this research work will be evidently seen.

5.5 Feature Selection

The process of feature selection from the original dataset. The original dataset contained 123,436 instances, 3 class labels and 16 features (including the MClass attribute), the selection of optimal features was conducted in order to achieve lower computation cost, less processing time for model development (since we are having a large number of instances), and avoidance of overfitting of the developed model.

Table 5.3: Details of Original Features

S/N	Name of Feature	Data type
1.	Source	Nominal
2.	Destination	Nominal
3.	ICMPv6Type	Nominal
4.	PacketsNumber	Numeric
5.	BytesNumber	Numeric
6.	SAT	Numeric
7.	BytesRatio	Numeric
8.	L_Diversity	Binary
9.	TC_Diversity	Numeric
10.	HL_Diversity	Binary
11.	FL_Diversity	Numeric
12.	NH_Diversity	Binary

13.	CS_Diversity	Binary
14.	PL_Diversity	Binary
15.	SH	Numeric
16.	MClass	Nominal

As seen in above Table 5.3, the features of the dataset revealed the flow-based representation of the captured network traffic, as well as the characteristics of each feature. To select best features, attributes with constant labels, used for indexing and those with binary labels were removed considering their redundancy and their potential impact of increasing computational cost. More so, the information gain feature selection algorithm was used in conjunction with the rank search method to finally identify optimal features that will not produce an overfitted model and will also result in the development of the models with high detection rate.

Using the aforementioned feature selection algorithm, the features were ranked as seen in following Table.

Table 5.4: Information Gain Attribute Evaluator and Ranker Search Method Feature Ranking

S/N	Name of Feature	Score	Feature Index	Rank
1.	Source	0.98465	1	1
2.	SH	0.9731	8	2
3.	Destination	0.96736	2	3
4.	ICMPv6Type	0.37018	3	4
5.	BytesNumber	0.26729	5	5
6.	PacketsNumber	0.15822	4	6
7.	SAT	0.14115	6	7
8.	BytesRatio	0.00159	7	8

Table 5.4 indicates that the Source and SH features are the best two features with score values of 0.98465 and 0.9731, respectively, along with its ability to supply high information that may lead to the development of an overfitted model, as both such features were removed while others were considered to produce robust models.

To Summarise, of the original sixteen features, seven features were selected for conducting experiments. The selected features were used to create a subset dataset from the original dataset and served as the input for the classification algorithms during the model development phases. As mentioned earlier, the importance of the resulting seven features lies in the facts that redundant and binary values features were removing in order to reduce computational time and potential impairment of model performance and also features supplying high information that may be lead to overfitting were also removed in over to obtain properly fitted models.

5.6 Results and Discussion of Locally Weighted Learning Models on Reduced Dataset

IDS models for NDPv6 attacks are developed with the data serving as input for locally weighted learning of all classifiers as a base learner. The classifiers used as base learners are as follows: Bayesian Network (BN), Decision Tree (DT), and Naïve Bayes (NB). As discussed in previous chapters, these selected base learners possess distinct learning methods (and are quite popularly used in developing IDS) which in turn provides heterogeneity to the implementation of the locally weighted learning method as used in this work. More so, the base learners provide simple, robust, and understandable models unlike black-boxed models produced by some other machine learning algorithm which are not quite interpretable in human's knowledge.

The dataset used in this research was network traffic represented in a flow-based method. The dataset was newly created for this research work as discussed in previous chapter. Following the antecedent of Elejla et al., (2018) research work that firstly produced novel flow-based features for representing packet, the need for creating a new flow-based network data for this research work is grounded upon this research's data being bespoke and also the increment of available flow-based dataset for research purposes by others researchers so as to create an environment where comparative analysis and evaluated can be made using several flow-based IPv6 datasets.

The importance of the resulting seven features lies in the facts that redundant and binary values features were removing in order to reduce computational time and potential impairment of

model performance and also features supplying high information that may be lead to overfitting were also removed in over to obtain properly fitted models.

5.6.1 Dataset Description

The models developed by the means of Locally Weighted Learning method for the selected base learners (BN, DT and NB) were all evaluated using the metrics discussed in the previous sections. Thus, the results of each model are discussed in this section.

Quick recall of the data distribution as depicted in Figure 5.1 shows that there were 74,517 Normal instances, 238 Replayed attacks, and 48,681 DDoS attacks, summing up to 123, 436 instances.

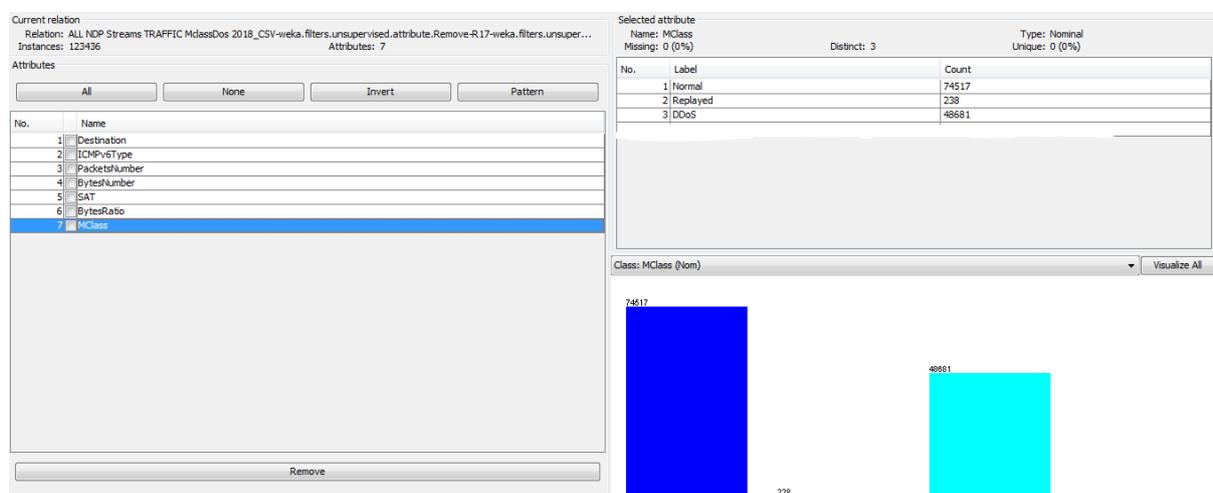


Figure 5.1: Normal and attacks label distribution.

More so, a pictorial representation of the research model is being depicted in Figure 5.2. This represents the step by step procedure taken while conducting this research work.

5.6.2 Model Development using Cross-Validation Technique

This is a mechanism in machine learning that trains and tests an algorithm by dividing the dataset into random N-partitions. The N partitioned dataset is used for testing the algorithm for correct prediction while all other partitions are used to train the algorithm. The training and testing phases are repeated iteratively for N times.

Conventionally, the value for N is mostly set to be 10 (Jung and Hu, 2015), which was adopted in this research work. Thus, the first 9 partitioned datasets train the algorithm while the other

group is used for testing. The process is repeated until all groups are covered and the performance is measured as the aggregate of all the N-folds, i.e. in this research 10-folds stratified cross validation (CV) is utilised.

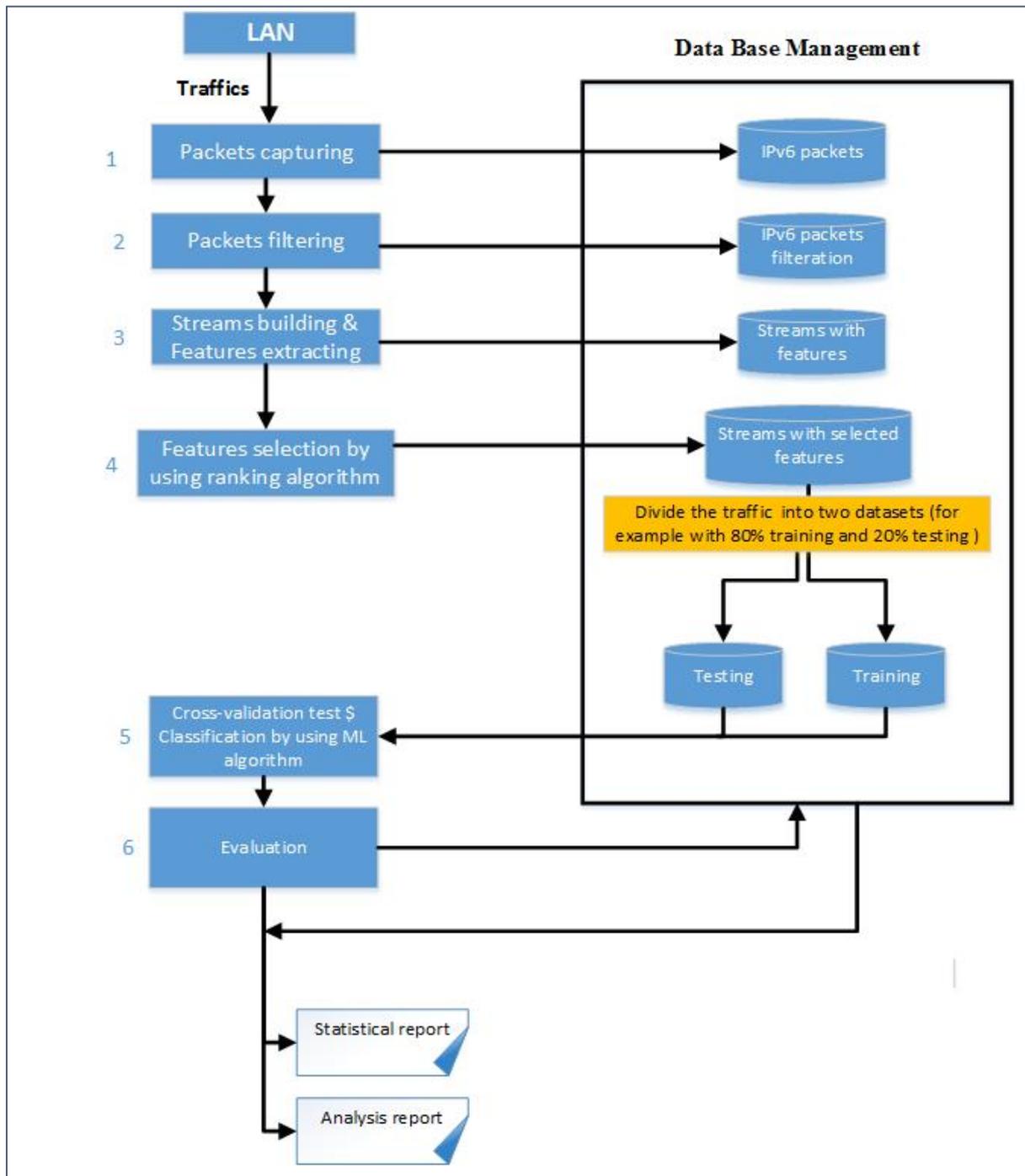


Figure 5.2: Research model

5.6.3 Evaluation of Locally Weighted Learning Models

This section presents the performance evaluation of the all developed LWL models as well as the comparative analysis of the models developed by this research work.

5.6.3.1 Evaluation of LWL-BN's Model

BN was used as the first base learner for the LWL algorithm in order to learn from the flow-based dataset and build its IDS model as shown in Figure 5.2.

The developed model achieved a very high overall accuracy of 96.48%. In total, the model correctly classified 119,098 of 123,436 instances and displayed only 4,342 incorrectly classified instances, indicating a very low false positive value. The results after evaluating the performance of the model using the mentioned metrics are presented in Table 5.5.

Table 5.5: Performance Evaluation of LWL-BN's model.

Model \ Metrics	LWL – BN
Detection Rate	96.4825
True Positive	0.9421
False Positive	0.0004
True Negative	0.999
False Negative	0.0579
Kappa Statistic	0.9282
F-Measure	0.967
Time Taken	0.2 sec

As shown in Table 5.5, LWL-BN's model achieved a detection rate of approximately 97% and a kappa value of 0.9282. Overall, (considering Normal as normal and both Replayed and DDoS as an attack), the TP value of the model was 0.9421 while the FP value of the model was

0.0004. Also, the TN value of the model was 0.99, its FN value was 0.0579, achieving f-measure value of 0.967 with 0.2secs to build the model.

The confusion matrix for this model is shown in Table 5.6, illustrating the correct and incorrect classification of instances for each class label.

Table 5.6: Confusion Matrix of LWL-BN’s Model

Classes	Normal	Replayed	DDoS
Normal	70204	785	3582
Replayed	18	217	3
DDoS	5	0	48676

From Table 5.6, this model achieved a very high detection rate of 99.98% for DoS attacks as it correctly classified 48,676 of 48,681 ‘DDoS’ instances, 94.2% detection rate for ‘normal’ traffic as it correctly classified 70,204 of 74,517 ‘Normal’ instances and 91.17% for detecting 217 of 238 ‘replayed’ type of attack. The high performance achieved by this model is as a result of the feature selection process which produced strong indicative features without noise or redundant features for the machine learning algorithm to develop such a predictive model during the model development phase.

More so, the sensitivity, true positive and false positive for each attack label were computed and depicted in Table 5.7.

From Table 5.7, the Replayed attack had a sensitivity value of 0.912 while the DDoS attack sensitivity value equalled 1. For the true positive value, Replayed attack label had 0.912 while DDoS had 0.9998, and lastly, the false positive value for the model on Replayed attack label was 0.006 and 0.047 for DDoS attack label.

Table 5.7: Sensitivity, True and False Positive values for LWL-BN

Attack Labels Metrics	Replayed	DDoS
True Positive Ratio	0.912	0.9998
False Positive Ratio	0.006	0.047
Sensitivity	0.912	0.9998

5.6.3.2 Evaluation of LWL-DT's Model

Using the decision tree classification algorithm as the base learner, a locally weighted learning method was developed. The J48 decision tree classification algorithm was used from the decision tree family to learn from the flow-based dataset and thus build its IDS model. In previous chapter, the selection of J8 decision tree algorithm as a base learner for the LWL method is that this algorithm requires no domain knowledge in the construction of its tree and also it handles both discrete and categorical variables. An evaluation of the performance model was then calculated. As seen from Table 5.6, the LWL-DT's model achieved a high overall detection rate of approximately 93% having correctly classified 114,868 of 123,436 instances but had a large number of incorrectly classified instances, amounting to 8,572 instances and thus emphasizing the strong predictive capability of the model.

Table 5.8: Performance Evaluation of LWL-DT's model.

Model Metrics	LWL – DT
Detection Rate	93.0557
True Positive	0.885
False Positive	0.013
True Negative	0.999
False Negative	0.1145
Kappa Statistic	0.8614
F-Measure	0.937
Time Taken	0.2sec

As shown in Table 5.8, considering Normal as normal instance and both Replayed and DDoS as an attack, the model yielded a TP value of 0.885, and the FP value of the model was 0.013. Also, the TN value of the model was 0.999, its FN value was 0.1145, achieving f-measure value of 0.937 with 0.2secs to build the model. This model also recorded a kappa statistic value of 0.8614.

The confusion matrix for this model is shown in Table 5.9, which helps to further investigate into the result of this model's performance in order to discover the predictive strength for each class value. From Table 5.9, this model achieved a very high detection rate of 98.61% for DDoS attacks by correctly classifying 48,005 out of 48,681 instances while 671 instances were mis-classified, 87.39% for detecting 'replayed' type of attack and 88.5% detection rate for 'normal' traffic respectively.

Table 5.9: Confusion Matrix of LWL-DT's Model

Classes	Normal	Replayed	DDoS
Normal	65982	784	7751
Replayed	24	208	6
DDoS	611	60	48005

More so, the sensitivity, true positive and false positive for each attack label were computed and depicted in Table 5.10. From Table 5.10, the true positive value for Replayed attack label was 0.874 while DDoS had 0.986, while the false positive value for the model on Replayed attack label was 0.007 and 0.104 for DDoS attack label, and lastly, the Replayed attack had a sensitivity value of 0.874 and was 0.986 for the DDoS attack label.

Table 5.10: Sensitivity, True and False Positive values for LWL-BN

Metrics \ Attack Labels	Replayed	DDoS
	True Positive Ratio	0.874
False Positive Ratio	0.007	0.104
Sensitivity	0.874	0.986

5.6.3.3 Evaluation of LWL-NB's Model

The third base learner, as discussed and justified in the Methodology chapter, is the Naïve Bayes machine learning algorithm which was used to in this research work to develop a locally weighted learning model as opposed to the usual global approximation function. The LWL-NB model achieved an overall detection rate of approximately 96% having correctly classified 118, 532 of 123,436 instances and misclassified 4,908 instances.

Table 5.11: Performance Evaluation of LWL-NB's model.

Model \ Metrics	LWL – NB
Detection Rate	96.024
True Positive	0.935
False Positive	0.01
True Negative	0.99
False Negative	0.065
Kappa Statistic	0.9193
F-Measure	0.964
Time Taken	0.1sec

As shown in Table 5.11, considering Normal as normal and both Replayed and DDoS as attack, the LWL-NB model fits on the dataset achieving at least 96% overall detection rate, having TP value of 0.935, FP value of 0.01, TN value of 0.99, FN value of 0.065, a kappa statistic value of 0.9193, f-measure of 0.964 and with 0.1sec to build the model.

The confusion matrix for this model is shown in Table 5.12, which helps to further investigate into the result of this model's performance in order to discover the predictive strength for each class value.

Table 5.12: Confusion Matrix of LWL-NB's Model

Classes	Normal	Replayed	DDoS
Normal	69648	1341	3528
Replayed	29	206	3
DDoS	443	5	48233

From Table 5.12, the confusion matrix of LWL-NB's model achieved high detection rate of 99.08% for detecting DDoS attacks having correctly classified 48,233 instances, 93.47% accuracy for correctly classifying 69,648 Normal traffic and 86.55% for correctly classifying 206 Replayed type of attack

Table 5.13: Sensitivity, True and False Positive values for LWL-NB

Metrics	Attack Labels	
	Replayed	DDoS
True Positive Ratio	0.87	0.985
False Positive Ratio	0.011	0.047
Sensitivity	0.87	0.985

More so, the sensitivity, true positive and false positive for each attack label were computed and depicted in Table 5.13. From Table 5.13, the true positive value for Replayed attack label was 0.87 while DDoS had 0.985, the false positive value for the model on Replayed attack label was 0.011 and 0.047 for DDoS attack label, and lastly, the Replayed attack had a sensitivity value of 0.87 and was 0.985 for the DDoS attack label.

5.6.4 Summative Evaluation of the developed Locally Weighted Models on the Reduced Dataset.

Locally Weighted Learning algorithm was deployed using linear weighting kernels and 3 neighbours with classifiers serving as the base learner for the implementation of the algorithms. The dataset contained 74,517 normal packets, 238 replayed packets, and 48,681 DDoS packets. BN was used as the first base learner for the LWL algorithm in order to learn from the flow-based dataset and build its IDS model. The developed model achieved a very high overall accuracy of 96.48%. In total, the model correctly classified 119,098 of 123,440 instances and displayed only 4,342 incorrectly classified instances, indicating a very low false positive value. The Decision Tree algorithm was also used as a base learner for the LWL algorithm to learn from the flow-based dataset and build its IDS model. An evaluation of the performance model was then conducted. The Decision Tree model achieved a high overall accuracy of 93%, having

correctly classified 114,868 of 123,440 instances but had a large number of incorrectly classified instances at 8,572. And lastly, Naïve Bayes (NB) was also used as a base learner for the LWL algorithm to learn from the flow-based dataset and build its IDS model. The developed model achieved an overall detection rate of 96.024%.

More so, the LWL-DT model had the least time taken for developing its model – a time of 0.01seconds, as against other two models (i.e. LWL-BN and LWL-NB) which took the same time to build their respective models – 0.02 seconds respectively.

Table 5.14: Comparative analysis of each model’s overall accuracies.

Models Metrics	LWL – BN	LWL – DT	LWL – NB
Detection Rate	96.4825	93.0557	96.024
True Positive	0.9421	0.885	0.935
False Positive	0.0004	0.013	0.01
Time Taken	0.02 seconds	0.01 seconds	0.02 seconds

As shown in Table 5.14, the LWL-BN model achieved the highest detection rate of 96.48% with the lowest false positive value of 0.0004%. LWL-NB model is the next best model, with an accuracy rate of 96.024% and a false positive value of 0.01, while the LWL-DT model had the lowest overall detection rate of 93%. The LWL-DT model also had the highest false positive value of 0.013%. Comparatively, all developed IDSs are capable of detecting DDoS and Replayed attacks based on NDP-based network traffic as well as the detection of anomalies. They all demonstrated strong predictive ability, however, the LWL-BN’s model proved to have the most favourable overall performance. However, the LWL-DT model took the least time to develop a locally weighted IDS model among the three models.

In comparative analysis with the base classifiers models that this research work sought to improve (see Section 5.4), it is not just clearly seen but with evidence that the research model implemented in this research work greatly and significantly improved the performances of the selected base classifiers. Beforehand, it should be stated that the LWL-BN model still took the lead in performances among other developed models due to its underlying characteristic of the Bayesian Network algorithm of not considering data attributes to be independent of each other (unlike the Naïve Bayes algorithm handling data as having independent attributes (Salih et al., 2015)) but it also takes to consideration the relationship among attributes (Blum et al., 2018). More so, the acceptable performance of LWL-DT model is also owed to intrinsic characteristic of the decision tree algorithm that extract relational details from data attribute through the application of entropy (David & Thomas, 2019).

It should be recalled that the base classifier model produced by Bayesian algorithm produced an accuracy of 83.5% whereas the LWL-BN model produced an accuracy of 96.48% – a significant accuracy increase of 12.98%. The least performing Naïve Bayes base classifier model achieved an accuracy of 63.5% but was significantly improved using the LWL method to the accuracy of 96.024% - a massive increase of 33%. And lastly, the decision tree base classifier model which produced an accuracy of 84.5% was also improved as the LWL-DT model achieved a 93% accuracy - a significant accuracy increase of 8.5%.

Table 5.15: Comparative accuracies per attack type.

Model	Accuracies of Classes (%)		
	Normal	Replayed	DDoS
LWL-BN	94.21	91.17	99.98
LWL-DT	88.56	87.39	98.62
LWL-NB	93.47	86.55	99.08

As depicted in Table 5.15, and also considering the DDoS attack type, all models are capable of detecting this attack at a very high rate, the least being 98%. As such, any of the models can be deployed for the purpose of detecting a DDoS attack. Furthermore, the accuracy of the model in classifying normal packets varied with LWL-DT achieving the lowest detection rate of 88.56%, while LWL-NB could identify a normal packet better than LWL-DT with a detection rate of 93.47%. In this category of attack, the LWL-BN model triumphed with the highest detection rate of 94.21%. Also, LWL-BN model outclassed other models in the detection of Replayed attacks. Detecting replayed attack at the rate 91.17%, the LWL-BN model proved to have the strongest predictive capability for this attack type. The LWL-DT model outperformed the LWL-NB model with a detection rate of 87.39% while LWL-NB had the lowest rate of 86.55%. In conclusion, it is safe to infer that all the models are strong enough to classify a DDoS attack, however, the LWL-BN model is more effective in detecting all attack types followed by the LWL-NB model.

Thus, a pictorial representation of all models' accuracies per class is depicted in Figure 5.3.

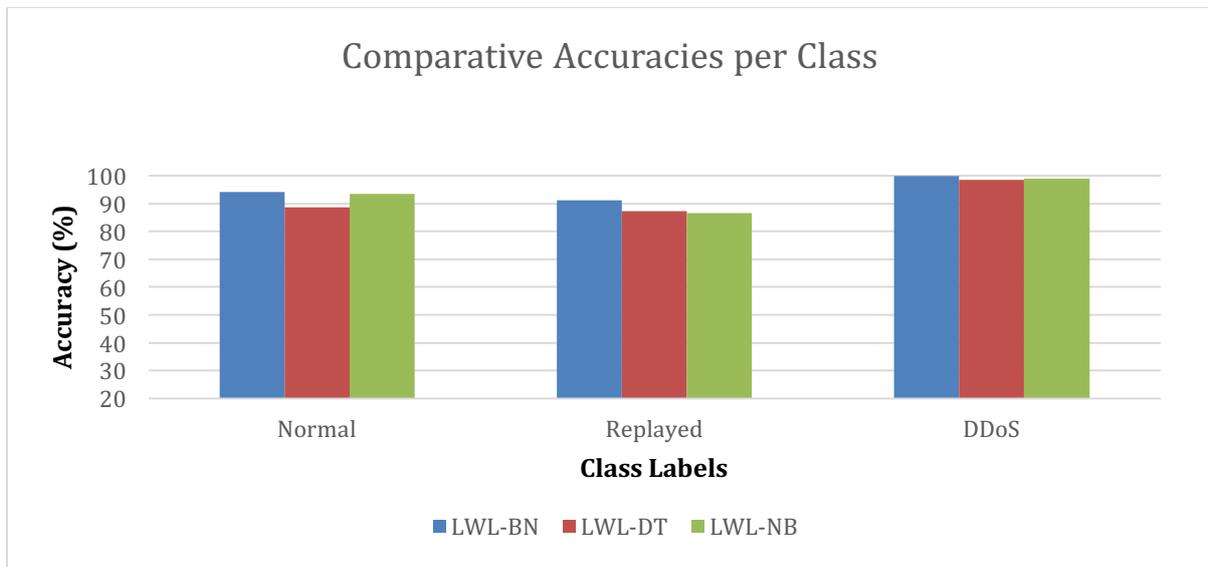


Figure 5.3: Comparative Accuracies of Models per Class Label.

5.7 Evaluation by Comparison with Current Research Studies

The development of the IDS model is a common research ground for several studies, as IDS is one of the essential components in the network and cyber security as a whole. This study is

being evaluated with other closely related works, first with flow-based IDS and then with some other general IDS developed for IPv6 network.

The study of (O. E. Elejla, Anbar, Belaton, & Alijla, 2018) presented the development of several IDS models for detecting ICMPv6-based DDoS attacks having used a flow-based representation of network traffic. The study made use of the following machine learning algorithms: C4.5 decision tree, SVM, Naïve Bayes, KNN, Neural Networks, Random forest, and Conjunctive rules. The models were developed and tested using cross-validation and further tested using a separate test set.

Comparatively, (O. E. Elejla et al., 2018) the best model was outperformed by the models of this study based on the cross-validation method of model development. In their study, the best model, after using the cross-validation method of model development, was the Random forest tree model which achieved an accuracy of 85.67% and consistently had the best accuracy when tested on a separate test set. Whereas, the least performing model in this research study achieved an accuracy of 93%. As a result of these results, the accuracy of 93% achieved by the LWL-DT model in this study is greater than the 85.67% accuracy achieved by the best model in Elejla et al. (2018) study though both research studies used a flow-based representation of network packets and cross-validation method of model development.

The IDS model by Niyaz et al. (2016) was developed using the Stacked Auto-Encoder (SAE) algorithm. The study implemented a multi-vector deep learning method for detecting DDoS. The Niyaz et al. (2016) study evaluated the developed model using accuracy among other metrics, and it achieved an accuracy (equivalent to the detection rate in this study) of 95.65%, whereas all models in this study comparatively outperformed the detection of DDoS, as the lowest detection rate for DDoS produced in this research work was 98.62% which exceeded Niyaz et al. (2016) 95.65% detection rate. In this case, it is evident that developing a locally weighted model (i.e. local approximation function) for detecting DDoS attack type in a typical IPv6 network outperforms a multi-vector deep learning method for detecting DDoS.

This study is compared with the research carried out by Li et al. (2018) which was rooted on detecting and defending against DDoS attack based on deep learning (Bi-directional RNN, LSTM and CNN) algorithms in a typical open Flow-based SDN. The result of the experimental work carried out indicated deep learning models, if correctly implemented, can effectively detect and defend against DDoS with accuracy as high as 99% on the training set and 98% on the test set, having used the ISCX2012 dataset. Comparing the performance of their model on

the test set with our work, it competed well enough with LWL-DT's model but was outperformed by the LWL-BN and LWL-NB models.

5.8 Summary

All models developed in this study achieved a very high detection rate, however, comparatively they outperformed each other as discussed in Section 5.2. More so, aside from the overall detection rate for each developed LWL-model, various detection rates for each class label (Normal, DDoS and Replayed) were presented.

In addition, the developed models were evaluated with existing and current related research studies in various capacities and it was evident that the locally weighted learning method for developing a local approximation function for detecting IPv6 Normal, DDoS and Replayed network packet is a viable technique as it outperformed most existing methods.

In conclusion, it is safe to infer that all deployed models are capable of classifying a DDoS attack at over 95% accuracy; more so, the LWL-BN model is more effective in detecting all attack types as it is closely followed by the LWL-NB model. Succinctly, the experimental results of this research work evidently show that the locally weighted learning method is a viable means to develop Network IDS with an advantage of little time taken for model development.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter contains the general conclusions as derived at the end of the completed work undertaken in this thesis, as well as including the main contributions of this work to the body of knowledge, particularly in the domain of network and cyber security in a typical IPv6 network. More so, in this chapter, recommendations and future works are also being highlighted.

6.1 Conclusion

In previous chapters, the importance of Internet Protocol (IP) addresses had being significantly discussed in regards to our modern society. The numerous advantages derived from the IP network either via Intranet or Internet or even Extranet, as used by people, governmental bodies and agencies, public and private organisations etc, had turned the IP network into one of the core components of our society (Levin & Schmidt 2014). It was established in previous chapters that the IP network's wide usage and popularity started with its version 4 which is being referred to as IPv4, however, the network addresses provided by this IP network were exhausted by human usage and thus the need to provide a viable solution (Shiranzaei & Khan 2015).

Among the numerous solutions provided to the exhaustion challenges faced by IPv4 is the creation and usage of a more robust IP network with large coverage of all humans and devices on earth even now and future generations to come (Frankel & Green 2008). Thus, careful development of IPv6 provides a unique, viable and long-lasting solution to the present IPv4 exhaustion problem. Though IPv6 development is with a security scheme called IPsec unlike IPv4 network, IPv6 has its peculiar security issues which have been discussed extensively in previous chapters (Ahmed, Hassan & Othman 2017).

In this research work, the susceptibility of IPv6 NDP due to the weak nature of its authentication process was discussed, as well as various attacks that are executed based on the highlighted weakness of NDP. The essential task performed by NDP in IPv6, which operates in the link layer for auto configuration of a node's address, made it impossible to remove or replace this protocol for IPv6 network infrastructure during the developmental stage. As such,

finding solutions to combat the vulnerabilities of this critical protocol, i.e. making NDP secure, becomes essential and thus, the focus of this study.

The literature review chapter presented a thorough discussion of core topics on network security with respect to the focus of this study. Essentially, the literature review chapter presented several solutions with respect to making NDP secured. It presented some state-of-art machine learning solutions serving as Intrusion Detection System (IDS) in a typical IPv6 network. The solutions were reviewed accordingly, by highlighting the algorithms used, the particular dataset used for developmental and testing phases, the aim of the developed IDS in particular, the strength of the development solution and as well as the limitation and or future work of each reviewed solution. This review helped in the identification of the research gap, which was investigated in this research.

Through the literature review conducted in this study, various mechanisms for securing a network were identified (having already been designed, developed and deployed previously), these mechanisms include firewalls, security expert system, anti-viruses and artificial intelligence Intrusion Detection System (Hemant, Sarkhedi & Vaghamshi 2013). These mechanisms possess unique strengths and weaknesses, which were, discussed in previous chapters, however, an artificial intelligence IDS mechanism proved to show more strength in network defence. Usually the network data are voluminous in size – thereby making it largely impossible for humans to uncover the underlying structure of data and/or hidden information within the network data which can be used to for the development of optimal security expert systems and fire-wall mechanisms. Also, network attacks are dynamic as attackers continually change their method of executing attacks, thus, rendering anti-viruses useless in the case of new forms of attack which had not being registered in the anti-virus signature repository. However, artificial intelligence shows great strength as typically, it extracts hidden or previously unknown patterns and data structure from (large) data which are then usable for making viable decisions (Mitchell & Vora 2013). This great strength of AI is the reason behind our adoption of this method to provide IDS that secures IPv6 NDP. An artificial intelligence IDS will extract hidden information about previously known attacks from the large deluge of network data to detect such attacks in the future, it will also help to detect new form of attack that shares similarity with previously known attack types and also, an artificial intelligence IDS through learning from a typical normal network packet, will be able to detect if a packet is normal, or an anomaly which is highly useful in detecting a new form of attack that has not

been previously identified as long as it is not normal network traffic (Jaiganesh, Mangayarkarasi & Sumathi 2013).

6.1.1 Comparison with other researches:

Also, through the literature review, it was identified that the representation of network packet determines the performance level of an artificial intelligence IDS. Most developed IDS made use of network data that are based on packet-based representation of the traffic characteristics and corresponding features to develop an IDS model using some machine learning algorithms. This packet-based representation of network traffic is unsuitable and does limit the detection of attacks, particularly DDoS attack as revealed by the recent study of Elejla et al., (2018). Thus, a new method for representing network packets was discovered, implemented, and evaluated against the packet-based representation, this method is referred to as flow-based representation of traffic. In the Elejla et. al., (2018) study, flow-based representation was proven to enable superior performances of all developed artificial intelligence IDS having conducted the experiment on real datasets and resulted in 61% detection rate as the best accuracy achieved through packet-based IDS when evaluated on a separate test set, while the flow-based IDS achieved 85.83% detection rate at its best model. With this proof, this research adopted flow-based representation of network traffic and it can be seen that the detection rate of all developed models was higher than that of the original author of the method.

In line with the research gap identified through the review of existing solutions for securing NDP, this research work presented a framework that detects anomalies on NDP. Particularly, this study worked closely on detecting two types of anomalies, Distributed Denial-of-Service (DDoS) and Replayed types of attack, and also it worked on identifying normal packets. The solution presented in this study is herein named “Flow-Based Locally Weighted Neighbour Discovery Protocol Attacks Detector (FB-LWNDPAD)”. Central to this proposed solution is to secure NDP by using a flow-based method of features extraction from a typical IPv6 network packet, then carefully best-selected features are used to provide usable information for detecting DDoS and Replayed attack types as well as identifying normal packets, within a short time and with reduced computational complexity.

6.1.2 Research Methodology:

The methodology chapter of this research work presented a novel proposed FB-LWNDPAD framework which was implemented and evaluated in this research work. The research methodology adopted the information gain feature selection techniques, and some machine

learning algorithms (i.e. Local weighted learning method, Naïve Bayes, Decision tree and Bayesian Network algorithms). A locally weighted learning method was proposed in order to develop IDS with a local approximation function as a means for lower computational cost, less time taken, and avoidance of overfitting while building a robust IDS with high detection rate.

6.1.3 The Novelty:

Our FB-LWNDPAD is a novel method for detecting anomalies (DDoS and Replayed attack) and to identify normal packets in a typical IPv6 network. It used a locally weighted classifier model, developed using optimal features that were extracted using a flow-based method and processed using feature selection technique. For comparison, three machine-learning algorithms (Bayesian Network, Decision tree and Naïve Bayes) were used to develop the locally weighted classification models. As seen in the results that were discussed in the evaluation methodology, our FB-LWNDPAD models, after being evaluated achieved 93% detection rate for the LWL-DT model, 96% detection rate for the LWL-NB model and lastly, 96.5% detection rate for the LWL-BN model. Overall, all models achieved a very high accuracy (i.e. detection) rate and with low false positive values which proved that the proposed FB-LWNDPAD solution accurately detected DDoS and Replayed anomalies and also, it accurately identified a normal packet of an IPv6 network.

6.2 Contributions

This research aim is to detect anomalies in an IPv6 network thereby securing NDP. Having implemented the framework of this study and thus evaluated the developed models, DDoS and Replayed attacks on NDP are accurately detected and also, normal packets are also being identified accurately using minimal resources and with improved computational time. FB-LWLNDPAD will help network administrators to ably know the nature of network packets according to the predefined category (i.e. normal, DDoS and Replayed packets).

One of the major contributions of this research work to the body of knowledge is the created dataset. The dataset used in this research work is newly created, its network packets are obtained from a typical IPv6 network. A flow-based representation of the extracted IPv6 network was made and then pre-processed into a usable format (i.e. table of rows and columns) for the machine learning algorithms to identify normal and attack packets respectively.

This work contributes to research by revealing the set of new features based on flow-based extraction of features that are capable of detecting anomalies in an IPv6 network. More

importantly, the process of feature selection which was conducted allowed a few sets of flow-based features of a network packet to be used to detect anomalies attacks. These sets of features are relevant – providing useful information to all machine learning used in this study; optimal –providing required information that does not cause the developed model to overfit, and redundant.

Another contribution is the method of learning used while developing the model. Through the literature review, it was established that most machine learning algorithms used in the development of IDS use a global function for approximating the probability of an instance belonging to a predefined class. However, this study introduced a locally weighted learning method for the development of IDS models. This method enables the development of models that are local to the point of query. Contextually, the model developed used the local information available around the nodes to classify as either normal or anomalies (DDoS or Replayed attack).

6.3 Limitation and Future works

This research implemented the FB-LWLNDPAD and it achieved viable results in the detection of NDP anomalies. However, the detection of NDP anomalies is limited to DDoS and Replayed type of attack. It is thus suggested that the future work of this research can be extended into detecting several other forms of NDP attacks, as this current research study only considered Distributed Denial of Service and Replayed types of NDP attacks. Other NDP attacks that may be considered include NA Spoofing, Man-in-the middle, NS Spoofing. Thus, developing IDS for other forms of attack using the proposed research methodology of this study is a viable research venture.

Also, in the future work the testing dataset will be different in terms of the included traffics, where not necessary to have all the included traffics of the training dataset. The testing dataset scenarios are varied by including all the targeted attacks, including some of the targeted attacks, including only normal traffic, including only malicious traffic. Using different types of attacks as well.

The FB-LWLNDPAD framework is also limited in detecting NDP anomalies available in a typical IPv6 network. It is also suggested that this research can be continued in the future to consider other anomalies belonging to other protocols of an IPv6 network, an example of this

protocol is Multicast Listener Discovery Protocol, by adopting the same methodology used in this research. More so, a combination of this implemented framework and signature-based IDS is potentially going to be a good fit and thus, it is also considered as a future work.

This research is covering all the NDP protocols with the five types that are RS, RA, NS, NA and RM for all the operating systems but as a future work this could be extended to cover IP Mobile operating systems.

With IDS, the detection of anomalies as well as profiling of normal packets are possible. However, IDS is only capable of detecting anomalies and reporting to network administration but does not automatically take action to combat the detected anomalies. In this regard, the author considered as a future work the integration of developed IDS with other network monitoring systems that are capable of taking action against detected anomalies. Furthermore using this method to detect the anomalies in Multi-agent IDS.

Lastly, deep learning methods are one of the prevailing methods as applied to solving numerous societal problems facing our modern society. Network security being one of the problems facing our society, the implementation of deep learning methods is also considered as a viable future work. Deep learning had been implemented by Niyaz, Sun & Javaid (2016) to detect DDoS in a SDN environment using Stacked Auto Encoder (SAE) model and it achieved high accuracy of 99.82%, and also by (Aziz & Okamura 2017) who used a FlowIDS approach to detect anomalies in SMTP traffic also using a Deep Learning algorithm; however, using a deep learning method in conjunction with flow-based representation of network traffic for securing NDP, is yet to be carried out and thus considered as work in the foreseeable future as it is more likely to also produce IDS with a high detection rate.

References

- Ahmed, A.S.A.M.S., Hassan, R. and Othman, N.E., 2017. IPv6 Neighbor Discovery Protocol Specifications , Threats and Countermeasures : A Survey. IEEE, pp.18187–18210.
- Al-Ani, M.S. and Haddad, R.A., 2012. IPv4/IPv6 Transition. International Journal of Engineering Science and Technology, 4(12), pp.4815-4822.
- Alaidaros, H., Mahmuddin, M. and Al-Mazari, A., 2011. An overview of flow-based and packet-based intrusion detection performance in high speed networks.
- Aleesa, A.M., Hassan, R. and Kamal, S.U.M., 2016. A rule-based technique to detect router advertisement flooding attack against biobizz web application. Advanced Science Letters, 22(8), pp.1887-1891.
- Anbar, M., Abdullah, R. and Saad, R.M.A., 2016. Review of Security Vulnerabilities in the IPv6 Neighbor Discovery Protocol Review of Security Vulnerabilities in the IPv6 Neighbor Discovery Protocol. Information Science and Applications (ICISA), (June), pp.603–612.
- Anbar, M., Abdullah, R., Al-Tamimi, B.N. and Hussain, A., 2018. A Machine Learning Approach to Detect Router Advertisement Flooding Attacks in Next-Generation IPv6 Networks. Cognitive Computation, 10(2), pp.201–214.
- Anon (2016) Weka : Decision Trees – J48.
- Anon (2017) Machine learning : the power and promise of computers that learn by example.
- Atkeson, C.G., Moore, A.W. & Schaal, S. (1996) Locally Weighted Learning. 1–52.
- Ayodele, T.O. (n.d.) Types of Machine Learning Algorithms.
- Aziz, M.Z.A. and Okamura, K., 2017. Leveraging SDN for Detection and Mitigation SMTP Flood Attack through Deep Learning Analysis Techniques. Ijcsns, 17(10), pp.166–172.
- Balogun, A.O., Balogun, A.M., Sadiku, P.. & Adeyemo, V.E. (2017) Heterogeneous Ensemble Models for Generic Classification. Anale. Seria Informatică. XV, 92–98.
- Barbhuiya, F.A., Biswas, S. and Nandi, S., 2011. Detection of neighbor solicitation and advertisement spoofing in IPv6 neighbor discovery protocol. Proceedings of the 4th

international conference on Security of information and networks - SIN '11, p.111.

Besold, T.R., Garcez, A. d'Avila, Bader, S., Bowman, H., et al. (2017) Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. ArXiv, cs. 1–58.

Bhosale, D. & Ade, R. (2014) Feature Selection based Classification using Naive Bayes , J48 and Support Vector Machine. 99 (16), 14–18.

Bishop, C.M., 2006. Pattern recognition and machine learning, Springer.

Budhathoki, D.R., n.d. Computer Network, 2011.

Caicedo, C.E. and Joshi, J., 2008. Security issues in ipv6 networks. International Telecommunications Research and Education Association (ITERA).

Caicedo, C.E., Joshi, J.B.D. and Tuladhar, S.R., 2009. IPv6 Security Challenges. IEEE Computer Society, pp.36–42.

Chai, K., Hn, H.T. & Cheiu, H.L. (2002) Naive-Bayes Classification Algorithm. Bayesian Online Classifiers for Text Classification and Filtering. 97–104.

Chao, W. (2011) Machine Learning Tutorial.

Chuangchunsong, N., Kamolphiwong, S., Kamolphiwong, T., Elz, R. and Pongpaibool, P., 2014. Performance evaluation of IPv4/IPv6 transition mechanisms: IPv4-in-IPv6 tunneling techniques. International Conference on Information Networking, pp.238–243.

Cisco Systems, 2011. NetFlow Services Solutions Guide, <http://www.cisco.com>.

David, J. & Thomas, C. 2019, 'Efficient DDoS Flood Attack Detection using Dynamic Thresholding on Flow-Based Network Traffic', Computers & Security.

Educba, 2019. What is DDoS Attack?. EDUCBA. [Online]. Available:<https://www.educba.com/what-is-ddos-attack/>.

Elejla, O., Belaton, B., Anbar, M. & Alnajjar, A. 2016, 'A Reference Dataset for ICMPv6 Flooding Attacks', Journal of engineering and applied sciences, vol. 11, no. 3, pp. 476–81.

Elejla, O.E., Anbar, M., Belaton, B. & Alijla, B.O. 2018, 'Flow-based IDS for ICMPv6-based DDoS Attacks Detection', Arabian Journal for Science and Engineering.

- Elejla, O.E., Belaton, B., Anbar, M. and Alnajjar, A., 2016. Intrusion Detection Systems of ICMPv6-based DDoS attacks. *Neural Computing and Applications*, pp.1-12.
- Englert, P. (2012) Locally Weighted Learning. Seminar Class on Autonomous Learning Systems. 1 (1), 1–9.
- Faltin, F. and Kenett, R., 2007. Bayesian Networks. *Encycl. Stat. Qual. Reliab.*, 1(1), p.4.
- Frankel, S. and Green, D., 2008. Internet Protocol Version 6. *IEEE Security & Privacy*, 6(3), pp.1–4.
- Frankel, S. and Krishnan, S., 2011. IP security (IPsec) and internet key exchange (IKE) document roadmap. *Request for Comments, 6071*.
- GNS3. 2017, source: <https://www.gns3.com>
- Goralski, W., 2009. IPv4 and IPv6 Headers. *The Illustrated Network*, pp.165–188.
- Guerra, L., MCGarry, L.M., Bielza, C., Larran, P., et al. (2010) Comparison Between Supervised and Unsupervised Classifications of Neuronal Cell Types : A Case Study.
- Hamed, T., Ernst, J.B. and Kremer, S.C., 2018. A Survey and Taxonomy of Classifiers of Intrusion Detection Systems.
- Hauser V., T.H.C., 2006. Attacking the IPv6 Protocol Suite.
- He, Z., Zhang, T. and Lee, R.B., 2017. Machine Learning Based DDoS Attack Detection from Source Side in Cloud. 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), pp.114–120.
- Hemant, P., Sarkhedi, B. and Vaghamshi, H., 2013. Intrusion Detection in Data Mining With Classification Algorithm. 2(7), pp.3063–3070.
- Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., et al. (2017) Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey. 1–43.
- Hogg, S., Karpenko, J., Miller, D. and Vyncke, E., 2009. IPv6 Security: Information assurance for the next-generation Internet Protocol. Cisco Press.
- Holzinger, A. (2016) Interactive machine learning for health informatics : when do we need the human-in-the-loop ? *Brain Informatics*. 3 (2), 119–131.

- Holzinger, A. (2017) Introduction to MACHine Learning & Knowledge Extraction (MAKE)
Introduction to MACHine Learning & Knowledge Extraction (MAKE).
- Holzinger, A. (2019) From Machine Learning to Explainable AI. 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA). (August 2018), 55–66.
- Hutter, F., Kotthoff, L. & Vanschoren, J. (2019) Automatic Machine Learning: Methods, Systems, Challenges.
- Jaiganesh, V., Mangayarkarasi, S. and Sumathi, P., 2013. Intrusion Detection Systems : A Survey and Analysis of Classification Techniques. 2(4), pp.1629–1635.
- Jain, A. & Rana, J.L. (2016) Classifier Selection Models For Intrusion Detection System (IDS). 4 (1), 1–11.
- Jain, Y.K. (2012) An Efficient Intrusion Detection Based on Decision Tree Classifier Using Feature Reduction. 2 (1), 1–6.
- Jung, Y. and Hu, J., 2015. AK-fold averaging cross-validation procedure. Journal of nonparametric statistics, 27(2), pp.167-179.
- Karim, M. and Rahman, R.M., 2013. Decision Tree and Naïve Bayes Algorithm for Classification and Generation of Actionable Knowledge for Direct Marketing. Journal of Software Engineering and Applications, 06(04), pp.196–206.
- Kumar, M.A., M. Hemalatha, P. Nagaraj and S. Karthikeyan, 2010. A new way towards security in Tcp/Ip protocol suite. pp: 46-50.
- Ladid, L., McGibney, J., Ronan, J. and Foghlú, M.Ó., 2005. Security and Privacy with IPv6.
- Levin, S.L. and Schmidt, S., 2014. IPv4 to IPv6: Challenges, solutions, and lessons. Telecommunications Policy, 38(11), pp.1059–1068.
- Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W., Li, X. & Gong, L. 2018, ‘Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN’, International Journal of Communication Systems, vol. 31, no. 5, pp. 1–15.
- Li, Q., Jimmel, T. and Shima, K., 2010. IPv6 Core Protocols Implementation.
- Li, Y., Li, Z.T. and Liu, S., 2006. A fuzzy anomaly detection algorithm for IPv6. 2006 2nd International Conference on Semantics Knowledge and Grid, SKG, pp.4–7.

- Liu, Z. and Lai, Y., 2009, June. A data mining framework for building intrusion detection models based on ipv6. In International Conference on Information Security and Assurance (pp. 608-618). Springer, Berlin, Heidelberg.
- Lu, B., Charlton, M., Brunson, C. and Harris, P., 2016. The Minkowski approach for choosing the distance metric in geographically weighted regression. *International Journal of Geographical Information Science*, 30(2), pp.351-368.
- Mabayoje, M.A., Balogun, A.O., Ameen, A.O. & Adeyemo, V.E. 2016, 'Influence of Feature Selection on Multi - Layer Perceptron Classifier for Intrusion Detection System', *Computing Information Systems, Development Informatics & Allied Research Journal*, vol. 7, no. 4, pp. 87–94.
- Maharaj, N., 2014. A Comparative Analysis of Different Classification Techniques for Intrusion Detection System. 95(17), pp.22–26.
- Mahjabin, T., Xiao, Y., Sun, G. and Jiang, W., 2017. A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks*, 13(12), p.1550147717741463.
- Mansingka, V.K., Schaechtle, U., Handa, S., Radul, A., et al. (2018) Probabilistic Programming with Programmable Inference. In: *Proceedings of 39th ACM SIGPLAN Conference on Programming Language Design and Implementation*. 2018 pp. 1–14.
- McHugh, J., 2000. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Transactions on Information and System Security (TISSEC)*, 3(4), pp.262-294.
- Min, C., 2011. Research on Network Security Based on IPv6 Architecture. *International Conference on Electronics and Optoelectronics (ICEOE 2011)*, (Iceoe), pp.4–6.
- Mitchell, D. and Vora, D., 2013. Comparative Study of Data Mining Techniques to Enhance Intrusion Detection. 3(1), pp.1267–1275.
- Motzev, M. (2018) Statistical Learning Networks in Simulations for Business Training and Education. 45 (2000), 295–301.
- Nazario, J. and Kristoff, J., 2012. Internet Infrastructure Security. *IEEE Computer and Reliability Societies*, pp.24–25.

- Niyaz, Q., Sun, W. and Javaid, A.Y., 2016. A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN).
- Niyaz, Q., Sun, W., Javaid, A.Y. & Alam, M. 2016, A Deep Learning Approach for Network Intrusion Detection System.
- Potluri, S. and Diedrich, C., 2016. Accelerated deep neural networks for enhanced Intrusion Detection System. IEEE International Conference on Emerging Technologies and Factory Automation, ETFA.
- Rasmussen, C.E. (1996) Evaluation Of Gaussian Processes And Other Methods For Non-Linear Regression.
- Reddy, E.K., Reddy, V.N. and Rajulu, P.G., 2011. A Study on Intrusion Detection Based on Data Mining. World Congress on Engineering, 3, pp.8–15.
- Saad, R., Manickam, S., Alomari, E., Anbar, M. And Singh, P., 2014. Design & Deployment Of Testbed Based On Icmpv6 Flooding Attack. Journal Of Theoretical & Applied Information Technology, 64(3).
- Saad, R.M., Ramadass, S. and Manickam, S., 2013. A study on detecting ICMPv6 flooding attack based on IDS. Australian Journal of Basic and Applied Sciences, 7(2), pp.175-181.
- Saad, R.M.A., Anbar, M., Manickam, S. and Alomari, E., 2016. An intelligent ICMPv6 DDoS flooding-attack detection framework (V6IIDS) using back-propagation neural network. IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India), 33(3), pp.244–255.
- Salih, A., Ma, X. and Peytchev, E., 2015. Detection and Classification of Covert Channels in IPv6 Using Enhanced Machine Learning.
- Salih, A., Ma, X. and Peytchev, E., 2017. Implementation of Hybrid Artificial Intelligence Technique to Detect Covert Channels Attack in New Generation Internet Protocol IPv6. pp.173–190.
- Sathya, R. & Abraham, A. (2013) Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. 2 (2), 34–38.
- Shiranzaei, A. and Khan, R.Z., 2015. A comparative study on IPv4 and Ipv6. International Journal of Advanced Information Science and Technology, 33(33), p.9.

- Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A. and Stiller, B., 2010. An overview of ip flow-based intrusion detection. *IEEE Communications Surveys and Tutorials*, 12(3), pp.343-356.
- Stalph, P. (2014) Introduction to F unction Approximation and Regression. [Online]. Available from: doi:10.1007/978-3-658-04937-9.
- Tan, P.N., Steinbach, M. and Kumar, V., 2006. Classification: basic concepts, decision trees, and model evaluation. *Introduction to data mining*, 1, pp.145-205
- Vinitha, V.J.P.S. a, 2013. Classification Algorithms in Intrusion Detection System : A Survey. 4(5), pp.746–750.
- Wang, Y., Anderson, J.A., Baciu, G. & Budin, G. (2012) Perspectives on eBrain and Cognitive Computing. *International Journal of Cognitive Informatics and Natural Intelligence*.
- Wireshark User's Guide :for Wireshark 1.7 by Ulf Lamping, Richard Sharpe, Ed.Warnicke Copyright © 2017; Source: <https://www.wireshark.org/about.html>.
- Ye, J., Cheng, X., Zhu, J., Feng, L. and Song, L., 2018. A DDoS Attack Detection Method Based on SVM in Software Defined Network. 2018.
- Yuan, X., Li, C. and Li, X., 2017. DeepDefense: Identifying DDoS Attack via Deep Learning. 2017 IEEE International Conference on Smart Computing, SMARTCOMP 2017, pp.1–8.
- Zekri, M., Kafhali, S. El, Aboutabit, N. and Saadi, Y., 2017. DDoS attack detection using machine learning techniques in cloud computing environments. 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), (February 2018), pp.1–7.
- Zulkiflee, M., Azmi, M.S., Ahmad, S.S.S., Sahib, S. and Ghani, M.K.A., 2015. A Framework of Features Selection for IPv6 Network Attacks Detection. 14(January 2015), pp.399-408.
- Ahmed, A.S.A.M.S., Hassan, R. and Othman, N.E., 2017. IPv6 Neighbor Discovery Protocol Specifications , Threats and Countermeasures : A Survey. *IEEE*, pp.18187–18210.
- Al-Ani, M.S. and Haddad, R.A., 2012. IPv4/IPv6 Transition. *International Journal of Engineering Science and Technology*, 4(12), pp.4815-4822.
- Alaidaros, H., Mahmuddin, M. and Al-Mazari, A., 2011. An overview of flow-based and packet-based intrusion detection performance in high speed networks.
- Aleesa, A.M., Hassan, R. and Kamal, S.U.M., 2016. A rule-based technique to detect router

- advertisement flooding attack against biobizz web application. *Advanced Science Letters*, 22(8), pp.1887-1891.
- Anbar, M., Abdullah, R. and Saad, R.M.A., 2016. Review of Security Vulnerabilities in the IPv6 Neighbor Discovery Protocol Review of Security Vulnerabilities in the IPv6 Neighbor Discovery Protocol. *Information Science and Applications (ICISA)*, (June), pp.603–612.
- Anbar, M., Abdullah, R., Al-Tamimi, B.N. and Hussain, A., 2018. A Machine Learning Approach to Detect Router Advertisement Flooding Attacks in Next-Generation IPv6 Networks. *Cognitive Computation*, 10(2), pp.201–214.
- Anon (2016) Weka : Decision Trees – J48.
- Anon (2017) Machine learning : the power and promise of computers that learn by example.
- Atkeson, C.G., Moore, A.W. & Schaal, S. (1996) Locally Weighted Learning. 1–52.
- Ayodele, T.O. (n.d.) Types of Machine Learning Algorithms.
- Aziz, M.Z.A. and Okamura, K., 2017. Leveraging SDN for Detection and Mitigation SMTP Flood Attack through Deep Learning Analysis Techniques. *Ijcsns*, 17(10), pp.166–172.
- Balogun, A.O., Balogun, A.M., Sadiku, P.. & Adeyemo, V.E. (2017) Heterogeneous Ensemble Models for Generic Classification. *Anale. Seria Informatică*. XV, 92–98.
- Barbhuiya, F.A., Biswas, S. and Nandi, S., 2011. Detection of neighbor solicitation and advertisement spoofing in IPv6 neighbor discovery protocol. *Proceedings of the 4th international conference on Security of information and networks - SIN '11*, p.111.
- Besold, T.R., Garcez, A. d'Avila, Bader, S., Bowman, H., et al. (2017) Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. *ArXiv*, cs. 1–58.
- Bhosale, D. & Ade, R. (2014) Feature Selection based Classification using Naive Bayes , J48 and Support Vector Machine. 99 (16), 14–18.
- Bishop, C.M., 2006. *Pattern recognition and machine learning*, Springer.
- Budhathoki, D.R., n.d. *Computer Network*, 2011.
- Caicedo, C.E. and Joshi, J., 2008. Security issues in ipv6 networks. *International Telecommunications Research and Education Association (ITERA)*.

- Caicedo, C.E., Joshi, J.B.D. and Tuladhar, S.R., 2009. IPv6 Security Challenges. IEEE Computer Society, pp.36–42.
- Chai, K., Hn, H.T. & Cheiu, H.L. (2002) Naive-Bayes Classification Algorithm. Bayesian Online Classifiers for Text Classification and Filtering. 97–104.
- Chao, W. (2011) Machine Learning Tutorial.
- Chuangchunsong, N., Kamolphiwong, S., Kamolphiwong, T., Elz, R. and Pongpaibool, P., 2014. Performance evaluation of IPv4/IPv6 transition mechanisms: IPv4-in-IPv6 tunneling techniques. International Conference on Information Networking, pp.238–243.
- Cisco Systems, 2011. NetFlow Services Solutions Guide, <http://www.cisco.com>.
- Cooper, A., Gont, F., & Thaler, D. (2016). Security and privacy considerations for ipv6 address generation mechanisms.
- David, J. & Thomas, C. 2019, ‘Efficient DDoS Flood Attack Detection using Dynamic Thresholding on Flow-Based Network Traffic’, Computers & Security.
- Elejla, O., Belaton, B., Anbar, M. & Alnajjar, A. 2016, ‘A Reference Dataset for ICMPv6 Flooding Attacks’, Journal of engineering and applied sciences, vol. 11, no. 3, pp. 476–81.
- Elejla, O.E., Anbar, M., Belaton, B. & Alijla, B.O. 2018, ‘Flow-based IDS for ICMPv6-based DDoS Attacks Detection’, Arabian Journal for Science and Engineering.
- Elejla, O.E., Belaton, B., Anbar, M. and Alnajjar, A., 2016. Intrusion Detection Systems of ICMPv6-based DDoS attacks. Neural Computing and Applications, pp.1-12.
- Englert, P. (2012) Locally Weighted Learning. Seminar Class on Autonomous Learning Systems. 1 (1), 1–9.
- Faltin, F. and Kenett, R., 2007. Bayesian Networks. Encycl. Stat. Qual. Reliab., 1(1), p.4.
- Frankel, S. and Green, D., 2008. Internet Protocol Version 6. IEEE Security & Privacy, 6(3), pp.1–4.
- Frankel, S. and Krishnan, S., 2011. IP security (IPsec) and internet key exchange (IKE) document roadmap. *Request for Comments, 6071*.
- GNS3. 2017, source: <https://www.gns3.com>

- Goralski, W., 2009. IPv4 and IPv6 Headers. *The Illustrated Network*, pp.165–188.
- Guerra, L., McGarry, L.M., Bielza, C., Larran, P., et al. (2010) Comparison Between Supervised and Unsupervised Classifications of Neuronal Cell Types : A Case Study.
- Hamed, T., Ernst, J.B. and Kremer, S.C., 2018. A Survey and Taxonomy of Classifiers of Intrusion Detection Systems.
- Hauser V., T.H.C., 2006. *Attacking the IPv6 Protocol Suite*.
- He, Z., Zhang, T. and Lee, R.B., 2017. Machine Learning Based DDoS Attack Detection from Source Side in Cloud. 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), pp.114–120.
- Hemant, P., Sarkhedi, B. and Vaghamshi, H., 2013. Intrusion Detection in Data Mining With Classification Algorithm. 2(7), pp.3063–3070.
- Hinden, R. (2017). *Internet protocol, version 6 (IPv6) specification*.
- Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., et al. (2017) Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey. 1–43.
- Hogg, S., Karpenko, J., Miller, D. and Vyncke, E., 2009. *IPv6 Security: Information assurance for the next-generation Internet Protocol*. Cisco Press.
- Holzinger, A. (2016) Interactive machine learning for health informatics : when do we need the human-in-the-loop ? *Brain Informatics*. 3 (2), 119–131.
- Holzinger, A. (2017) *Introduction to MACHine Learning & Knowledge Extraction (MAKE)*
- Holzinger, A. (2019) From Machine Learning to Explainable AI. 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA). (August 2018), 55–66.
- Hutter, F., Kotthoff, L. & Vanschoren, J. (2019) *Automatic Machine Learning: Methods, Systems, Challenges*.
- Jaiganesh, V., Mangayarkarasi, S. and Sumathi, P., 2013. Intrusion Detection Systems : A Survey and Analysis of Classification Techniques. 2(4), pp.1629–1635.
- Jain, A. & Rana, J.L. (2016) Classifier Selection Models For Intrusion Detection System (IDS). 4 (1), 1–11.

- Jain, Y.K. (2012) An Efficient Intrusion Detection Based on Decision Tree Classifier Using Feature Reduction. 2 (1), 1–6.
- Jung, Y. and Hu, J., 2015. AK-fold averaging cross-validation procedure. *Journal of nonparametric statistics*, 27(2), pp.167-179.
- Karim, M. and Rahman, R.M., 2013. Decision Tree and Naïve Bayes Algorithm for Classification and Generation of Actionable Knowledge for Direct Marketing. *Journal of Software Engineering and Applications*, 06(04), pp.196–206.
- Kumar, M.A., M. Hemalatha, P. Nagaraj and S. Karthikeyan, 2010. A new way towards security in Tcp/Ip protocol suite. pp: 46-50.
- Ladid, L., McGibney, J., Ronan, J. and Foghlú, M.Ó., 2005. Security and Privacy with IPv6.
- Levin, S.L. and Schmidt, S., 2014. IPv4 to IPv6: Challenges, solutions, and lessons. *Telecommunications Policy*, 38(11), pp.1059–1068.
- Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W., Li, X. & Gong, L. 2018, ‘Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN’, *International Journal of Communication Systems*, vol. 31, no. 5, pp. 1–15.
- Li, Q., Jimmel, T. and Shima, K., 2010. IPv6 Core Protocols Implementation.
- Li, Y., Li, Z.T. and Liu, S., 2006. A fuzzy anomaly detection algorithm for IPv6. 2006 2nd International Conference on Semantics Knowledge and Grid, SKG, pp.4–7.
- Liu, Z. and Lai, Y., 2009, June. A data mining framework for building intrusion detection models based on ipv6. In *International Conference on Information Security and Assurance* (pp. 608-618). Springer, Berlin, Heidelberg.
- Lu, B., Charlton, M., Brunson, C. and Harris, P., 2016. The Minkowski approach for choosing the distance metric in geographically weighted regression. *International Journal of Geographical Information Science*, 30(2), pp.351-368.
- Mabayoje, M.A., Balogun, A.O., Ameen, A.O. & Adeyemo, V.E. 2016, ‘Influence of Feature Selection on Multi - Layer Perceptron Classifier for Intrusion Detection System’, *Computing Information Systems, Development Informatics & Allied Research Journal*, vol. 7, no. 4, pp. 87–94.
- Maharaj, N., 2014. A Comparative Analysis of Different Classification Techniques for Intrusion

Detection System. 95(17), pp.22–26.

Mahjabin, T., Xiao, Y., Sun, G. and Jiang, W., 2017. A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks*, 13(12), p.1550147717741463.

Malik, M., & Dutta, M. (2018). Defending DDoS in the Insecure Internet of Things: A Survey. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems* (pp. 223-233). Springer, Singapore.

Mansinghka, V.K., Schaechtle, U., Handa, S., Radul, A., et al. (2018) Probabilistic Programming with Programmable Inference. In: *Proceedings of 39th ACM SIGPLAN Conference on Programming Language Design and Implementation*. 2018 pp. 1–14.

McHugh, J., 2000. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Transactions on Information and System Security (TISSEC)*, 3(4), pp.262-294.

Min, C., 2011. Research on Network Security Based on IPv6 Architecture. *International Conference on Electronics and Optoelectronics (ICEOE 2011)*, (Iceoe), pp.4–6.

Mitchell, D. and Vora, D., 2013. Comparative Study of Data Mining Techniques to Enhance Intrusion Detection. 3(1), pp.1267–1275.

Motzev, M. (2018) Statistical Learning Networks in Simulations for Business Training and Education. 45 (2000), 295–301.

Nazario, J. and Kristoff, J., 2012. Internet Infrastructure Security. *IEEE Computer and Reliability Societies*, pp.24–25.

Niyaz, Q., Sun, W. and Javaid, A.Y., 2016. A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN).

Niyaz, Q., Sun, W., Javaid, A.Y. & Alam, M. 2016, A Deep Learning Approach for Network Intrusion Detection System.

Potluri, S. and Diedrich, C., 2016. Accelerated deep neural networks for enhanced Intrusion Detection System. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*.

- Rasmussen, C.E. (1996) Evaluation Of Gaussian Processes And Other Methods For Non-Linear Regression.
- Reddy, E.K., Reddy, V.N. and Rajulu, P.G., 2011. A Study on Intrusion Detection Based on Data Mining. World Congress on Engineering, 3, pp.8–15.
- Saad, R., Manickam, S., Alomari, E., Anbar, M. And Singh, P., 2014. Design & Deployment Of Testbed Based On Icmpv6 Flooding Attack. Journal Of Theoretical & Applied Information Technology, 64(3).
- Saad, R.M., Ramadass, S. and Manickam, S., 2013. A study on detecting ICMPv6 flooding attack based on IDS. Australian Journal of Basic and Applied Sciences, 7(2), pp.175-181.
- Saad, R.M.A., Anbar, M., Manickam, S. and Alomari, E., 2016. An intelligent ICMPv6 DDoS flooding-attack detection framework (V6IIDS) using back-propagation neural network. IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India), 33(3), pp.244–255.
- Salih, A., Ma, X. and Peytchev, E., 2015. Detection and Classification of Covert Channels in IPv6 Using Enhanced Machine Learning.
- Salih, A., Ma, X. and Peytchev, E., 2017. Implementation of Hybrid Artificial Intelligence Technique to Detect Covert Channels Attack in New Generation Internet Protocol IPv6. pp.173–190.
- Sathya, R. & Abraham, A. (2013) Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. 2 (2), 34–38.
- Shiranzaei, A. and Khan, R.Z., 2015. A comparative study on IPv4 and Ipv6. International Journal of Advanced Information Science and Technology, 33(33), p.9.
- Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A. and Stiller, B., 2010. An overview of ip flow-based intrusion detection. IEEE Communications Surveys and Tutorials, 12(3), pp.343-356.
- Stalph, P. (2014) Introduction to Function Approximation and Regression. [Online]. Available from: doi:10.1007/978-3-658-04937-9.
- Tan, P.N., Steinbach, M. and Kumar, V., 2006. Classification: basic concepts, decision trees, and model evaluation. Introduction to data mining, 1, pp.145-205

- Vinitha, V.J.P.S. a, 2013. Classification Algorithms in Intrusion Detection System : A Survey. 4(5), pp.746–750.
- Wang, Y., Anderson, J.A., Baciou, G. & Budin, G. (2012) Perspectives on eBrain and Cognitive Computing. International Journal of Cognitive Informatics and Natural Intelligence.
- Wireshark User's Guide :for Wireshark 1.7 by Ulf Lamping, Richard Sharpe, Ed.Warnicke Copyright © 2017; Source: <https://www.wireshark.org/about.html>.
- Ye, J., Cheng, X., Zhu, J., Feng, L. and Song, L., 2018. A DDoS Attack Detection Method Based on SVM in Software Defined Network. 2018.
- Yuan, X., Li, C. and Li, X., 2017. DeepDefense: Identifying DDoS Attack via Deep Learning. 2017 IEEE International Conference on Smart Computing, SMARTCOMP 2017, pp.1–8.
- Zekri, M., Kafhali, S. El, Aboutabit, N. and Saadi, Y., 2017. DDoS attack detection using machine learning techniques in cloud computing environments. 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), (February 2018), pp.1–7.
- Zulkiflee, M., Azmi, M.S., Ahmad, S.S.S., Sahib, S. and Ghani, M.K.A., 2015. A Framework of Features Selection for IPv6 Network Attacks Detection. 14(January 2015), pp.399-408.