Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

Detection of Wormhole Attack in Vehicular Ad-hoc Network over Real Map using Machine Learning Approach with Preventive Scheme

Shahjahan Ali*

*Corresponding author, Assistant Professor, Department of Computer Science & Engineering at SRMSCET, Bareilly (UP) India, Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh, India. E-mail: shahjahansrms@gmail.com

Parma Nand

Professor and Dean Academic Affairs, Sharda University Greater Noida (U.P.) India, Pin Code:201306. E-mail: parmaastya@gmail.com

Shailesh Tiwari

Professor and Director, KEC, Ghaziabad (U.P.) India, Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh, India. E-mail: shail.tiwari@yahoo.com

Abstract

VANET (Vehicular Ad-hoc Network) is a developing technology, which is a combination of cellular technology, ad-hoc network & wireless LAN to improve the safety of vehicle as well as driver. VANET communication can be of two types, first one is broadcast and second one is unicast. Either communication may be broadcast or unicast both are sensitive to different types ofassaults, for example message forgery, (DOS) denial of service, Sybil assault, Greyhole, Blackhole & Wormhole assault. In this paper machine learning method is used to detect the wormhole assault in VANET's multi-hop communication. We have created a scenario of VANET by using AODV routing protocol on NS-3.24.1 simulator, which utilizes the overall mobility traces generated by the simulator SUMO-0.32.0 to model the wormhole assault. The simulation is performed by using NS-3.24.1 simulator, and the statistics created by flow monitor are collected. The collected data is pre-processed and the k-NN & Random Forest algorithms are applied on this data, to make the model such type so that it can memorize the wormhole attack. The novelty of this research work is that with the help of proposed detection & prevention technique, vehicular ad-hoc network can be made free from wormhole assault by using ML approach. The performance of proposed machine learning models is compared with existing work. In this way it is clear that our proposed approach by using ML is powerful tool by which the wormhole assaults can be detected in VANETs. A scheme based on packet lease and cryptographic techniques is used to prevent the wormhole attack in VANET.

Keywords: VANET, AODV, Broadcast, Unicast, k-NN, Random Forest, SUMO-0.32.0, NS-3.24.1, Packet leash,, Cryptography.

DOI: https://doi.org/ 10.22059/jitm.2022.86658 Manuscript Type: Research Paper University of Tehran, Faculty of Management

Received January 12, 2021 Accepted March 25, 2021

Introduction

a. Vehicular Ad-hoc Network

(VANET)Vehicular Ad-hoc Network is a subset of (MANET) Mobile Ad-hoc Network), having frequently changing topology, high mobility and a few other intricacies (Bakhouya et al., 2011). VANETs are susceptible to big range of security threads due to the openness of network, lacking of fixed infrastructure, in which the participating nodes (vehicles) can create a network freely with no requirement of pre-deployed communication framework (Ali et al. 2020). The structure of VANET is specified is given below:

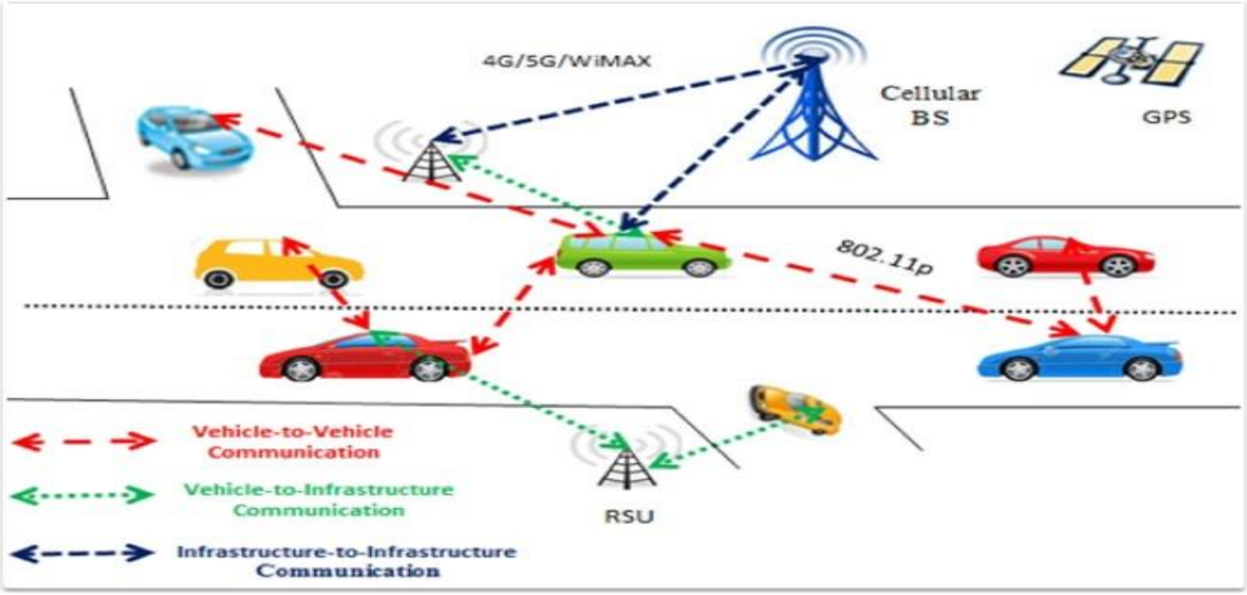

4G/5G/WiMAX
Cellular BS
GPS
802.11p
RSU
Vehicle-to-Vehicle Communication
Vehicle-to-Infrastructure Communication
Infrastructure-to-Infrastructure Communication

Figure 1. Architecture of VANET (Chahal et al., 2017)

b. Wormhole Attack

Wormhole attack is such type of attack which can interrupt the routing mechanism of AODV and DSR routing protocol in VANET (Albouq & Fredericks, 2017). The wormhole contains at least two malicious vehicles. These malicious vehicles form a tunnel (Singh et al., 2019). Following figure 2 shows a wormhole assault:

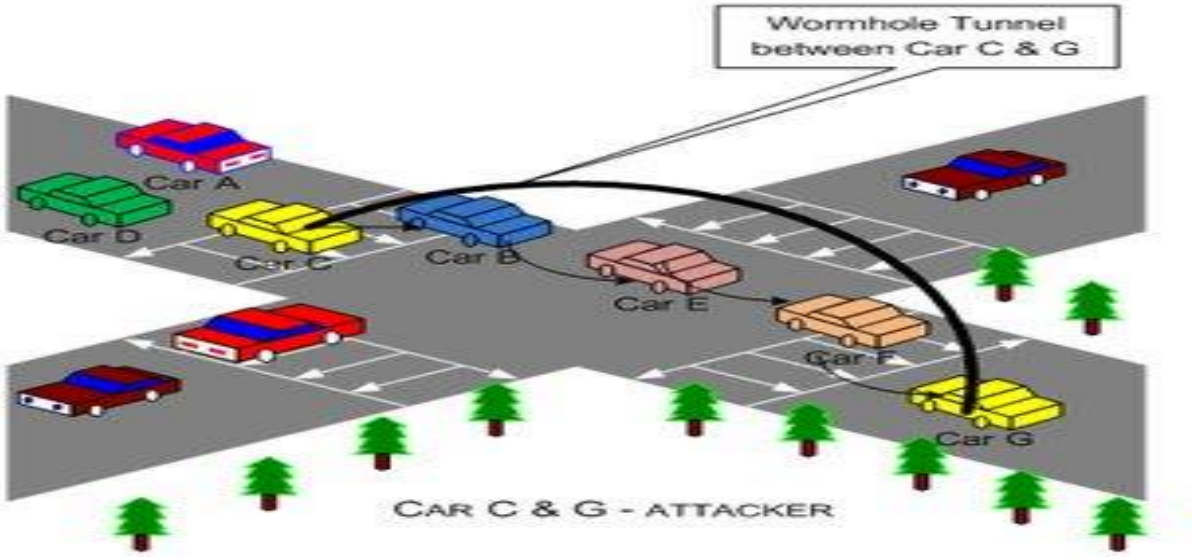


Figure 2.Wormhole assault (Upadhyaya & Shah, 2018)

The intention of these malicious vehicles is to mislead permissible vehicles and pretend them as they are neighbors (Singh et al., 2019). The present security structure, based on public key encryption is invulnerable to outsider assaults. To detect such type of misbehaving nodes inside the network is very difficult (Singh et al., 2019). In this research paper, a machine learning based approach is used, to detect the wormhole assault in VANET. A scheme which is combination of packet leash (Ali et al., 2017) and cryptography is also proposed to stave off the wormhole assault. Here, research work depends on wormhole assault with Ad-hoc On-Demand Distance Vector (AODV) routing protocol. First, machine learning models are trained, and then these machine learning models are used to foretell the deportment of every vehicle in VANET according to rules those machine learning paradigm have learned. In this research work the performance of proposed ML models is also compared with existing work.

c. Literature survey

In (Bakhouya et al., 2011), the authors given an adjustive approach for information metastasizing in VANETs. In this method, every participating node stabilizes the values of local parameters utilizing local data (such as number of superfluous messages, inter-arrival time), related to the messages which are obtained from neighboring vehicles with no attempt.

Strengths:

1. Every node adjusts local parameters by using local information at run time.
2. The performance of proposed approach is measured on different traffic scenario & mobility.

Weaknesses*:*

Redundant broadcast messages indirectly affect the performance of network.

In (Albouq & Fredericks, 2017), the authors introduced a lightweight wormhole protocoldetector (WPD) protocol to discover as well as alleviate wormhole assaults.

Strengths:

1. The proposed protocol can be used on highway.
2. WPD can watch and detect out-of range packets.
3. It can also guess the hop count between source plus destination vehicles.
4. It is a lightweight protocol

Weaknesses:

1. The proposed protocol is not significant to detect the encapsulation wormhole assault
2. Packets are not authorized in detection phase.

In (Singh et al., 2019), a scheme is proposed based on ML to detect wormhole attack in VANETs.

Strengths:

1. Proposed ML based approach having a big potential.
2. It can be used to detect the assaults on various layers of VANET protocol stack.
3. In this research paper a self created data set is used.
4. The proposed model find out the wormhole assault at very high accuracy.

Weakness:

In this research paper it is not discussed how the data is pre-processed.

(Hu et al., 2003) presented wormhole assault as well as cited how dangerous this assault is in MANETs. It's not easy to protect towards wormhole assault, and can form a big threat particularly towards several ad-hoc routing protocols.

Strengths:

1. In this paper a packet leash is used which can restrict the transmission distance of packet to detect & defend the wormhole attack.
2. The TIK provides the authentication of received packets.

Weaknesses:

In this proposed approach there is a problem of time synchronization node movement with speed of light.

In (Grover et al., 2011), a ML-based approach is discussed which is used to differentiate the vehicle’s behaviour.

Strengths:

1. In this research work various misbehaviours are implemented: like J-48, Naive Bayes, Rand Forest and Ada Boost1.
2. The proposed approach is efficient in classifying different nodes, which are not co-ordinating in the communication.

Weakness:

Here WEKA tool is used, which cannot handle a big data set.

In (Kang & Kang, 2016), the authors suggested a (DNN) deep neural network supported process to find out.

Strength:

The proposed technique provides the real-time response for assault.

Weakness:

The proposed approach does not give the efficient result if data set is very big.

In (Loukas et al., 2018), authors suggested IDS based on deep learning, which is implemented on cloud to find cyber-physical assaults within the vehicle.

Strengths:

1. Proposed approach uses lightweight ML technique.
2. The accuracy of this approach is very high

Weaknesses:

1. In proposed approach there is a problem of security.
2. There is no security measure corresponding to communication jamming.

(Ali et al., 2016) proposed ML-based approach for detecting gray hole as well as rushing assault in vehicular plan.

Strengths:

1. In this research paper intelligent Intrusion Detection System(IDS) is proposed, which is useful to protect the external communication system.
2. Here feed-forward neural network & SVM is used, which are having more advantages as compare to other techniques.

Weaknesses:

1. Proposed approach is prone towards over fitting.
2. The proposed approach is also taking more time to converge.

At different plane, different machine learning based methods have been implemented to find out the assaults. But machine learning based models, to detect the wormhole assault in VANET over real map are used in few research works which do not have good detection performance. In this research work machine learning based approach is used to find out the wormhole assault in VANET with better performance as compare to existing work. So this work will have a significant role in research.

d. Problem Definition

VANET is latest technology by which the journey can be made more easy and secure. In VANET participating nodes are the vehicles. Due to mobility of vehicles the topology changed very rapidly. So routing is main issue. Due to wireless nature of vehicles the VANET is more suspected to the attacks. The VANET can be made more secure by detecting and preventing various attacks such as wormhole attack. In this research paper ML based approach is used to detect the wormhole assault to make the VANET more secure. An approach based on cryptographic concept and packet leash is also proposed to stave off the wormhole assault.

System Proposed

a. Overview

VANET is a subset of MANET, in which V2V & V2I communication takes place to enhance traffic and security applications. Real map scenario is considered to detect the wormhole assault in VANET by using machine learning based approach. For the same VANET scenario, an approach based on cryptographic technique and packet leash is also given to prevent the wormhole attack.

In this work for V2V and V2I correspondence, AODV routing protocol is considered. By proposed approach of detection & prevention of wormhole attack, VANET can be made more secure in future.

b. Proposed approach to detect the wormhole attack based on machine learning in vehicular ad-hoc network

Here, a machine learning based wormhole assault discovery system is proposed which find out the conduct of participating vehicles in vehicular ad-hoc networks. The approach utilizes the data collected from simulator which contains both normal as well as abnormal conduct (under wormhole attack) of vehicles in VANETs.

The steps which are followed in this proposed approach are given as follow:

i. Scenario generation by using SUMO-032.0 simulator: With the help of real map and SUMO-0.32.0 simulator, the scenario is created.

The scenario for VANET is given as follow:

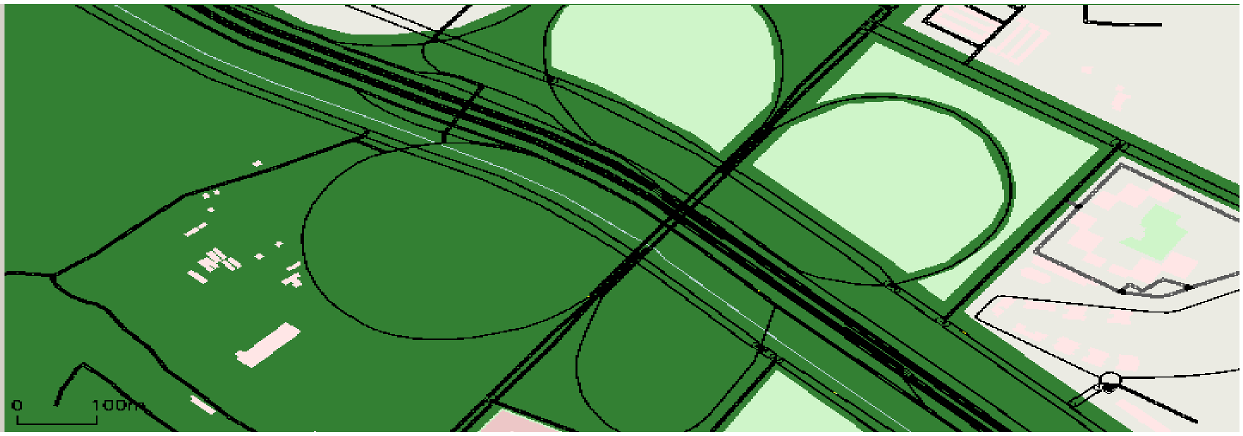


Figure 3. Scenario for VANET

ii. Machine Learning Models for Detection Wormhole Attack

K-nearest neighbors (k-NN) and Random Forest machine learning models are used in this research work.

The k-NN algorithm (Altman, 1992) may also resolve each categorization & retrogression issues, but primarily being applied for categorization work. For each data point, we have assumed k closest training datums as well as predict the most happening class for any test datum. Here k is a hyper parameter which expresses training datum numbers to be assumed for labelling test datum.

Definitely Random Forest is a group classifier, which creates a group of self-sufficient as well as non-identical decision trees according to the idea of randomization. Random forest may be defined as $\{h(x,\theta_k),k=1,2,\ldots\ldots L\}$, in which θ_k is a kind of mutual self-sufficient random vector variables, as well as x is the input data (Provost et al., 2016). The random forest having following advantages:

1. The random forest model can also deal with huge dataset.
2. It can also balance datasets automatically.

c. Proposed Scheme to Prevent Wormhole Attack

An attacker can access to global parameter n and g which isn't having adequate information to possibly calculate K_s. However, an assault might be able to intercept and modify all messages as well as negotiate a secret key K_A with A and K_B with B. The standard Diffie-Hellman key distribution scheme is prone to such man-in-the middle attacks.

The proposed scheme to prevent wormhole attack is divided into following two algorithms:

Proposed algorithm to distribute the shared key K_s

Algorithm proposed, to stave off wormhole assault in VANET

In proposed algorithm to distribute the shared key K_s, man-in-middle attack is eliminated by keeping parameters n and g secret; means the encrypted form of n and g tend to be pre-initialized in vehicle OBU’s memory before coming on the road. In case any vehicle is captured, hacker will be unable to read n and g.

i. **Proposed algorithm to distribute the shared key K_s:** Proposed algorithm to distribute the shared key K_s having two phases:

Phase 1: (Before coming the vehicle on road): Design& deploy the same random number generator to every vehicle before coming on road. Assign the secret number S and the common number n and g to each vehicle manually, before using the vehicle on road in an encrypted form ($n \oplus R_i$ && $g \oplus R_i$), in the memory of OBU’s, where:

$$S \rightarrow \boxed{\text{Random Number}}$$

Phase 2 (key generation):

1. Vehicle generates a secret random number q repeatedly and calculates $Q = (q \oplus S)$.
2. Broadcast Q to all other vehicles.
3. Every vehicle on receiving Q calculates the following:

(a).Secret number $X' = Q \oplus S = q$

(b).Decrypt the common number n and g.

4. Then every vehicle calculate key $K_s = n^{X'}$ mod g.

The above algorithm (Kumar & Singh, 2016) is proposed to distribute the shared key in WSN. But in this research paper it is proposed for VANET. The above shared key (K_s) is used for the further communication to prevent the wormhole attack.

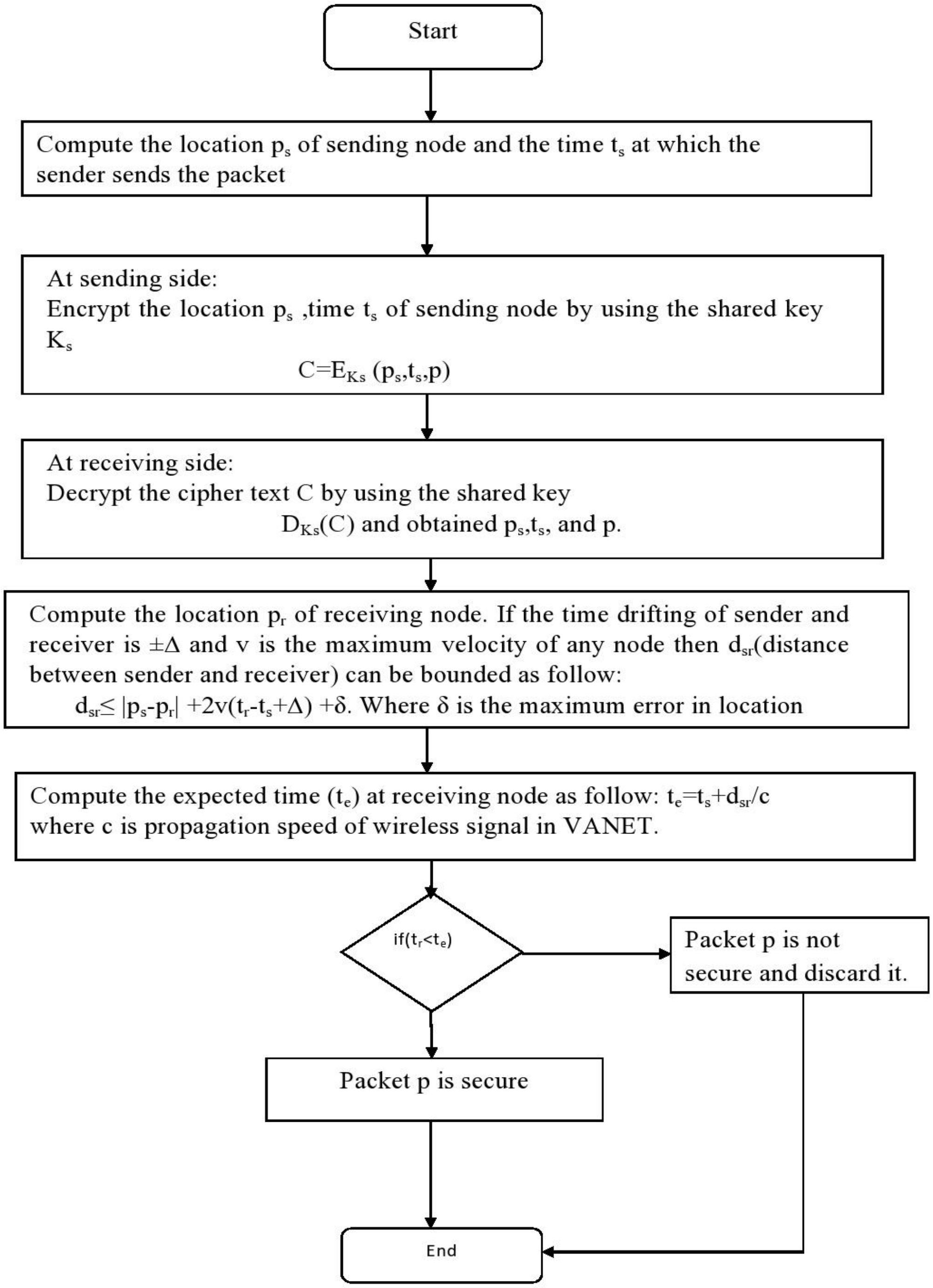

Start
Compute the location p_s of sending node and the time t_s at which the sender sends the packet
At sending side:
Encrypt the location p_s ,time t_s of sending node by using the shared key K_s
$C=E_{Ks}(p_s,t_s,p)$
At receiving side:
Decrypt the cipher text C by using the shared key
$D_{Ks}(C)$ and obtained p_s,t_s, and p.
Compute the location p_r of receiving node. If the time drifting of sender and receiver is ±Δ and v is the maximum velocity of any node then d_{sr}(distance between sender and receiver) can be bounded as follow:
$d_{sr} \leq |p_s-p_r| +2v(t_r-t_s+\Delta) +\delta$. Where δ is the maximum error in location
Compute the expected time (t_e) at receiving node as follow: $t_e=t_s+d_{sr}/c$ where c is propagation speed of wireless signal in VANET.
if(t_r<t_e)
Packet p is not secure and discard it.
Packet p is secure
End

Figure 4: Flowchart of proposed approach

ii. Algorithm proposed, to stave off Wormhole Assault in VANET:

1. Compute the position p_s of transmitting node, the sending time t_s at which the packet is sent.
2. Compute $C=E_{Ks}(p_s,t_s,p)$ at sending node where p is a packet.
3. Send cipher text C to the receiving vehicle.
4. Decrypt the C by using $D_{Ks}(C)$, and obtain p_s , t_s and packet p at receiving site.
5. Compute the position p_r of receiving vehicle.
6. At receiving vehicle compute:

(d_{sr}) distance between sender and receiver as follow:

$d_{sr} \leq |p_s-p_r| +2v(t_r-t_s+\Delta) +\delta$.

in which

$\pm\Delta$=time drifting of sender and receiver

v=maximum speed of any vehicle

δ= the maximum error in location information

7. Compute the expected time (t_e) at accepting node as:

$t_e=t_s+d_{sr}/c$.

where:

c=wireless signal's propagation speed in VANET.

8. if($t_r<t_e$)

then packet is secured

else

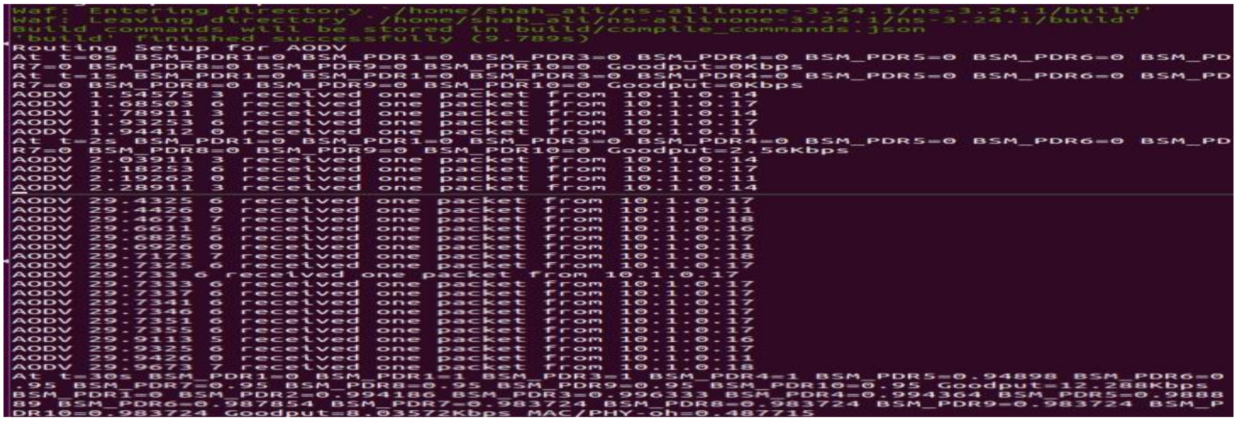
if($t_r \geq t_e$)

then packet is not secure and discard it.

Above algorithm is proposed in (Ali et al., 2017) research paper. But in (Ali et al., 2017) the key is distributed by using public key crypto system like RSA which requires more computations.

d. Generation of Dataset

A dataset is created from simulation to learn the assault. The screenshot of simulation is given below:



```
Waf: Entering directory `/home/shah_ali/ns-allinone-3.24.1/ns-3.24.1/build'
Waf: Leaving directory `/home/shah_ali/ns-allinone-3.24.1/ns-3.24.1/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (9.789s)
Routing Setup for AODV
At t=0s BSM_PDR1=0 BSM_PDR1=0 BSM_PDR3=0 BSM_PDR4=0 BSM_PDR5=0 BSM_PDR6=0 BSM_PD
R7=0 BSM_PDR8=0 BSM_PDR9=0 BSM_PDR10=0 Goodput=0Kbps
At t=1s BSM_PDR1=0 BSM_PDR1=0 BSM_PDR3=0 BSM_PDR4=0 BSM_PDR5=0 BSM_PDR6=0 BSM_PD
R7=0 BSM_PDR8=0 BSM_PDR9=0 BSM_PDR10=0 Goodput=0Kbps
AODV 1.54575 3 received one packet from 10.1.0.14
AODV 1.68503 6 received one packet from 10.1.0.17
AODV 1.78911 3 received one packet from 10.1.0.14
AODV 1.93253 6 received one packet from 10.1.0.17
AODV 1.94412 0 received one packet from 10.1.0.11
At t=2s BSM_PDR1=0 BSM_PDR1=0 BSM_PDR3=0 BSM_PDR4=0 BSM_PDR5=0 BSM_PDR6=0 BSM_PD
R7=0 BSM_PDR8=0 BSM_PDR9=0 BSM_PDR10=0 Goodput=2.56Kbps
AODV 2.03911 3 received one packet from 10.1.0.14
AODV 2.18253 6 received one packet from 10.1.0.17
AODV 2.19262 0 received one packet from 10.1.0.11
AODV 2.28911 3 received one packet from 10.1.0.14
AODV 29.4325 6 received one packet from 10.1.0.17
AODV 29.4426 0 received one packet from 10.1.0.11
AODV 29.4673 7 received one packet from 10.1.0.18
AODV 29.6611 5 received one packet from 10.1.0.16
AODV 29.6825 6 received one packet from 10.1.0.17
AODV 29.6926 0 received one packet from 10.1.0.11
AODV 29.7173 7 received one packet from 10.1.0.18
AODV 29.7325 6 received one packet from 10.1.0.17
AODV 29.733 6 received one packet from 10.1.0.17
AODV 29.7333 6 received one packet from 10.1.0.17
AODV 29.7337 6 received one packet from 10.1.0.17
AODV 29.7341 6 received one packet from 10.1.0.17
AODV 29.7346 6 received one packet from 10.1.0.17
AODV 29.7351 6 received one packet from 10.1.0.17
AODV 29.7355 6 received one packet from 10.1.0.17
AODV 29.9113 5 received one packet from 10.1.0.16
AODV 29.9325 6 received one packet from 10.1.0.17
AODV 29.9426 0 received one packet from 10.1.0.11
AODV 29.9673 7 received one packet from 10.1.0.18
At t=30s BSM_PDR1=0 BSM_PDR1=1 BSM_PDR3=1 BSM_PDR4=1 BSM_PDR5=0.94898 BSM_PDR6=0
.95 BSM_PDR7=0.95 BSM_PDR8=0.95 BSM_PDR9=0.95 BSM_PDR10=0.95 Goodput=12.288Kbps
BSM_PDR1=0 BSM_PDR2=0.994186 BSM_PDR3=0.996333 BSM_PDR4=0.994364 BSM_PDR5=0.9888
89 BSM_PDR6=0.987854 BSM_PDR7=0.983724 BSM_PDR8=0.983724 BSM_PDR9=0.983724 BSM_P
DR10=0.983724 Goodput=8.03572Kbps MAC/PHY-oh=0.487715
```

Figure 5. Simulation of Scenario

Animation of scenario on NetAnim is given as follow:

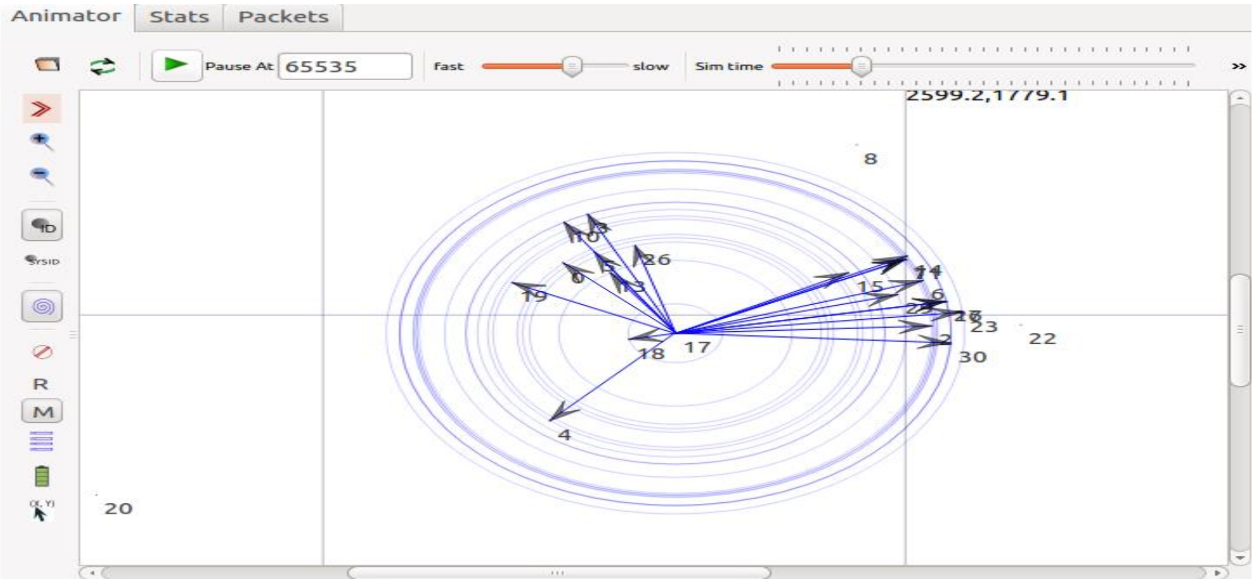


Figure 6: Animated view of scenario

Various features which are given by flow monitor tool of NS-3.24.1 simulator are source and destination IP, timeFirstTxPacket, timeFirstRxPacket, timeLastTxPacket, timeLastRxPacket, delaySum, jitterSum, lastDelay, txBytes, rxBytes, txPackets, rxPackets, and lostPackets etc. But in this research work txPackets(Transmitted Packets), rxPackets(Received Packets), lostPackets(lost Packets) and jitterSum(Jitter Sum) features are considered to create the dataset. The AODV routing protocol is modified to create the scenario for wormhole assault in NS-3.24.1for VANET.

e. Normalization of Data Set

The dataset which is created from data obtained from flow monitor of animator of NS-3.24.1 simulator is not in normalized form. To apply the ML model the normalized dataset is required. The function in python to normalize the dataset is given as follow:

normalized_df=(df-df.min())/(df.max()-df.min())

After normalization, the dataset is fetched in the form of excel sheet. The function in python to fetch the dataset is given as follow:

normalized_df.to_excel(writer, sheet_name='Sheet1')

Simulation

a. Simulation Tools

(SUMO-0.32.0) Simulation of Urban Mobility Model (Khan et al., 2019) and (NS-3.24.1) Network Simulator (R.Henderson et al., 2008) are used as simulators in this research work. We have taken Noida (Near Sector 93A Park) (U.P.) India map with the help of OpenStreetMap (OSM) to generate various files.

The VANET scenario generated by the SUMO-03.2.0 with the help of Real Map is given below:

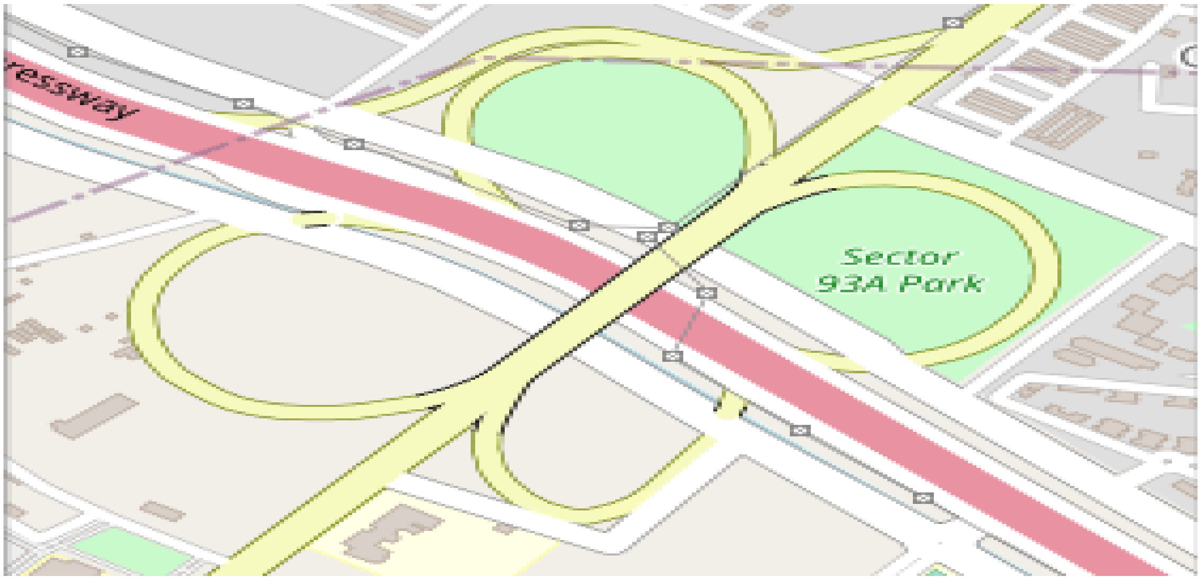


Figure 7. Scenario with the help of Real Map

The steps used to generate various files are given in the following figure:

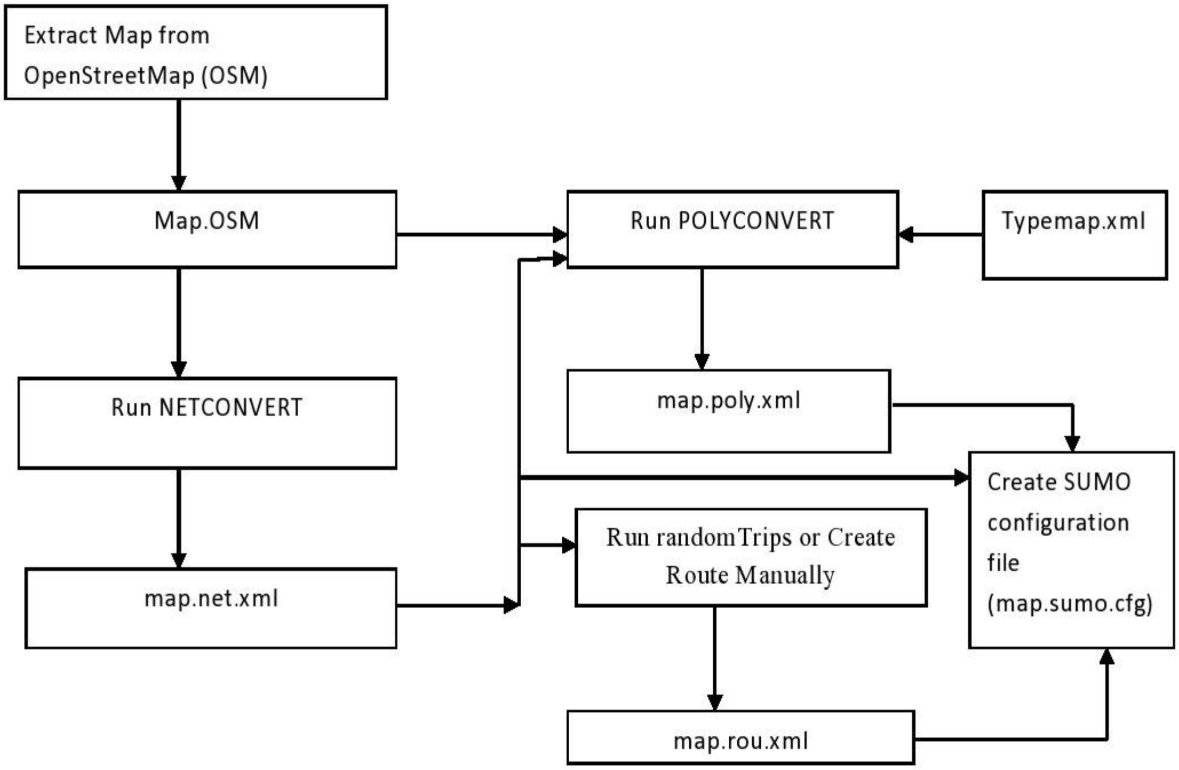

Extract Map from OpenStreetMap (OSM)
Map.OSM
Run POLYCONVERT
Typemap.xml
map.poly.xml
Run NETCONVERT
Create SUMO configuration file (map.sumo.cfg)
Run randomTrips or Create Route Manually
map.net.xml
map.rou.xml

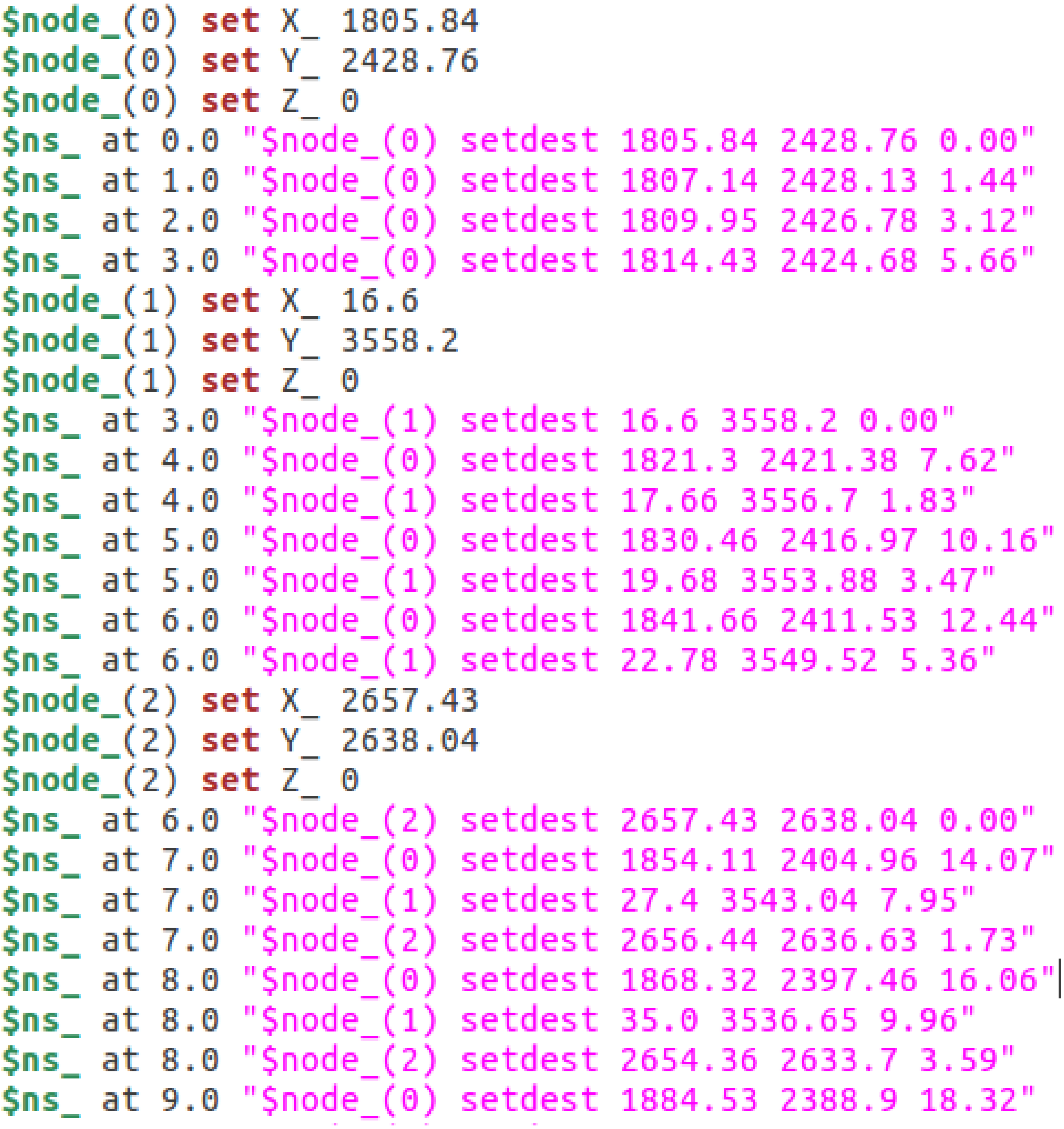
Figure 8. Steps to generate various files

Figure 8 shows the steps by which SUMO.cfg file is created with the help of OSM file (Singh et al., 2018). The mobility.tcl file which is created by the SUMO-0.32.0 simulator is used in the NS-3.24.1 simulator to provide the mobility to the VANET scenario.

b. Configuration of Simulation

Various steps which are followed to create scenario are shown as follow:

i. Generating the tcl file for the mobility

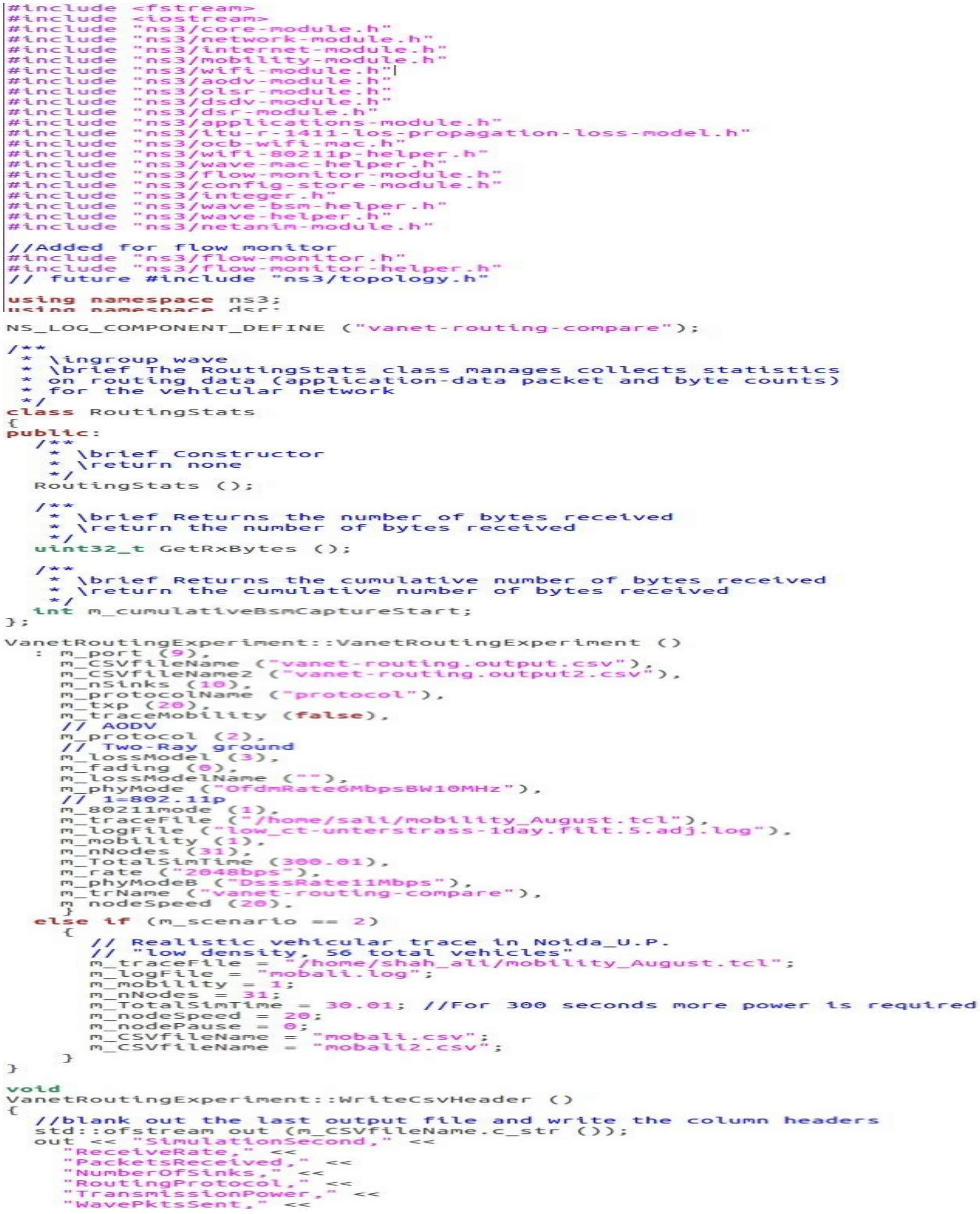


```
$node_(0) set X_ 1805.84
$node_(0) set Y_ 2428.76
$node_(0) set Z_ 0
$ns_ at 0.0 "$node_(0) setdest 1805.84 2428.76 0.00"
$ns_ at 1.0 "$node_(0) setdest 1807.14 2428.13 1.44"
$ns_ at 2.0 "$node_(0) setdest 1809.95 2426.78 3.12"
$ns_ at 3.0 "$node_(0) setdest 1814.43 2424.68 5.66"
$node_(1) set X_ 16.6
$node_(1) set Y_ 3558.2
$node_(1) set Z_ 0
$ns_ at 3.0 "$node_(1) setdest 16.6 3558.2 0.00"
$ns_ at 4.0 "$node_(0) setdest 1821.3 2421.38 7.62"
$ns_ at 4.0 "$node_(1) setdest 17.66 3556.7 1.83"
$ns_ at 5.0 "$node_(0) setdest 1830.46 2416.97 10.16"
$ns_ at 5.0 "$node_(1) setdest 19.68 3553.88 3.47"
$ns_ at 6.0 "$node_(0) setdest 1841.66 2411.53 12.44"
$ns_ at 6.0 "$node_(1) setdest 22.78 3549.52 5.36"
$node_(2) set X_ 2657.43
$node_(2) set Y_ 2638.04
$node_(2) set Z_ 0
$ns_ at 6.0 "$node_(2) setdest 2657.43 2638.04 0.00"
$ns_ at 7.0 "$node_(0) setdest 1854.11 2404.96 14.07"
$ns_ at 7.0 "$node_(1) setdest 27.4 3543.04 7.95"
$ns_ at 7.0 "$node_(2) setdest 2656.44 2636.63 1.73"
$ns_ at 8.0 "$node_(0) setdest 1868.32 2397.46 16.06"
$ns_ at 8.0 "$node_(1) setdest 35.0 3536.65 9.96"
$ns_ at 8.0 "$node_(2) setdest 2654.36 2633.7 3.59"
$ns_ at 9.0 "$node_(0) setdest 1884.53 2388.9 18.32"
```

Figure 9. The snapshot of mobility.tcl file is shown as follow:

ii. Generating the VANET scenario file

The snapshot of VANET scenario file is shown as follow:



```
#include <fstream>
#include <iostream>
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/mobility-module.h"
#include "ns3/wifi-module.h"
#include "ns3/aodv-module.h"
#include "ns3/olsr-module.h"
#include "ns3/dsdv-module.h"
#include "ns3/dsr-module.h"
#include "ns3/applications-module.h"
#include "ns3/itu-r-1411-los-propagation-loss-model.h"
#include "ns3/ocb-wifi-mac.h"
#include "ns3/wifi-80211p-helper.h"
#include "ns3/wave-mac-helper.h"
#include "ns3/flow-monitor-module.h"
#include "ns3/config-store-module.h"
#include "ns3/integer.h"
#include "ns3/wave-bsm-helper.h"
#include "ns3/wave-helper.h"
#include "ns3/netanim-module.h"

//Added for flow monitor
#include "ns3/flow-monitor.h"
#include "ns3/flow-monitor-helper.h"
// future #include "ns3/topology.h"

using namespace ns3;
using namespace dsr;

NS_LOG_COMPONENT_DEFINE ("vanet-routing-compare");

/**
 * \ingroup wave
 * \brief The RoutingStats class manages collects statistics
 * on routing data (application-data packet and byte counts)
 * for the vehicular network
 */
class RoutingStats
{
public:
  /**
   * \brief Constructor
   * \return none
   */
  RoutingStats ();

  /**
   * \brief Returns the number of bytes received
   * \return the number of bytes received
   */
  uint32_t GetRxBytes ();

  /**
   * \brief Returns the cumulative number of bytes received
   * \return the cumulative number of bytes received
   */
  int m_cumulativeBsmCaptureStart;
};

VanetRoutingExperiment::VanetRoutingExperiment ()
  : m_port (9),
    m_CSVfileName ("vanet-routing.output.csv"),
    m_CSVfileName2 ("vanet-routing.output2.csv"),
    m_nSinks (10),
    m_protocolName ("protocol"),
    m_txp (20),
    m_traceMobility (false),
    // AODV
    m_protocol (2),
    // Two-Ray ground
    m_lossModel (3),
    m_fading (0),
    m_lossModelName (""),
    m_phyMode ("OfdmRate6MbpsBW10MHz"),
    // 1=802.11p
    m_80211mode (1),
    m_traceFile ("/home/sali/mobility_August.tcl"),
    m_logFile ("low_ct-unterstrass-1day.filt.5.adj.log"),
    m_mobility (1),
    m_nNodes (31),
    m_TotalSimTime (300.01),
    m_rate ("2048bps"),
    m_phyModeB ("DsssRate11Mbps"),
    m_trName ("vanet-routing-compare"),
    m_nodeSpeed (20),
    }
  else if (m_scenario == 2)
    {
      // Realistic vehicular trace in Noida_U.P.
      // "low density, 56 total vehicles"
      m_traceFile = "/home/shah_ali/mobility_August.tcl";
      m_logFile = "mobali.log";
      m_mobility = 1;
      m_nNodes = 31;
      m_TotalSimTime = 30.01; //For 300 seconds more power is required
      m_nodeSpeed = 20;
      m_nodePause = 0;
      m_CSVfileName = "mobali.csv";
      m_CSVfileName = "mobali2.csv";
    }
}

void
VanetRoutingExperiment::WriteCsvHeader ()
{
  //blank out the last output file and write the column headers
  std::ofstream out (m_CSVfileName.c_str ());
  out << "SimulationSecond," <<
    "ReceiveRate," <<
    "PacketsReceived," <<
    "NumberOfSinks," <<
    "RoutingProtocol," <<
    "TransmissionPower," <<
    "WavePktsSent," <<
```

Figure 10. Screenshot of VANET Scenario file

c. Simulation Parameters

Various parameters used in simulation are given as follow:

Table1. Parameters & Values used in Simulation

Parameter	**Value**
Simulator	NS-3.24.1
Time of Simulation	30 s
Vehicles Strength	31
Type of Traffic	(CBR)Constant Bit Rate
Territory Dimension	5198.32m ∗ 3558.2 m
Transport Protocol	512 Bytes
Transport Protocol	UDP
Protocol for Routing	AODV
MAC Protocol	IEEE 802.11p
Radio Propagation Model	Two ray ground
Maximum Speed	20 m/s
Mobility Model	Noida (Near Sector 93A Park) Model

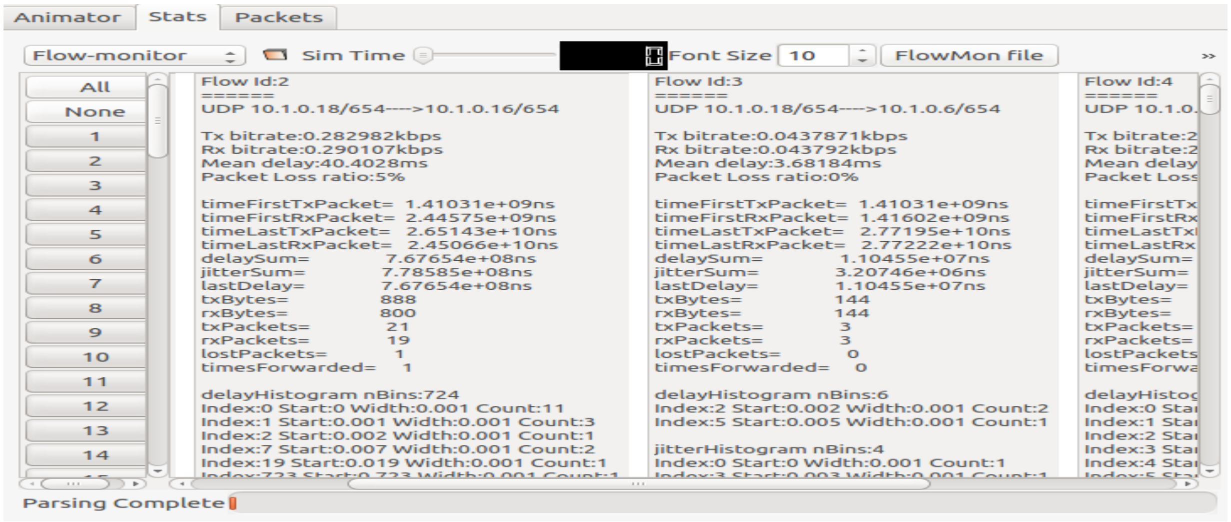
For generating the dataset, simulation is carried out multiple times with a distinctive couple of vehicles as assaultive vehicles. We mark the overall transmission taking place between the pair of assaultive vehicles as abnormal and the remaining as normal.

Table 2. Attributes of Dataset

Variable	**Value**
Count of Features	4
Categories	2 (For Normal 0, for Abnormal 1)
Total Cases	249
0-Tag	212
1-Tag	37
Size of Training	174
Size of Testing	75

Result and Discussion

Discussion regarding the results which are achieved after the simulation, training and testing a machine from created data set is carried out, in this section. The data set is created by collecting the required data values from the flow monitor of animator. The screenshot of flow monitor flow id and normalized dataset is given as follow:



```
Flow Id:2
======
UDP 10.1.0.18/654---->10.1.0.16/654

Tx bitrate:0.282982kbps
Rx bitrate:0.290107kbps
Mean delay:40.4028ms
Packet Loss ratio:5%

timeFirstTxPacket=  1.41031e+09ns
timeFirstRxPacket=  2.44575e+09ns
timeLastTxPacket=   2.65143e+10ns
timeLastRxPacket=   2.45066e+10ns
delaySum=           7.67654e+08ns
jitterSum=          7.78585e+08ns
lastDelay=          7.67654e+08ns
txBytes=            888
rxBytes=            800
txPackets=           21
rxPackets=           19
lostPackets=          1
timesForwarded=       1

delayHistogram nBins:724
Index:0 Start:0 Width:0.001 Count:11
Index:1 Start:0.001 Width:0.001 Count:3
Index:2 Start:0.002 Width:0.001 Count:1
Index:7 Start:0.007 Width:0.001 Count:2
Index:19 Start:0.019 Width:0.001 Count:1
Index:723 Start:0.723 Width:0.001 Count:1
```

```
Flow Id:3
======
UDP 10.1.0.18/654---->10.1.0.6/654

Tx bitrate:0.0437871kbps
Rx bitrate:0.043792kbps
Mean delay:3.68184ms
Packet Loss ratio:0%

timeFirstTxPacket=  1.41031e+09ns
timeFirstRxPacket=  1.41602e+09ns
timeLastTxPacket=   2.77195e+10ns
timeLastRxPacket=   2.77222e+10ns
delaySum=           1.10455e+07ns
jitterSum=          3.20746e+06ns
lastDelay=          1.10455e+07ns
txBytes=            144
rxBytes=            144
txPackets=           3
rxPackets=           3
lostPackets=          0
timesForwarded=       0

delayHistogram nBins:6
Index:2 Start:0.002 Width:0.001 Count:2
Index:5 Start:0.005 Width:0.001 Count:1

jitterHistogram nBins:4
Index:0 Start:0 Width:0.001 Count:1
Index:3 Start:0.003 Width:0.001 Count:1
```

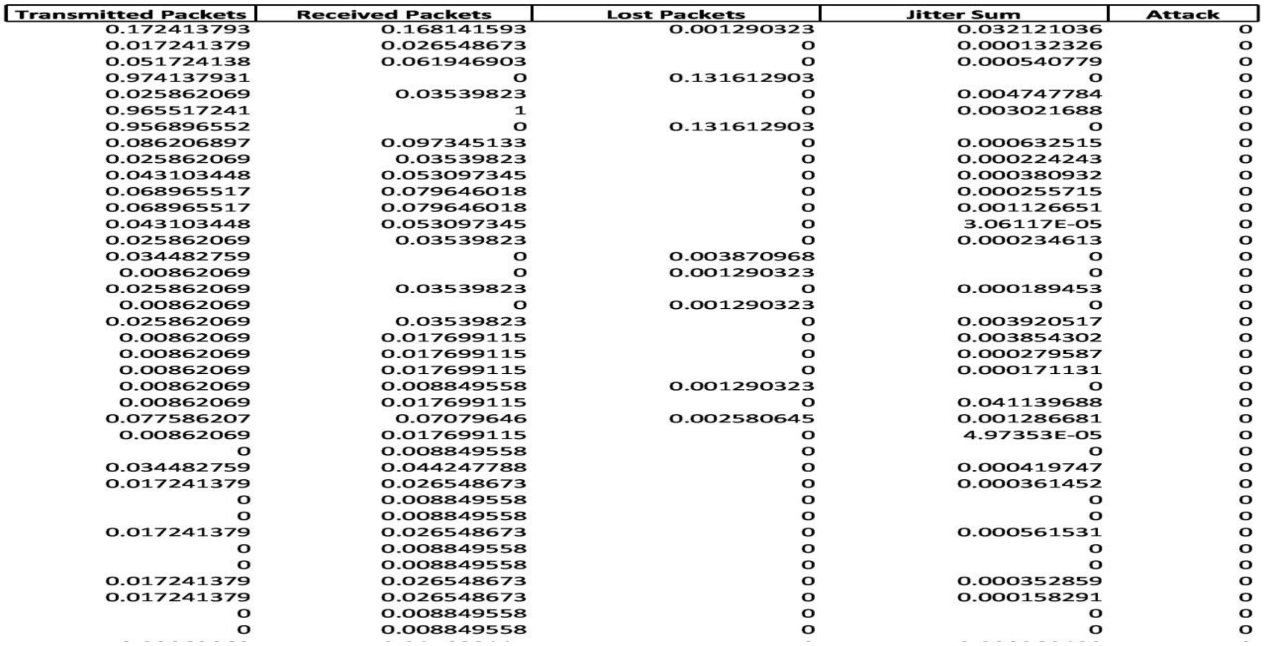
```
Flow Id:4
======
UDP 10.1.0.

Tx bitrate:2
Rx bitrate:2
Mean delay
Packet Loss

timeFirstTx
timeFirstRx
timeLastTx
timeLastRx
delaySum=
jitterSum=
lastDelay=
txBytes=
rxBytes=
txPackets=
rxPackets=
lostPackets
timesForwa

delayHistog
Index:0 Sta
Index:1 Sta
Index:2 Sta
Index:3 Sta
Index:4 Sta
Index:5 Sta
```

Figure 11(a). Screenshot of flow monitor with flow Ids



Transmitted Packets	Received Packets	Lost Packets	Jitter Sum	Attack
0.172413793	0.168141593	0.001290323	0.032121036	0
0.017241379	0.026548673	0	0.000132326	0
0.051724138	0.061946903	0	0.000540779	0
0.974137931	0	0.131612903	0	0
0.025862069	0.03539823	0	0.004747784	0
0.965517241	1	0	0.003021688	0
0.956896552	0	0.131612903	0	0
0.086206897	0.097345133	0	0.000632515	0
0.025862069	0.03539823	0	0.000224243	0
0.043103448	0.053097345	0	0.000380932	0
0.068965517	0.079646018	0	0.000255715	0
0.068965517	0.079646018	0	0.001126651	0
0.043103448	0.053097345	0	3.06117E-05	0
0.025862069	0.03539823	0	0.000234613	0
0.034482759	0	0.003870968	0	0
0.00862069	0	0.001290323	0	0
0.025862069	0.03539823	0	0.000189453	0
0.00862069	0	0.001290323	0	0
0.025862069	0.03539823	0	0.003920517	0
0.00862069	0.017699115	0	0.003854302	0
0.00862069	0.017699115	0	0.000279587	0
0.00862069	0.017699115	0	0.000171131	0
0.00862069	0.008849558	0.001290323	0	0
0.00862069	0.017699115	0	0.041139688	0
0.077586207	0.07079646	0.002580645	0.001286681	0
0.00862069	0.017699115	0	4.97353E-05	0
0	0.008849558	0	0	0
0.034482759	0.044247788	0	0.000419747	0
0.017241379	0.026548673	0	0.000361452	0
0	0.008849558	0	0	0
0	0.008849558	0	0	0
0.017241379	0.026548673	0	0.000561531	0
0	0.008849558	0	0	0
0	0.008849558	0	0	0
0.017241379	0.026548673	0	0.000352859	0
0.017241379	0.026548673	0	0.000158291	0
0	0.008849558	0	0	0
0	0.008849558	0	0	0

Figure 11(b). Screenshot of normalized dataset

The dataset is splitted into 70% training and 30% testing dataset. In python following syntax is used, first:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

After importing the data set, the partitioning is carried out by using the following code:

```
From sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test=train_test_split(X, y,test_size=30)
```

The graph of training dataset and testing dataset is given as follow by using python programming on jupyter Notebook.

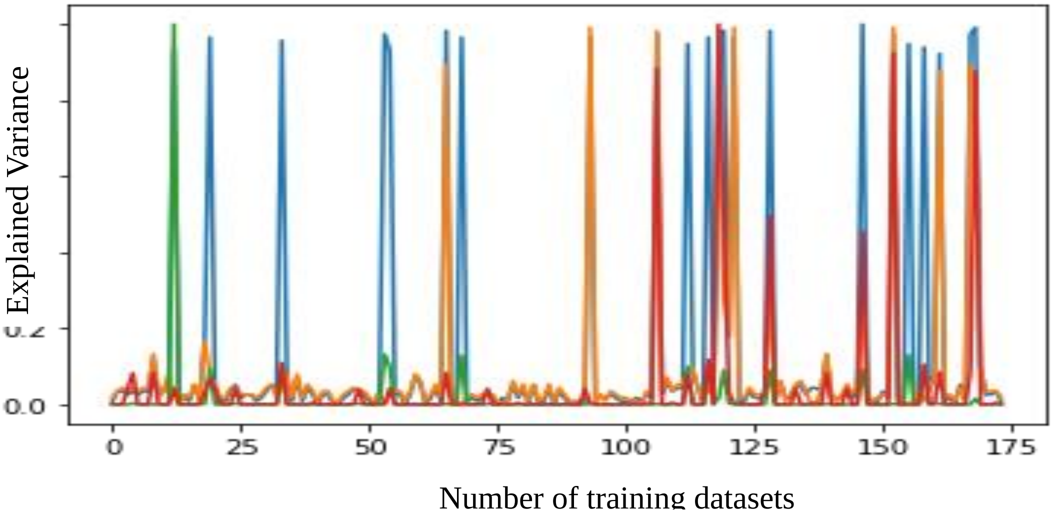


Figure 12(a). Graph for training dataset for various features.

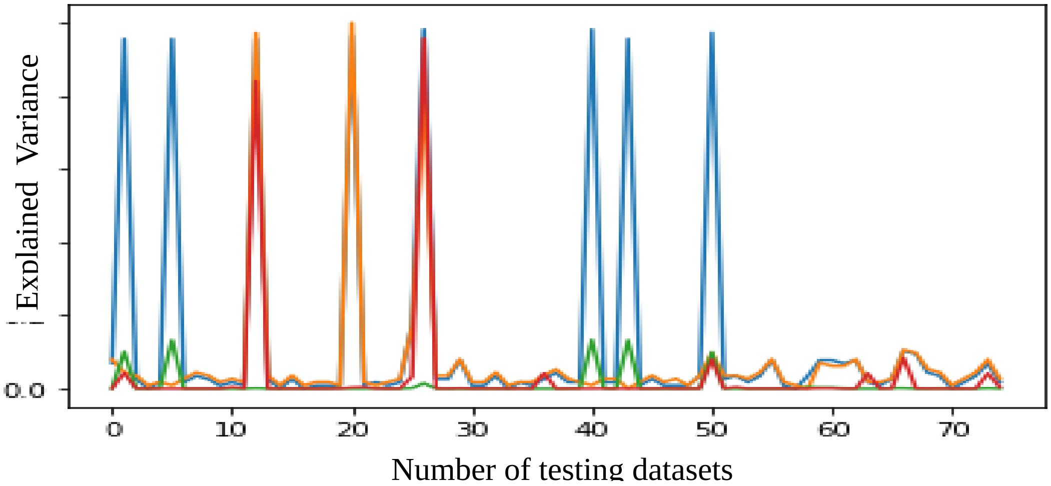
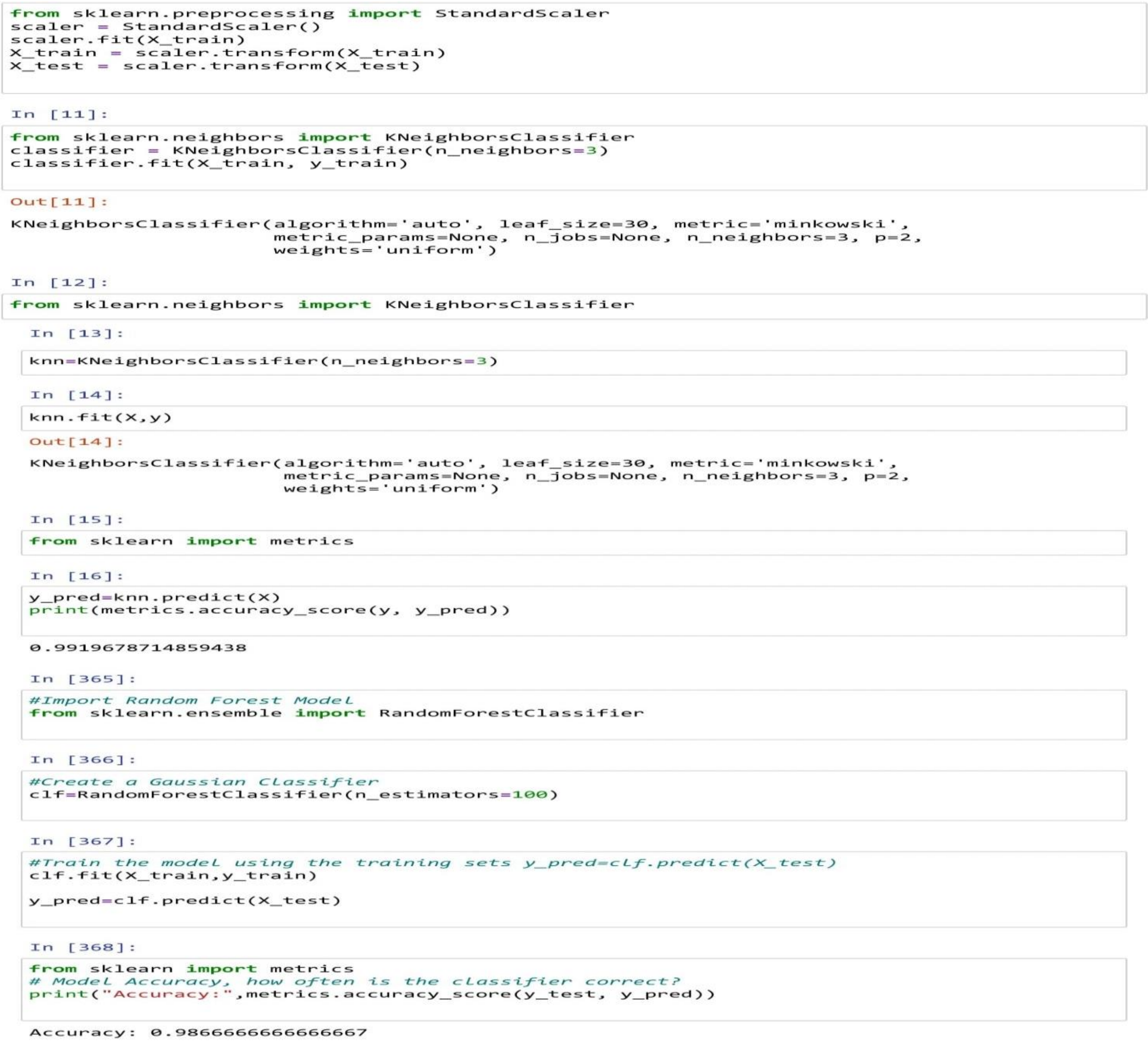


Figure 12(b). Graph of testing dataset for various features.

In this research work the value 3 is assigned to k in k-NN. The results show that the k-NN having the accuracy of 99.196% and random forest is having the accuracy 98.666% respectively. The screenshot of execution is given as follow:



```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

In [11]:
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=3)
classifier.fit(X_train, y_train)

Out[11]:
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='uniform')

In [12]:
from sklearn.neighbors import KNeighborsClassifier

In [13]:
knn=KNeighborsClassifier(n_neighbors=3)

In [14]:
knn.fit(X,y)
Out[14]:
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='uniform')

In [15]:
from sklearn import metrics

In [16]:
y_pred=knn.predict(X)
print(metrics.accuracy_score(y, y_pred))

0.9919678714859438

In [365]:
#Import Random Forest Model
from sklearn.ensemble import RandomForestClassifier

In [366]:
#Create a Gaussian Classifier
clf=RandomForestClassifier(n_estimators=100)

In [367]:
#Train the model using the training sets y_pred=clf.predict(X_test)
clf.fit(X_train,y_train)

y_pred=clf.predict(X_test)

In [368]:
from sklearn import metrics
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

Accuracy: 0.9866666666666667
```

Figure 13. Screenshot of results computation

In existing research paper (Singh et al., 2019) the k-NN model is having the accuracy 99% to detect the wormhole attack, but in our research work the k-NN model is having the accuracy 99.196% to detect the wormhole attack in VANET due to proper normalization of data. In existing research random forest ML model is not used to detect the wormhole attack in VANET. In this research work the random forest ML model is also used due to its various advantages as compare to other ML models as discussed previously. The random forest ML model also performed well with a detection accuracy of 98.666%.

And by analytically it is proved that the scheme based on packet leash and cryptographic technique is useful to prevent the wormhole attack in vehicular ad-hoc network.

Conclusion

In this research work an approach based on ML concept is proposed to detect the wormhole assault in VANET. By using methodology & results it has been demonstrated that how efficient ML is in finding wormhole assault in VANET. The depicted model finds the assault with accuracy 99.196% and 98.666%. ML can be utilized at various layers of protocol stack of VANET to detect the wormhole assault. An approach based on packet leash and cryptographic concept, is also proposed to stave off the wormhole assault in vehicular ad-hoc network.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article

References

Albouq, S.S., & Fredericks, E.M. (2017). Detection and avoidance of wormhole attacks in connected vehicles. *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*. https://doi.org/10.1145/3132340.3132346

Ali Alheeti, K.M., Gruebler, A., & McDonald-Maier, K. (2016). Intelligent intrusion detection of grey hole and rushing attacks in self-driving vehicular networks. *Computers, 5(3), 16*. https://doi.org/10.3390/computers5030016

Ali, S., Nand, P. & Tiwari, S. (2017). Secure message broadcasting in VANET over wormhole attack by using cryptographic technique. *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 520-523. doi: 10.1109/CCAA.2017.8229856.

Ali, S., Nand, P., & Tiwari, S. (2020). Impact of wormhole attack on AODV routing protocol in vehicular ad-hoc network over real map with detection and prevention approach. *Int. J. Vehicle Information and Communication Systems*, 5(3), 354–373. doi: 10.1504/IJVICS.2020.110997

Altman, N.S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. Journal of the American Statistical Association published by Taylor & Francis Ltd., 46(3), 175–185. https://doi.org/10.2307/2685209

Argyroudis, P.G., & O'mahony, D. (2005). Secure routing for mobile ad hoc networks. *IEEE Commun. Surv.Tutor*, 7(3), 1-21.

Bakhouya, M., Gaber, J., & Lorenz, P. (2011). An adaptive approach for information dissemination in vehicular ad hoc networks. *Journal of Network and Computer Applications*, 34(6), 1971–1978. https://doi.org/10.1016/j.jnca.2011.06.010

Bellare, M., Canetti, R., & Krawczyk, H. (1996). Keying hash functions for message authentication. *In Advances in Cryptology – CRYPTO '96* edited by Neal Koblitz, volume 1109 of Lecture Notes in Computer Science, 1–15. Springer-Verlag, Berlin Germany, 1996. https://doi.org/10.1007/3-540-68697-5_1

Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., & Pinkas, B.(1999). Multicast security: a taxonomy and some efficient constructions. *In Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 2, 708-716 doi: 10.1109/INFCOM.1999.751457.

Chahal, M., Harit, S., Mishra, K., Sangaiah, A., & Zheng, Z. (2017). A Survey on software-defined networking in vehicular ad hoc networks: Challenges, applications and use cases. *International Journal of Sustainable Cities and Society*, 35, 830–840. https://doi.org/10.1016/j.scs.2017.07.007

Chung, T., & Roedig, U. (2007) 'Poster: efficient key establishment for wireless sensor networks using elliptic curve Diffie-Hellman', *September 2008 IEEE 5th International Conference on Mobile Adhoc and Sensor Systems, MASS 2008*, 29 September - 2 October 2008, Atlanta, Georgia, USA, DOI: 10.1109/MAHSS.2008.4660127.

Grover, J., Prajapati, N.K., Laxmi, V., & Gaur, M.S. (2011). Machine learning approach for multiple misbehavior detection in VANET. *International Conference on Advances in Computing and Communications*, 644–653. https://doi.org/10.1007/978-3-642-22720-2_68

Hu, Y.C., Perrig, A., & Johnson, D.B. (2003). Packet leashes: A defense against wormhole attacks in wireless networks. *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, 3, 1976–1986. doi: 10.1109/INFCOM.2003.1209219

IEEE Standard for Information technology-Local and metropolitan area networks- Specific requirements-

- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments. *In IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)* , 1-51, 15 July 2010. doi: 10.1109/IEEESTD.2010.5514475.

Kang, M.J., & Kang, J.W. (2016). Intrusion detection system using deep neural network for in-vehicle network security. *PLOS ONE* 11(6), e0155781(2016). https://doi.org/10.1371/journal.pone.0155781

Khalil, I., Bagchi, S., Shroff , N.B. (2007). LiteWorp: Detection and isolation of the worm-hole attack in static multi hop wireless networks. *Comput.Netw.,51(13*), 3750–3772. https://doi.org/10.1016/j.comnet.2007.04.001

Khan, T., & Singh, K. (2021). TASRP: a trust aware secure routing protocol for wireless sensor networks. *International Journal of Innovative Computing and Applications*, 12(2), 108-122. doi:10.1504/IJICA.2021.113750

Khan, T., Singh, K., Le, H. S., Mohamed Abdel-Basset, Hoang, V. L., Singh, S. P., & Manjul, M. (2019). A novel and comprehensive trust estimation clustering based approach for large scale wireless sensor networks. *IEEE Access* 7 (2019):58221-58240. doi: 10.1109/ACCESS.2019.2914769.

Kumar, S., & Singh, R.K. (2016). Secure authentication approach using Diffie-Hellman key exchange algorithm for WSN. *Int. J. Communication Networks and Distributed Systems*, 17(2), 189–201. https://doi.org/10.1504/IJCNDS.2016.079102

Loukas, G., Vuong, T., Heartfield, R., Sakellari, G., Yoon, Y., & Gan, D.(2018). Cloud- based cyber-physical intrusion detection for vehicles using Deep Learning. *In IEEE Access, vol. 6*, 3491-3508. doi: 10.1109/ACCESS.2017.2782159.

Perkins, C.E., & Royer, E.M. (1999). Ad-Hoc on-Demand Distance Vector Routing. *Proc. of IEEE Workshop Mobile Computing Systems and Applications*, 90-100. doi: 10.1109/MCSA.1999.749281

Provost, F., Hibert, C., Malet, J.-P., Stumpf, A., & Doubre, C. (2016). Automatic classification of endogenous seismic sources within a landslide body using random forest algorithm. *EGU General Assembly 2016*, held 17-22 April, 2016 in Vienna Austria, id. EPSC2016-15705.

R.Henderson, T., Lacage, M., Riley, G.F., Dowell, C., & Kopena, J. (2008). Network simulations with the ns-3simulator. *SIGCOMMDemonstr.*14(14),527, ACM 978-1-60558-175-0/08/08.

Rawat, G., & Singh, K. (2020). Joint beacon frequency and beacon transmission power adaptation for internet of vehicles. *Transactions on Emerging Telecommunications Technologies*, e4124, https://doi.org/10.1002/ett.4124

Safi, S., Movaghar, A., & Mohammadizadeh, M. (2009). A Novel approach for avoiding Wormhole Attacks in VANET. *Second International Workshop on Computer Science and Engineering*, 160- 165. doi: 10.1109/WCSE.2009.787

Sanzgiri, K., Dahill, B., Levine, B.N., Shields, C., & Belding-Royer, E.M. (2002). A secure routing protocol for ad hoc networks. *10th IEEE International Conference on Network Protocols, 2002*. *Proceedings,* 78-87. doi: 10.1109/ICNP.2002.1181388

Singh, P.K., Dash, M.K., Mittal, P., Nandi, S.K., & Nandi, S.(2018). Misbehavior detection in C-ITS using deep learning approach. *In: 18th International Conference on Intelligent Systems Design and Applications (ISDA),Springer(2018),* 641-652. doi:10.1007/978-3-030-16657-1_60

Singh, P.K., Gupta, R.R., Nandi, S.K., & Nandi, S. (2019). Machine learning based approach to detect wormhole attack in VANETs. *Web, Artificial Intelligence and Network Applications, Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019)*. https://doi.org/10.1007/978-3-030-15035-8_63

Singh, P.K., Sharma, S., Nandi, S.K., & Nandi, S. (2018). Multipath TCP for V2I communication in SDN controlled small cell deployment of smart city. *Veh. Commun.* (2018), https://doi.org/10.1016/j.vehcom.2018.11.002.

Stalings, W. (2005). Cryptography and Network Security, Principles and Practices. *Prentice Hall* 2005, 268-270 & 296-297.

Taylor, A., Leblanc, S., & Japkowicz, N. (2016). Anomaly detection in automobile control network data with long short-term memory networks. *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 130-139. doi: 10.1109/DSAA.2016.20

Upadhyaya, A., & Shah, J. (2018). Attacks on VANET Security. *International Journal of Computer Engineering & Technology (IJCET)*, 9(1), 8–19.

Bibliographic information of this paper for citing:

Ali, Shahjahan; Nand, Parma& Tiwari,Shailesh (2022).Detection of Wormhole Attack in Vehicular Ad-hoc Network over Real Map using Machine Learning Approach with Preventive Scheme. *Journal of Information Technology Management,* Special Issue, 159-179.