DETECTION OF DISTRIBUTED DENIAL OF SERVICE ATTACK ON VEHICULAR ADHOC NETWORKS USING VERY FAST DECISION TREE

By

Amman Durrani

A thesis submitted to the Faculty of Information Security Department, Military College of Signals, National University of Science and Technology, Pakistan in partial fulfillment of the requirements for the degree of MS in Information Security

December 2016

SUPERVISOR CERTIFICATE

This is to certify that **NS Amman Durrani** student of **MSIS-12** course has completed her MS thesis titled **"Detection Of Distributed Denial Of Service Attack On Vehicular Adhoc Networks Using Very Fast Decision Tree"** under my supervision. I have reviewed her final thesis copy and I am satisfied with her work.

_____

Dr. Seemab Latif

ABSTRACT

Smart technologies have seeped into every aspect of human communication. The increase in population has escalated the numbers of vehicles on road as well as probability of collision among them. The development of intelligent vehicles for improving driver experience have introduced the Mobile Adhoc Technologies for transport systems in the form of VANETs or more recently referred to as the Inter Vehicular Communication Network. This is an inherently wireless system which brings its own set of security challenges dependent on its lack of infrastructure, short connection times and high mobility. VANETs aim to provide user with optimum driving experience as well as safety on road. The most significant service for the vehicular networks is the availability of ubiquitous information to the legitimate users because a delay in this life-critical and time-sensitive network can be fatal in some scenarios. Hence, Denial of Service is the most imminent security threat for this system. Although, some research has been done for the mitigation of this issue, data mining approach towards this aspect is minimum. Furthermore, the available schemes are deficient in one aspect or the other to provide impeccable detection.

The capability of decision trees to timely identify behavioral changes in traffic with low error rate makes it a powerful detection scheme. Very fast decision tree is a data mining mechanism which can handle high speed streaming data, suitable for VANETs. The ultimate aim of this research is to explore a resilient diagnostic technique to mitigate the denial of service attack in VANETs. The technique will be computationally efficient and error free, to provide a secure environment for optimum delivery of VANETs applications. Appropriate simulation methodology for VANETs is imperative for acurate reproduction of attack on the vehicular adhoc system. Real time mobility simulator and traffic generator have been employed for this purpose. The research aims to carry out a performance evaluation of the proposed detection scheme through simulations

to test its competence for high speed, sensor data in vehicular adhoc systems and benchmark its dexterity against other detection schemes. Furthermore, it provides quantitative comparison of various types of detection schemes. The comparative analysis confirms the ascendancy of the applied technique with respect to the earlier research in this field.

DECLARATION

I hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere.

_____

Amman Durrani

DEDICATION

"In the name of Allah, the most Beneficent, the most Merciful"

This thesis is dedicated to my parents, husband and beautiful daughters who have graciously endured my busy schedule during the degree.

ACKNOWLEDGMENT

I am deeply humbled by the benevolence of almighty Allah for providing me the strength to attempt and conclude this endeavor, which he gave me the courage to undertake.

I would like to express deep regard and gratitude for my supervisor, Dr. Seemab Latif for her persevering support and guidance throughout the thesis. The invaluable expertise on her subject and commitment to her students is an inspiration and a virtue that have greatly inspired me.

I am highly indebted to Dr. Rabia Latif for her contributions and unwavering guidance throughout the journey of this research. From the selection of the topic, to the experimental and review analysis, her constructive comments, suggestions and motivation have been instrumental to this research.

I am very grateful to my committee and evaluation members, Dr. Imran Rashid and Lt. Col. Muhammad Mubashir Quddoos for their support and constructive criticism for the amelioration of the research work.

In the end, I would like to mention the constant and unflinching support of my parents specially my mother who encouraged me to join the master's program and supported me to complete it. I would also like to thank my husband for taking time out from his work to review my thesis and provide critical analysis for its amendments.

TABLE OF CONTENTS

LIST OF FIGURES

LSIT OF TABLES

GLOSSARY

VANETs Vehicular adhoc Networks

MANETs Mobile aadhoc Networks

WSN Wireless sensor networks

DDoS Distributed Denial of Service Attack

RWP Random Way Point Model

ITS Intelligent Transport Systems

IVS Inter Vehicular Systems

TCL Tool Command Language

SUMO Simulation of Urban Mobility

MOVE Mobility Model Generator

CBR constant bit rate

NS2 Network Simulator 2

AODV Ad-Hoc On-Demand Distance Vector

DSR Dynamic Source Routing

DSDV Destination Sequenced Distance Vector

ACK ACKnowledgment

CBR Constant Bit Rate

DSDV Distance Sequenced Distance Vector

FCC Federal Communications Commission

GUI Graphical User Interface

NCTUns National Chiao Tung University Network Simulator

NS2 Network Simulator

RSU Road Side Unit

RREP Route Replies RREQ Route Requests

SNR Signal to Noise Ratio

UDP User Datagram Protocol

RX Receiver

TX Transmitter

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

QoS Quality of Service

PDR Packet Delivery Ratio

HB Hoeffding Bound

IDS Intrusion Detection Systems

Chapter 1

INTRODUCTION

1.1 Overview

Vehicle to vehicle communication also known as V2V is part of the growing trend towards pervasive computing. The concept is also part of the emergent area known as the Internet of Things (IoT). This is being developed to allow autonomous decision making on the part of vehicles to facilitate an optimum road experience and avert traffic predicaments like congestion and casualties. Though the research work in this area is gaining momentum, still the information is quite fragmented and requires comprehensive work, especially in the provision of security for this adhoc network.

Machine learning, data mining and pattern recognition have gained immense interest in latest scientific literature for numerous applications. These techniques have also been employed for intrusion detection in communication networks. Although some research has been performed to evaluate their efficiency in denial of service attack in wireless networks, yet they have been minimally applied in technology involving Vehicular Adhoc Networks (VANETs). Decision trees are a very effective in finding irregularities in large amount of streaming data sets, hence their application in VANETs seems to be a practical solution for the detection of DoS attacks.

The chapter is organized as follows. Section 1.1 provides an overview of this research, whereas section 1.2 introduces the concept of Vehicular Adhoc Network and understanding of its communication flow. Section 1.2.1 briefs about the characteristics of VANETs significantly susceptible

to attack. The security requirements of VANETs have been described in section 1.3 . Section 1.4 enumerates the attacks on Vehicular adhoc networks. Section 1.5 comprehensively describes the Denial of Service attack and its effects on the applications of VANETs. Section 1.6 points out the deficiency of current research in this area and illustrates the motivation of our research. The contributions of our research towards the progressive field of securing the vehicular adhoc network communications have been highlighted in Section 1.7. In the end, Section 1.8 gives a complete breakdown of the chapters in this dissertation.

1.2 VANETs

VANETs are one of the most promising public adhoc network applications due to their human and economic impact. National French Road Safety Observatory, reported 65,556 accidents in 2012 as compared to 65,024 in 2011 [1]. The research conducted by National Highway Traffic Safety Administration (NHTSA) [2] concluded that V2V communications could help to avoid 80% of traffic accidents. According to this investigation, V2V technology could prevent 592,000 crashes and save 1,083 lives annually from these calamities. Almost 50 applications have been submitted by major car manufacturers, Daimler-Chrysler, General Motors, BMW and Ford pertaining to the dedicated short range channel for inter vehicular communications.

VANET is a type of wireless ad hoc network that supports ubiquitous connectivity between vehicles. Two standards are currently been advocated in the vehicular adhoc network, Dedicated Short Range Communication (DSRC) and wave. In October 1999, the United States Federal Communications Commission (FCC) assigned 75 MHz of spectrum in the 5.9 GHz band to be used by vehicular adhoc networks. ETSI, the European Telecommunications Standards Institute granted 30 MHz spectrum in the 5.9 GHz band for VANETs, in August 2008. This is the Dedicated short-range communications wireless channel specifically designed for automotive use. It has a

communication range from 300m to 1Km and provides 6-27 Mbps data transferring rate. The DSRC spectrum is divided into 7 channels where each channel is of 10 MHz.

Another standard is IEEE 1609-standards for Wireless Access in Vehicular Environments also known as WAVE or IEEE 802.11p. The frequency range of WAVE is 5.85-5.925 GHz. It uses Orthogonal Frequency Division Multiplexing technique and provides a data transferring rate of 3- 27 Mbps in 10 MHz channels. VANETs make use of short range wireless technologies [3] like WLAN, both standard Wi-Fi and the vehicle-specific IEEE 802.11p, Bluetooth, Infrared, and ZigBee as well as cellular technologies like, LTE, or WiMAX IEEE 802.16, UMTS etc.

1.2.1 Vulnerabilities of VANETs

The security issues mandatory for Inter Vehicular Communication Systems (IVS) network are similar to traditional networks but due to the wireless medium of VANETs, there are certain intrinsic vulnerabilities of such systems [4], which produce challenges in the provision of network protocols and security systems [5]. These characteristics include high mobility of the nodes in this inherently wireless system that results in rapidly changing network topology, signal fading and collusion due to impediments on the roads. Security challenges arise due to the availability of routing information dependent on the predictable road configurations in urban street scenario.

The wireless medium of this network makes it vulnerable to numerous attacks like jamming, eavesdropping and interference [4]. Attackers can flood the system with high frequency signals rendering the system unavailable for legitimate users. They can also introduce false information into the system causing data to become irrelevant or useless resulting in nuisance to the users or triggering accidents. For example, malicious nodes could send fake congestion messages or fake emergency events for their own benefit like clearing a lane for themselves.

Security requirements for VANETs have a huge impact on market penetration and public acceptance of this technology and is imperative for an approval by the transport industry. VANETs also termed as Inter Vehicular Communication Networks (IVC), are a subclass of Mobile Adhoc Networks (MANETS), where vehicles are the mobile nodes of this network. There are three types of communication levels in VANETs shown in figure 1.1.

V2V Communication: This is the connection between the vehicles on road [1].

V2I communication: This is the connection between the On Board Unit (OBUs) in the vehicle and the Road Side Units RSUs. This architecture is also known as cellular/WLAN network.

Communication between roadside infrastructure: This is the communication between the RSUs and the WiMax or 3G Infrsastructure.

VANETs have different applications which can be applied by Peer-to-Peer (P2P) communication

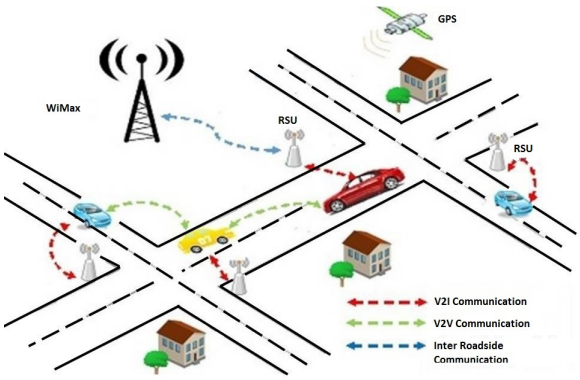

Figure 1.1: Vehicular adhoc networks

or via multi-hop communication [6]. The services provided by VANETs can be entertainment or comfort provision for the consumer such as, weather forecasting, and broadcasting information like advertisements for some goods, commodity and online services. VANETs are most significantly used for critical services [3] including and certainly not limited to the following:

Collision prevention ensures that beacons on cars and motorcycles help maintain minimum separation to avoid collisions between vehicles.

Accident reporting handles broken down cars to send simple reports to central servers.

Intersection assistance assures pairs of cars to automatically coordinate complex maneuvers at intersections.

Lane assistant or transport efficiency manages simple roadside beacons for support while changing lanes or overtaking [2].

Christoph Summer and Dressler describe the various applications of Vehicular adhoc networks for road users [3] . We have only mentioned the services of VANETs in figure 1.2 taken from this book which are time sensitive and require robust quality of service to assist in a quick reaction by the driver. The assortment of applications in VANETs generate a new class of vehicular support network known as Intelligent Transportation System (ITS) [15].

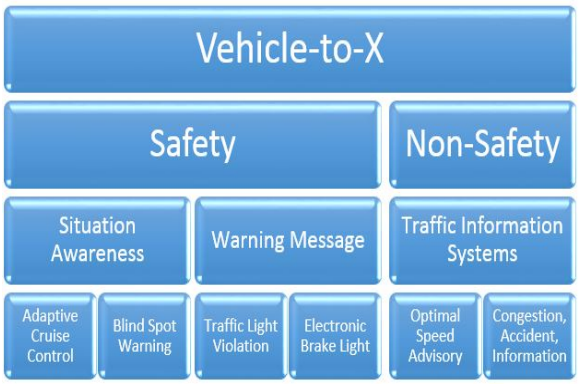

Figure 1.2: Time sensitive application of VANETs

1.3 Security Requirements of VANETs

There is a valid threat to vehicular adhoc networks by malicious entities. As with other communication networks VANETs has specific requirements described as follows [7, 8, 9]:

Confidentiality, Integrity and Availability (CIA): The CIA triage is the basis for any communication network and hence is the basic security challenge for IVS as well.

Authentication: The vehicles should respond only to legitimate events. Senders of messages should be authenticated.

Vehicle Privacy and Anonymity: Privacy of drivers should be maintained against unauthorized observers.

Access Control: The vehicles should be able to access available services offered by remote nodes.

Data Non-Repudiation: No entity should be able to claim false actions or deny true ones. It is necessary so that the impostor who sends wrong information on purpose can be called to account e.g. in a post-paid charging and billing.

Verification of data consistency: : Data should be correct as legitimate senders can send false messages resulting in a reaction by the driver which can even become fatal.

1.4 Security Attacks on VANETs

The security attacks on VANETs can be classified as either active or passive. The former describes the actions of an attacker to impersonate a legitimate driver/RSU and introduce false information into the network causing other drivers to redirect their routes, evoke traffic jams, collisions or to clear a certain path for the attacker's benefit. The latter is eavesdropping by a malicious in-

sider/outsider to gain information about the network. The attacks can be further described as follows:

Integrity Attack: The saboteur vehicle can alter the real data, changing the real sender of the packets.

Message Replay Attack: The fraudulent user replays sending past/false messages in order to jam the network.

Message Spoofing: The attacker sends false messages to other vehicles to impede efficient transfer of information between the nodes of the network.

Impersonation Attack: An eavesdropping vehicle takes up the identity of a legitimate vehicle to introduce false messages in the network.

Movement Tracking: A snooping vehicle collects information about the other vehicles to track their position and speed. Thereafter, it can detect the future behavior of those vehicles and affect their transmission performance.

Denial of Service (DoS) attack: Denial of Service attack aims to clog the resources and bandwith of the network, thus rendering the system unavailable for legitimate users.

1.5 Denial of Service attack in VANETs

According to Verma et. al, the most serious threat to VANETs is Denial of Service [10]. This attack makes the network unavailable for the legitimate users. Compared to other wireless networks, VANETs are extremely sensitive to availability of information. Dissemination of information at appropriate intervals is extremely critical for this time sensitive network, since a delay in information can result in accidents and casualties such as in the case of sending information of sharp

braking or collision to vehicles coming behind. In DoS attack, the attacking vehicle transmits useless messages into the network to waste large bandwidth of the channel and devour resources of other vehicles. In Distributed denial of service attack (DDoS), the attacker first interrogates the network to find the vulnerable entities and then gains root access to these entities in order to perpetrate DDoS attacks from them by making them slaves/zombies. This methodology not only makes trace back to the attack originator difficult for Security investigators but also increases the attack capabilities of the master manifold. The infrastructure of DDoS attack can be viewed in figure 1.3 In the case of VANETs, DDoS can be of three types. It can be enacted upon the vehicles, it can affect the Road Side Units by reducing their resources, or it can occur on the channel itself by congesting the network. Following are the most significant types of DoS attacks relevant

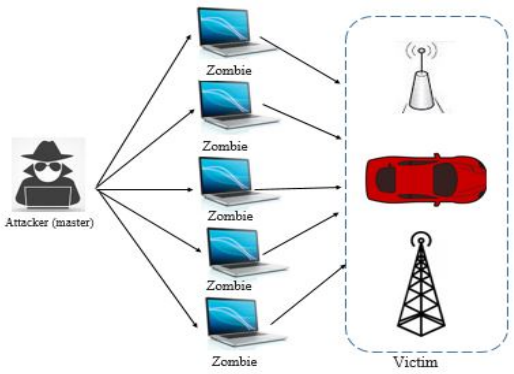

Figure 1.3: DDoS attack in VANETs

to vehicular adhoc networks:

(a) ***Jamming:*** In this physical layer attack, malicious vehicles inject a large no. of useless messages to lower the signal to noise ratio and hence, drop communication between legitimate vehicles [11]. This taxes the bandwidth of the network disallowing the dissemination of critical information required for the smooth running of a vehicular adhoc network. The specification of DSRC is partially responsible for such attacks as it specifies that a vehi-

cle must not transmit until it determines that the channel is idle. This lapse in the system accedes continuous transmission from an adversary resulting in a jamming attack [12]. Jamming can be of three types:

- **Jamming the Node in V2V Communication:** As the node will be busy in acknowledging the messages/packets it is being continuously sent hence it will become unavailable for other purposes.
- **Jamming the RSU in V2I Communication:** Here RSU is rendered unavailable due to the same problem of acknowledging continuously sent packets by malicious user.
- **Jamming the Channel:** A high frequency signal is sent to congest the network so it becomes clogged for legitimate cars or RSUs in its path.

(b) ***Black hole Attack:*** This is basically an attack on the availability of the network whereby the malicious entity shows itself to be part of a network but doesn't participate in forwarding information of the route thus disrupting routing tables, redirecting data, loss of data and reducing robustness of the system.

(c) ***Greyhole Attack:*** This is a variant of black hole attack which drop packets of certain application and allows others. This affects efficiency and availability of the communication network.

(d) ***Sinkhole Attacks:*** Sinkhole is used to propagate black hole and greyhole attacks by attracting the packets from neighboring nodes to pass it along the route. This allows the attacker to, not only collect information about the packets but also drop legitimate packets rendering them unavailable for the nodes.

(e) ***Wormhole Attack:*** In a wormhole attack, an attacker receives packets at one point in the

network and "tunnels" them to another point in the network by broadcasting itself as a neighbor node. This creates a shortcut known as wormhole. This shortcut is used in the network to gather all the traffic in order to collect information about the nodes. This attack also results in additional routes which are not present in the routing table. The attacker can even replay this collected information into the network to exhaust resources and cause DoS attack or drop packets.

(f) ***Broadcast Tampering:*** The attacker which has already been authenticated as legitimate introduces false messages into the system resulting in hiding the correct information and having sever consequences, along with loss in network efficiency [12].

(g) ***Spamming Attack:*** Just as in a traditional communication network spamming results in loss of robustness of the system by reducing the bandwidth. This is done by injecting useless packets into the network and is hard to detect because of the lack of a centralized infrastructure in VANETs [12].

(h) ***Syn Flooding Attack:*** This is an application layer DoS attack, where many half opened TCP connections are created between two nodes resulting in loss of traffic [6].

(i) ***Session hijacking:*** This is an application layer DoS attack where a malicious entity spoofs IP address and session number of a legitimate user to perform DoS attack on other vehicles, so that the users whose identities are being used cannot access the network. Session hijacking occurs because in most communications credentials are authenticated at the beginning of the session and not afterwards.

(j) ***TCP Ack storm:*** This is another example of application layer DoS attack which is started after session hijacking. The attacker sends a session data to a victim A which further send

an ack packet to the target node B with incorrect sequence no. The node B tries to establish a connection by sending the correct sequence number but A replies with the same ack packet and this process is repeated numerous times to cause a DoS attack.

(k) **Congestion:** Network is congested when the rate of the injected packets exceeds its processing capacity, resulting in an accumulative packets loss [14].

(l) **Impersonation:** Some vehicles masquerade as legitimate nodes in order to attract other vehicles to start communication with them during multi-hop. This malicious vehicle can collect information about the nodes, send false information or drop genuine traffic causing Denial of Service for legitimate users [6].

(m) **Sybil Attack:** In this attack, the attacker uses spoofing to adopt myriad identities creating an illusion of additional vehicles on road, thus impairing the network by injecting false information for the benefit of the attacker, e.g. causing other vehicles to clear a route on the basis of false information about additional traffic on that route.

1.6 Motivation and Problem Statement

Significant research is being conducted for standardization, improvement of communication and quality of service in the realm of Vehicular adhoc networks. There are many open issues regarding the security of VANETs. As it has been established in the previous sections, DDoS attack is the most critical attack threatening the service of VANETs, because the availability of an ubiquitous network is imperative, not only for smooth operation of transportation services employing VANET technology but also for the safety of the consumers.

The schemes more frequently used in recent researches, for attack detection in VANETs are either based on cryptographic means, vehicle location, variations in packet and network features or

using trust based reputation. Apart from the burden of complexity in cryptographic techniques, these schemes have various deficiencies which makes them incompatible for attack detection in VANETs. The weaknesses have been discussed in our literature review in chapter 2.

Data mining and most significantly decision trees have not been effectively exploited for the purpose of attack detection in vehicular networks. The ultimate aim of this research is to explore a resilient detection scheme to mitigate the DDoS attack in VANETs. The technique should be computationally efficient and error free, to provide a secure environment for optimum delivery of VANETs applications. Very fast decision tree (VFDT) is a data mining technique which can be employed for the detection of DDoS attack in wireless networks. It can handle high speed streaming data which is suitable for VANETs. The research aims to carry out a performance evaluation of the proposed detection scheme through simulations to test its competence for high-speed sensor data in VANETs and benchmark its dexterity against other detection schemes. Furthermore, it strives to implement a scalable technique to detect the DDoS attack in VANETs with high accuracy, lesser detection time and reduced computational cost.

The current techniques employed for detection of DDoS attack in VANETs do not employ parameters specific to VANETs. This is necessary to ensure that a standard is maintained for comparison with the current research in VANETs among the scientific community as well as for future work. This research intends to adapt particular VANET parameters to amend this situation and ensure standardization among detection techniques in this field.

1.7 Contributions

The research contributions of this research are as follows:

- ***Contribution # 1***: A framework is proposed to acquire and adopt a real time map for use in the simulation tests. Vehicular traffic is generated on this map by a microscopic and continuous road traffic generator and its trace file is introduced into a real world mobility model to create an adaptable output for use in a network simulator for detailed packet level simulation required for our research. This design aids in a standard process of traffic and network simulation to ensure ease of use and scalability while adopting the detection scheme for different landscapes and traffic scenarios. An approach of creating a simulation without these foundations not only renders incorrect real time simulation but is also cumbersome and time consuming.

- ***Contribution # 2***: A thorough scrutiny of simulation parameters, network protocols and map/traffic generators has been conducted to guarantee the selection of the best possible research criteria. Appropriate bit rate traffic, packet size, antenna direction etc were chosen for correct simulation of VANETs. Moreover, identification of the possible attack points in VANETs has been established.

- ***Contribution # 3***: The most critical DDoS attacks were selected to maximize the range of attack types that come under the umbrella of DDoS attack. These were imitated in the simulator and the appointed data mining technique was adapted to detect these attacks. Performance indicators were further selected for performance evaluation of this detection scheme in terms of accuracy, false alarms rates, detection time, memory usage, computational cost, sensitivity and specificity.

- ***Contribution # 4***: An extensive literature review has been conducted on existing attacks on VANETs and current detection schemes to thwart these attacks. A systematic literature review research paper has been submitted in an ISI indexed journal, titled "Ad Hoc & Sensor

Wireless Networks", having index factor of 0.587. The paper was submitted on August 13, 2016. Further work in the dissertation has also been organized in the form of research papers and ready for submission to peer-reviewed journal or conference.

- ***Contribution # 4***: The most important contribution of the research presented in this dissertation is that it proposes a DDoS attack detecton scheme for VANETS which circumvents all the issues arising in the techniques recommended in the literature review. The proposed scheme is not only light weight and scalable but also presents a methodology which avoids cumbersome cryptographic procedures and reputation based procedures which can themselves fall victim to malicious agents.

1.8 Thesis Outline

The thesis has been divided into six chapters. A concise outline of the chapters is as follows:

Chapter 1 provides an overview of this research and introduces the concept of Vehicular adhoc Network and a description of its communication flow. It also briefs about the characteristics of VANETs significantly susceptible to attack. Security Requirements of VANETs have been enumerated and the attacks on Vehicular adhoc networks have also been discussed in this chapter. It comprehensively describes the Denial of Service attack and its effects on the applications of VANETs, while pointing out the deficiency of current research in this area and illustrating the motivation of our research. It also highlights the contributions of our research towards the progressive field of securing the vehicular adhoc network communications. In the end, it gives a complete breakdown of the chapters in this dissertation.

Chapter 2 gives a complete overview of vehicular adhoc networks. The security requirements of this network along with the security threats have been reviewed and it has been established that

the most critical attack on VANETs is the Distributed denial of Service which has been thoroughly discussed. The chapter also analyzes the techniques and mechanisms which have been employed in recent literature for the detection of distributed denial of service attacks in VANETs. The pros and cons of these techniques and the issues which have been addressed in our research to compensate for the weaknesses in these mechanisms have been subsequently highlighted.

Chapter 3 explains the need for using simulation for testing the proposed scheme and to mirror appropriate vehicular conditions to examine the proposed technique. It also explains the selection of a map generator and the method of creating the road topologies. Moreover, the traffic mobility models have been discussed comprehensively. It shows the method to create this model as well as tailor it according to the requirements of simulating real time vehicular adhoc networks. The chapter also examines the formulation of tcl script required for compatibility with NS-2. In addition, it delineates the application of OSM for importing a real map of a busy city area of Pakistan.

Chapter 4 explains the process of communication protocol selection and delineates the characteristics of Ad hoc On-Demand Distance Vector protocol. It discusses the advantages of using 802.11p for VANETs and shows the comparison of performance features obtained by implementing both 802.11a and 802.11p. Moreover, it defines the algorithms and mechanism proposed for generating the cbr file for NS2 specifically for real-time map involvement. In addition, it illustrates the simulation of Distributed Denial of Service attack. A complete description of DDoS on VANETs, including the explanation of congestion and blackhole attacks have been provided.

Chapter 5 illustrates a comprehensive framework for the detection of denial of service attack on VANETs. It introduces the most critical performance features which describe the success or failure of data transmission in vehicular adhoc networks. It also explains the data mining and pattern

recognition mechanism while highlighting the characteristics of stream mining and decision trees respectively. The chapter provides an overview of the VFDT and its variants including including EVFDT, OVFDT and CVFDT. Moreover, it enumerates the performance metrics selected for the purpose of analyzing the detection schemes and provides the results of this comparative analysis.

Finally Chapter 6 concludes the thesis by summarizing the research work and providing a future direction in this area.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

In chapter 1, we have given a complete overview of vehicular adhoc networks. The security requirements of this network along with the security threats have been reviewed and it has been established that the most critical attack on VANETs is the Distributed denial of Service which has been thoroughly discussed. The chapter also analyzes the techniques and mechanisms which have been employed in recent literature for the detection of distributed denial of service attacks in VANETs. Furthermore, the pros and cons of these mechanisms and the issues which have been addressed in our research to compensate for the weaknesses in these techniques have been highlighted.

2.2 Question Formalization

The objective of the literature review is to answer the following research questions:

RQ1: Which types of DoS attacks are more frequently addressed in the literature regarding the service availability of a vehicular adhoc network?

RQ2: What are the current solutions or techniques being proposed to detect the attack mentioned in RQ1?

The guidelines proposed by Kitchenham et al. [15] were applied to generate a systematic review.

This review drew upon the published peer-reviewed papers that specifically considered the issues and challenges for vehiclular adhoc networks from a security attack perspective focusing on the Denial of Service attack.

The following keywords were used for the research questions in this review: Vehicular Adhoc Network, VANETs, DDoS Attack, Sybil Attack, jamming, greedy, data mining, classifier, decision trees, intrusion, malicious, wormhole, sink hole, blackhole.

2.2.1 Search Strings

According to the keywords, the following search strings were employed to search published articles that specifically mention DoS attacks and their countermeasures in VANETs.

Search String for RQ1: (VANETs OR Vehicular adhoc network) AND (DoS attack OR Security).

Search String for RQ2: (VANETs OR Vehicular adhoc network) AND (DoS attack OR Security) AND (Techniques OR Defence mechanism OR Solution).

2.2.2 Selection of Sources

The selected search questions were addressed in digital libraries to acquire related material. The search process encompassed conference papers and journal articles available in reliable, peer reviewed electronic databases including ACM, Springer, IEEE Xplore, Elsevier, Springer and Science Direct. The search process also draws upon grey literature, including government issued technical reports, white papers, articles, etc. In order to make a more pertinent review, we have narrowed down our search to articles published in the 2009 or later. The search was performed in February 2016, therefore, the research work published after this date have not been included in

this review.

2.2.3 Inclusion and Exclusion Criteria

The inclusion criteria for research work are as follows:

(a) Published from 2009 to 2016

(b) Clear focus on discussing the security issues facing VANETs, focusing on Denial of Service attacks

(c) Provided a defense mechanism for the above mentioned issue

The exclusion criteria for research work are as follows:

(a) Duplication of papers

(b) Publications related to attacks, but not specific to VANETs or Denial of Service attack

(c) Non-English contributions

2.2.4 Quality Assessment Checklist (QAC)

A quality assessment checklist was established to evaluate the research papers, based on Kitchenham et al. guidelines [15]:

1. Does the research paper specifically mention the research process ?

2. Is the research process appropriate for the problem statement ?

3. Is the analysis of the study properly done ?

If the study met the assessment criteria then it was given a yes. The complete selection process of research articles is given in figure 2.1.

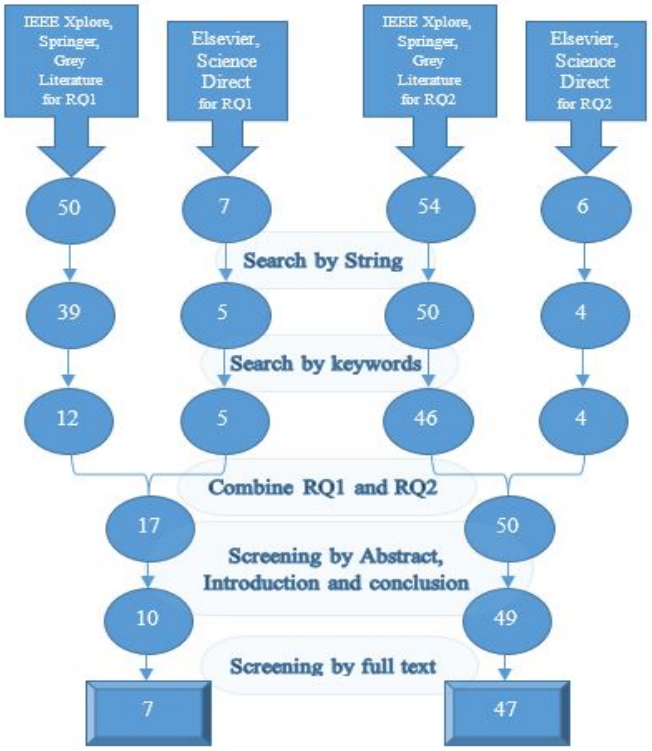

Figure 2.1: Selection Process for RQ1 and RQ2

2.3 Results and Discussions

We have already discussed the answer for RQ1 in detail in chapter 1. Frequencies of the attacks most commonly addressed in literature have been registered in table 2.1. The column headed DoS, pertains to those researches which have discussed general denial of service attack instead of focusing on a particular type of DOS attack.

The subsequent subsections focus on the results and discussions alongwith an analysis of the shortcomings existing in the reviewed techniques, pertaining to RQ2. Out of the total 54 papers that were selected according to the inclusion criteria, 47 researches discussed the detection

mechanisms for DOS on VANETs. The reviewed research papers have been divided into sections according to the type of attack they discuss.

2.3.1 Jamming

Hamieh et al. [11] use correlation coefficient to detect reactive jamming. The attacker waits on the start of transmission by legitimate nodes to initiate its attack. Thus, this dependency correlation is greater in case of jamming attacks than in regular data transmission. ***Shortcomings:*** Inadequate number of performance assessment indicators are stated in the research, for example, detection rate (i.e. how long it takes to detect the attack) is not addressed.

Malla and Sahu [51] have given an anti-jamming solution by using the frequency hopping technique where pseudo-random numbers are generated by cryptographic methods for the hopping algorithm. ***Shortcomings:*** As Orthogonal Frequency Division Multiplexing (OFDM) standard is being used for VANETs so this approach cannot be adopted presently.

The proposed solution in [16] describes the use of the factor, packet drop rate PDR (no. of packets dropped or total packet received) to detect jamming in the network. ***Shortcomings:*** It is imperative to note that PDR could also be low due to issues like collusion, low SNR or congestion, hence other network performance parameters can be used simultaneously with PDR, to effectively detect DoS due to jamming in VANETs.

The research in [18] presents a solution to jamming by detection of collusion of beacons at the beginning of Control Channel Interval (CCHI) because reactive jamming also occurs at this time. As the collusions requires at least two participants, value of the detection parameter must be greater than 2, if there is no attack. A variation in this value indicates a jamming attack. ***Shortcomings:*** This approach does not work efficiently in scenarios with a large number of neighboring nodes.

The authors of [19] propose a method to identify reactive jamming in Direct Sequence Spread Spectrum (DSSS). The chip error rate is compared at the beginning of the frame and the actual frame error rate. If there is a change in this value during transmission, a jamming attack is assumed. ***Shortcomings:*** This methodology is restricted to DSSS-based systems in VANETs.

The technique in [17] employs a detector which checks for the beacons from the vehicles. If a beacon is missing from a group, an alarm is generated. ***Shortcomings:*** The fault with this method is that the detection can occur only if one beacon is missing. This is problematic since jamming causes blockage of radio signals for many vehicles at the same time. Additionally it has strict dependency on fixed beaconing period. This can cause problems if the beaconing is set to be non-periodic. Furthermore this technique only focuses on jamming attack and does not cater for other DoS attacks.

2.3.2 Wormhole and Sinkhole

A Wormhole Attack Detection Protocol using Hound Packet (WHOP) is proposed in [47] to detect an attack in the network. It is based on AODV protocol. A hound packet is transmitted in the network after the route discovery process. This packet is processed by each node. It counts the difference of hops between the neighboring nodes which are located one hop away in the current path. If the hop difference between these nodes is more than a predefined threshold value then it is identified as a wormhole. ***Shortcomings:*** Although, this method has a higher detection rate, but the processing delay introduced due to this protocol makes it inefficient.

The authors of [6] have proposed a technique called, hop-by-hop efficient algorithm protocol (HEAP) to detect against Wormhole attacks in the vehicular adhoc network. Heap is based on the AODV protocol and uses packet authentication to authenticate packets at each hop by HMAC-

based algorithm. This algorithm employs two keys to authenticate the packets and rejects packets on this basis. ***Shortcomings:*** Since this algorithm applies geographical leashes instead of temporal, the traveling distance of packets is limited. This technique does not mention, network overhead, detection rate, false positive rate, false negative rate etc.

Kaushik [31] recommends using a cryptographic scheme to mitigate the threat of message suppression which causes unavailability of information to the vehicles. The technique uses the group key management system to manage the key distribution. ***Shortcomings:*** The dependency on central body in this scheme is inefficient as it can be the point of attack. Furthermore, the drivers are unwilling to be dependent upon a central supervising body due to privacy reasons.

Gandhewar et al. [52], present a technique to counter sinkhole attack in VANETs. AODV Broadcasts RREQ packet to all neighbors to discover the shortest and optimum route to destination. The parameters of this RREQ packet such as hop count, source/destination address and destination sequence number are stored and compared with the parameters of the new RREQ packets. If the current source sequence no. is very large then the node is considered faulty. ***Shortcomings:*** The problem with this scheme is that it has a very large end to end delay. Although the throughput is high but the detection rate, false positive and negative rates have not been determined.

2.3.3 Denial of Service

Ruj et al. [7], have proposed an attack detection scheme by observing the messages being sent by the nodes. V2V communication involves exchange of information like alert messages and regular beacon messages. There are certain alert messages that become expendable after some time such as changing lanes or slowing down in a particular location. If the distance becomes greater than a threshold value, it is considered useless. ***Shortcomings:*** Although this approach

seems workable but no performance testing or validation has been performed in the research. Only considering sent and received time for position verification is not sufficient as it can change due to road conditions.

Grover et al. [8], have proposed an approach for misbehavior detection using data mining techniques. They have concluded that J48 and random forest are appropriate for intrusion detection. ***Shortcomings:*** Random forest is known to take a long time to build models furthermore these classifiers depend on complete data sets before making their decisions. It is also noted that the performance efficiency on training phase is a major challenge for their work. We aim to apply a more recent data mining technique for intrusion detection which works on real time streaming to evaluate the traffic scenario.

VANETs that are using the IEEE 802.11p channel access EDCA mechanism are vulnerable to a synchronization-based Distributed Denial of Service attack. This occurs due to reiteration of transmissions and small contention window sizes [45]. ***Shortcomings:*** The problem here is that since broadcast communications in VANETS do not have acknowledgments, therefore, the sender and receivers of periodic transmissions will not find out about the attack.

Kaur and Mahajan [32], suggest using data analysis to detect the Denial of Service attack in VANETs, by asserting that DDoS has certain pattern behavior which can be used to distinguish it. When a node gets a request it examines the payload. If a large number of packets are coming in with the same payload, then it is considered an attack. ***Shortcomings:*** Although, the performance of the network after employing this technique is shown to improve but attack detection performance metrics like detection rate or detection time have not been mentioned. Signature based authentication is used here to thwart DoS.

Malhi et al. [28], have proposed a Decision Inference System (DIS). The approach applies XML

dependency trees. Certificate authority combines the private key of the nodes with the public key to generate the full key. If the vehicles alter their credentials they can be verified by the CA, because all the information is sent to the Certificate Authority. ***Shortcomings:*** This system depends upon the verification of digital signatures by an external entity i.e. Certificate Authority (CA). This not only poses a requirement for an extra infrastructure but also assumes RSUs and CA failures or corruptness to be negligible. Such assumptions are not security efficient. Furthermore, extra resources or bandwidth is used in the exchange of information.

As applications provided by VANETs are intrinsically time sensitive, hence timing attack is another issue which causes non availability of message or DoS by delaying the time of reception of information by the nodes in the network. The authors of [1] suggest using time stamping methodology to thwart this attack. ***Shortcomings:*** This technique requires the time synchronization between various nodes of the network.

The authors of [27] propose a methodology to make use of fuzzy inference engine to analyze various aspects of the network such as log files, routing tables. It further applies pattern recognition or data clustering techniques to detect an attack on the vehicular network. ***Shortcomings:*** This concept has not been tested or analyzed to show results by the authors.

In [22], Denial of Service attack is being detected using time slot. The Road side unit documents the time at which request is sent and also the number of vehicles which are sending these requests. If the number of packets sent by a node doubles then it is marked as a malicious node. ***Shortcomings:*** Although this technique improves the false positive rate, still the approach employed by this scheme is inadequate because it depends on the RSU to detect the DoS attack. If this node is not in working condition or is a victim of an attack itself, then the detection for DoS attack on other nodes becomes impossible.

2.3.4 Malicious Node

Khan et al. [9], propose a method of detection of Denial of Service attack based on their Detection of Malicious Nodes (DMN) algorithm. In order to improve the network throughput and performance, they select only a few verifier nodes based on parameters like distance and load. The malicious node is selected if they drop packets during transmission and repeat this behavior. ***Shortcomings:*** The same problem of neighbor node or RSU dependency, and loss of privacy due to the introduction of CA are observed in this technique.

Bansal et al. [20] divide the network into two sections Lower Level Nodes (LPN) and High Level Nodes, where the local protection nodes are selected from LPN to assess the attack on other nodes. This is basically a defensive or protective node for its neighbors. The LPN investigate the Packet Drop Rate value and compare it to a threshold value. When these two values become equal, the LPN sends a monitor mode message to other vehicles in the network to analyze them. If a vehicle injects a large number of false packets into the network, it will be marked as an attacker. Only packet drop rate is chosen to judge the maleficent nature of the nodes. ***Shortcomings:*** As discussed before, PDR can change in a network due to other network characteristics, hence sole dependency on PDR for this assessment is not efficient. Other performance metrics should be included for an effective detection process. The communication overhead created by the transmission of reports by the verifiers to the Certificate Authority lowers the network throughput and performance. Furthermore, detection of saboteur nodes depends on neighboring nodes and RSUs to make the final decision. An independent approach is better for an unerring evaluation.

A method called Efficient And Lightweight Intrusion Detection for Vehicular Networks (ELIDV) is proposed in [21]. The authors in this research paper recommend detecting intrusions in the network through trust, by assigning reputation scores to vehicles. In this case, the RSUs are

employed to compute these scores, while the Certificate Authority aggregates them. This process leads to detection of malicious nodes which threaten the network with DoS attack. ***Shortcomings:*** The dependency upon other nodes, vehicles and RSUs for detection of DoS attack is a cause of concern. This weakness or shortcoming should be addressed in the proposed scheme to enhance the potency of this strategy.

The approach used in [30] is the attacked packet detection algorithm proposed by Roselin et al. in their paper [23] to detect the attacked packet by setting a time stamp dependent threshold using RSU. A database of these nodes is maintained at the Road Side Radio Transductor (RSDT). It discards new request coming from the same node. ***Shortcomings:*** This method has huge dependency on the innocence of the RSU or RSDT which is a single point of failure. As applications provided by VANETs are intrinsically time sensitive, hence timing attack is another issue which causes non availability of message or DoS by delaying the time of reception of information by the nodes in the network.

Chen et al. [53], suggest to employ an additional authority other than the certificate authority such as the vehicle manufacturer to work in a group for the purpose of managing the security of the vehicular network by a group signature based methodology. ***Shortcomings:*** There is an assumption of trust on the vehicle which is a cause for concern since such an assumption can cause insecurity in the network. This approach not only requires the expenditure of new hardware, but also might be unacceptable by drivers as it impairs their privacy. Another problem of this scheme is that different messages, with similar semantics due to same urban traffic scenario, cannot be handled by it.

The research in [26] adopts the approach identified in [24] to create a DoS detection scheme known as Malicious Node Detection Algorithm (MVND). ***Shortcomings:*** The methodology has

the same discrepancies as established with the approach employed by [25] and [24].

The authors of [25], like earlier discussed schemes, uses packet drop ratio (PDR) to categorize nodes as malicious. ***Shortcomings:*** As we have noted before PDR can be high due to network conditions, high mobility of nodes as well as varying geographical road conditions. Hence, only using Packet Drop Rate as the main criteria for identification of an attack is insufficient.

2.3.5 Sybil

Grover et al. [44], propose a neighbor based cooperative information scheme to detect Sybil attack. In this scheme, , each node keeps a record of its neighboring nodes and a threshold value is set. If some nodes observed that they have similar neighbors for a period of time, then these stagnant neighbors are classified as Sybil nodes. ***Shortcomings:*** This technique requires periodic communication between vehicles, but it cannot detect the Sybil nodes in the scenario where the attack duration is shorter than the threshold value . This method is a detection paradox for the saboteur since it depends highly on the loyalty of the neighbors and inaccurate assumption that all neighbor nodes are legitimate.

The authors of [43] adopt social network to counter the threat of Sybil attack. It suggests that true neighbors will quickly build trust relationship with each other while sybil nodes which have forged identities will be unable to do so. ***Shortcomings:*** Using social networking for detecting this attack has problems, since the ever changing topology of the vehicular adhoc network results in fast alteration of the neighborhood plus very few evaluation parameters are given e.g. no indication of detection time has been provided.

Yan et al. advise the use of GPS information in order to construct the key for the detection scheme of Sybil attack [49]. ***Shortcomings:*** The shortcoming in this technique is that, GPS information

is publicly available and an attacker can use it to launch a sybil attack.

The technique in [34], uses elliptic curve cryptography and the use of pseudonyms to establish a basis of detecting malicious vehicle in the vehicular network. A government authority is used as a certificate authority to manage the pseudonyms of all the nodes. Two types of road side units are required here. One of these, issues the pseudonyms which is responsible for normal communication and the other one provides services for authentication. It will overhear the communication messages among vehicles for a particular road segment. ***Shortcomings:*** Not only does this approach require extra infrastructure, but it also requires this infrastructure to be uncompromised.

The authors of [35] employ a technique called P2DAP, where the interaction of department of motor vehicles (DMV) is required to manage the security of the system. Pseudonyms for each vehicle are generated and hashes for each pseudonym are calculated using a global key. Some bits are selected from this hash value and are called coarse grained hash value. The same coarse grains hash values are added to a group. Global key that is used to generate full key is sent to all the RSUs, but the private key is kept secret. There is a lack of safeguarding the privacy of the nodes in many researches attempting to thwart Sybil attack, but here it is protected. ***Shortcomings:*** The drawback of this technique is, that it uses the standard, time-consuming Public Key Infrastructure (PKI) model, albeit in a disparate form. Moreover, it does not cater for stolen keys and pseudonyms. The exchange of information between the DMV and RSUs can exhaust the network performance and cause latency. Furthermore, if the RSU is compromised there is a danger of privacy leakage.

The technique in [50] uses cryptographic method for detection of Sybil attack by confirming encryption function with the authentication key upon reception of the message and comparing it with the signature. ***Shortcomings:*** This approach has high cost, high bandwidth usage and does

not work efficiently in urban environment with complex routes, due to large number of vehicles.

The mechanism employed in [5] can detect spoofing for application level DoS attack. A Bloom Filter based ipchock reference (BFICR) approach is employed for the detection. A data structure is used to record applicable traffic information in a table of fixed length, after which the BFICR method is used to discover sudden variations in the features of traffic. These variations are due to flooding attack. ***Shortcomings:*** The detection rate of this approach is high but it has a high data storage overhead. Furthermore it is unable to detect other types of DoS attacks e.g. jamming.

Chang et al. [39], recommend a Sybil node detection scheme based on vehicle trajectories. Each vehicle demands an authorized data packet from the RSU, with a clear time stamp. This time stamp is used to prove the vehicles location at a certain time aiming to preserve its privacy with this identification. ***Shortcomings:*** The drawback of this approach is the fact that Road side units use long term identities to generate signatures and this leaks the location of the vehicle compromising the privacy of the nodes. Since, it uses the RSUs to get a signature as a proof for the vehicle to be at a certain location at a particular time, hence this technique is predisposed to being dependent upon a forged RSU. Moreover, it is susceptible to colluding attack.

Feng at al. [40] assert that Sybil attack leads to Denial of Service attack. It further recommends using a technique called Event Based Reputation System (EBRS) to preserve the privacy of vehicles by using a pseudo identity and alleviates the need of using neighbor information by independently detecting Sybil nodes. Furthermore it has the capability of detecting multiple Sybil attacks from forged or stolen identities. ***Shortcomings:*** This technique largely depends on the Road Side Unit for issuing the certificates of the vehicle in its communication range. This dependency might be problematic since the RSU might already be compromised.

Vehicles attains motion trajectory information by requesting the RSU signature and employ the

Time Stamp Approach (TSA). If the motion trajectory information is similar for more than one vehicle, then they might be Sybil nodes. The technique in [41] has the dilemma of loss of privacy, which is a bone of contention for drivers. Furthermore, it cannot rule out the Sybil attack with embezzled movement trajectories also known as the case of the conspired Sybil attack. ***Shortcomings:*** A critical glitch in using time stamp dependent techniques is the assumption that two vehicles with similar trajectories cannot be at the same RSU at the same time. This modus operandi can be adopted in highways but it is rendered useless in heavy traffic areas of urban roads.

The authors of [48] propose a group-signature-based schemes. They create a group of vehicles and assume that the messages received from them are reliable. As duplicate signatures cannot be signed by the same node, hence Sybil attack is prevented. ***Shortcomings:*** The problem with this method is that different messages with comparable attributes may be disregarded from thwarting this attack.

The authors of [42] use the signatures broadcasted regularly form Road side units to infer the motion trajectory information and employ it in the same manner as described in [41]. ***Shortcomings:*** This technique also has the same problems faced by [41]. Furthermore, using graphs and analyzing complex trajectories due to the nature of traffic in urban areas becomes complicated. Some compromised RSUs will also introduce a new malicious factor to the approach.

Yu et al. [36], charter signal propagation model and Received Signal Strength Indicator (RSSI) to estimate the position of the vehicle in this paper. This value, is then compared with the position that the vehicle is announcing. If the calculated location of the vehicle is different from the one it is advocating, it might be a Sybil node. Since, it depends on the signal strength values for detection, it cannot guarantee accurate results because if Sybil nodes are near an innocent vehicle,

it is hard to distinguish them. ***Shortcomings:*** As RSSI value is dependent upon complex road conditions, hence this method might become inaccurate. This technique depends upon neighbor vehicles to be present and in some cases, there might be no neighboring vehicles to provide evidence. Furthermore, there is a privacy violation of the nodes due to the fact that the identity and position of the nodes is shared with other nodes. It should be further noted that malicious nodes can collaborate with each other and give incorrect evaluation.Moreover, transmission rate might be required to be increased or decreased by legitimate nodes as well. Hence, this technique cannot be concluded as a sole method of detecting DDoS attacks.

A signature based intrusion detection method has been proposed in [37]. The technique relies on the existence of few legitimate vehicles in the network to isolate a single forged vehicle, which is aiming to mislead the legitimate nodes. ***Shortcomings:*** As the recommended arrangement is projected to take help from other vehicles which might be illegitimate themselves, hence it is not suitable to detect DoS due to Sybil attack. Other issues with using neighbor nodes for the detection process are explained in the problems for [36].

The attack detection technique presented in [38] depends on the Road Side Unit to collect the beacon packets and calculate the distance of the vehicles. It selects the Sybil nodes by analyzing the difference of this distance value. ***Shortcomings:*** This scheme has a large overhead and it gives a high false positive rate for a small network, though it works well in a large network.

2.3.6 Malicious Data

The authors of [29], propose a technique called Misbehavior Discovering (MisDis). This method can detect the misbehaving vehicle by executing state automata and supervision. It also utilizes a distinct security log to observe the behavioral features of a specific selected node. It implements

ideas of peer view system that provides accountability in distributed systems. Any node can demand the log of another node from the Department of Transportation (DoT) and then judge accordingly if it has strayed from normal behavior. ***Shortcomings:*** The drawbacks of this technique are that the validity of this data is not accounted for and there is no assurance of non-malignant logs. VANET is an ephemeral network and a detection mechanism based on cooperation is not efficient. The transmission of logs for each node will cause a communication overhead reducing the network throughput or performance. Furthermore, no evaluation or performance test has been conducted to appraise the performance of this scheme.

The scheme in [33] provides a pre-authentication mechanism before a signature verifying process. The pre-authentication process makes use of one-way hash chain and a group re-keying scheme. In this scheme, if the message has passed the one-way hash chain-based authentication, the signature verification process of the message will be carried out by the receiver, otherwise it rejects to verify it. ***Shortcomings:*** This technique has a large authentication delay and does not mention false positive, false negative and detection rate etc. Furthermore, it fails to detect attacks form insiders.

An Attacked Packet Detection Algorithm (ADPA) is used in [23]. This algorithm calculates the frequency of the broadcast packets by using the velocity of the vehicles. It uses frequency and velocity to judge the position of the vehicle. If both these values are high, it is assumed to be regular traffic otherwise if the values are low, the vehicle is considered static and fraudulent. ***Shortcomings:*** This algorithm is only run on the RSUs instead of all the nodes, independently. Furthermore, this algorithm only works on one car at a time and cannot manage invalid requests sent from multiple vehicles. Moreover, it depends on RSUs for threat detection, if RSU is not functioning or is compromised then this algorithm cannot work effectively. It also has a delay

overhead issue which needs to be resolved for prompt detection.

In [24], Quyoom et al. have employed the same methodology of Roselin et. al in [23] to detect Denial of Service attack. The only difference is that instead of depending upon a single threshold value to determine a false packet, they have kept a lower and upper base line. If the frequency and velocity values of the vehicles are between these two threshold values, then the vehicle is considered safe, otherwise they claim it is malignant based on their Malicious and Irrelevant Packet detection algorithm (MIPDA). ***Shortcomings:*** This approach not only lacks in effective detection methodology due to dependency on RSUs for running the algorithm but also, it is unable to handle multiple malicious packets at one time.

2.3.7 Greedy Behavior

Mejri et al. [13], have used the linear regression and watch dog technique to detect greedy behavior of nodes in the network. In a normal network a node connects to the support for a certain period of time and the next node has to wait for that time after which it transmits. Greedy nodes violate this rule, thus increasing the coefficient value calculated by this time dependency. ***Shortcomings:*** This technique can only detect greedy behavior and cannot identify other Denial of Service attacks.

A metric called packet entropy has been used in [46] to detect DoS in VANETs. Packet entropy is calculated on the basis of the emissions of data and acknowledgment packets. Since these values are higher for greedy packets, therefore entropy of these packets will be higher. ***Shortcomings:*** The parameters chosen for computing the entropy are not adequate for detecting other Denial of Service attacks.

2.4 Conclusion

We have inferred from the discussion in chapter 1 and from table 2.1 that the most critical and frequent attack on vehicular adhoc networks is denial of service attack, whether it is in the form of sybil attack, or blackhole attack, or any other form of attack disturbing the availability and utilizing the resources of the networks rendering it useless for legitimate consumers. DDoS attack is an even more severe form of DoS attack since it employs bots to propagate the attack and hence the attack radius and destruction increases manifold. Furthermore, it becomes extremely difficult to find the source of attack.

An overview of table 1 shows that almost 31% of the selected researches focus on Sybil attack followed by general DoS at 19%, malicious node 17%, jamming 14.8%, malicious data 12.7%, wormhole/sinkhole 6.4% and greedy behavior 4.2%. As mobility of vehicles in VANETs contributes to the difficulty in ascertaining the fraudulent vehicle location during a Sybil attack, hence it is the most pertinent attack on VANETs. This might have led maximum numbers of researchers to select âĂIJDetection Schemes Against Sybil Attack in VanetsâĂİ as their research field, thus reflecting the result in Table 1.

Figure 2.2 makes a comparison between the number of related research papers vs. year they are published in to get an idea about the progress of work in this field. These are the papers that have been approved after the selection process described in figure 2.1. It can be seen from figure 2.2, that maximum research work which has been addressed in this literature review is from the Period, 2013-2015. This assures that latest research in the area has been selected for the purpose of accurate and systematic investigation. After an overview of table 1, it is clear that the most frequently examined attack in this specific period is Sybil attack. A thorough analysis of the defense mechanisms for Denial of Service attacks in vehicular adhoc networks has been performed

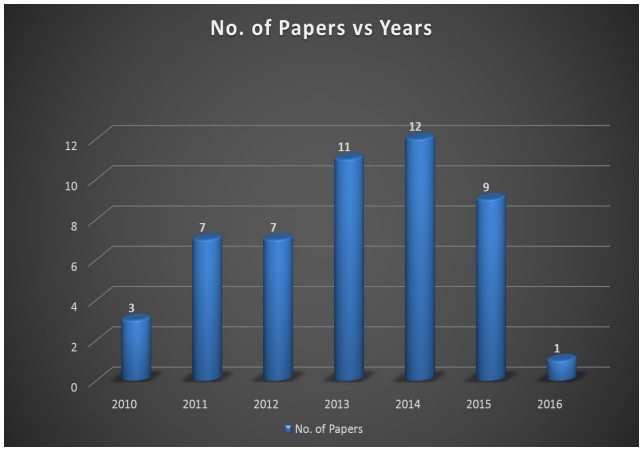

Figure 2.2: No. of papers vs. year they are published

which establish that there are certain shortcomings in the proposed techniques that hinder the process of provision of an ubiquitous Inter Vehicular Communication System. The schemes include vehicle location/trajectory, variations in packet/network features, use of trust based relationship/reputation or implementation of extra infrastructure for the process of detection of distributed denial of service attack in VANETs. Although a number of cryptographic methods have also been used for the detection and prevention of saboteur attack in Inter Vehicular Communication network but these come with the burden of complexity and loopholes, which have been thoroughly analyzed in Section 2.3. On the other hand, data mining and most significantly decision trees have not been effectively exploited for the purpose of attack detection in vehicular networks.

Table 2.1: Frequencies of Occurrence of Attacks in Current Literature

Reference	Jamming	Worm hole/sink hole	DoS	Malicious Node	Sybil	Malicious data	Greedy Be-haviour
[38]				x	x		
[6]		x					
[43]				x	x		
[13]							x
[46]							x
[29]						x	
[7]			x				
[8]			x				
[9]				x			
[20]				x			
[24]						x	
[45]			x				
[32]			x				
[28]			x				
[1]			x				
[21]				x		x	
[34]					x		
[11]	x						
[36]					x		
[35]					x		
[51]	x						
[33]						x	
[50]					x		
[31]		x					
[52]		x					
[5]					x		
[30]				x			
[27]			x				
[22]			x				
[39]					x		
[40]					x		
[53]				x			
[16]	x						
[41]					x		
[49]				x			
[17]	x						
[18]	x						
[19]	x				x		
[48]					x		
[42]					x		
[14]	x						
[44]					x		
[47]		x					
[23]						x	
[25]				x			
[26]						x	

Chapter 3

MAP GENERATION AND MOBILITY SIMULATIONS

3.1 Requirement of Simulation

VANET is a popular nascent research area. Testing of emergent solutions and proposed schemes is essential before its deployment to assess the amount of trust that investors can build in the success of this technology.

The chapter is organized as follows. Section 3.1 explains the need for using simulation for the testing of the proposed scheme and to mirror appropriate vehicular conditions to examine our technique. The selection of a map generator and the method of creating the road topologies has been explained in section 3.2. Section 3.3 discusses the traffic mobility model. It shows the method to create this model as well as tailor it according to our requirements. It also examines the formulation of tcl script required for compatibility with NS-2. Section 3.4 delineates the use of OSM for importing a real map of a busy city area of Pakistan. Finally the concluding remarks are given in Section 3.5.

The testing would involve and require the following:

(a) expenditure of transportation services

(b) provision of vehicles

(c) real road conditions (traffic,signals,parking etc)

(d) road topology

(e) road traffic flow

(f) modeling driver behavior in various conditions

(g) road side infrastructure

(h) provision of radio signal propagation

(i) provision of network protocols

In order to achieve the above mentioned objectives, it would become extremely expensive if such conditions were to be met in real time scenario. Specifically, in the case of a large scale implementation the budget would sky rocket. Simulation of VANETs is an optimum approach towards acquiring the same results in a more frugal and controlled environment. Simulations come with the shortcoming of introducing errors in the final results. This can be reduced by experimenting and collecting the data repeatedly over the same scenario conditions to rule out erroneous conditions.

Many simulation tools have already penetrated the market which combine communication and networking to evaluate the performance of network protocols in different types of networks. These include OPNET, Qualnet and NS-2 simulators [54]. The problem with these simulators is that they provide a conventional simulation arrangement and do not cater for the nuances of vehicular and transportation platform. There are certain tools that have been specifically categorized and created for use in research involving transportation scenario. These include CORSIM, PARAMICS and VISSIM. This effort still left a need to integrate these two kinds of simulators to bring forth a framework that is able to evaluate the network protocols and schematics over a certain

transportation scenario. The process of imitating a vehicular network and the interaction between a traffic simulator and a network simulator is shown in figure 3.1.

The first step in simulating VANETs, is to generate a map for the type of road and area that is

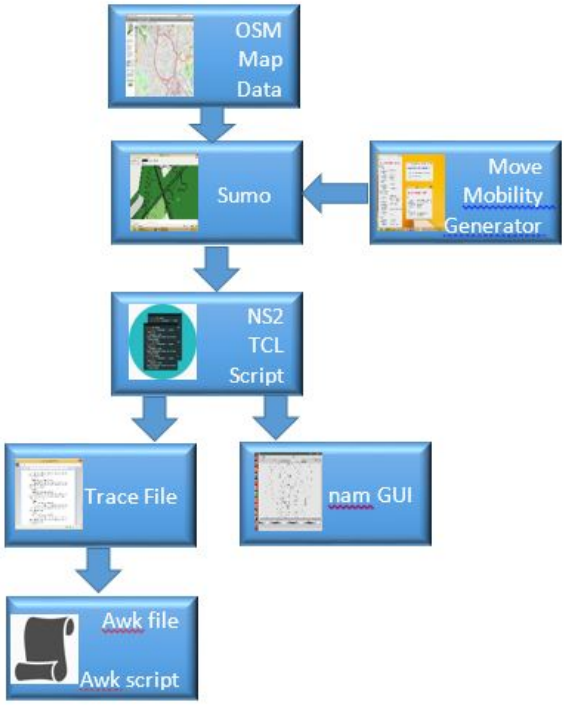

Figure 3.1: Interaction between Traffic and Network Simulator

to be considered for the traffic [55]. Without the formulation of a proper road map, the created network would reflect an ordinary wireless adhoc network, instead of an appropriate vehicular adhoc network. The second step is to represent the reactions of the drivers based on the road and traffic conditions. The third step is to measure the traffic origin, destination and time it takes for the trip. Many research papers have included simulations of VANETs based on a conventional wireless adhoc networks by applying road topology models like Manhattan grid model or Random Way Point model (RWP) [56]. Application of these road topology models is appropriate for research on MANETs and previously mentioned WSNs but VANETs have specific topology re-

quirements. In order to satisfy the demands of this unique network, the ability of the simulation to show independent traffic flows on roads and junction is necessary to emulate the physical logistics of transportation in VANETs.

Map and topology generation first depends on the definition of traffic models. The traffic models are divided into three groups levels based on level of detail they represent [55]:

Microscopic: represents individual view with all vehicles having independent speed and reactions.

Mesoscopic: represents individual view but all vehicles on the same road have the same speed.

Macroscopic: represents an aggregate view, as opposed to individual flow of vehicles. It is based on hydrodynamic phenomenon.

This suggests that the most realistic representation of traffic modeling can be achieved by the microscopic level as respective vehicles will require precision of speed and position reporting, for reaping the benefits of codependent driving and the general advantages of VANET applications. The location measurements have to be aligned to meters and accuracy is required to ensure less error in data set. Furthermore, the densities of roads and highways, obstacles on these roads, speed limits and allowance of one-way or two-way traffic has to be delineated.

3.2 Map Generation by SUMO

Generating a map is the first process to be considered for the realistic simulation of VANETs. We have used two methods to depict road infrastructure in our research. The first method is to create our own map. This was achieved by applying two lane traffic, straight lines with curves and junctions. The edges represent the road and the vertices represent the junctions. The road map was created by using a mobility model generator MOVE, which allows the generation of a road

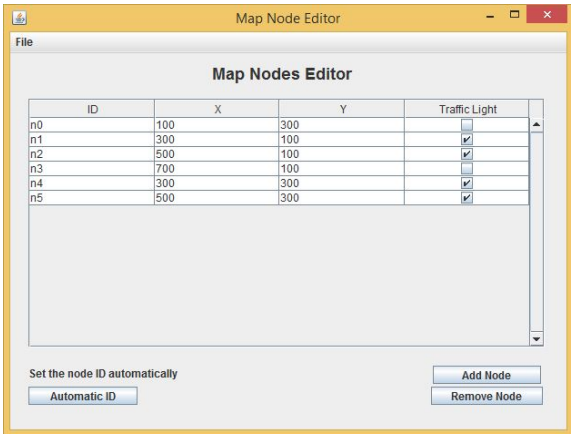

Figure 3.2: Node Editor

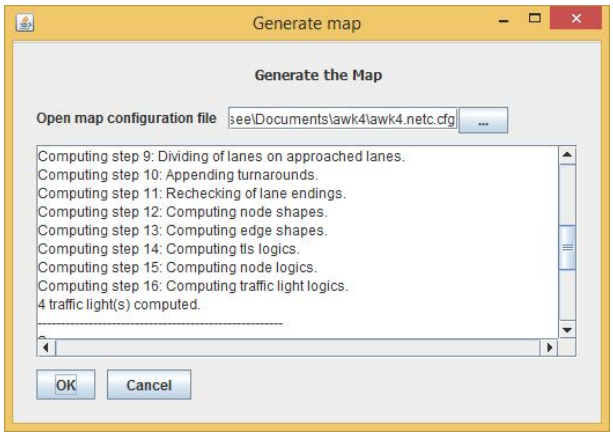

Figure 3.3: Traffic Signals

map, either manually or by input of an open source like google maps or TIGER database from the US Censor Bureau [57]. Basic system requirement for MOVE is the JDK package. There is another option of automatic map generation which offers three types of random maps i.e. spider, grid and random networks.

MOVE is built in JAVA and runs over an open source traffic simulator SUMO. MOVE is compatible with SUMO version 0.12.3. It primarily consists of two generators, map editor and the vehicle movement editor. First we have to select the mobility model generator to run the map editor. For creating the manual map, we chose the "node" button in the mobility model generator dialog box. Node represents a point on the map e.g. a junction or road end. This could also be a traffic light. This takes us to map nodes editor shown in figure 3.2. We chose a road topology consisting of six nodes, with a total area spanning 600 units for x-axis and 200 for y-axis. Traffic lights were also installed to simulate real time reactions of vehicles in case of a pause at these traffic signals. These lights were placed at the four junctions n1, n2, n4 and n5. The assignment of these traffic lights has to be confirmed after the creation of the configuration file, as there is an anomaly in MOVE which renders the traffic lights ineffective on the first run of the configuration file. We

remake them in this case and save the configuration file again shown in figure 3.3.

The road editor is opened to lay out the parameters for the road infrastructure and specify the edges. These edges represent the road between two points on the map. Identities of the roads are mentioned by depicting the direction of movement of the vehicles. For example, in case of the road between node 0 and 1, two directions are mentioned namely, U01 and D01 which represent up and down movements respectively. Similarly, if the nodes lay such that the movement is horizontal, the depiction can be L12 and R12 for left and right movements between nodes 1 and 2. This can become clear by viewing figure 3.11. Number of lanes are set as 2 in our case in order to portray urban city scenario. Speed is set at 20, which is about 70km/hr, representing the nominal speed in a city or highway. Moreover, priority is set to be same for all the vehicles to randomize the simulation. Road editor is shown in figure 3.4. For map configuration file, we go

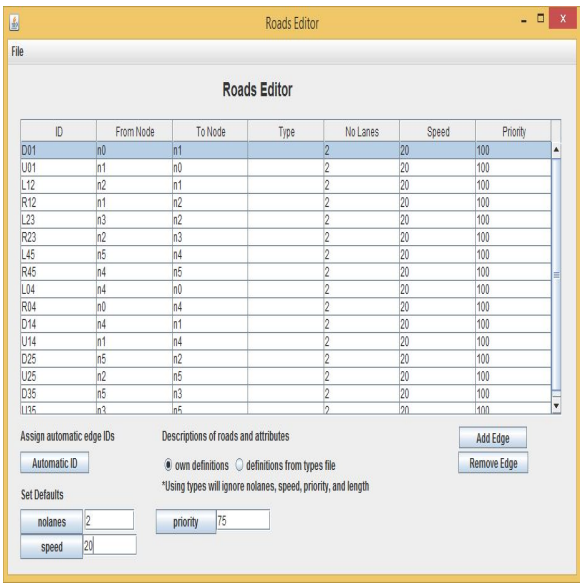

Figure 3.4: Roads Editor

to the map configuration editor. Here the nodes file is selected as nodes.xml , roads file is selected as edg.xml and the output file is set as .net.xml. The lane numbers, speed and priority have to be again specified in this editor. Next the file is saved as .netc.cfg. This process is depicted in figures

3.5 and 3.6.

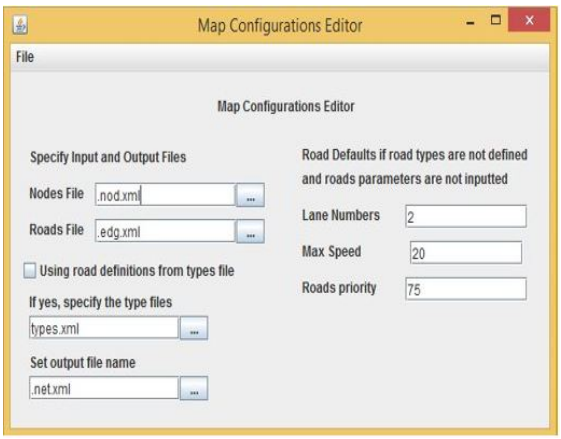

Figure 3.5: Map Config Editor

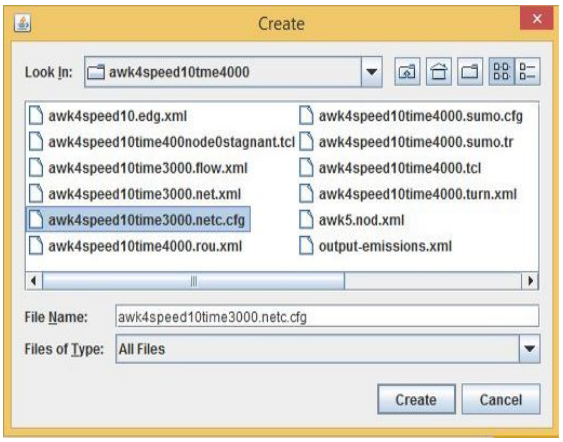

Figure 3.6: Saving the config file

To generate the map, this .netc.cfg file has to be selected from the folder where all these files are being saved. Next, the Vehicles Flow Definitions Editor is selected from the Vehicle Movement Editor. This allows us to select the flow of the vehicles in a particular direction of the prioritized routes as per our requirement.

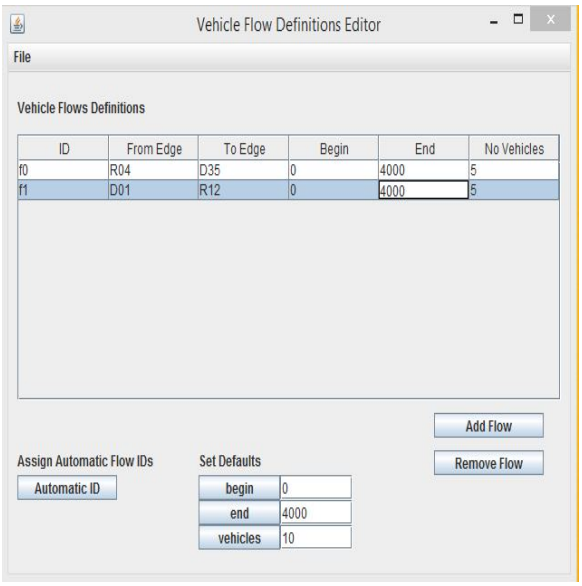

Figure 3.7: Vehicles Flow Definitions Editor

For automatic movement, vehicle flow has to be defined where a fleet of vehicles move in the same direction. Starting position, destination, start time, end time, number of vehicles, and the inter-departure time of the vehicles form the starting position have to be delineated. Probability of turning in different directions can also be specified. For manual description, we can specify various attributes such as number of vehicles, acceleration, deceleration, start and end time, duration of trip, start and destination positions, pause times etc. As can be seen from figure 3.7, we have selected flows from R04 to D35 and from D01 to R12. The number of vehicles has been kept small to run a controlled simulation to assess the changes of the road topology and protocols on the network parameters. The time of the simulation had to be increased to 4000 seconds, in order to get maximum data, since data mining algorithms require a huge amount of data for testing purposes. With a small data collection, the results of the simulations are not guaranteed to be optimum for correct simulation or results. From edge R04 to edge R45, the junction turning probability is set at 20% and from edge R04 to D14, it is set at 80%, shown in Figure 3.8.

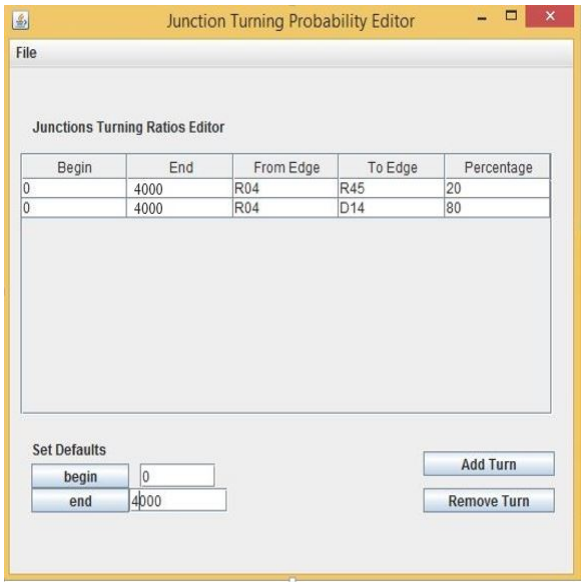

Figure 3.8: Junction Turning Probability

This process is followed by manually setting the movement for each vehicle, its parking, acceleration or declaration and pause times. Instead we selected the automatic vehicle movements generated for random movement selection for each vehicle, since this level of detail is not expected to influence our research criteria. After the assignment of the flow definitions file, turn definitions file and the map file, the output file is saved as as rou.xml. This can be seen from figure 3.9.

As already discussed we need a large amount of input data for data mining algorithm. We in-

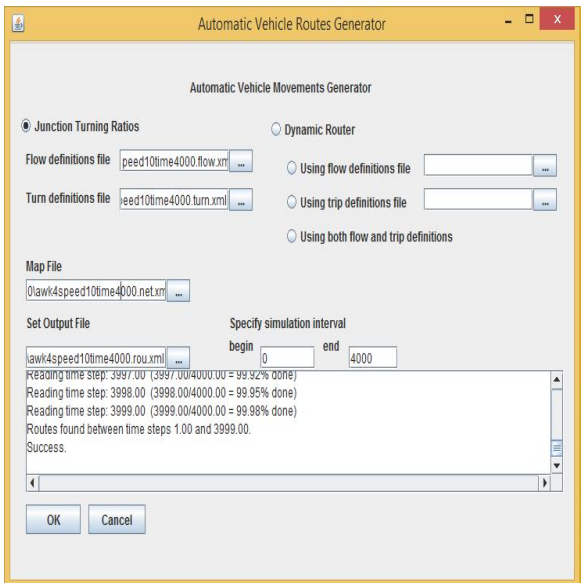

Figure 3.9: Automatic Vehicles Routes Generator

creased the simulation time in vehicle flow definitions editor to 4000 seconds. But after running the subsequent steps of traffic mobility program, we found out that the vehicles disappear after a few hundred seconds, thus rendering our assumptions about maximum data, pointless. In order to fulfill this requirement, we open the rou.xml file in a text editor and increase the route edges for each vehicle. This increases the distance that each vehicle has to cover before it disappears from the map. This is shown in figure 3.10.

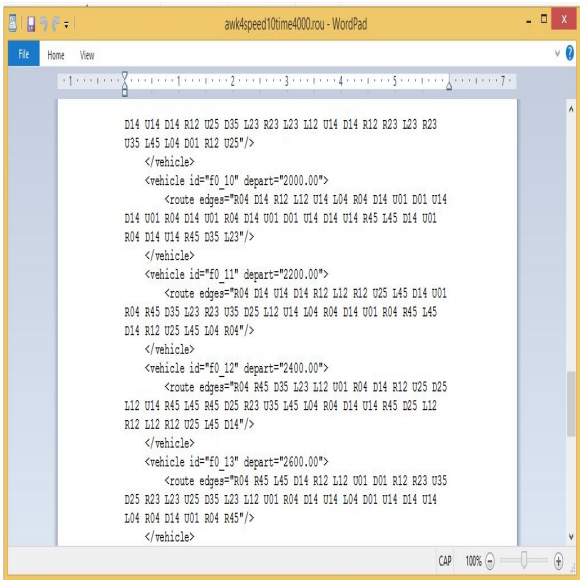

Figure 3.10: Editing the route file

After selecting the map file (.net.xml) and routes file (.rou.xml), the traffic simulation configurations editor sets an output trace file (rou.xml.sumo.tr) and saves the traffic file as sumo.cfg. By selecting the option of "Run Simulation" a new window is opened where the sumo.cfg configuration file is loaded to show the map that we have created by the whole process. This can be viewed in figure 3.11.

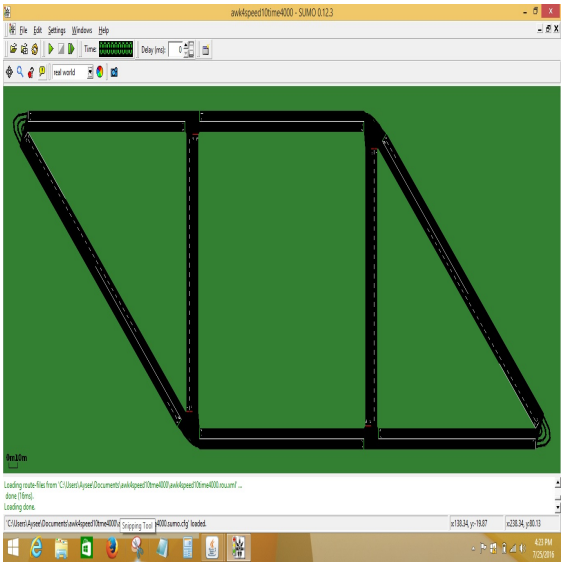

Figure 3.11: Final generated map

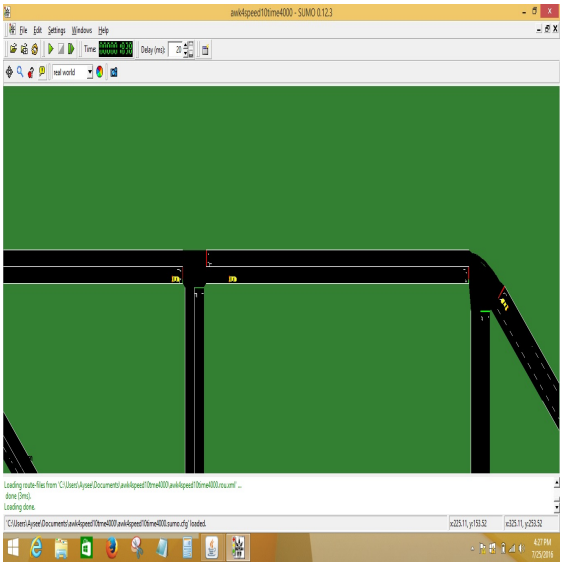

Figure 3.12: Vehicles stopped at red light

The play button is used to start the simulation. The delay option box allows to slow down the simulation in order to view the movement of the vehicles easily. The real world is selected for optimum viewing. As can be seen in figure 3.12 vehicles are also paused at the traffic lights.

3.3 Traffic Model Generator

Now the map has been produced along with the vehicles movement. Next, the network traffic is generated by the Static Traffic Model Generator for NS-2 shown in figure 3.13. The MOVE tracefile has to be imported from the file menu, The click boxes are selected for agent trace, MAC trace, Router trace, Movement trace, NAM trace. The NAM trace file, and the trace output file has to be designated in the list boxes and their output destination is assigned. When this is done, the mobile nodes (vehicles) starting positions with their time and IDs are filled automatically. Here onwards, the "node" depicts the vehicles. To set up the connections each source and destination node along-with start/end time and protocol, is filled manually. We want data traffic to flow from each node to node 0, as we will calculate the selected traffic parameters at node 0. The output file

is saved as net.xml and then finally tcl file is generated.

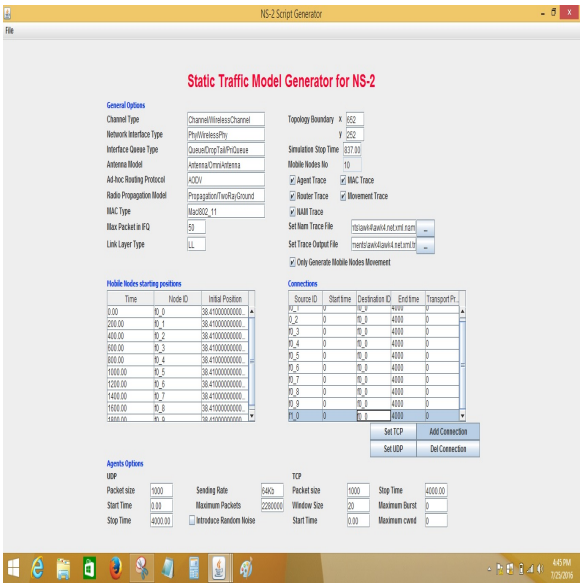

Figure 3.13: Traffic Model generator

The dominant factor influencing the choice of MOVE and SUMO for the mobility and traffic generation for our research was the support they provide for the automatic creation of script to run the required network protocols. The output script which is achieved from SUMO is compatible for input into NS-2 and Qualnet. Vissim, NCTUns, BonnMotion [58] were previously inspected for this purpose but the combination of MOVE+SUMO gives the most user-friendly interface for manual/automatic map production, mobility model and TCL (Tool Command Language) script. The other mentioned programs are either difficult to navigate or do not provided compatibility with popular network simulators, e.g. NCTUns does not support the trace format of NS2 [59]. Furthermore, they do not have ample map/topology options required for realistic simulation.

3.4 Map Generation Using OSM

In Section 3.2, we have created a map by introducing six junctions and a self-made road topology. This was done to investigate our results on a small scale to rule out any errors in our framework.

This was followed by loading a real map of a dense city area of Rawalpindi and Islamabad. For this purpose Open street map was applied to capture the respective area's geographical data. OpenStreetMap [60] is af collaboratively developed, user generated map, built by a community of mappers who dispense data about roads, road side infrastructure and popular destinations etc. to populate this map for free usage all over the world.

The site for open street map is equipped with a search toolbar. This allows the user to input, either the name or the geographical longitude and latitude parameters of the area, in order to gather the geo-data of desired location. We input "Faizabad Rawalpindi" in the search box. It resulted in the option "Faizabad, Islamabad, Islamabad Capital Territory, 44000, Pakistan". By clicking on the hyperlink, it gave a graphical depiction of the demanded region. By clicking the "export" button, it allows the option to either opt for the automatically appointed coordinates, or manually select the preferred areas from the pictorial depiction. We opted for the latter to expand the specific region. This process gives the output in the form of .osm file.

An exhaustive road scenario can be produced by using the server.py script. In this case, the road network is put together with options which are appropriate for the prerequisite traffic cases. In our case, we required more control over the generation process, hence we used the NETCONVERT call to import the OSM files. The following typemap file shown in figure 3.14 is required to interpret the OSM file which provides additional polygons.

```
<polygonTypes>
    <polygonType id="waterway"               name="water"       color=".71,.82,.82" layer="-4"/>
    <polygonType id="natural"                name="natural"     color=".55,.77,.42" layer="-4"/>
    <polygonType id="natural.water"          name="water"       color=".71,.82,.82" layer="-4"/>
    <polygonType id="natural.wetland"        name="water"       color=".71,.82,.82" layer="-4"/>
    <polygonType id="natural.wood"           name="forest"      color=".55,.77,.42" layer="-4"/>
    <polygonType id="natural.land"           name="land"        color=".98,.87,.46" layer="-4"/>

    <polygonType id="landuse"                name="landuse"     color=".76,.76,.51" layer="-3"/>
    <polygonType id="landuse.forest"         name="forest"      color=".55,.77,.42" layer="-3"/>
    <polygonType id="landuse.park"           name="park"        color=".81,.96,.79" layer="-3"/>
    <polygonType id="landuse.residential"    name="residential" color=".92,.92,.89" layer="-3"/>
    <polygonType id="landuse.commercial"     name="commercial"  color=".82,.82,.80" layer="-3"/>
    <polygonType id="landuse.industrial"     name="industrial"  color=".82,.82,.80" layer="-3"/>
    <polygonType id="landuse.military"       name="military"    color=".60,.60,.36" layer="-3"/>
    <polygonType id="landuse.farm"           name="farm"        color=".95,.95,.80" layer="-3"/>
    <polygonType id="landuse.greenfield"     name="farm"        color=".95,.95,.80" layer="-3"/>
    <polygonType id="landuse.village_green"  name="farm"        color=".95,.95,.80" layer="-3"/>

    <polygonType id="tourism"                name="tourism"     color=".81,.96,.79" layer="-2"/>
    <polygonType id="military"               name="military"    color=".60,.60,.36" layer="-2"/>
    <polygonType id="sport"                  name="sport"       color=".31,.90,.49" layer="-2"/>
    <polygonType id="leisure"                name="leisure"     color=".81,.96,.79" layer="-2"/>
    <polygonType id="leisure.park"           name="tourism"     color=".81,.96,.79" layer="-2"/>
    <polygonType id="aeroway"                name="aeroway"     color=".50,.50,.50" layer="-2"/>
    <polygonType id="aerialway"              name="aerialway"   color=".20,.20,.20" layer="-2"/>

    <polygonType id="shop"                   name="shop"        color=".93,.78,1.0" layer="-1"/>
    <polygonType id="historic"               name="historic"    color=".50,1.0,.50" layer="-1"/>
    <polygonType id="man_made"               name="building"    color="1.0,.90,.90" layer="-1"/>
    <polygonType id="building"               name="building"    color="1.0,.90,.90" layer="-1"/>
    <polygonType id="amenity"                name="amenity"     color=".93,.78,.78" layer="-1"/>
    <polygonType id="amenity.parking"        name="parking"     color=".72,.72,.70" layer="-1"/>
    <polygonType id="power"                  name="power"       color=".10,.10,.30" layer="-1" discard="true"/>
    <polygonType id="highway"                name="highway"     color=".10,.10,.10" layer="-1" discard="true"/>

    <polygonType id="boundary" name="boundary"    color="1.0,.33,.33" layer="0" fill="false" discard="true"/>
    <polygonType id="admin_level" name="admin_level"    color="1.0,.33,.33" layer="0" fill="false" discard="true"/>
</polygonTypes>
```

Figure 3.14: typemap

After saving this in notepad file as typemap.xml, we had to remove the power id for error free execution of the succeeding commands. The following call to POLYCONVERT imports polygons from OSM data and produces a Sumo-polygon file by using the typemap file in figure 3.14.

Polyconvert – net-file berlin.net.xml –osm-file berlin.osm –type-file typemap.xml -o berlin.poly.xml

By using the following NETCONVERT command, the road network is converted from ".osm.xml" to ".net.xml".

d:\sumo-0.26.0\bin\netconvert --osm-files _map.osm -o _map.net.xml

Various characteristics of the created network need to be altered to satisfy our requirements and improve the topography of the resulting map.

d:\sumo-0.26.0\bin\polyconvert --net-file _map.net.xml --osm-files

_map.osm --type-file _typemap.xml -o _map.poly.xml

The route file "map.rou.xml" is generated by calling the following command:

python d:\sumo-0.26.0\tools\randomTrips.py -n d:\sumo-0.26.0\bin

_map.net.xml -r d:\sumo-0.26.0\bin_map.rou.xml -e 10000 -l

This is similar to the file that has been discussed in section 3.2 and provides the edges and co-ordinate locations of each road. It is important to mention here that, since we import the map of the Faizabad junction by manual click-and-drag selection, hence there are some dead-ends or empty edges which appear in the final map. We discuss the solution to this problem after the final generation of the topological area. The SUMO compatible "map.sumo.cfg" file is required as input to SUMO, we need to type in this last command to achieve this purpose:

d:\sumo-0.26.0\bin\sumo-gui.exe d:\sumo-0.26.0\bin_map.sumo.cfg

This will run the configuration file in the sumo-gui and generates the map for the area of faizabad flyover shown in figure 3.15

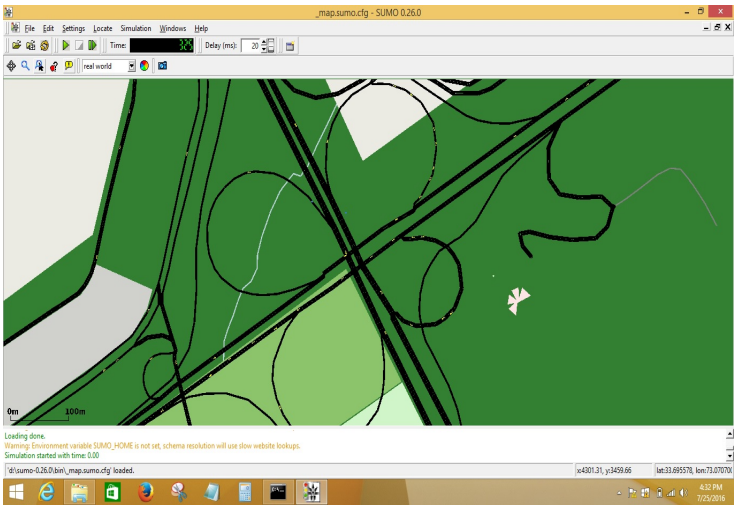

Figure 3.15: Faizabad map imported from OSM

In order to simulate network traffic on the map generated through this whole process and make

it compatible to NS-2 (the selected network simulator), we need to create the trace file of the vehicular data received from the SUMO configuration file. The following command was called for the said purpose:

```
d:\sumo-0.26.0\bin\sumo -c _map.sumo.cfg --fcd-output _sumoTrace.
xml
```

This provides the trace file in an xml format. It defines the vehicles, their start and end times, vehicle movements at particular time and destination etc. The auto generation of TCL script can be executed by the command:

```
d:\sumo-0.26.0\tools\traceExporter.py --fcd-input _sumoTrace.xml
--ns2mobility-output ns2mobility.tcl
```

As we had to run these commands numerous times to learn the capabilities of the system, hence bat files were created for ease of use. Now we address the problem of incomplete edges and routes in the final generated vehicular simulation which causes it to break due to errors. For this purpose, we had to manually find and delete the broken edges, routes and dead ends from the map. This improves the continuity of the routes and causes correct flow of the vehicles in the right direction. It is important to mention here that SUMO supports right hand drive. A change to left hand drive has been tried by many researchers before as well but it causes improper vehicle movements i.e. the simulation shows two vehicles going over each other. For this purpose, this characteristic of SUMO had to be left as it is. It can be changed in the future work, but as it does not influence the criteria of our main research problem, hence it is not a critical issue that has to be addressed immediately.

After these alterations, the vehicular simulation runs smoothly, yet it has the same problem, which we encountered while creating a self configured map, referred in section 3.2, page 41. We need to have a large amount of data for optimum data mining, and the output configuration/trace file

from SUMO does not serve this purpose. The data is very little to satisfy our requirements. To overcome this problem, we have increased the time of the simulation and number of vehicles.

3.5 Conclusion

Testing of a new technology is an imperative part of research methodology. In case of vehicular adhoc networks, it becomes even more relevant, since many industrial and economic factors depend on the deployment of the proposed methods. Testing and evaluation pose a heavy burden on the researchers, since it is very expensive and time consuming to deploy experiments in real time with actual vehicles and road side infrastructures. This cumbersome situation is avoided by appointing simulation softwares and programs. There are many such programs available for the simulation of vehicles and their movements. SUMO and MOVE are the currently popular simulation software which allow the researcher to simulate traffic and mobility generation for vehicular networks. They provide numerous options and can be tailored according to the needs of the user. We have created two road topologies and generated vehicular traffic mobility on the routes. The process has to be improved manually to remove the errors caused during automatic generation and optimum parameters have to be set to achieve the desired results. A skeleton TCL script is automatically generated as an end product.

Chapter 4

PROPOSED DETECTION MODEL

4.1 Overview

Chapter 3 explains the generation of tcl script to run the simulation in NS2, but the code has to be configured in order to cater for our requirements. This chapter describes this procedure and its preliminaries in detail. It has been organized as two parts. The first part details the selection procedure of communication protocol and wireless standard, while the second part focuses on the framework and attack detection scheme.

Chapter 4 has been organized as follows: Section 4.2 explains the process of communication protocol selection and delineates the characteristics of Ad hoc On-Demand Distance Vector protocol. The advantages of using 802.11p for VANETs have been discussed in section 4.3. Section 4.4 shows the comparison of performance features obtained by implementing both 802.11a and 802.11p. Section 4.5 defines the algorithms and mechanism proposed for generating the cbr file for NS2 specifically for real time map involvement. The simulation of DDoS attack have been illustrated in section 4.6. A complete description of DDoS on VANETs, including the explanation of congestion and black hole attack have been detailed. Section 4.13 ends with the conclusive remarks. Section 4.7 illustrates a comprehensive framework for the detection of denial of service attack on VANETs. Section 4.8 introduces the most critical performance features which describes the success or failure for data transmission in vehicular adhoc networks. Section 4.9 describes the data mining and pattern recognition mechanism while section 4.10 and section 4.11 highlight

the characteristics of stream mining and decision trees respectively. Section 4.12 provides an overview of the very fast decision trees including EVFDT, OVFDT and CVFDT.

4.2 Routing Protocol Selection

We have to include the protocols and parameters reflecting the conditions of a vehicular network. A lot of research has already been conducted on the routing protocols for VANETs. This has been done based on the criterias of, connectivity, routing, Quality of Service (QoS) and security [61]. As seen from figure 4.1, the protocols can be divided on the basis of geo cast, topology, position, broadcast and cluster. Topology based routing protocols can be classified into three denominations, Proactive, Reactive and Hybrid. Ad-Hoc On-Demand Distance Vector (AODV)

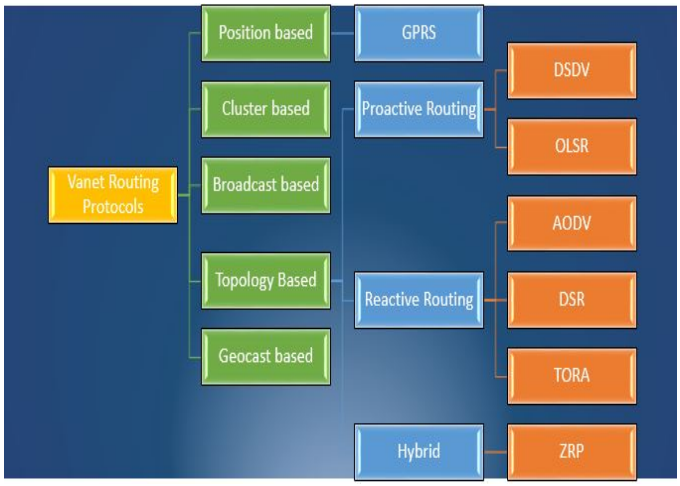

Figure 4.1: Heirarchy of Routing protocols in VANETs

is a reactive protocol, and hence when a node requests a route to another node, the route processing unfolds and routing information is exchanged among the nodes. As only the currently in-use routes are maintained, this reduces the burden on the node, which is imminent in the case of proactive, table-driven protocols. VANETs performance definetely benefits from the use of reactive protocol with less bandwidth usage. There are three phases of AODV protocol:

Route discovery phase: When data has to be transmitted to a node, the source node sends control packet, RREQ (Route Request) by broadcasting it to all neighbors. This is done by a process called flooding. The Broadcast ID and Source IP address are together joined into a unique identifier. If the receiving node is not the destination node, it again forwards the packet to all of its neighbors. When the packet is received by the destination node, it sends back an RREP to the source node. Every RREQ carries a Time To Live (TTL) value to show the lifetime of the packet, indicating the number of hops it has to be forwarded.

Data Transmission phase: When the source node receives the Route Reply packet (RREP), it starts the data transmission through the route with the least number of hop counts.

Route maintenance phase: The process of route maintenance is initiated by the node where the link is broken. This occurs when data transmission fails due to link failure.

Although it has been shown in research that proactive protocols have a low latency for real-time applications, the route table has to be maintained for unused routes as well, thereby occupying an appreciable amount of the network bandwidth [62]. This is definitely problematic in high speed adhoc networks like VANETs. The research by Shastri et al. considered VANETs while making a comparison between two reactive protocols DSR and AODV, on the basis of Packet Delivery Ratio, End-to-end Average Delay, number of received packets, reception time of the first packets and number of dropped packets [63]. They concluded that AODV performs better than Dynamic Source Routing protocol (DSR) in all cases except Packet Delivery Ratio where the value shows a minimum difference as compared to its value in DSR. Another research also uses similar parameters to establish the superiority of AODV for VANETs, which produces highest

throughput, because of using HELLO messages and gratuitous RREPs [64]. DSR results in highest end-to-end delay because of first checking route cache even for low density node networks. Gamess et al. claim that AODV performs better in massive topological changes as compared to DSDV [59].

4.3 Selection of 802.11p

After some amendments in the previous standard i.e. IEEE802.11a 4.1 which was created for wireless communications, IEEE has created a new protocol, IEEE802.11p to handle the challenges of high speed Vehicular networks and to add wireless access in vehicular environments. It manages the vehicle-vehicle and vehicle-RSU data exchange in 5.9 GHz (5.85-5.925 GHz) ITS band. IEEE 1609 is a higher layer standard based on the IEEE 802.11p [65]. The half bandwidth of 802.11p as compared to that of 802.11a ensures that the receiver is able to perform well in the special characteristics of vehicular communications e.g. in the case of ignoring the echoes of signals from nearby buildings or other vehicles.

Before implementing this new protocol, we experimented with the simulation parameters specific to 802.11p, shown in table 4.3 in order to establish this preference with conclusive results. We ran the simulation for 10 vehicles by using the tcl script discussed in Chapter 3, section 3.3. Node 0 was allotted to be the recipient node and packet features are analyzed at this node. The first simulation is run by applying 802.11a parameters specified in table 4.1. We get a trace file with a .tr extension and a nam file with a .nam extension. The nam file is a GUI file of the running simulation, which not only shows the movements of the nodes but also, the flow of data packets seen in figure 4.2. The trace file is in the following format:

```
s1.000535000_1_MAC---0AODV106[0ffffffff1800]-------[1:255-1:
```

Table 4.1: Simulation Parameters of 802.11a

Paramater	Value	Explanation
set val(chan)	Channel/WirelessChannel	channel type
set val(prop)	Propagation/TwoRayGround	radio-propagation model
set val(netif)	Phy/WirelessPhy	network interface type
set val(mac)	Mac/802_11	MAC type
set val(ifq)	Queue/DropTail/PriQueue	interface queue type
set val(ll)	LL	link layer type
set val(ant)	Antenna/OmniAntenna	antenna model
set val(ifqlen)	50	max packet in ifq
set val(nn)	10	number of mobilenodes
set val(rp)	AODV	routing protocol
set opt(x)	652	x coordinate of topology
set opt(y)	252	y coordinate of topology
set stopTime	4000.00	time

```
255][0x211[00][14]](REQUEST)

r1.001383000_2_MAC---0AODV48[0ffffffff1800]-------[1:255-1:
255][0x211[00][14]](REQUEST)
```

This trace shows the type of event that has occurred, which could be send received, or dropped etc. This is followed by the time of the event, sending and receiving nodes, type of data packet, packet size, flags, source address, destination address, sequence no, packet ID etc. Although UDP implementations do not use sequence number, NS2 still manages UDP packet sequence number to ease in analysis. In order to transform this data into comprehensible information and to properly analyze the results, we need to run a script called AWK.

4.4 Feature Extraction for 802.11

Named after its authors, AWK [66] is a data-driven scripting language which processes text and is used for extracting data from text intensive files. This results in the ease of analysis and investigation of such files. The trace file is provided as an input to the script. Separate awk scripts were

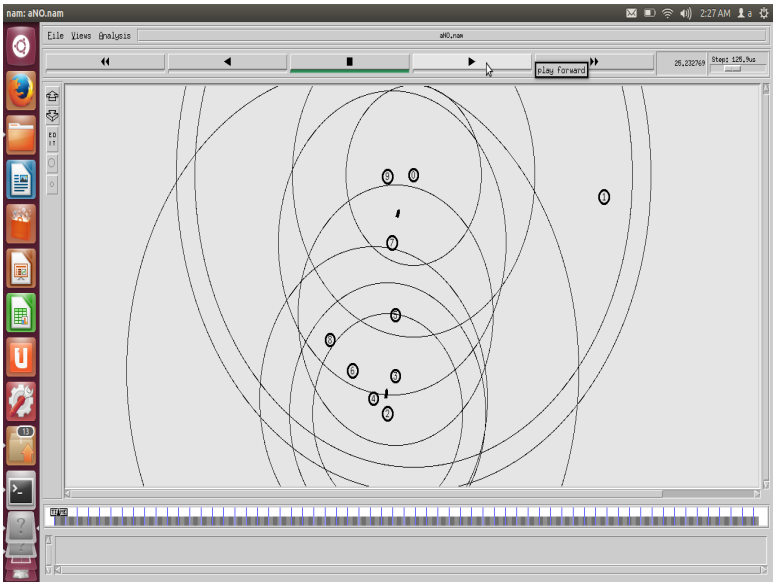

Figure 4.2: data flow in nam GUI

created for each type of statistical features that we wanted to analyze. We have already established in Chapter 2 that the most critical features for wireless networks in general and vehicular networks in particular are the following:

1. **Throughput:** Throughput is the rate of successful message delivery over a communication channel. It can be calculated by the number of bytes transferred per unit time from source node to destination node. It is measured in bits per second (bps). During Denial of Service attack, the throughput is reduced for legitimate users and increased for the malicious entities. The defense against DDoS aims to increase the throughput for the legitimate users. It can be calculated by using following equation:

$$Throughput = \frac{\sum_{i=0}^{n}(Size\ of\ Received\ Packet)}{\sum_{i=0}^{n}(Start\ Time - Stop\ Time)} \tag{4.1}$$

2. **Average Delay:** Average delay reflects the latency issues of a network. It is essentially the value of time a packet takes to be delivered from source node to destination node. It is

dependent on the amount of traffic being transmitted. If their is an attack, the congestion increases, therefore the delay also increases. It is measured in seconds or multiples of seconds. A low value of this metric is preferred for optimum performance of the network. Delay can be calculated by:

$$AverageDelay = \sum_{i=0}^{n}(Packet\ Arrival\ Time_i - Packet\ Sent\ Time_i) \tag{4.2}$$

3. **Drop Ratio:** The significance of measuring the dropped packets is to check the transmission quality (QoS) of the network. If the numbers of dropped packets increases, it indicates an attack on the packet The legitimate packets are dropped due to DDoS attack causing traffic overload at the destination nodes. The drop ratio can be calculated as follows:

$$DropRatio = \frac{\sum_{i=0}^{n}(Dropped\ Packets_i)}{\sum_{i=0}^{n}(Sent\ Packets_i)} \tag{4.3}$$

4. **Packet Delivery Ratio:** The ratio of packets that are successfully delivered to a destination compared to the number of packets that have been sent out by the sending node is called the packet delivery ratio. It is basically the number of packets or bytes received by the destination per unit time. It is an important metric for analyzing the efficiency of network protocols and has been used by the detection schemes discussed in chapter 2. Therefore, it can be used by us to evaluate the competence of our detection mechanism. Packet Delivery Ratio can be calculated by the following equation:

$$PacketDeliveryRatio = \frac{\sum_{i=0}^{n}(Packets\ Received_i)}{\sum_{i=0}^{n}(Packets\ sent_i)} \tag{4.4}$$

In order to extract these features from the trace files, we have written the AWK scripts to separately calculate the different parameters but the pseudo code is being shown cumulatively in table 4.2 .

Table 4.2: Algorithm 1 AWK Script

Require:
T: start time for sending packets
PS: packet size
ES: Event of sending data
ER: Event of Receiving data
D: Event of dropped packet
Trace file of NS2 simulation Output: **performance features of the RX node**
1: Procedure Feature(*T,PS, ES, ER*)
2: **BEGIN**
3:set *S*T:start time=x
4:set *ET*:end time = 0
5:set *WIN*=0
3: **if** r = 0 **then**
5: Initial round *E* selection
6: **end if**
7: **for** (maximum number of rounds r) **do**
8: Choose r rounds AN randomly
9: T< do ST=T, WIN++
10: T> do ET = T, PS++
12: *D*++, *ES*++, *ER*++
13: **end for**
14: BEGIN
15: WIN % 500 do delay = *ET-ST*
16: WIN % 500 do throughput=*PS/(ET-ST)*
17: WIN % 500 do *D/WIN*
18: WIN % 500 do *ER/ES*
19: **end for**
20: END Procedure

The data mining mechanism requires the testing data to be divided into windows of pre-determined sizes to get the packet information. For this purpose, the trace file for processed accordingly.

As we discussed in Section 4.3, the simulation is carried out first by using the parameters for the 802.11a protocol. The similar procedure is carried out using the 802.11p parameters show in in table 4.3. The rest of the parameters are same as in table 4.1. 802.11p consistently

Table 4.3: Simulation Parameters of 802.11p

Paramater	Value
Mac/802.11 set dataRate	6.0e6
Mac/802.11 set basicRate	6.0e6
Mac/802.11 set CCATime	0.000004
Mac/802.11 set CWMax	1023
Mac/802.11 set CWMin	15
Mac/802.11 set PLCPDataRate	6.0e6
Mac/802.11 set PLCPHeaderLength	50
Mac/802.11 set PreambleLength	16
Mac/802.11 set SIFS	0.000016
Mac/802.11 set SlotTime	0.000009
Phy/WirelessPhy set RXThresh	6.72923e-11
Phy/WirelessPhy set freq	5.15e9
Phy/WirelessPhy set Pt	0.281838

shows better results as compared to 802.11a in a vehicular environment. As 802.11p uses 10 MHz frequency bandwidth which is half of bandwidth of 802.11a, therfore the signal becomes robust against fading. Furthermore, the tolerance for multipath propagation effects of signals also increases.Figure 4.3 shows that the throughput at node 0 is much higher when the former protocol is applied. Figure 4.4 displays the values of average delay in the transmission of data from nodes 1-9 to node 0. As can be seen, the delay is high when 801.11a is applied. Figure 4.5 compares the packet delivery values when using the 802.11a and 802.11p. Similarly, drop ratio of packets at node 0 is considerably low when 802.11p is tested shown in figure 4.6. These results confirm the superiority of 802.11p for data communication in high speed vehicular adhoc networks.

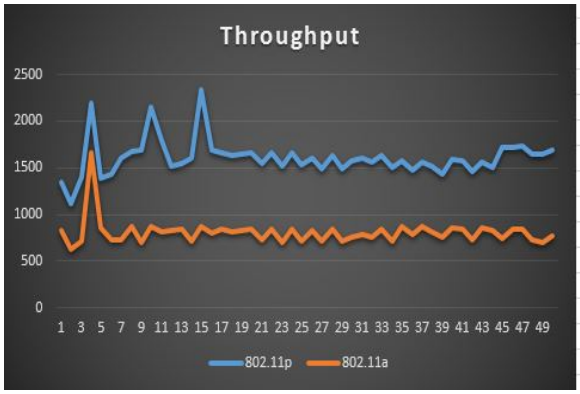

Figure 4.3: Comparison of throughput between 802.11a and 802.11p

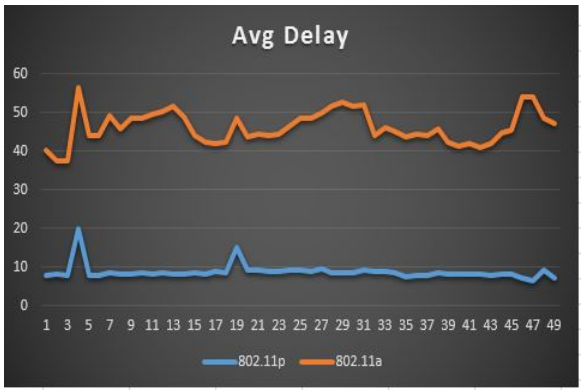

Figure 4.4: Comparison of average delay between 802.11a and 802.11p

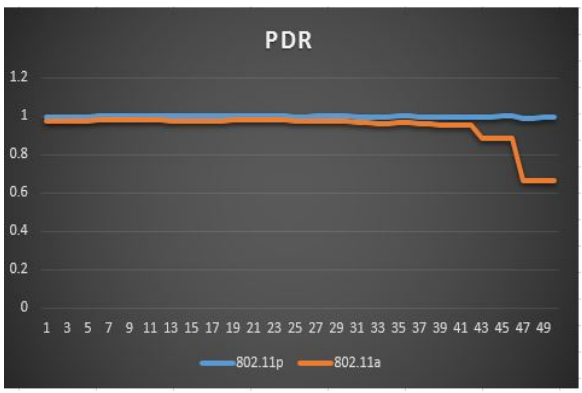

Figure 4.5: Comparison of PDR between 802.11a and 802.11p

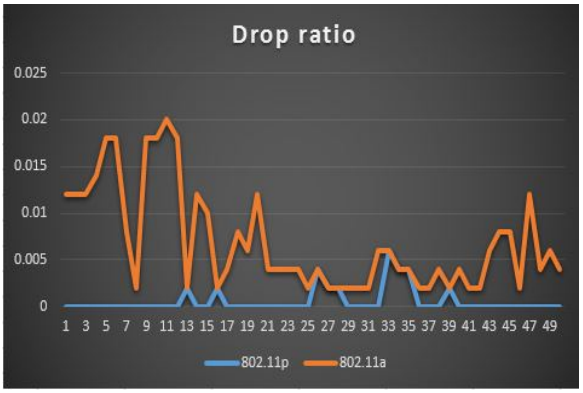

Figure 4.6: Comparison of drop ratio between 802.11a and 802.11p

4.5 CBR Generation

In order to simulate real time vehicular traffic for testing and evaluation, we have to increase the size of the road topology as well as the number of vehicles. For this purpose, real map data is used which was converted into SUMO trace files in Chapter 3. This resulted in an area of 9000 by 6500 units, as compared to the previous values of 652 by 252 units. VANET specific 802.11p protocol is used to warrant optimum results in data delivery. The resulting trace file of NS2 does not have enough data to fulfill the pre-requisite of data mining detection schemes. In order to augment this data, we have to extend the simulation time. A trial and error methodology ensued, as increasing the number of vehicles caused the mobility and traffic generators to hang. A balance between the number of vehicles on road and maximum simulation time is achieved by keeping the number of vehicles at 246. The SUMO graphical simulation for these specifications, completes the simulation at 403 seconds. Although, the traffic mobility generators provide the position and movement of each vehicle for every second of the simulation, it does not create the calls for setting communication protocols and the time for starting and ending this communication at each node.

This data has to be inferred from the traffic file generated by the mobility generator. Due to the

profusion of this information in the sumo trace file, we use AutoIt [67], a BASIC-like scripting language designed for automating general scripting. The start and stop time of cbr traffic for each vehicle will also be set at the same time as they enter or exit the perimeters of the map. The pseudo code for the program shown in table 4.4.

Table 4.4: Algorithm 2 Generating cbr for Nodes

Require:
ID: node ID
R: Number of rounds
ST: start time
ET: end time
SD: set destination
mobility generator output file, set dest **Output:** CBR data for data transmission among nodes
1: Procedure **CBR generation(SD,R)**
2: BEGIN
3: **if** *r = 0* **then**
4: Initial round *SD* selection
5: **end if**
6: **for** *(maximum number of rounds r)* **do**
7: searchstring "(" , trim string: left
8: trim string at ()
9: Find Node 0 and set array ID=0
10: Increment and set *ID*
11: Set *ST*, Set *ET*
12: Set UDP(r)
13: Attach agent to *ID*
14: Set cbr(*r*)
15: Set packet size=512
16: Attach agent udp with cbr
17: **end for**
18: END Procedure

The mobility file and cbr file are attached in a single tcl code and processed to get nam and trace files reflecting data transmission employing AODV protocol for nodes in the Faizabad map. In

order to extract performance features, we designated one node as the destination node. This can be either RSU or a vehicle.

In DDoS, the main aim of the attacker is to deplete the bandwidth and energy resources for the legitimate user. This can cause disruption in the dissemination of information and data transmission to other vehicles, not only resulting in a cessation in VANETs services but even major/fatal accidents on road. The distinctive formula for attack for denial of service, is to congest the assets of the network by either injecting useless messages or dropping useful information relevant to the network process.

This divides the modus operandi of DDoS attack into mainly two groups. Congestion attacks include, jamming, greedy attack, spamming, syn flooding, TCP ack storm, routing table overflow, routing cache poisoning , rushing atack and sybil attacks. Jamming attack can be on the node in V2V Communication, on the RSU in V2I communication and can also be directly targeted towards the channel by sending high frequency signals.

On the other hand, dropping useful information also accounts for DoS in a network. In VANETs these include blackhole, sinkhole, wormhole, greyhole, Broadcast Tampering and impersonation attack. The attack category mentioned latter is an active attack, where the malicious entity advertises itself to be the optimum route to the destination or show itself as the destination, and then either drops or does not forward the data packets.

4.6 DDoS Attack Simulation

To cater for the different types of attacks discussed in 4.5, we simulated congestion and blackhole attack on the nodes in order to accommodate for all types of DDoS attacks on VANETs discussed in the research. The pseudo code is shown in table 4.5.

Table 4.5: Algorithm 3 Blackhole attack algorithm

Require:
ID: node ID
R: Number of rounds
Simulation Parameters of AODV protocol **Output**: DDOS Attack Dataset
1: Procedure **DDoS Algorithm(ID,R)**
2: BEGIN
3: **if** $r = 0$ **then**
4: Initial round *SD* selection
5: **end if**
6: **for** *maximum number of rounds r* **do**
7: set attack times to malicious nodes
8: initiate malicious node in aodv constructor
9: set argument=hacker for nodes i,j after setting node position
10: **if** malicious=true
11: **Reply** rq > rq-src of source
12:Reply hopcount = 1
13: reply rq > rq-dst
14:reply maximum seq no
15: reply route time out=lifetime
16: reply rq greater than rq-timerstamp
17: **end for**
18: END Procedure

In order to replicate the first type of attack scenario, i.e. congestion, the data rate was increased in the algorithm for all the transmitting nodes. The stagnant destination node, can be either a vehicle or road side infrastructure. For the purpose of reproducing the latter mentioned attacks, a blackhole attack agent was attached to nodes near or adjacent to the receiving node. To mimic real time situation on road, one blackhole node is kept stationery and near the destination node. This is also done to represent the distributed nature of the attack. Another blackhole node is moving at a speed similar to the rest of the vehicles. This is to generalize all types of vehicle levels that can be faced in real on-road circumstances.

Figure 4.7, shows the account of a blackhole attack in VANETs whereby the malicious node is sending back RREP packet to the source node with forged sequence number and hop count towards a destination node, although it does not have route information for it.

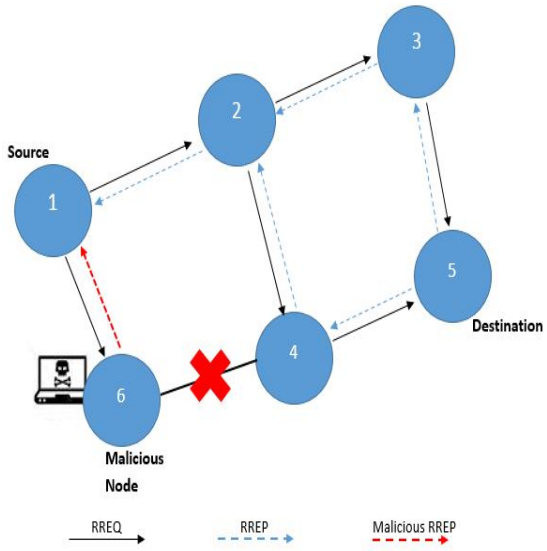

Figure 4.7: Blackhole attack in VANETs

4.7 Proposed Framework and Detection Model

The distributed manner of attack on availability of network services requires an ingenious detection scheme to cater for the attack spectrum. For this purpose, a comprehensive framework has been proposed for the detection of DDoS attack on VANETs, which has been illustrated in figure 4.8. The detection system is installed at the the receiving/destination node. The data packets are collected at this node and packet features are extracted for the data in an online database. A rule set is created for training and testing data and the pre-processing for labelling the data is performed in an offline database. Results are normalized for the classifier and the data is separated into testing and training categories. The resulting data is fed into the data classifier and detection module. Incase of attack, incoming packets are dropped, while the non-attack packets are forwarded and

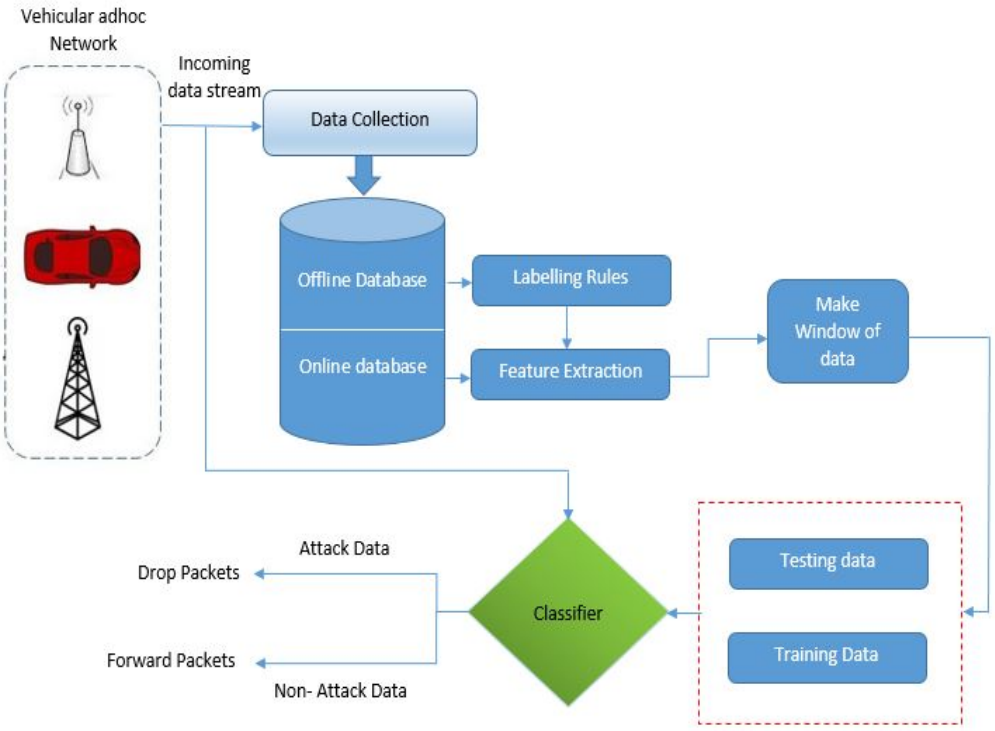

Figure 4.8: Proposed Detection Model

utilized. The incoming stream of data is divided into segments of a particular window size. The window size is determined by a trial method as the VFDT algorithm does not run properly if a substantially sufficient amount of data is not fed into it. As far as our simulation data is considered, we have conducted experiments on 5000, 10000, 15000 and 20000 data values. 20% of data is kept as testing data while 80% is the training data in each experiment. The window at which the performance features are calculated is adjusted to 500 values. The features are extracted from the input data and labeled, as they are the attributes for the classification process. If the feature shows attack data, then it is labeled as 1, otherwise it is marked 0. It can be in numeric, binary, nominal, or mixed form.

On the basis of the literature review conducted in chapter 2, we have selected the following performance features of the data to be employed for the detection of attack on VANETs. These features

require the extraction of the following parameters from the trace files: the numbers of packets sent, number of packets received at the destination node, size of the packets, time at which the packets are sent from the source, time at which they reach the destination (the time interval) and number of packets dropped during transmission. When the node receives the input data stream, it takes in the amount of data based on the window size and then forwards the attributes to the decision tree, which in turn detects the attack based on the training data and forwards the non-attack packets to the next node.

4.8 Performance Features

As mentioned in 4.7, we have selected the most critical performance features which describe the success or failure for data transmission in vehicular adhoc network i.e throughput, packet delivery ratio, drop ratio and average delay. They have been described thoroughly in section 4.4. These features are attributes which have to be classified by the decision tree in order to detect the DDoS attack on VANETs.

4.9 Data Mining

Data mining is a field of computer science which performs pattern discovery and computation learning in large data sets by methods using AI, machine learning and statistics. Data mining algorithms not only aim to learn from the large amount of data but also to make predictions on it. The predictions are not based on strict statistical programming instructions, instead the data mining process makes models from testing data to make decisions. Data classification is an important data mining technique which assigns predefined classes to raw data for better organization and research purposes. For example, emails are considered as input data and the classes "spam" and "not spam" can be used to classify them.

4.10 Stream Mining

Stream mining is a new generation of data mining, which trains the decision model to extract large amounts of knowledge from continuous stream of data by just one pass on the infinite data set, while the new data streams are being passed on run-time [68]. This reduces the computational and memory resources required for the processing.

4.11 Decision Trees

Decision Trees have been often applied in classification and prediction process of data mining. Their strength lies in their ability to clearly define the interaction between variables. Decision trees work like flow charts. Here, the internal nodes represent the test on an attribute, the branches stand for the outcome of the test and the leaf nodes are the class labels and the topmost node in the tree is the root node. This process clearly delineates the knowledge that is gathered in the process of classification. Therefore it renders a much easier rule generation process [69].

There are many popular variants of machine learning techniques. A comparison of VFDT with C4.5, SVM, K-NN, GA and NeuroFuzzy schemes has been made in [70] which shows that VFDT is dominant in terms of using less memory consumption as compared to GA, KNN, C4.5, and no need of prior knowledge of data distribution (compared to GA, neuro Fuzzy, C4.5) when handling large amounts of data. Furthermore, it has low complexity as compared to KNN, GA and Neuro Fuzzy. Only Neuro Fuzzy and VFDT have incremental learning capability or real time stream data mining dexterity which is undermined in Neuro fuzzy due to its complexity in making a decision. Moreover, VFDT has the highest classification accuracy as compared to the rest which makes it as the best choice for handling streaming data in detecting a DDoS attack on VANETs.

4.12 VFDT

VFDT is a well-known decision tree algorithm in data stream mining. It is a lightweight technique which makes a tree while large stream of data is incoming at run-time. It does not require the full data set before making a decision, instead it performs a test and train process as soon as the new packets of data are received. It does not need memory for storing the example data, instead it only requires space equal to the size of the tree.

4.12.1 Hoeffding Bound

In VFDT, Hoeffding bound is used to regulate the node splitting process dynamically. This is based on data statistics at the leaf. A tie-threshold is selected to split and control the tree size. The Hoeffding Bound is used to convert the tree leaf to a tree node by collecting a substantial amount of statistics from new samples, so that VFDT constructs a decision tree from the huge incoming data set at run-time. If we have N independent observations of a random variable r with a bounded range R, the Hoeffding bound gives the mean of variable r to be atleast $r - \epsilon$ with a confidence level $1 - \delta$. ϵ is calculated by:

$$\epsilon = \sqrt{\frac{R^2 ln(1/\delta)}{2N}} \tag{4.1}$$

4.12.2 Information Gain

Information gain is a heuristic evaluation function which is employed to calculate the the upper and lower bounds with high confidence. The lower bound is given by $G(.)^-$ and the upper bound can be stated as $G(.)^+$. The lower bound is calculated by equation 4.12.2:

$$G(A,T)^+ = \sum_{v \varepsilon A} P(T,A,v) + \sqrt{\frac{ln(1/\delta)}{2N}} H(Sel(T,A,v))^+ \tag{4.2}$$

The upper bound can be calculated by equation 4.12.3:

$$G(A,T)^{-} = \sum_{v \varepsilon A} P(T,A,v) + \sqrt{\frac{ln(1/\delta)}{2N}} H(Sel(T,A,v))^{-} \tag{4.3}$$

In these equations A is an attribute in the T set of training samples. $P(T,A,v)$ is a part of the training samples in set T where T keeps the value v for attribute A. Function $Sel(T,A,v)$ basically picks all the training samples with the value v for attribute A from this set T.

VFDT composes a decision tree by employing constant memory and constant time per sample. A heuristic evaluation process regulates the split attributes which performs the conversion from leaves to nodes. The decision tree requires the incoming sample to evaluate the attributes at each node while passing from the root to the leaf. When the sample arrives at the leaf, the mentioned statistics are updated. The conversions are performed according to the attribute values. A leaf is converted to decision tree if the statistics support one test over the other. VFDT consists of a tree Initializing process starting with a single leaf and a tree growing function that uses the heuristic evaluation $G(.)$ and Hoeffding bound to perform splitting. This has been shown in figure 4.9.

First, a complete set of training samples is observed. Each sample has different attributes and is represented as (X,y) where X is a vector of n attribute given by (x1, x2, xn) and y is a class where the classification problem is to construct a model that defines a mapping function f(X)$\rightarrow$ y in order to predict accurate samples of x. As shown in the flow chart of VFDT in figure 4.9 a certain number of counts are collected and n_{ijk} is considered sufficient for the calculation of information gain required for the decision making process. Let N be the number of samples observed on a leaf node, X_a is the highest value attribute in $G(.)$ if following two conditions are met: $\Delta G > \epsilon$ and Hoeffding bound states with the confidence level $1 - \delta$.

The difference between the two top gain value are calculated by $\Delta G = G(X_a) - G(X_b)$ where

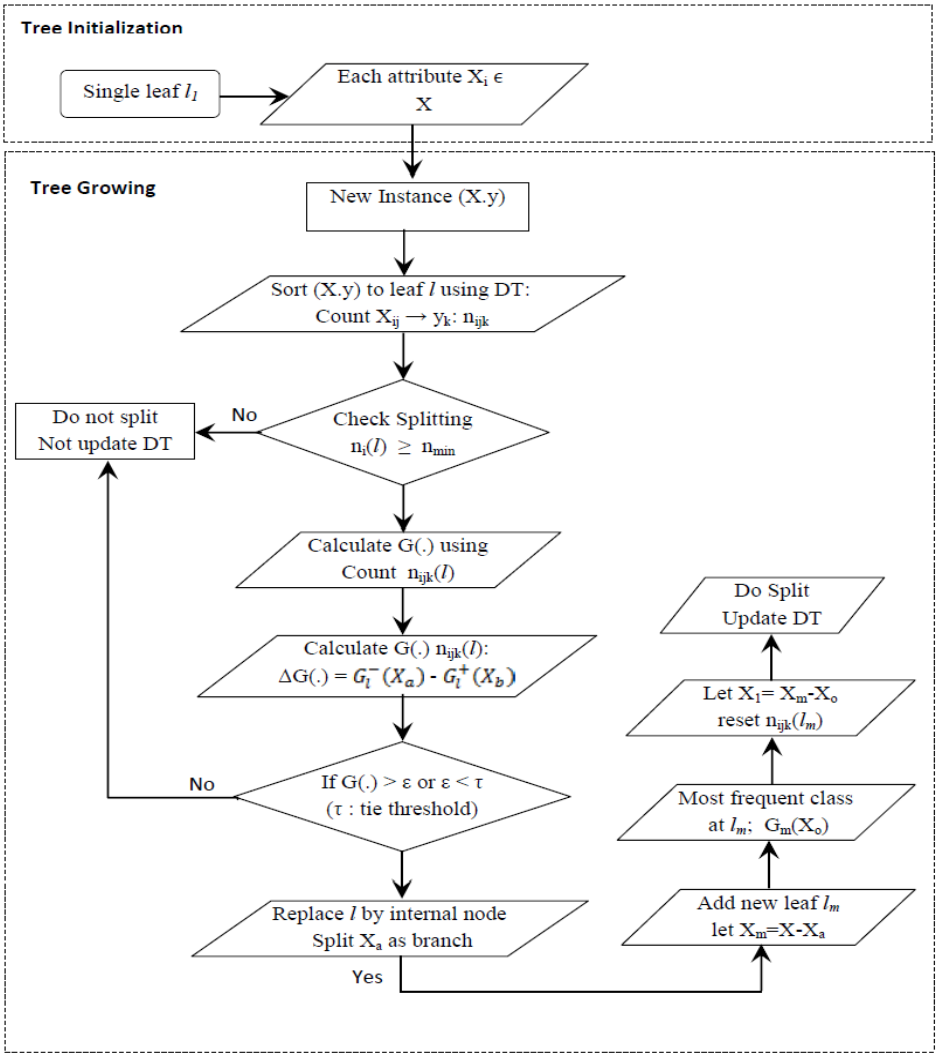

Figure 4.9: Very Fast Decision Tree Algorith (VFDT)

X_a is the attribute with the highest gain and X_b is the attribute with the second highest gain. This leads to the node splitting on attribute X_a, and the conversion of this leaf into a decision node. The next step is the tree induction phase where the succeeding training samples are passed down to the corresponding leaves using $X_m = X - X_a$ and the correct right splitting attribute is selected so on and so forth. As X_a has already converted into decision node therefore it is excluded from this process. The counter in VFDT is incremented whenever a training sample N arrives at a leaf node and hence storage requirement for this counter is not required. This in-turn reduces the computational resource burden for the decision making process.

4.12.3 OVFDT

Optimized Very Fast Decision Tree (OVFDT) [68] is dependent on an adaptive tie-breaking threshold calculated from Hoeffding Bound (HB) mean. This process reduces the accuracy of the detection scheme as the value of HB mean fluctuates with the increase in the noise level.

4.12.4 CVFDT

Concept Adaptive VFDT (CVFDT) [71] maintains two trees at the same time. The tree which has the shortest depth is kept and the other one is neglected. Due to the maintenance of two trees at a time, this technique consumes more memory and time.

4.12.5 EVFDT

Enhanced Very Fast Decision Tree (EVFDT) [72] is an improvement of VFDT in terms of accuracy. The origincal VFDT and its variants pre-configure the value of τ and this value is kept same during the tree building process. Brute force has to be applied to find the best value of τ. Testing such a huge number of τ values is not appropriate for a real time scenario. EVFDT assigns an adaptive tie-breaking threshold for splitting, which is equal to the mean of difference between HB values.

A java code for VFDT, EVFDT [72], CVFDT [71] and OVFDT [68] is used to run the decision tree on the data gathered in Chapter 4. The simulation time is arranged to generate data instances equal to 5000, 10000, 15000 and 20000 in order to evaluate the variance and efficiency of the respective decision trees for the detection of DDoS attack on VANETs. The code is run in Visual Studio 2010 and requires three files for each labeled data which has been stored in comma separated values. These values are converted to .arff format using weka 3.7 [73] and saved as

"DATA" file for the training data, "TEST" file for the testing data and "NAMES" file for the labeling, where DATA files contains 80% training data and TEST file contains 20% testing data. Weka automatically generates numeric valued labels in the NAMES file. These have to be altered to "real" values while the labels have to be given the actual value that have been used in the data. Here, it was applied "1" for attack and "0" for non-attack data.

4.13 Conclusion

The chapter has described in detail the process of routing protocol selection based on its comparison with other protocols keeping in mind the characteristics and requirements of VANETs. The description of wireless local network specifications involving 802.11 a and 802.11p has been given and a systematic comparison has been made to show the advantage of using 802.11p for VANETs, by simulating both the specifications in NS2. The results show a marked increase in the packet delivery ratio and throughput of the data. On the other hand , the drop ratio and the delay is much less in 802.11p as compared to 802.11a. This resulted in the selection of 802.11p for attack simulations and detection scheme.

CBR traffic generation was required for the nodes and an algorithm has been proposed to ease the process for a large number of nodes. In the end a complete description of DDoS attack in VANETs is given and the most critical attacks are selected to be implemented. Congestion attack and blackhole actually cover the entire range of DDoS attack impeding the successful communications in VANETs. Furthermore a framework for the detection of denial of service attack on VANETs has been discussed in this chapter and a commensurate description of data mining/pattern recognition mechanisms including VFDT, EVFDT, OVFDT and CVFDT has been provided.

Chapter 5

EVALUATION AND RESULTS

5.1 Introduction

The inherent wireless medium of the vehicular adhoc network, not only renders it vulnerable to multitude communication attacks as discussed in Chapter 1, but also makes the attack detection process difficult. QoS is a basic requirement of any network and specifically important for life and time critical VANETs. A major setback to QoS is caused by DDoS attack. As discussed in Chapter 2, the detection schemes mentioned in recent literature pose some drawbacks in the investigation of DDoS attack on VANETs, and have technical flaws that lead to unsatisfactory results.

The chapter is organized as follows: Section 5.2 enumerates the performance metrics selected for the purpose of analyzing the detection schemes while section 5.3 and 5.4 make a comparative analysis of the vfdt variant schemes. Finally the concluding remarks are given in Section 5.7.

5.2 Performance Evaluation

In order to evaluate the efficiency of decision trees for detecting the DDoS attack on VANETs, a number of performance metrics are selected. These evaluation parameters are essential in benchmarking the efficacy of the proposed framework for detection of DDoS on VANETs. The following parameters are chosen for this purpose: Accuracy, sensitivity, specificity, false alarm rate, memory usage, computational time and tree size.

5.2.1 Accuracy

Accuracy is the foremost requirement in analyzing the competence of the attack detection scheme. It is defined as the ratio of the true predictions to the total number of tested incidences. The true predictions are the total of true positives (correct predictions of attack) and true negatives (correct prediction of non-attack). Accuracy is calculated by:

$$Accuracy = \frac{True\ Positives\ +\ True\ Negatives}{Total\ no.\ of\ incidences} \tag{5.1}$$

As the number of data samples increase, the accuracy tends to increase for decision trees, as the amount of data used for predicting makes it easier to make correct judgments.

5.2.2 Sensitivity

Sensitivity is a statistical measure of performance of a classification test. It basically measures the proportion of true positives that are correctly predicted in a classifier and quantifies avoiding the false negatives. In our case, sensitivity refers to the test's ability to correctly identify attacks. It is calculated by the following equation:

$$Sensitivity = \frac{True\ Positives}{True\ Positives\ +\ False\ Negatives} \tag{5.2}$$

5.2.3 Specificity

Specificity is also a statistical measure of the performance of a classification test. In contrast to sensitivity, it measures the proportion of true negatives that are correctly predicted in a classifier and hence quantifies avoiding the false positives. In our scenario, specificity refers to the test's ability to correctly identify non-attacks or legitimate packets. It is calculated by the following equation:

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives} \tag{5.3}$$

5.2.4 False Alarm Rate

This metric calculates the erroneous outcomes of the statistical tests. False Alarm Rate include the false positive rate and the false negative rate. A comparative analysis of the false alarm rates can identify the precedence of one detection scheme over the other, as along with accuracy, it can measure the probability of the classifier making a correct prediction. A high false negative rate is problematic in detection schemes for identifying attacks on communication.

- ***False positive rate:*** This false alarm rate is the ratio of the legitimate packets that have been incorrectly identified or predicted by the classifier as attack packets. A high false packet rate causes legitimates packets to be dropped and hence results in low packet delivery rate and low throughput. This impedes the flow of complete information to the destination node. It is critical to VANETs, as a substantial amount of information is required for VANET applications to perform on road, specially in case of emergency situations. The false positive rate is calculated as follows:

$$False\ Positive\ Rate = \frac{Legitimate\ Packets\ incorrectly\ identified\ as\ attack}{Total\ number\ of\ packets} \quad (5.4)$$

- ***False Negative Rate:*** The false negative rate indicates the ratio of attack packets incorrectly identified as legitimate packets. This is a very critical performance metric for IDS since a high false negative rate indicates the failure of the detection scheme to successfully classify attack on the communication network. The false negative rate is calculated as follows:

$$False\ Negative\ Rate = \frac{Attack\ Packets\ incorrectly\ identified\ as\ Legitimate}{Total\ number\ of\ packets} \quad (5.5)$$

5.2.5 Memory

Memory Usage by the detection scheme has to be competitively minimal to allow other VANETs applications to perform multiprocessing effectively. Decision Trees along with other stream mining schemes already have the benefit of not requiring the full data set for classification purposes. Instead, they perform labeling and attack detection in real time. The total memory demand of decision trees is for the learning and training processes of the classifier. It depends on the total number of attributes, number of decision nodes in a tree, number of values in each attribute and the number of classes.

5.2.6 Time

Similar to memory, minimum time requirement is also a feasible performance metric for bench marking a detection mechanism. It is specifically imperative for time sensitive characteristics of vehicular adhoc networks due to high mobility of its nodes. The time taken by decision tree for the detection is dependent upon the total number of attributes, length of a pruned tree, number of classes and the number of values in each attribute.

5.2.7 Tree Size

The performance of decision tree with respect to memory and time is hugely dependent on the tree size it builds during the pruning. A smaller tree size is a desirable metric to ensure high speed detection while maintaining accurate results. A comparative analysis of decision trees on the basis of tree size has been performed in order to decide the efficiency of these schemes comprehensively.

5.3 Comparative Analysis for Congestion Attack

The results for the congestion attack are compared for the VFDT variants in the following subsection:

5.3.1 Accuracy

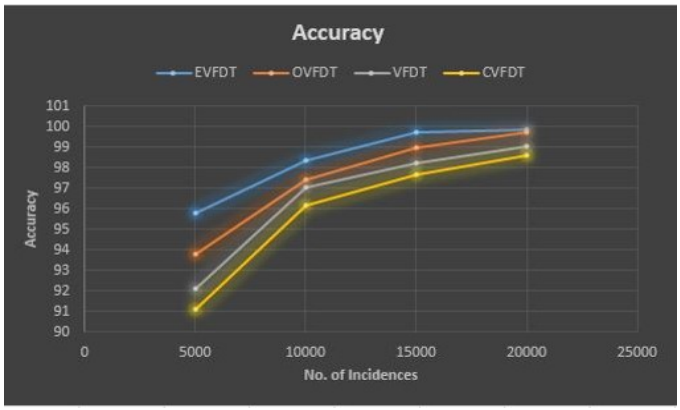

Figure 5.1: Accuracy Analysis for Congestion Attack

The accuracy of the decision tree is calculated by using equation 5.2.1. The comparison accuracy for vfdt, evft, cvfdt and ovfdt are given in figure 5.1. The accuracy for each decision tree is analyzed by correlating it with the number of incidences. As we can see in figure 5.1, the accuracy of detecting an attack tends to increase with increase in the size of data set, for example for VFDT, the value of accuracy increases from 92% for 5000 incidences to about 98.5% for 20,000 testing data. It is evident from figures that evfdt shows the best accuracy for detecting a denial of service attack, followed by ovfdt, vfdt and cvfdt in this order.

5.3.2 False Positive Rate

The false alarms for the detection scheme based on classification algorithms can be calculated by equation 5.2.4 and 5.2.5. Generally for an IDS, the false positive rate is more critical to detect

the true attacks on the network. An IDS should generate less false alarms to accurately detect an attack. Figure 5.2 shows the calculated value of false alarm for different variants of decision tree. The graph shows that the value of FPR decreases with an increase in the size of the data set. Furthermore, EVFDT shows lowest values of FPR e.g. for 5000 incidences its value is 3.12. It can be seen that the values of FPR for the various VFDT algos decrease with the increase in incidences and come very close at 20,000 data set.

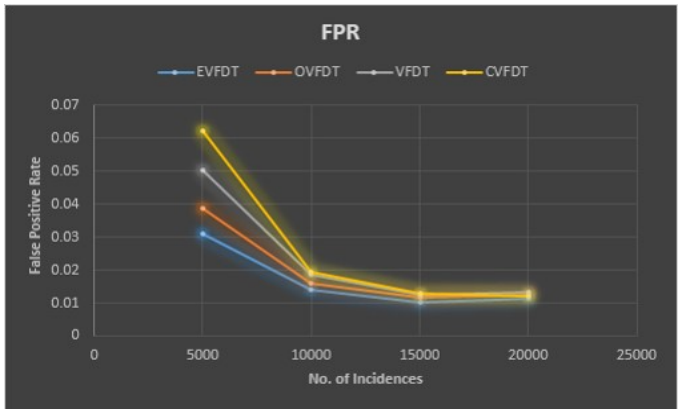

Figure 5.2: FPR Analysis for Congestion Attack

5.3.3 Sensitivity

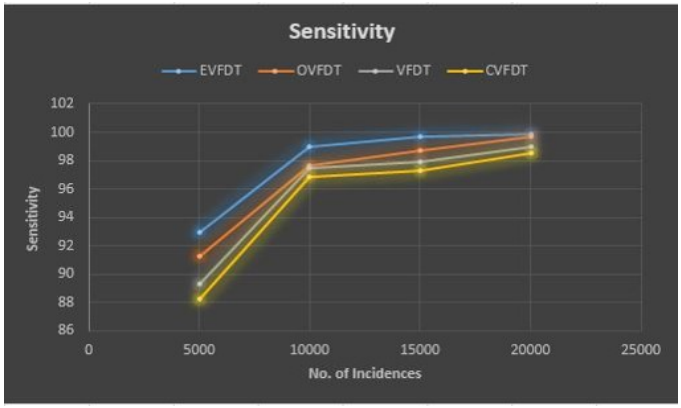

Figure 5.3: Sensitivity Analysis for Congestion Attack

Sensitivity is a prime feature for analysis of a detection scheme. It is calculated by equation

5.2.2. A high value of sensitivity is a requirement for an ideal classification mechanism. Figure 5.3 makes a comparative analysis of Sensitivity between VFDT, EVFDT, OVFDT and CVFDT. It is evident, that EVFDT has highest sensitivity for all sizes of data set and goes up to 98.7 for 20000 values, while CVFDT shows lowest sensitivity, due to more false alarms.

5.3.4 Specificity

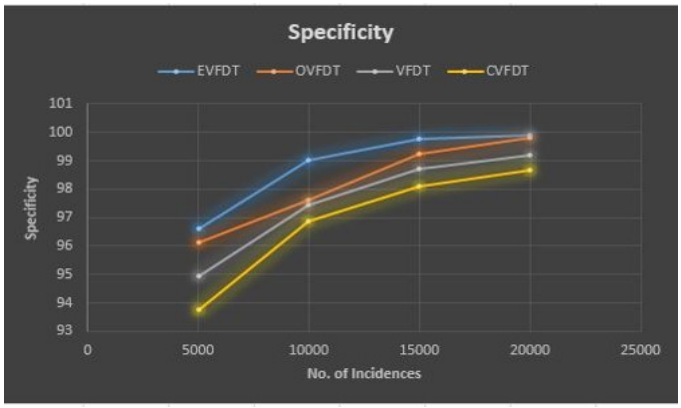

Figure 5.4: Specificity Analysis for Congestion Attack

Specificity is another critical statistical feature for analysis of a detection scheme. It is calculated by equation 5.2.3. A classification mechanism should show high specificity and avoid the false positives. Figure 5.4 makes a comparative analysis of specificity between VFDT, EVFDT, OVFDT and CVFDT. It can be seen that EVFDT has highest specificity but the values come very close as the size of data set increases.

5.3.5 Computational Memory

Computational Memory is the total resource required by CPU for training and testing the complete data set. As the size of the data set increases, the memory requirement also increases. Figure 5.5 makes a comparative analysis of memory for the tested classification algorithms. Although the tree size for CVFDT is smaller than OVFDT, it can be seen from the figure, that CVFDT uses

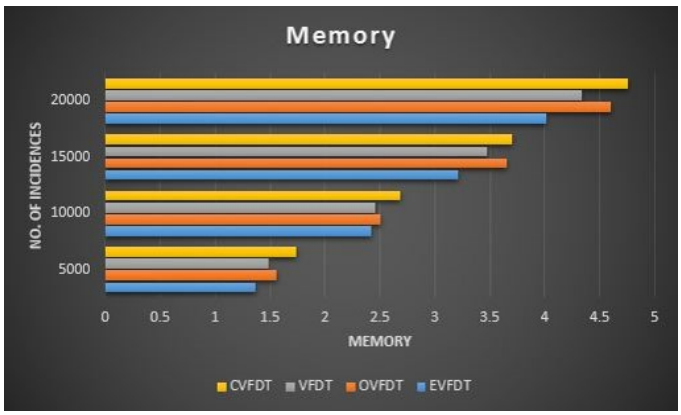

Figure 5.5: Memory Analysis for Congestion Attack

maximum memory resources, due to its requirement of maintaining two data sets simultaneously, while EVFDT shows minimum memory usage.

5.3.6 Time

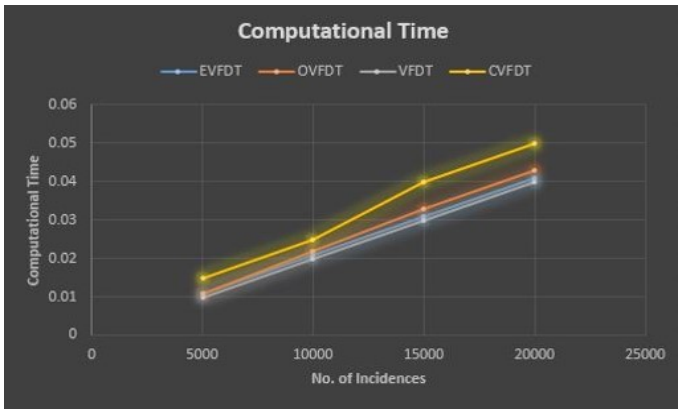

Figure 5.6: Time Analysis for Congestion Attack

Total resource usage also includes the time for training and testing the complete data set and is measured in seconds. This time actually refers to the detection time of an IDS. As the size of the data set increases, the detection time also increases. Figure 5.6 compares the time required by the the different classification algorithms. It can be seen from the figure, that CVFDT takes the most time while EVFDT shows minimum value of detection/classification time due to run-time

computation of tie-breaking threshold.

5.3.7 Tree Size

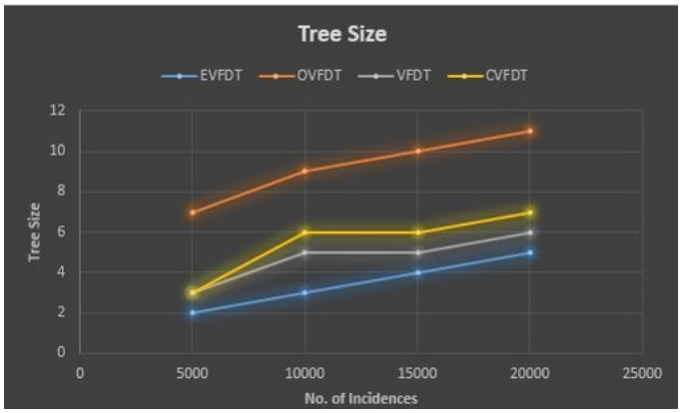

Figure 5.7: Tree Size Analysis for Congestion Attack

A small tree size of the classification algorithm, results in lesser computational resource usage and detection time. Figure 5.7 compares the tree size of the different vfdt classification algorithms for varying data set sizes. The tree size increases with the increase in data set size and shows maximum value for OVFDT.

5.4 Comparative Analysis for Black hole Attack

The results for the black hole attack are compared for the VFDT variants in the subsequent sub-sections:

5.4.1 Accuracy

The comparison accuracy for vfdt, evft, cvfdt and ovfdt are given in figure 5.8. As we can see in the graph, the accuracy of detecting a black hole attack tends to increase with increase in the size of data set, for example for VFDT, the value of accuracy increases from 92.1% for 5000 incidences to about 99.08% for 20,000 testing data.

It is evident from figures that evfdt shows the best accuracy for detecting a denial of service

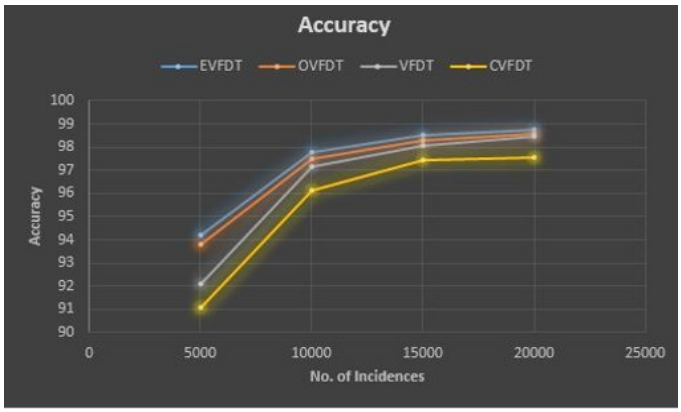

Figure 5.8: Accuracy Analysis for Black hole Attack

attack, followed by ovfdt, vfdt and cvfdt in this order.

5.4.2 False Positive Rate

Figure 5.9 shows the calculated value of false alarm for different variants of decision tree. It can be seen that, EVFDT shows lowest values of FPR at all data set sizes e.g. 3.37 as compared to 6.24 of CVFDT for 5000 incidences. It can be seen that the values of FPR for the various VFDT algorithms decrease with the increase in incidences and come close at 20,000 data set.

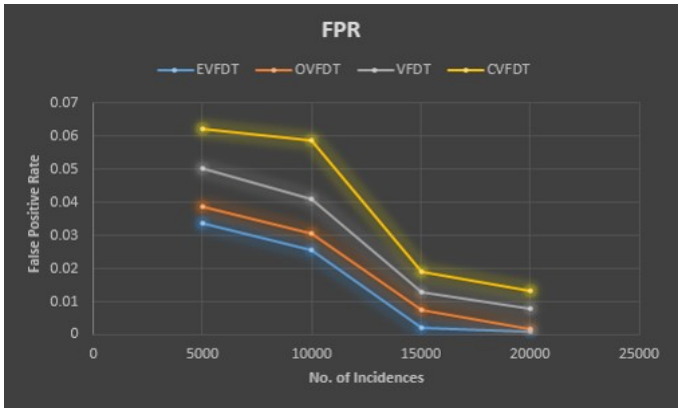

Figure 5.9: FPR Analysis for Black Hole Attack

5.4.3 Sensitivity

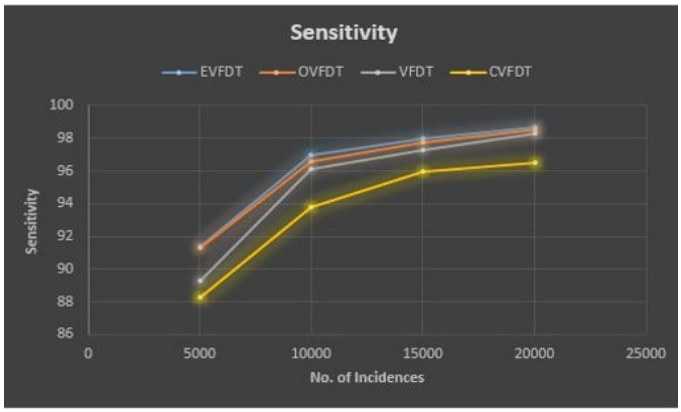

Figure 5.10: Sensitivity Analysis For Black Hole Attack

A high value of sensitivity is a requirement for an ideal classification mechanism. Figure 5.10 makes a comparative analysis of Sensitivity between VFDT, EVFDT, OVFDT and CVFDT. It is evident, that EVFDT has highest sensitivity for all sizes of data set and goes up to 99.91 for 20000 values, while CVFDT shows lowest sensitivity, due to more false alarms.

5.4.4 Specificity

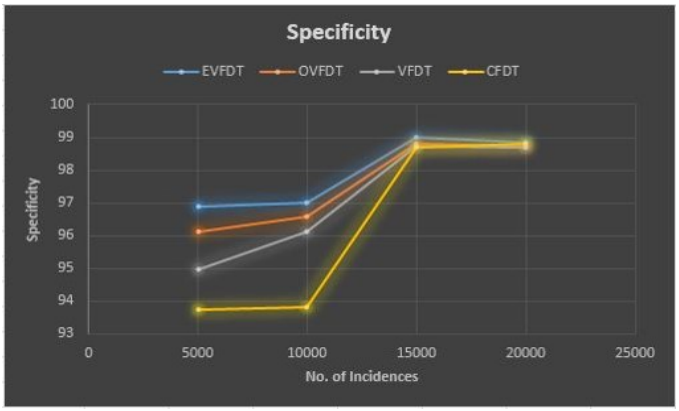

Figure 5.11: Specificity Analysis for Black Hole Attack

Figure 5.11 makes a comparative analysis of specificity between VFDT, EVFDT, OVFDT and CVFDT. It can be seen that EVFDT has highest specificity but the value of OVFDT comes close

to EVFDT's specificity as the size of data set increases.

5.4.5 Computational Memory

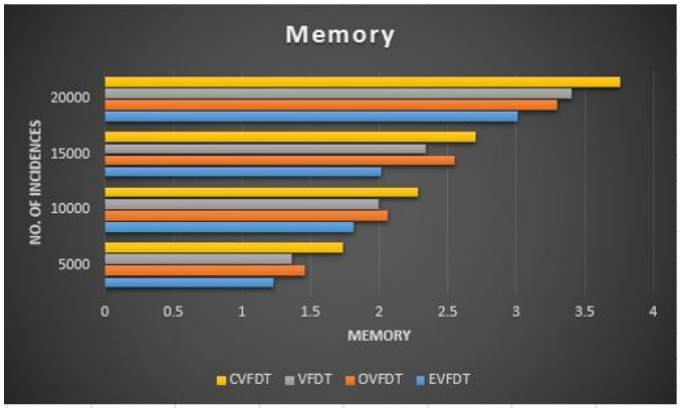

Figure 5.12: Memory Analysis for Black Hole Attack

As the size of the data set increases, the memory requirement also increases. Figure 5.12 makes a comparative analysis of specificity for the tested classification algorithms. It can be seen from the figure, that CVFDT uses maximum memory resources, due to its requirement of maintaining two data sets simultaneously, while EVFDT shows minimum memory usage.

5.4.6 Time

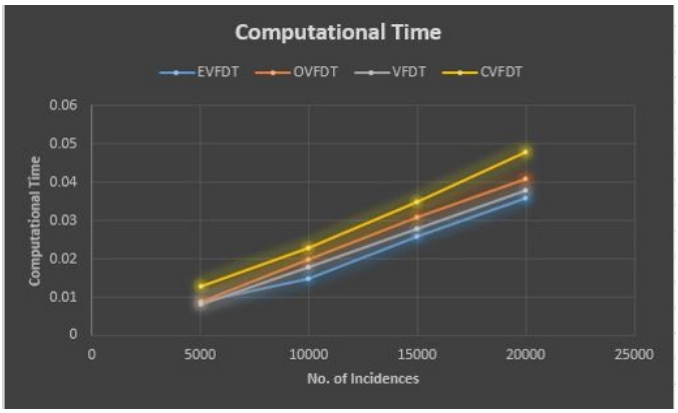

Figure 5.13: Time Analysis for Black Hole Attack

The computational/detection time is the time taken by CPU for training and testing the complete data set and is measured in seconds. As the size of the data set increases, the detection time also increases. Figure 5.13 compares the time required by the the different classification algorithms. It can be seen from the figure, that CVFDT takes the most time while EVFDT shows minimum value of detection/classification time due to run-time computation of tie-breaking threshold.

5.4.7 Tree Size

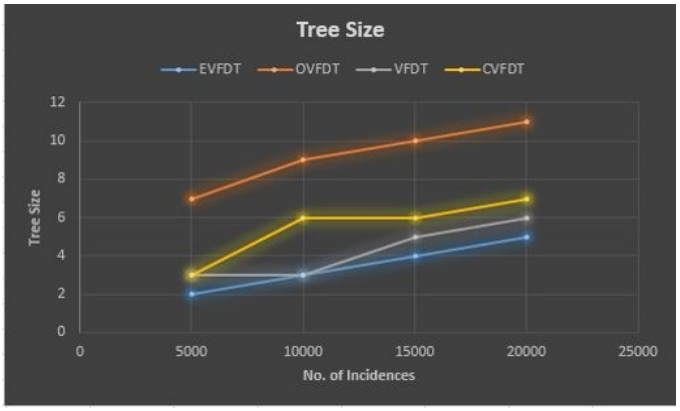

Figure 5.14: Tree Size Analysis for Black Hole Attack

A small tree size of the classification algorithm, results in lesser computational resource usage and detection time. Figure 5.14 compares the tree size of the different vfdt classification algorithms for varying data set sizes. The tree size increases with the increase in data set size and shows maximum value for OVFDT.

5.5 Qualitative Analysis VFDT Classification Algorithms

The results of the previous section show the performance of our detection model by using VFDT and its variant. A qualitative analysis to judge of the competence of each of these variants can be conducted by observation of the values in the table shown in figure 5.15. VFDT and EVFDT show high accuracy, sensitivity and specificity as compared to CVFDT and OVFDT. They have less FPR

and uses less memory and smaller tree size as compared to CVFDT and OVFDT. CVFDT takes more time due to fact that it builds two trees for the classification process.

Features	VFDT	CVFDT	OVFDT	EVFDT
Detection Accuracy	high	Low	Good; Does not handle outliers	Excellent
Time	Less time	More time in building two trees	Less time	Less time
Memory	less memory	More memory	Less memory	Less memory
Sensitivity/ Specificity	low	low	High	High
FPR	high	high	less	less

Figure 5.15: Qualitative Analysis VFDT Classification Algorithms

5.6 Comparison with existing techniques

The literature review in Chapter 2 depicted the shortcomings in existing detection schemes for DoS attack in VANETS due to the insufficient performance parameters, time consuming techniques, dependency on fixed infrastructure for framework employment etc. Our technique is not only lightweight and scalable, it can be employed at any destination node to satisfy the requirements of the ephemeral VANET environment. In order to make a comparative analysis, table in figure 5.6 depicts the ascendancy of our technique with reference to the existing detection mechanisms.

VFDT and its variants show high sensitivity and the benefit of handling multiple vehicles as compared to previous schemes. Moreover they have low FPR and take less time for classification. These parameters are essential for high performance of an attack detection function.

Features	VFDT	CVFDT	OVFDT	EVFDT	Other schemes
Sensitivity %	98.99	98.56	99.69	99.90	20-80 [Verma et al]
Time	0.023	0.029	0.025	0.02	7 [Roselin et al.] 0.23 [Bansal et al.] 7.5 [Singh et al.] 10-16 [Grover et al.] 50 -200 [Rahbari et al.]
Handle multiple vehicles	yes	yes	yes	yes	No [Quyoom et al.]
FPR	0.03	0.032	0.02	0.016	0.59 [Roselin et al.]

Figure 5.16: Comparison with existing techniques

5.7 Conclusion

A comprehensive analysis has been conducted to check the performance of very fast decision tree variants. The results of simulating distributed denial of service attack on AODV protocol for VANETs were gathered in section 4.8 and the data was labeled and fed into the classifier. The results are analyzed by the help of performance metrics discussed in section 5.2. The outcome of these experiments is exhaustively delineated in the subsequent sections by observing the behavior of the decision trees for both congestion attack (this includes jamming and flooding) and the black hole attacks. The performance analysis of the variants of VFDT schemes and their comparison with the existing techniques shows their preeminence for attack detection in VANETs.

Chapter 6

CONCLUSION

6.1 Contributions

Intrusion Detection systems are the heart of any communication system, due to the intrinsic vulnerabilities of attack on such networks. Detecting an attack is the first step in ensuring mitigation of attack for the successful transmission of data in a communication network. This dissertation deals with the issue of attack on Vehicular adhoc networks, an important subclass of Mobile adhoc networks. Due to the intrinsic nature of VANETs, specifically its wireless medium and user applications, VANETs are susceptible to a number of attacks which have been comprehensively discussed in chapter 1. An analysis on the frequencies of attacks on VANETs discussed in recent literature was carried out and it was found that the Denial of Service attack was mentioned more than the other violations on communication of VANETs. DDoS is a critical attack rendering the network unavailable, bereaving the legitimate users from using VANET applications or even resulting in critical accidents on roads.

The schemes more frequently used in recent research papers, for attack detection in VANETs are analyzed exhaustively in chapter 2, ensuring that a systematic literature review is conducted based on peer-reviewed conference papers and journals from recent years. These schemes were found to be either based on cryptographic means, vehicle location/trajectory, variations in packet/network features or using trust based relationship/reputation which come with numerous open issues. Apart from the burden of complexity in cryptographic techniques, these schemes have various

deficiencies which makes them incompatible for attack detection in VANETs. The weaknesses have been comprehensively discussed in the literature review. Future research pertaining to security issues other than DoS can also benefit from the analysis in this review.

A destination based attack detection scheme is explored in the thesis which should be computationally efficient and error free, to provide a secure environment for optimum delivery of VANETs applications. For this purpose, data mining and in particular stream mining have been chosen as the detection scheme to impede the denial of service attack on VANETS. Decision Trees are a lesser explored subclass of stream mining which have not been effectively exploited for the purpose of attack detection in vehicular adhoc networks. A number of simulation based experiments have been carried out in chapter 3 to evaluate the performance of various decision trees to test their competence in detection DDoS attack on high speed VANETs. The first step in the process was to simulate the vehicular adhoc network including the mobility generation of the vehicles in the network. This required the selection of performance metrics, protocols and parameters reflecting the conditions of a vehicular network.

A research was carried out to find out the best possible and currently standardized network parameters to ensure correct simulation of VANETs. This resulted in the selection of 802.11p parameters discussed in 4. Before implementing this new protocol, simulations were run for both 802.11a and 802.11p in order to establish this preference with conclusive results. The TCL script was written to include the mentioned parameters and simulated on 10 vehicles speeding at 70km per hour to emulate a city/urban scenario and the nam trace files were obtained. Based on the research compilations of chapter 2 four performance features were picked to compare the parameters. These included, throughput, average delay, drop ratio and packet delivery ratio, for which separate awk scripts were written to extract the results from the nam trace files.

In order to simulate real time vehicular traffic for testing and evaluation, both the size of the road topology as well as the number of vehicles were augmented. This was done by employing MOVE and SUMO, whereby the resulting trace files were input into NS2 for network simulation over the generated map and mobility traces. A real time map of a busy city flyover (Faizabad flyover) of Pakistan was imported from Open Street Map to perform experiments on real time basis. This map was imported into MOVE and SUMO. Moreover, a framework was established to import map data into simulation software, so that evaluations can be implemented on real city/urban scenario. The mobility file and cbr file (explained in this section) are attached in a single tcl code and processed to get a nam and trace file reflecting data transmission employing AODV protocol for nodes in the Faizabad map. It should be mentioned here that pseudo codes for all the steps that have been discussed above, have been provided in the aforementioned chapter. This design aids in a standard process of traffic and network simulation to ensure ease of use and scalability while adopting the detection scheme for different landscapes and traffic scenarios.

It has been shown in chpater 4 that distributed denial of service attack has two active attack methods which include the various attacks mentioned in literature review of chapter 2. To cater for both these categories, we simulated both types of attack on the nodes in order to accommodate for all types of DDoS attacks on VANETs. In order to replicate the first type of attack scenario, i.e. congestion/jamming, the data rate was increased in the algorithm for all the transmitting nodes. For the purpose of reproducing the second type of attacks, a blackhole attack agent was attached to nodes near or adjacent to the receiving node. The pseudo codes for both these attack has been provided in chapter 4. It is safe to assume here that this almost completely covers the range of Distributed denial of service attacks impeding the successful communications in VANETs.

A comprehensive framework has been proposed for the detection of DDoS attack on VANETs

in chapter 4 which not only address all the issues discussed in the DDoS attack detection techniques reviewed in 2, but also proposes a scheme which is light weight and scalable. It presents a methodology which avoids cumbersome cryptographic procedures and reputation based procedures which can easily fall prey to inside attacks and malicious agents. The chapter includes the learning and training steps of the detection scheme and a detailed description of data mining has been provided alongwith a comparison of decision trees with other data mining techniques, authorizing the prepotence of decision trees over the other pattern discovery mechanisms. Four decision tree, pattern recognition algorithms were applied in the detection framework and a comparative analysis was carried out to compare them. The applied schemes are not only light weight and scalable but are not dependent upon extra infrastructure which can become a liability incase it is already compromised. The technique can be easily employed in the on-board unit (OBU) of the vehicles or the RSU.

To benchmark the dexterity of decision trees against other detection schemes, a quantitative analysis is executed by testing the results of the detection scheme with performance metrics including accuracy, false alarm rates, computational resource usage, sensitivity and specificity. Various amounts of data sets were used for an encompassing investigation. The audit is performed for data from both the types of DDoS attacks quoted in chapter 4 separately. It was concluded for both these types that the attack detection accuracy, sensitvity, specificity, memory and tree size increase with the increment in the number of incidences whereas the false alarm rates tend to decrease with this accretion. EVFDT has performed best over the board with all these performance metrics although it does show increased values of memory, tree size and time usage. The prepotence of the rest are as follows in descending order, OVFDT, VFDT, CVFDT. A comparative analysis of VFDT based detection scheme for DDoS in VANETs shows its preeminence as compared to the existing schemes and satisfy the shortcomings of the existing techniques discussed in

the literature review.

6.2 Future Work

The research work has contributed towards providing a framework for the simulation of vehicular adhoc networks using real, urban/city map scenario and to implement a new detection scheme that has not been previously explored for VANETs. A detection Model and Framework has been presented for this purpose. In the future other attacks which have been discussed in chapter 2 on Vehicular adhoc networks could be detected by applying decision trees and a trace back mechanism to apprehend the source of the malicious attack can be implemented. Furthermore, the proposed frameworks and the mentioned decision trees or data mining mechanisms can be refined to garner improved results. It is important to mention that the establishment of a robust and secure transport system, requires the VANETs technology to be deployed in maximum number of vehicles and infrastructure to ensure that the security mechanisms work efficiently to improve the drivers experience and safety.

BIBLIOGRAPHY

[1] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on vanet security challenges and possible cryptographic solutions," *Vehicular Communications*, vol. 1, no. 2, pp. 53–66, 2014.

[2] J. Harding, G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons, and J. Wang, "Vehicle-to-vehicle communications: Readiness of v2v technology for application," tech. rep., 2014.

[3] C. Sommer and F. Dressler, *Vehicular networking*. Cambridge University Press, 2014.

[4] A. Dhamgaye and N. Chavhan, "Survey on security challenges in vanet 1," 2013.

[5] K. Verma and H. Hasbullah, "Ip-chock (filter)-based detection scheme for denial of service (dos) attacks in vanet," in *Computer and Information Sciences (ICCOINS), 2014 International Conference on*, pp. 1–6, IEEE, 2014.

[6] S. M. Safi, A. Movaghar, and M. Mohammadizadeh, "A novel approach for avoiding wormhole attacks in vanet," in *2009 First Asian Himalayas International Conference on Internet*, pp. 1–6, IEEE, 2009.

[7] S. Ruj, M. A. Cavenaghi, Z. Huang, A. Nayak, and I. Stojmenovic, "On data-centric misbehavior detection in vanets," in *Vehicular technology conference (VTC Fall), 2011 IEEE*, pp. 1–5, IEEE, 2011.

[8] J. Grover, N. K. Prajapati, V. Laxmi, and M. S. Gaur, "Machine learning approach for multiple misbehavior detection in vanet," in *International Conference on Advances in Computing and Communications*, pp. 644–653, Springer, 2011.

[9] U. Khan, S. Agrawal, and S. Silakari, "Detection of malicious nodes (dmn) in vehicular ad-hoc networks," *Procedia Computer Science*, vol. 46, pp. 965–972, 2015.

[10] K. Verma, H. Hasbullah, and A. Kumar, "Prevention of dos attacks in vanet," *Wireless personal communications*, vol. 73, no. 1, pp. 95–126, 2013.

[11] A. Hamieh, J. Ben-Othman, and L. Mokdad, "Detection of radio interference attacks in vanet," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pp. 1–5, IEEE, 2009.

[12] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (vanets): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[13] M. N. Mejri and J. Ben-Othman, "Detecting greedy behavior by linear regression and watchdog in vehicular ad hoc networks," in *2014 IEEE Global Communications Conference*, pp. 5032–5037, IEEE, 2014.

[14] S. Djahel and Y. Ghamri-Doudane, "A robust congestion control scheme for fast and reliable dissemination of safety messages in vanets," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2264–2269, IEEE, 2012.

[15] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering–a systematic literature review," *Information and software technology*, vol. 51, no. 1, pp. 7–15, 2009.

[16] L. Mokdad, J. Ben-Othman, and A. T. Nguyen, "Djavan: Detecting jamming attacks in vehicle ad hoc networks," *Performance Evaluation*, vol. 87, pp. 47–59, 2015.

[17] N. Lyamin, A. V. Vinel, M. Jonsson, and J. Loo, "Real-time detection of denial-of-service attacks in ieee 802.11 p vehicular networks.," *IEEE Communications letters*, vol. 18, no. 1, pp. 110–113, 2014.

[18] H. Nguyen-Minh, A. Benslimane, and A. Rachedi, "Jamming detection on 802.11 p under multi-channel operation in vehicular networks," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on*, pp. 764–770, IEEE, 2015.

[19] D. Giustiniano, V. Lenders, J. B. Schmitt, M. Spuhler, and M. Wilhelm, "Detection of reactive jamming in dsss-based wireless networks," in *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, pp. 43–48, ACM, 2013.

[20] P. Bansal, S. Sharma, and A. Prakash, "A novel approach for detection of distributed denial of service attack in vanet," *International Journal of Computer Applications*, vol. 120, no. 5, 2015.

[21] H. Sedjelmaci, S. M. Senouci, and M. A. Abu-Rgheff, "An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks," *IEEE Internet of Things Journal*, vol. 1, no. 6, pp. 570–577, 2014.

[22] A. Singh and P. Sharma, "A novel mechanism for detecting dos attack in vanet using enhanced attacked packet detection algorithm (eapda)," in *2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS)*, pp. 1–5, IEEE, 2015.

[23] S. RoselinMary, M. Maheshwari, and M. Thamaraiselvan, "Early detection of dos attacks in

vanet using attacked packet detection algorithm (apda)," in *Information Communication and Embedded Systems (ICICES), 2013 International Conference on*, pp. 237–240, IEEE, 2013.

[24] A. Quyoom, R. Ali, D. N. Gouttam, and H. Sharma, "A novel mechanism of detection of denial of service attack (dos) in vanet using malicious and irrelevant packet detection algorithm (mipda)," in *Computing, Communication & Automation (ICCCA), 2015 International Conference on*, pp. 414–419, IEEE, 2015.

[25] A. Temurnikar and D. Sharma, "Secure and stable vanet architecture model," *International Journal of Computer Science and Network*, vol. 2, no. 1, pp. 37–43, 2013.

[26] S. Ghorsad, P. Karde, V. Thakare, and R. Dharaskar, "Dos attack detection in vehicular ad-hoc network using malicious node detection algorithm," *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)*, vol. 3, p. 36, 2014.

[27] K. Sahare and D. L. Hasbullah, "Review- technique for detection of attack in vanet," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 2, pp. 580–584, 2014.

[28] A. K. Malhi and S. Batra, "Decision inference system for misbehavior detection in vanets," in *Electronics and Communication Systems (ICECS), 2015 2nd International Conference on*, pp. 1558–1563, IEEE, 2015.

[29] T. Yang, W. Xin, L. Yu, Y. Yang, J. Hu, and Z. Chen, "Misdis: An efficent misbehavior discovering method based on accountability and state machine in vanet," in *Asia-Pacific Web Conference*, pp. 583–594, Springer, 2013.

[30] U. D. Gandhi and R. M. Keerthana, "Request response detection algorithm for detecting dos

attack in vanet," in *Optimization, Reliabilty, and Information Technology (ICROIT), 2014 International Conference on*, pp. 192–194, IEEE, 2014.

[31] S. S. Kaushik, "Review of different approaches for privacy scheme in vanets," *Int. J. Adv. Eng. Technol*, vol. 5, pp. 2231–1963, 2013.

[32] M. Kaur and M. Mahajan, "A novel security approach for data flow and data pattern analysis to mitigate ddos attacks in vanets," *International Journal of Hybrid Information Technology*, vol. 8, no. 8, pp. 113–122, 2015.

[33] L. He and W. T. Zhu, "Mitigating dos attacks against signature based authentication in vanets," in *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*, vol. 3, pp. 261–265, IEEE, 2012.

[34] M. S. Naveed and M. H. Islma, "Detection of sybil attacks in vehicular ad hoc networks," 2015.

[35] T. Zhou, R. R. Choudhury, P. Ning, and K. Chakrabarty, "P2dap sybil attacks detection in vehicular ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 582–594, 2011.

[36] B. Yu, C.-Z. Xu, and B. Xiao, "Detecting sybil attacks in vanets," *Journal of Parallel and Distributed Computing*, vol. 73, no. 6, pp. 746–756, 2013.

[37] N. Bissmeyer, C. Stresing, and K. M. Bayarou, "Intrusion detection in vanets through verification of vehicle movement data," in *Vehicular Networking Conference (VNC), 2010 IEEE*, pp. 166–173, IEEE, 2010.

[38] J. Grover, M. S. Gaur, and V. Laxmi, "A novel defense mechanism against sybil attacks in

vanet," in *Proceedings of the 3rd international conference on Security of information and networks*, pp. 249–255, ACM, 2010.

[39] S. Chang, Y. Qi, H. Zhu, J. Zhao, and X. Shen, "Footprint: detecting sybil attacks in urban vehicular networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 1103–1114, 2012.

[40] X. Feng, C.-y. Li, D.-x. Chen, and J. Tang, "A method for defensing against multi-source sybil attacks in vanet," *Peer-to-Peer Networking and Applications*, pp. 1–10, 2016.

[41] S. Park, B. Aslam, D. Turgut, and C. C. Zou, "Defense against sybil attack in the initial deployment stage of vehicular ad hoc network based on roadside unit support," *Security and Communication Networks*, vol. 6, no. 4, pp. 523–538, 2013.

[42] C. Chen, X. Wang, W. Han, and B. Zang, "A robust detection of the sybil attack in urban vanets," in *Distributed Computing Systems Workshops, 2009. ICDCS Workshops' 09. 29th IEEE International Conference on*, pp. 270–276, IEEE, 2009.

[43] A. Mohaisen and S. Hollenbeck, "Improving social network-based sybil defenses by rewiring and augmenting social graphs," in *International Workshop on Information Security Applications*, pp. 65–80, Springer, 2013.

[44] J. Grover, M. S. Gaur, V. Laxmi, and N. K. Prajapati, "A sybil attack detection approach using neighboring vehicles in vanet," in *Proceedings of the 4th international conference on Security of information and networks*, pp. 151–158, ACM, 2011.

[45] S.-w. Chang, J. Cha, and S.-s. Lee, "Adaptive edca mechanism for vehicular ad-hoc network," in *The International Conference on Information Network 2012*, pp. 379–383, IEEE, 2012.

[46] M. N. Mejri and J. Ben-Othman, "Entropy as a new metric for denial of service attack detection in vehicular ad-hoc networks," in *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pp. 73–79, ACM, 2014.

[47] S. Gupta, S. Kar, and S. Dharmaraja, "Whop: Wormhole attack detection protocol using hound packet," in *Innovations in information technology (IIT), 2011 international conference on*, pp. 226–231, IEEE, 2011.

[48] Q. Wu, J. Domingo-Ferrer, and U. González-Nicolás, "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 559–573, 2010.

[49] G. Yan, *Providing location security in vehicular Ad Hoc networks*. PhD thesis, Old Dominion University, 2010.

[50] M. Rahbari and M. A. J. Jamali, "Efficient detection of sybil attack based on cryptography in vanet," *arXiv preprint arXiv:1112.2257*, 2011.

[51] A. M. Malla and R. K. Sahu, "Security attacks with an effective solution for dos attacks in vanet," *International Journal of Computer Applications*, vol. 66, no. 22, 2013.

[52] N. Gandhewar and R. Patel, "Detection and prevention of sinkhole attack on aodv protocol in mobile adhoc network," in *Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on*, pp. 714–718, IEEE, 2012.

[53] L. Chen, S.-L. Ng, and G. Wang, "Threshold anonymous announcement in vanets," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 605–615, 2011.

[54] C.-M. Huang, *Telematics Communication Technologies and Vehicular Networks: Wireless Architectures and Applications: Wireless Architectures and Applications*. IGI Global, 2009.

[55] P. Manzoni, M. Fiore, S. Uppoor, F. J. M. Domínguez, C. T. Calafate, and J. C. C. Escriba, "Mobility models for vehicular communications," in *Vehicular ad hoc Networks*, pp. 309–333, Springer, 2015.

[56] A. Hesham, A. Abdel-Hamid, and M. A. El-Nasr, "A dynamic key distribution protocol for pki-based vanets," in *Wireless Days (WD), 2011 IFIP*, pp. 1–3, IEEE, 2011.

[57] OpenStreetMap, "Tiger products - geography - u.s. census bureau." lhttp://www.census.gov/geo/maps-data/data/tiger.html, 2015. [Online; accessed 18-February-2016].

[58] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "Bonnmotion: a mobility scenario generation and analysis tool," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, p. 51, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010.

[59] E. Gamess and M. Chachati, "Analyzing routing protocol performance with nctuns for vehicular networks," *Indian Journal of Science and Technology*, vol. 7, no. 9, pp. 1391–1402, 2014.

[60] OpenStreetMap, "Openstreetmap." https://www.openstreetmap.org/map, 2004. [Online; accessed 18-February-2016].

[61] M. Altayeb and I. Mahgoub, "A survey of vehicular ad hoc networks routing protocols," *International Journal of Innovation and Applied Studies*, vol. 3, no. 3, pp. 829–846, 2013.

[62] Z. Wu, Y. Yang, X. Guo, and J. An, "Analysis of collision probability in ieee 802.11 based vanets," *Chinese Journal of Electronics*, vol. 19, no. 1, pp. 187–190, 2010.

[63] A. Shastri, R. Dadhich, and R. C. Poonia, "Performance analysis of on-demand routing protocols for vehicular ad-hoc networks," *International Journal of Wireless & Mobile Networks (IJWMN) Vol*, vol. 3, pp. 103–111, 2011.

[64] W. Arshad, N. Javaid, R. Khan, M. Ilahi, U. Qasim, and Z. Khan, "Modeling and simulating network connectivity in routing protocols for manets and vanets," *arXiv preprint arXiv:1306.0757*, 2013.

[65] ITSstandard, "Intelligent transportation systems - u.s. department of transportation." https://www.standards.its.dot.gov/Factsheets/Factsheet/80, 2009. [Online; accessed 25-March-2016].

[66] A. V. Aho, B. W. Kernighan, and P. J. Weinberger, *The AWK programming language*. Addison-Wesley Longman Publishing Co., Inc., 1987.

[67] A. T. Jonathan Bennett, "Autoit scripting language." https://www.autoitscript.com/site/autoit/, 2015. [Online; accessed 19-May-2016].

[68] H. Yang and S. Fong, "Moderated vfdt in stream mining using adaptive tie threshold and incremental pruning," in *International Conference on Data Warehousing and Knowledge Discovery*, pp. 471–483, Springer, 2011.

[69] J. Yu, H. Kang, D. Park, H.-C. Bang, and D. W. Kang, "An in-depth analysis on traffic flooding attacks detection and system using data mining techniques," *Journal of Systems Architecture*, vol. 59, no. 10, pp. 1005–1012, 2013.

[70] R. Latif, H. Abbas, and S. Latif, "Distributed denial of service (ddos) attack detection using data mining approach in cloud-assisted wireless body area networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 23, no. 1-2, pp. 24–35, 2016.

[71] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 97–106, ACM, 2001.

[72] R. Latif, H. Abbas, S. Latif, and A. Masood, "Evfdt: an enhanced very fast decision tree algorithm for detecting distributed denial of service attack in cloud-assisted wireless body area network," *Mob. Inf. Syst*, pp. 1–13, 2015.

[73] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.