A game theory based multi layered intrusion detection framework for VANET

Basant Subba, Santosh Biswas, Sushanta Karmakar

PII: S0167-739X(17)31448-6
DOI: https://doi.org/10.1016/j.future.2017.12.008
Reference: FUTURE 3846

To appear in: *Future Generation Computer Systems*

Received date : 30 June 2017
Revised date : 1 December 2017
Accepted date : 3 December 2017

A game theory based multi layered intrusion detection framework for VANET

Basant Subba, Santosh Biswas, Sushanta Karmakar
s.basant@iitg.ernet.in, santosh_biswas@iitg.ernet.in, sushantak@iitg.ernet.in

Indian Institute of Technology, Guwahati
India, Assam, 781039

Santosh Biswas[a], IIT Guwahati*

[a]*Indian Institute of Technology, Guwahati*

Abstract

Vehicular Ad-hoc Networks (VANETs) are vulnerable to various type of network attacks like Blackhole attack, Denial of Service (DoS), Sybil attack etc. Intrusion Detection Systems (IDSs) have been proposed in the literature to address these security threats. However, high vehicular mobility makes the process of formulating an IDS framework for VANET a difficult task. Moreover, VANETs operate in bandwidth constrained wireless radio spectrum. Therefore, IDS frameworks that introduce significant volume of IDS traffic are not suitable for VANETs. In addition, dynamic network topology, communication overhead and scalability to higher vehicular density are some other issues that needs to be addressed while developing an IDS framework for VANETs. This paper aims to address these issues by proposing a multi-layered game theory based intrusion detection framework and a novel clustering algorithm for VANET. The communication overhead of the IDS is reduced by using a set of specification rules and a lightweight neural network based classifier module for detecting malicious vehicles. The volume of IDS traffic is minimized by modeling the interaction between the IDS and the malicious vehicle as a two player non-cooperative game and adopting a probabilistic IDS monitoring strategy based on the Nash Equilibrium of the game. Finally, the proposed clustering algorithm maintains the stability of the IDS framework, which ensures that the framework scales up well to networks with higher vehicular densities. Simulation results show that the proposed framework achieves high accuracy and detection rate across wide range of attacks, while at the same time minimizes the overall volume of intrusion detection related traffic introduced into the vehicular network.

Keywords: **Intrusion Detection System (IDS), Vehicular Ad-hoc Network (VANET), Game Theory**

1. Introduction

The concept of enabling vehicles with the capability to make transportation infrastructure more secure and efficient has received immense attention in recent years. This has lead to the emergence of Vehicular Ad-hoc Networks (VANETs), which are formed on the fly by a network of vehicles equipped with multiple sensors and On Board Units (OBUs). The OBUs enable vehicles to connect with Road Side Units (RSUs) through a wireless short-range direct communication link based on the IEEE 802.11p radio frequency channel. VANET uses various type of notification messages like Post Crash Notification (PCN), Road Hazard Condition Notification (RHCN), Stopped/Slow Vehicle Advisor (SVA) etc., to provide vehicular communication.

VANET uses 75 MHz of Dedicated Short Range Communications (DSRC) spectrum at 5.9 GHz to support IEEE 802.11p standard for communication among vehicles. DSRC provides a communication range of 300 to 1000 m with a data rate of more than 27 Mbps and supports a vehicular mobility as high as 200 Kmph [1]. The IEEE P1609 working group has proposed DSRC as IEEE 802.11p standard for Wireless Access in Vehicular Environment (WAVE) platform [2]. The DSRC based WAVE architecture supports two different protocol stacks namely, the WAVE Share Message Protocol (WSMP) and the traditional IPv6 protocol. Time sensitive and high priority communication are achieved using the WSMP, while the less demanding communication involving the UDP/TCP/IP data frames are achieved using the IPv6 protocol. As shown in the Fig. 1, the DSRC spectrum band is divided into seven channels of 10 MHz each [3]. Channel 178 is the Control Channel (CCH), which is used for transmission of emergency messages. The other six channels numbered 172, 174, 176, 180, 182 and 184 are Service Channels (SCHs), which are used for both both safety and non-safety applications. If the CCH channel is active, all vehicles are bound to stop their communication during CCH time frame to receive and transmit emergency messages on CCH channel.

VANETs use emergency broadcast messages for disseminating information about adverse road conditions and traffic accidents, which require communication between the member ve-

*Corresponding author
Email address: santosh_biswas@iitg.ernet.in (IIT Guwahati)
URL: http://www.iitg.ac.in/santosh_biswas/ (Santosh Biswas)

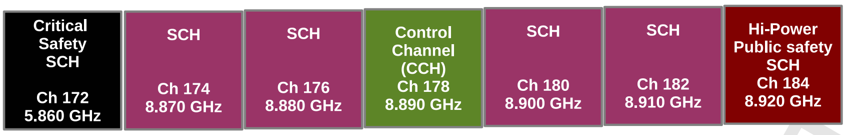


Figure 1: **Dedicated Short Range Communications (DSRC) spectrum with 7 channels of 10 MHz**

hicles through open wireless medium. The attacker can exploit this broadcast nature of VANET to carry out various type of attacks like eavesdropping, interference, jamming, masquerading, packet replay, Denial of Service (DoS), impersonation, identity disclosure etc. [4, 5, 6]. Preventive security measures like digital signature, authentication and encryption are usually employed as the first line of defense to prevent and detect such attacks in VANETs. However, these preventive measures have several limitations. The attacker can easily circumvent them by modifying the attack signatures to avoid detection. Moreover, an insider attacker with valid cryptographic keys used for secure communication can render the preventive security measures obsolete. Additionally, they use handshake based authentication mechanisms, which incur high communication overhead. All these factors make preventive security measures inadequate for providing a comprehensive protection against various type of attacks in VANETs.

To address the the drawbacks associated with the preventive security measures, an alternative security mechanism in the form of Intrusion Detection Systems (IDSs) have been proposed in the literature [7, 8, 9, 10, 11]. They complement the preventive security measures and act as a second line of defense against various type of attacks. IDSs have widely been deployed in wired networks with great results. However, unlike the wired networks with fixed infrastructure and static topology, VANETs are highly dynamic with intermittent network connectivity and constrained wireless bandwidth radio spectrum. All these issues make the task of formulating an efficient intrusion detection framework for VANET difficult and challenging. Therefore, any intrusion detection framework proposed for VANET must take the following key issues into consideration:

- *Bandwidth constraints and IDS traffic volume*: VANETs operate in a narrow bandwidth wireless radio spectrum. The entire bandwidth spectrum of the DSRC band (5.850 - 5.925 GHz) used for vehicular communication in VANET is only 75 MHz with a maximum theoretical throughput of 27 Mbps and a maximum transmission distance of 1000 m. Therefore, intrusion detection frameworks that introduce significant volume of IDS traffic and require pre stored information about the participating vehicles are not suitable for VANETs.

- *Dynamic network topology*: Network topologies in VANETs vary depending on the traffic density and vehicular mobility. This can cause high delays in dissemination of messages due to broadcast storm at high vehicular densities and disconnected network problems at low vehicular densities. Therefore, any intrusion detection framework proposed for VANET must adopt a suitable clustering algorithm for producing stable vehicular clusters to maintain the network's stability.

- *Communication overhead and scalability*: The association of a vehicle with other vehicles and RSUs in VANET is usually short lived and intermittent. Therefore IDS frameworks that require high communication overhead are not suitable for VANETs. In addition, VANETs consist of a network of hundreds of vehicles and are designed for supporting real time safety related applications, which require them to be up and running all the time. Therefore, IDS frameworks designed for VANETs must be scalable to vehicular networks with high vehicular densities.

A good trade-off must be maintained between gathering enough information for effectively detecting network intrusions and preventing the overburdening of IDS's logging component with high volume of IDS traffic in VANET. To achieve this trade-off, a novel **clustering algorithm**, a distributed **Cluster Head (CH) election algorithm** and a **game theory based multi layered intrusion detection framework** for VANET are proposed in this paper. In summary, the main contribution of this paper are as follows:

- We propose a distributed clustering algorithm that uses various vehicular information like velocities, reputation values, real time coordinates and direction of movement to generate stable vehicular clusters. Stable clusters enhance the robustness of the intrusion detection framework by providing vehicles enough time frame to exchange their data and thereby enabling them to make informed decisions.

- We propose a novel Cluster Head (CH) election algorithm that uses an incentive structure based on the Vickrey-Clarke-Groves (VCG) mechanism to motivate vehicles to actively participate in the CH election process by offering them payment in the form of reputation gain for taking up the role of the CH. Data packets of the reputed vehicles are given higher priority during data traffic routing.

- We propose a multi layered game theory based intrusion detection framework for VANET that uses a set of specification rules and a lightweight neural network based classifier module to detect various type of attacks in VANET.

- We model the interaction between the IDS and the malicious vehicle as a two player non-cooperative game, and adopt probabilistic monitoring strategies based on the NE of the game. Such game theoretic modeling minimizes the volume of IDS traffic in bandwidth constrained vehicular networks, without compromising the overall performance of the intrusion detection framework.

The rest of the paper has been organized in following way. Section 2 discusses related works on VANET intrusion detection frameworks and their drawbacks. Section 3 provides a detailed description of the proposed game theory based multi layered intrusion detection framework. Section 4 provides the experimental results and comparison analysis of the proposed framework with various other intrusion detection frameworks. Conclusion and future works are provided in Section 5.

2. Related Works

The work proposed in this paper primarily focuses on detection of insider attacks like blackhole attack, wormhole attack, selective forwarding etc., in VANETs. Therefore, we begin the related work section with description of various misbehaving detection mechanisms proposed in the literature for identifying insider attacks in various wireless networks like, Wireless Sensor Network (WSN), Mobile Ad-hoc Network (MANET) and VANET. We then state the drawbacks associated with these detection mechanisms, which provide motivations for the work carried out in this paper.

Various methods for augmenting the security level of the watchdog mechanism in the sensor network for preventing insider attacks is proposed in [12]. Similarly, security framework for detecting malicious nodes carrying out insider attacks through generation of malicious reports is proposed in [13]. On the other hand, effective intrusion detection mechanisms for detecting insider attacks in Mobile Ad-hoc Network (MANET) are proposed in [14, 15]. Although, these security mechanisms provide comprehensive protection against insider attacks in WSNs and MANETs, they cannot be applied to VANETs due to differences in the underlying network characteristics of VANETs with that of WSNs and MANETs.

A bandwidth-efficient protection mechanism for detecting insider attacks on *in-network* aggregation protocols in VANET is proposed in [16]. It uses combination of various data mining techniques to detect false information in the vehicular network and a filtering technique to limit the influence of attackers on the aggregated data. However, the drawback of this mechanism is the additional overhead required for maintaining the path list, which contains a fixed number of signatures. A framework that uses graph based metrics for insider attack detection in VANET multi-hop data dissemination protocols is proposed in [17]. It proposes a redundant data forwarding based mechanism to enable consistency checks in multi-hop data dissemination protocols to prevent insider attacks. However, the drawback of this framework is its high communication overhead and introduction of significant volume of IDS related traffic due to dissemination of redundant information.

Various cryptographic and authentication based preventive mechanisms have been proposed in the literature to address the security threats in VANET [3, 18]. A novel Authentication, Authorization and Accounting (AAA) access control schemes for application services in VANETs are proposed in [19, 20]. These schemes are based on IEEE 802.11i standard and use EAP-Kerberos model, wherein the vehicles willing to join the network send authentication request message through the intermediate vehicles and the RSU, until they reach a centralized authentication server that can grant access to the requesting mobile users. Although, AAA access control schemes based on IEEE 802.11i standards provide a promising solution for authentication and authorization between vehicles and service providers in VANETs, a full 802.11i based authentication requires a long authentication delay between 750 to 1200 ms due to lengthy round trip delay between the AAA server and the RSU [21].

An accurate and lightweight intrusion detection framework, called AECFV, which aims to protect VANET against various type of attacks is proposed in [7]. The framework uses a combination of specification rules and a Support Vector Machine (SVM) based classifier model to detect various type of attacks. However, the main drawback of the framework is the overhead involved in training the complex SVM classifier model. A novel approach for detecting wormhole attack in VANET is proposed in [22]. It uses authentication mechanism based on HEAP method [23] and can be easily implemented in AODV routing protocol without requiring any special hardware support. However, the drawback of the scheme is that it can only detect wormhole attacks in AODV routing protocol. A Learning Automata (LA) based IDS framework for VANET is proposed in [24], wherein the vehicles are equipped with Markov Chain Model based LA to capture different activities and states of the vehicles. This framework uses a classifier model based on parameter called the Collaborative Trust Index (CTI) to detect various attacks in the vehicular network. However, the drawback of this framework is the overhead involved in developing a complex Markov Chain Model based LA, which puts a limitation on its real time deployment.

A framework to identify and evict misbehaving faulty vehicles from the vehicular network are proposed in [4] [25]. These frameworks use dynamic rule-based IDS detection engines and revocation of certificate by the Certification Authority (CA) as the primarily tools to evict misbehaving vehicles from the vehicular network. Similarly, various other intrusion detection frameworks for identifying attacks like DoS, blackhole, greyhole etc., in VANET are proposed in [26, 27, 28]. However, all these frameworks suffer from various drawbacks like, latency involved in identification of the malicious vehicles, overhead in distribution of the revocation information across vehicular networks and generation of high volume of IDS traffic.

Data traffic aggregation models for wireless networks (including VANETs) to efficiently utilize the smallest unit of a Radio Access Network (RAN) and for fluently up-linking in 5G cellular networks using fixed Relay Nodes (RNs) are proposed in [29] [30, 31]. These schemes use a new 5G network slicing technique based on classification and measurement of the data traffic, to satisfy the Quality of Service (QoS) requirements of various smart system networks such as smart vehicular traffic monitoring systems. However, the main drawback of these schemes is the challenge associated with the full scale deployment of 5G radio networks over highly dynamic and mobile vehicular networks.

From our survey of related works, we found the following drawbacks associated with the existing VANET intrusion detection frameworks:

1. Some frameworks use complex markov chain and SVM based models for detecting intrusive vehicular activities [24] [7]. This increases the communication overhead involved in dissemination of intrusion detection related information across the vehicular network.
2. Authentication and cryptography based frameworks [19, 20] incur high communication overhead and large latency in dissemination of revocation information.
3. Some frameworks introduce significant volume of IDS traffic [17] [4] [25], which can cause congestion in a bandwidth constrained vehicular network.
4. Some frameworks are geared towards detection of specific type of attacks [22] and cannot be generalized for detecting other class of attacks.
5. Few IDS frameworks have scalability issues [24] [29] [30] as they suffer from broadcast storm and disconnected network problems at high and low vehicular densities, respectively.

In this paper, we aim to address these issues in existing IDS frameworks by proposing a **novel clustering algorithm**, a distributed **CH election algorithm** and a **game theory based multi layered intrusion detection framework** for VANET.

3. Multi layered game theory based hybrid intrusion detection framework

In this section, we first present an overview of the proposed intrusion detection framework for VANET and state various assumptions made by the framework. We then describe various type of attacks in VANETs followed by a detailed description of the proposed multi-layered game theory based intrusion detection framework for VANET.

The overall architecture of the proposed intrusion detection framework is shown in Fig. 2. In the proposed framework, the vehicles communicate with their respective Cluster Heads (CHs) using the IEEE 802.11p wireless standard and the CHs communicate with the Road Side Units (RSUs) using the Long-Term Evolution (LTE) wireless standard. Two different wireless standards were chosen to maintain a fair trade-off between latency and operational cost. Vehicular networks employing only the IEEE 802.11p standard encounter high delay in dissemination of safety messages due to broadcast storm and disconnected network problems at high and low vehicular densities, respectively. Cellular technologies based on LTE can mitigate this problem, as they have low latency and wide-range communication. However, a pure cellular-based VANET communication is not feasible due to high cost of communication between the vehicles and the CHs. Therefore, a hybrid vehicular network that uses a combination of IEEE 802.11p and LTE based wireless standards provide the best trade-off between the latency and the operational cost.

The proposed framework carries out the intrusion detection task at three different levels. At the lowest level, a set of agent nodes operate their Local Intrusion Detection System (LIDS) module and monitor vehicles in their neighborhood. The agent node's LIDS module uses a set of specification rules based on Received Signal Strength Indicator (RSSI), Packet Delivery Rate (PDR), Packet Forwarding Rate (PFR) and Duplicate Packet Rate (DPR) values to detect the malicious vehicles in the neighborhood. In addition, the agent nodes also monitor the CH for sign of maliciousness. If majority of the agent nodes find the CH to be malicious then a new CH is elected in its place. A detailed description about the agent nodes election algorithm and agent node's LIDS module is provided in Section 3.3.

At the intermediate level, the CH operates the Cluster Intrusion Detection System (CIDS) module to monitor vehicles in its cluster. The CIDS module uses a combination of specification rules and a lightweight neural network based classifier module to detect malicious vehicles in the cluster. It uses the information received from the agent nodes in its cluster to device a game theory based probabilistic monitoring strategy. This enables the CH to minimize the volume of IDS traffic introduced into the vehicular network. It also employs a payment mechanism for updating the agent nodes' reputation values. When the reputation of any agent node falls below the threshold value, it is replaced with a new agent node by the CH. A detailed description about the CH's CIDS module is provided in Section 3.4.

Finally, at the highest level, the RSU operates the Global Decision System (GDS) module, which receives input from multiple CHs within its radio range. The malicious vehicle reported by the CHs is assigned either to the *Malicious* list or to the *Suspicious* list based on the number of the CHs that reported the vehicle as malicious. The RSUs periodically broadcast the identities of these malicious vehicles to prevent other normal vehicles in the network from communicating with them. A detailed description of the RUS's GDS module is provided in Section 3.5.

Various algorithms running at different layers of the proposed IDS framework is also shown in Fig. 2. A brief description about these algorithms is provide below:

1. *CH election algorithm (Algorithm 1)*: This algorithm runs at every cluster of the vehicular network and elects the CH for the given cluster.
2. *Detection rule algorithm (Algorithm 2)*: This algorithm uses set of specification rules based on Packet Delivery Rate (PDR), Received Signal Strength Indicator (RSSI), Duplicate Packet Rate (DPR) and Packet Forwarding Rate (PFR) values to detect malicious vehicles in the cluster.
3. *Malicious CH detection algorithm (Algorithm 3)*: This algorithm is executed by the agent nodes to verify whether the CH is normal or malicious.
4. *IDS agent nodes election algorithm (Algorithm 4)*:This algorithm is executed at every cluster to elect a set of agent nodes, which are responsible for aiding the CH in monitoring and identifying malicious vehicles.

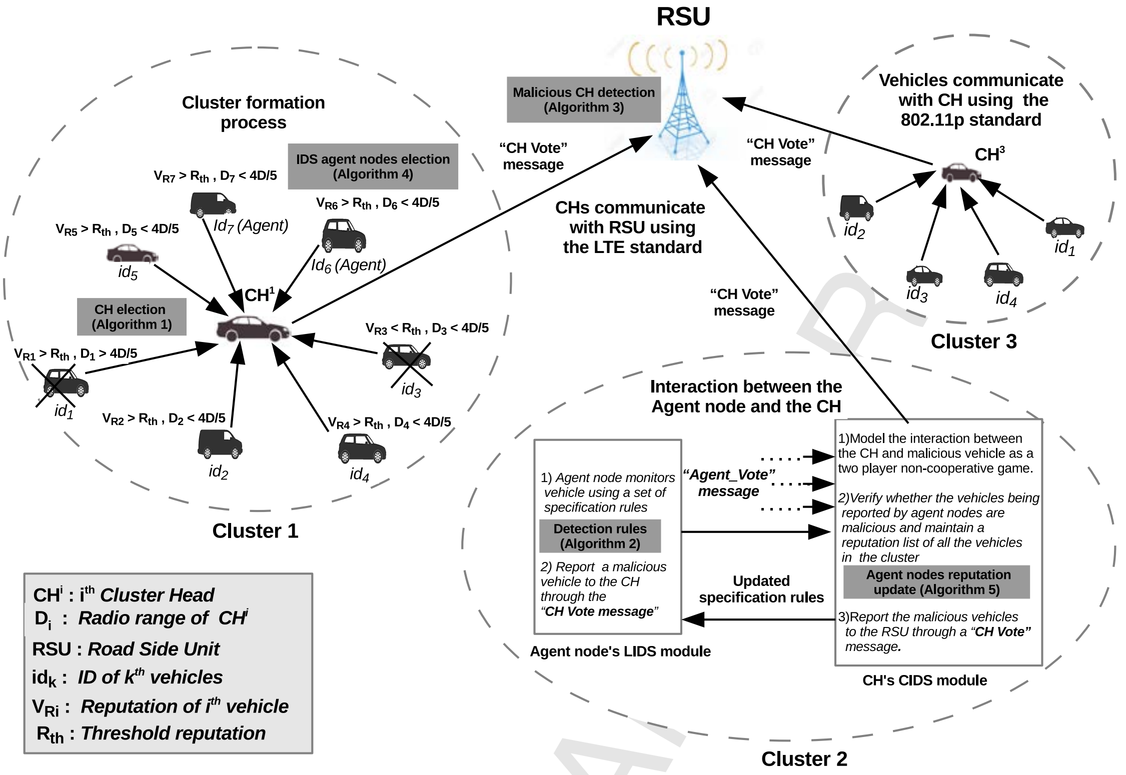

RSU
Malicious CH detection (Algorithm 3)
Vehicles communicate with CH using the 802.11p standard
Cluster formation process
IDS agent nodes election (Algorithm 4)
"CH Vote" message
"CH Vote" message
CH³
$V_{R7} > R_{th}$, $D_7 < 4D/5$
Id₇ (Agent)
$V_{R6} > R_{th}$, $D_6 < 4D/5$
Id₆ (Agent)
CHs communicate with RSU using the LTE standard
$V_{R5} > R_{th}$, $D_5 < 4D/5$
id₅
id₂
id₁
id₃
id₄
CH¹
CH election (Algorithm 1)
"CH Vote" message
Cluster 3
$V_{R3} < R_{th}$, $D_3 < 4D/5$
$V_{R1} > R_{th}$, $D_1 > 4D/5$
id₃
id₁
$V_{R2} > R_{th}$, $D_2 < 4D/5$
$V_{R4} > R_{th}$, $D_4 < 4D/5$
Interaction between the Agent node and the CH
id₂
id₄
Cluster 1
1) Agent node monitors vehicle using a set of specification rules
"Agent_Vote" message
Detection rules (Algorithm 2)
2) Report a malicious vehicle to the CH through the "CH Vote message"
Updated specification rules
1)Model the interaction between the CH and malicious vehicle as a two player non-cooperative game.
2)Verify whether the vehicles being reported by agent nodes are malicious and maintain a reputation list of all the vehicles in the cluster
Agent nodes reputation update (Algorithm 5)
3)Report the malicious vehicles to the RSU through a "CH Vote" message.
Agent node's LIDS module
CH's CIDS module
Cluster 2
CH^i : i^{th} Cluster Head
D_i : Radio range of CH^i
RSU : Road Side Unit
id_k : ID of k^{th} vehicles
V_{Ri} : Reputation of i^{th} vehicle
R_{th} : Threshold reputation

Figure 2: **Proposed multi layered VANET intrusion detection framework**

5. *Agent node reputation update algorithm (Algorithm 5)*: This algorithm updates the reputation values of the agent nodes in the cluster based on their observed behavior. The agent nodes found to be behaving maliciously are penalized with negative payment and removed from the cluster by the CH.

Elaborate details about each of these algorithms are provided in the later sections of the paper.

We make the following assumptions with respect to the proposed intrusion detection framework:

1. Vehicles are equipped with 802.11p enabled wireless DSRC radio, which enables them to communicate with each other. Vehicles use Global Positioning System (GPS) and digital map to determine their coordinates at real time. Additionally, vehicles employ public key based cryptographic solutions to ensure communication privacy and source authentication.
2. Only the vehicles that have stopped at the traffic signal and the vehicles approaching the road intersection point with relatively low speed are considered in the proposed framework. This allow vehicles enough time frame to form clusters and exchange their gathered information.
3. As shown in Fig. 3, the vehicular network is partitioned into multiple grid regions and each region is assigned a unique identity number (ID). In the figure these IDs are numbered **A** through **T**. In order to comply with the maximum transmission range under DSRC standard, each grid's dimension is set to 1000 m × 1000 m. The vehicles are grouped into clusters based on their grid IDs. Vehicles can only communicate with other vehicles in their own cluster. Any inter cluster communication has to be made via the CH. The CHs exchange their information containing the list of malicious vehicles when they come into each others radio range.
4. Prior to participating in the vehicular network, all the vehicles must initially register with one of the RSUs. The RSUs maintain a reputation list of all the registered vehicles in their radio range.

Due to the wireless nature and broadcast medium of communication in VANET, a malicious vehicle can disseminate false alert messages for its own selfish gain and disrupt the normal functioning of the network. In our study we have considered the following class of attacks in VANETs:

1. **Selective forwarding and black hole attacks**: In the selective forwarding attack, the malicious vehicle selectively forwards the data packets while dropping others. On the other hand, in the black hole attack the malicious vehicle drops all the packets that it receives without forwarding them further. A malicious vehicle or a CH performing these attacks can be detected by computing its Packet Delivery Rate (PDR) and Received Signal Strength Indicator (RSSI) values and comparing them with a threshold PDR ($T_{pdr_{sf}}$, $T_{pdr_{bh}}$) and RSSI ($T_{rssi_{bh}}$) values.

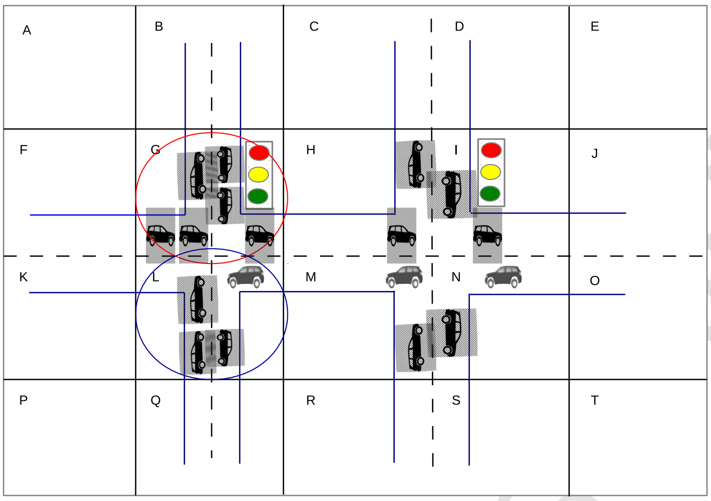


Figure 3: **An illustration of cluster formation in the proposed framework**

2. **Denial of Service (DoS) attack**: In this attack, the malicious vehicle inundates the network with a large number of fake alert messages about road accident and congestion in order to consume the network's bandwidth. A malicious vehicle performing a DoS attack can be detected by computing its Duplicate Packet Rate (DPR) and Packet Forwarding Rate (PFR) values. If its DPR and PFR exceed the threshold values $T_{dpr_{dos}}$ and $T_{pfr_{dos}}$, respectively then it is assumed to be carrying out the DoS attack.
3. **Wormhole attack**: In this attack, two malicious vehicles located at different locations collude together to form a private tunnel. To execute this attack, the malicious vehicle generates a high RSSI value signal to convince other normal vehicles in its neighborhood that it has the shortest path to destination or the CH. Thereafter, the malicious vehicle forwards all the received packets to another malicious vehicle at the other end of the tunnel, which in turn either drops the packets or modifies them before forwarding them to the destination. If the RSSI and PDR values of the vehicle being monitored exceed the threshold values $T_{rssi_{wh}}$ and $T_{pdr_{wh}}$, respectively then it is assumed to be carrying out the wormhole attack.
4. **Sybil attack**: In this attack, the malicious vehicle creates a multiple fake identities of itself in order to prevent detection when launching various other attacks like black hole and DoS attacks. Sybil attack can be detected by computing the RSSI value of the vehicle and then verifying whether it follows a normal distribution. To detect this attack, the mean (μ) and the standard deviation (σ) corresponding to the RSSI values of all the vehicles in the cluster are calculated. The 'Z-score' of RSSI value of vehicle v_i ($RSSI_{v_i}$) is then calculated using the formula $\frac{RSSI_{v_i}-\mu}{\sigma}$. If the 'Z-score' of RSSI value of the vehicle being monitored exceeds the value 2.5 ($T_{rssi_{syb}}$) in the normal distribution curve, then the vehicle is assumed to be carrying out the Sybil attack.

Before delving into the detailed description of the proposed multi-layered game theory based intrusion detection framework, we provide a description about the distributed clustering algorithm employed by the proposed framework for generating stable vehicular clusters. We also discuss a novel CH election algorithm along with a stimulus structure based on Vickrey-Clarke-Groves (VCG) mechanism [32] for motivating vehicles to actively participate in the CH election process.

3.1. Distributed cluster formation and CH election algorithms

The effectiveness of any cluster based VANET intrusion detection framework largely depends on the stability of the clusters produced by the clustering algorithms. Stable clusters reduce the overhead involved in cluster formation process and provide vehicles with sufficient time frame to exchange their data. To achieve this objective, a distributed clustering algorithm for VANET that produces stable vehicular clusters with enhanced connectivity among member vehicles is proposed in this paper.

The proposed clustering algorithm requires the vehicles to periodically broadcast beacon messages to inform other vehicles in the neighborhood about their presence. The beacon message comprises the vehicle's ID, its current region ID (coordinates) and its cluster membership status. The vehicles use the beacon messages received from their neighborhood vehicles to generate their Social Choice Function (SCF) tables. The SCF table of the vehicle comprises the IDs of all the vehicles within its radio range along their associated reputation values. Initially, the reputation values of the vehicles in the SCF tables are initialized to some default values. The vehicle v_i later updates the reputation of vehicles in its SCF table (SCF_{v_i}) based upon its observation about their behavior and the updates that are received from the CH and the RSU. When the reputation of any vehicle $v_j \in SCF_{v_i}$ falls below the threshold value (R_{th}), it is removed from the SCF_{v_i}.

Vehicles in VANET are constrained by road topologies, which require them to follow traffic lights and road signs leading to a predictable mobility pattern with restricted movement along predefined directions. The proposed clustering algorithm exploits these vehicular constraints to generate stable clusters. The clustering process starts at the road intersection point with vehicles broadcasting the cluster formation messages (Clu_{frm}) comprising their identities, current region IDs, velocities and their SCF table details. If the vehicle v_i receives a Clu_{frm} message from a vehicle v_j that is not in its SCF table, i.e., $v_j \notin SCF_{v_i}$ and if the region ID of v_i and v_j are same, then v_i adds v_j to its SCF table. Upon successful exchange of the Clu_{frm} messages, the vehicles compute the mean and standard deviation values of the vehicular velocities in their SCF tables. The vehicles are then grouped into clusters based on their velocities and region IDs. All the vehicles in a given cluster must belong to the same region ID and their velocities must be within three standard deviation of the mean cluster velocity. If the vehicle v_i cannot find any other vehicle in its neighborhood, i.e., it does not receive any Clu_{frm} messages in response to its own Clu_{frm} message, then it incrementally increases its transmission range to find new neighbors. The transmission range can be increased up to 1000 m, which is the maximum transmission range allowed in the DSRC standard, after which v_i declares itself as the CH with v_i as the sole member of the cluster.

After the successful completion of the cluster formation process, the CH election process is initiated at each cluster. Each vehicle v_i in the cluster C computes the utility function of every other vehicle $v_j \in SCF_{v_i}$ using the following rule:

$$U_{v_j}^{v_i} = \beta_1 R_{v_j^i} + \beta_2 \mid SCF_{v_j} \mid \quad \forall \quad v_j, v_i \in \ C \qquad (1)$$

where $R_{v_j^i}$ is the reputation of v_j in SCF_{v_i}. $\mid SCF_{v_j} \mid$ is the number of vehicles in the SCF list of v_j. β_1 and $\beta_2 \in [0,1]$ are the weight parameters used for specifying the significance of reputation and connectivity metrics of v_j in computation of the utility function $U_{v_j}^{v_i}$.

After computing the utility functions corresponding to every vehicle in their SCF lists, the vehicles exchange their utility function lists ($Utility_{list}$). The $Utility_{list}$ of vehicle v_i comprises the utility functions corresponding to every other vehicle in its SCF table. The $Utility_{list}$ of v_i is of the form $\{U_{v_a}^{v_i}, U_{v_b}^{v_i}, ..., U_{v_k}^{v_i}\}$, where v_a, v_b .., $v_k \in SCF_{v_i}$. v_i then computes the aggregated utility function corresponding to every vehicle $v_j \in SCF_{v_i}$ ($U_{v_{ij}}$) using the $Utility_{list}$ it received from other vehicles. Finally, the vehicle with the highest aggregated utility function is elected as the CH. It is to be noted that the vehicle assigns higher weight to its own utility function compared to utility functions received from other vehicles, while computing the aggregated utility function. A detailed description of the proposed CH election mechanism is given by Algorithm 1

Furthermore, to enhance the connectivity among vehicles within a given cluster and to ensure that malicious vehicles are not provided cluster memberships, the clustering algorithm places an additional constraint, which requires the vehicles to be within a four-fifth radio range of the elected CH ($\frac{4}{5}D$) and also have an average reputation greater than the predefined threshold value (R_{th}) to be the cluster members. As the vehicles on the boundary radio range of the CH are more likely to exit the cluster, this process ensures the stability of the cluster and minimizes the frequency of the cluster formation process. This concept is elaborated in *Cluster 1* of Figure 2. As shown in the figure, vehicles 2, 4 and 5 (id_2, id_4, id_5) are included in Cluster 1, as they satisfy all the constraints. On the other hand, vehicle 1 (id_1) and vehicle 3 (id_3) are excluded from the cluster as the distance of former lies outside the threshold radio range of the CH (greater than $\frac{4}{5}D$), while the average reputation of the latter is below the threshold value (R_{th}). The value of R_{th} is set to one-half of the average reputation of the agent nodes in the cluster. The agent nodes election process and their reputation update mechanism are provided in Section 3.3.

Algorithm 1 : Distributed CH election algorithm

Input : *Cluster C and SCF table details of each vehicle $v_i \in C$*
Output: *Cluster Head (CH) of the cluster C*

```
$v_i \xleftrightarrow{SCF\ \ table} cluster_{-v_i}^{C}$   /* Each vehicle $v_i \in C$ exchange its SCF table with every other vehicles in C */
for each $v_j$, $v_i \in C$ do
    $U_{v_j}^{v_i} = \beta_1 R_{v_j^i} + \beta_2 | \ SCF_{v_j} \ |$ , where $\beta_1, \beta_2 \in [0,\ 1]$ /* $v_i$ computes the utility function of $v_j$. */
end
$v_i \xleftrightarrow{Utility_{list}} cluster_{-v_i}^{C}$   /* Vehicle $v_i$ broadcasts its utility function list */
for each $v_i \in C$ do
    if $v_j \in SCF_{v_i}$
    $U_{v_{ij}} = \alpha_1 U_{v_j}^{v_i} + \frac{\alpha_2 \sum_{k=1}^{N} U_{v_j}^{v_k}}{N}$, where $\alpha_1, \alpha_2 \in [0, 1]$ and $N$ is the total number of vehicle from which $v_i$ received the utility function value of $v_j$
end
if $U_{v_{ij}} > U_{v_{ik}} \quad \forall\ v_k \in C$ then
    $cluster_{-v_j}^{C} \xrightarrow{CH_{elect}} v_j$   /* $v_j$ is elected as the CH */
    $v_j \xrightarrow{Confirmation} cluster_{-v_j}^{C}$
else
    $cluster_{-v_{j'}}^{C} \xrightarrow{CH_{elect}} v_{j'}$, where $U_{v_{ij'}} > U_{v_{ik}}\ \forall\ v_k \in C$
    $v_{j'} \xrightarrow{Confirmation} cluster_{-v_{j'}}^{C}$   /* $v_{j'}$ is elected as the CH */
end
```

When a new vehicle v_n wants to join one of the existing clusters, it sends out the cluster join request message (Clu_{join}) comprising its ID, direction of movement, current coordinates and velocity information. Upon receiving the Clu_{join} message from v_n, the CH verifies whether the coordinates, direction and velocity information of v_n meet the requirement for providing v_n with the cluster membership. If v_n satisfies all the criteria, then the CH makes an entry for v_n in its SCF table and sends a cluster join approval message (Clu_{apprv}) to v_n. However, if v_n does not receive any Clu_{apprv} message after waiting for a predefined time period, it moves to the next road intersection point and re-executes the cluster join request procedure there.

3.2. VCG mechanism based payment structure for CH

Since monitoring operation requires a substantial amount of computing and resources therefore, vehicles do not have any profitable incentive to act as the CH unless they are provided with some form of stimulus. Towards this end, we propose a stimulus based structure to encourage vehicles to participate in the CH election process by providing them payments in the form of reputation gain for carrying out the CH monitoring operation. Data packets of the reputed vehicles are given higher priority compared to those with lower reputation during traffic routing. Therefore, vehicles with higher reputation maintain greater throughput even during the congestion period. The cost function of the vehicle v_i for performing the monitoring operation after being elected as the CH of cluster C is given by the following equation:

$$Cst_{v_i} = \frac{R_{v^i_{avg}}}{\sum_{j=1}^{n} R_{v^j_{avg}}} * \frac{n}{U_{v^i_{avg}}} \tag{2}$$

where n is the total number of vehicles in C. $R_{v^i_{avg}} = \frac{\sum_{k=1}^{n-1} R_{v^k_i}}{n-1}$, $\forall\ v_k \in C$ is the average reputation value of v_i in C. $U_{v^i_{avg}} = \frac{\sum_{k=1}^{n-1} U_{v^k_i}}{n-1}$ is the average aggregated utility function value of v_i in C. Each vehicle v_i holds a private information about its type (Θ_{v_i}). The type Θ_{v_i} can be either *Normal* or *Malicious*. The reward function for vehicle v_i when it is elected as the CH is given by following equation:

$$Rwd_{v_i}(\Theta_{v_i}, \Theta_{-v_i}) = P_{v_i} - Cst_{v_i} \tag{3}$$

where Θ_{-v_i}, represents the type of all other vehicles except v_i. P_{v_i} is the payment made to v_i in the form of reputation gain by the mechanism when it is elected as the CH. Every vehicle $v_k \in C$ would want to maximize its reward function (Rwd_{v_k}). It signifies the reward value if it chooses the type Θ_{v_i}. However, v_i might not reveal its true cost function value (Cst_{v_i}) by either over valuing or under valuing it, if doing so leads to higher reward. Therefore, to address this issue, we propose a VCG mechanism based payment structure, wherein truthful revelation of the cost function is the dominant strategy [32] [33]. The primary objective of the proposed CH election mechanism is to elect the vehicle $v_i \in C$ with the least Cst_{v_i} value. Since, $Cst_{v_i} \propto \frac{1}{U_{v^i_{avg}}}$ therefore, electing vehicle with the least Cst_{v_i} value as the CH is equivalent to electing vehicle with the highest $U_{v^i_{avg}}$ value. We refer to this as the Social Choice Function (SCF) and define it as:

$$SCF = Min\{Cst_{v_i} \quad \forall \quad v_i \in C\} \tag{4}$$

If multiple vehicles have the same *SCF* value, then the vehicle with the highest reputation value is elected as the CH. The payment to the elected CH vehicle v_i is made using the VCG mechanism. The payment received by v_i (P_{v_i}) is equal to the second least cost function value of vehicle v_k (Cst_{v_k}) excluding the cost function of the elected CH v_i.

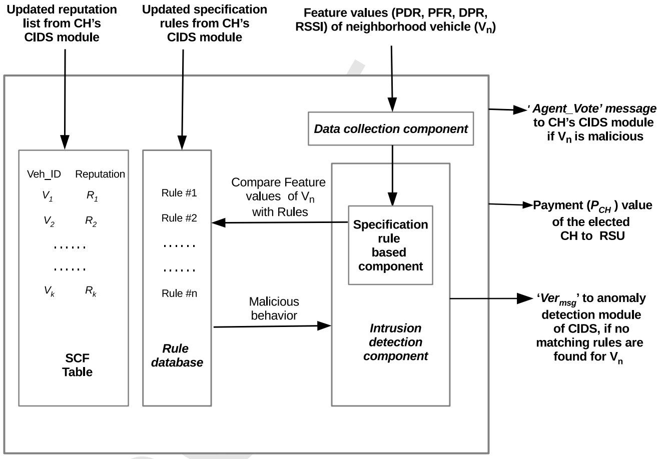

Updated reputation list from CH's CIDS module
Updated specification rules from CH's CIDS module
Feature values (PDR, PFR, DPR, RSSI) of neighborhood vehicle (V$_n$)
Data collection component
Veh_ID
Reputation
V_1
R_1
V_2
R_2
……
……
V_k
R_k
SCF Table
Rule #1
Rule #2
……
……
Rule #n
Rule database
Compare Feature values of V$_n$ with Rules
Malicious behavior
Specification rule based component
Intrusion detection component
'Agent_Vote' message to CH's CIDS module if V$_n$ is malicious
Payment (P_{CH}) value of the elected CH to RSU
'Ver_{msg}' to anomaly detection module of CIDS, if no matching rules are found for V$_n$

Figure 4: **Agent node's LIDS module**

$$P_{v_i} = Min\{Cst_{v_k} \quad \forall \quad v_k \in C \quad and \quad v_k \neq v_i\} \tag{5}$$

After vehicle v_i is elected as the CH and the payment (P_{v_i}) to be made to v_i is calculated, the agent nodes compute the reward function for v_i (Rwd_{v_i}) and inform the RSU to increment the reputation of v_i by Rwd_{v_i} value.

In the subsequent sub-sections, we provide a detailed description about various components of the proposed VANET intrusion detection framework namely, the agent node's LIDS module, CH's CIDS module and the RSU's GDS module. These intrusion detection components interact with each other to identify the malicious vehicles and provide a comprehensive security to the vehicular network.

3.3. Agent node's Local Intrusion Detection System (LIDS) module

At the lowest level of the proposed framework, the agent nodes operate the LIDS module to monitor vehicles in their neighborhood for sign of maliciousness. The agent node maintains a *SCF Table* comprising the identities and the reputation values of the vehicles in its radio range. The *SCF Table* also contains the identities of the vehicles blacklisted by the RSU, which should be excluded from the vehicular communication.

The LIDS module uses a set of specification rules based on RSSI, PDR, PFR and DPR values to detect malicious vehicles in the cluster. The overall architecture of the agent node's LIDS module is shown in Fig. 4. The *Data collection component* of the LIDS module computes the RSSI, PDR, PFR and DPR values of the vehicle being monitored. These information are then forwarded to the *Intrusion detection component*, which uses a set of specification rules (Algorithm 2) stored in its *Rule database* to detect various type of attacks. When a malicious vehicle is detected by the agent node, it sends a '*Agent_Vote*' message comprising the identity of the malicious vehicle along with the detected attack type to the CH. The CH gathers ‘*Agent_Vote*’ messages from multiple agent nodes to determine whether the vehicle being reported is indeed malicious. Since the network dynamics in VANET changes frequently, the agent nodes receive updated specification rules from the CH at regular intervals. When there are no matching specification rules against the vehicle being monitored, a verification message (Ver_{msg}) is forwarded to the CH's neural network based ‘*Anomaly detection component*’ for further analysis.

Additionally, the agent nodes also monitor the CH for sign of maliciousness. Since CHs perform many vital tasks like data aggregation and monitoring, they are attractive targets for attacker as compromising them can provide the attacker with a huge payoff. The attacker can disrupt the vehicular network through compromised CH by propagating false information and ignoring to act against the malicious vehicles reported by the agent nodes. Therefore, a cooperative detection mechanism (Algorithm 3) is adopted by the agent nodes to detect the malicious CH. Each agent node maintains a binary variable called *CH_Status*, which is initially set to 0. However, when the agent node finds the CH to be acting maliciously, it sets the *CH_Status* variable to 1. The agent node uses the same set of detection rules given in Algorithm 2 to detect the malicious CH. When more than one half of the agent nodes in the cluster find the CH to be misbehaving, they report the malicious CH to the RSU. The RSU then blacklists the malicious CH and broadcasts a message asking vehicles in the cluster to elect a new CH.

Algorithm 2 : Detection rules for various attacks

```
Input : Identity (Node_ID), PDR, RSSI, DPR and PFR values of the vehicle
Output: Prediction whether Node_ID is malicious or normal

if ($PDR_{Node\_ID} > T_{pdr_{sf}}$) then
    // node is performing selective forwarding attack
    send Agent_Vote(Node_ID, selective forwarding) message to the CH
end
if ( $RSSI_{Node\_ID} > T_{rssi_{syb}}$) then
    // node is performing sybil attack
    send Agent_Vote(Node_ID, sybil attack) message to the CH
end
if ($PDR_{Node\_ID} > T_{pdr_{bh}}$ & $RSSI_{Node\_ID} > T_{rssi_{bh}}$) then
    // node is performing black hole attack
    send Agent_Vote(Node_ID, black hole attack) message to the CH
end
if ($DPR_{Node\_ID} > T_{dpr_{dos}}$ & $PFR_{Node\_ID} > T_{pfr_{dos}}$) then
    // node is performing DoS attack
    send Agent_Vote(Node_ID, DoS attack) message to the CH
end
if ($RSSI_{Node\_ID} > T_{rssi_{wh}}$ & $PDR_{Node\_ID} > T_{pdr_{wh}}$) then
    // node is performing worm hole attack
    send Agent_Vote(Node_ID, worm hole attack) message to the CH
end
```

Algorithm 3 : Distributed cooperative mechanism for detecting malicious CH

```
Input : Agent nodes' CH_Status variables
Output: Prediction whether the CH is malicious or normal

$Agt_i \xleftrightarrow{CH\_Status} Agt_{k-i}$    /* ‘k’ agent nodes exchange their CH_Status variables */
if Count(CH_Status == 1) < k/2; then
    CH is normal
else
    CH is malicious
    Report the malicious CH to RSU
    RSU informs vehicles in the cluster to initiate a new CH election process
end
```

3.3.1. Distributed agent nodes election algorithm

The performance of the proposed intrusion detection framework largely depends upon the agent nodes election algorithm. Selecting few agent nodes degrade the detection rate of the IDS, while selecting too many agent nodes introduce a large volume of intrusion detection related traffic in the network. Therefore, to maintain a good trade-off between the detection rate and the IDS traffic volume, we propose a distributed agent node election algorithm (Algorithm 4) that elects an appropriate number of highly reputed vehicles as the agent nodes. The agent nodes election process of cluster C starts with the vehicle $v_k \in C$ broadcasting the *IDS_Agent_Elect* (ID_{v_k}, SCF_{v_k}) message comprising its ID and SFC table details. Upon receiving the *IDS_Agent_Elect ()* message from v_k, every other vehicle in the cluster C broadcast their own *IDS_Agent_Elect ()* messages. Each vehicle v_j then calculates the average aggregated reputation of every other vehicle $v_i \in C$ ($Agg_{R_{v_{ji}}}$) using the SCF table details it received in the *IDS_Agent_Elect ()* messages. The algorithm then elects the top k vehicles with the highest aggregated reputation values as the agent nodes. Through various rounds of simulations, it was found that electing between 25% to 30% of the vehicles in the cluster as the agent nodes gives the best trade-off between high detection rate and low IDS traffic volume.

3.4. CH's Cluster Intrusion Detection System (CIDS) module

At the intermediate level of the proposed framework, the CH uses *Cluster Intrusion Detection System (CIDS)* module to detect malicious vehicles. The overall architecture of the CH's

Algorithm 4 : Algorithm for electing k agent nodes

```
Input : Cluster C and IDS agent election messages (IDS_Agent_Elect ( ))
Output: 'k' elected IDS agents of cluster C

$v_k \xleftrightarrow{IDS\_Agent\_Elect(ID_{v_k}, SCF_{v_k}))} Cluster^{C}_{-v_k}$   /* Vehicles in C exchange the agent nodes election messages */

Each vehicle $v_j$ computes the average aggregated reputation of every other vehicle $v_i \in C$ ($Agg_{R_{v_{ji}}}$) using the SCF table information obtained from the IDS_Agent_Elect ( ) messages

Let $\{R_{k_{th}}\}$ be the set of 'k' vehicles in C with the highest aggregated reputation values

if $v_i \in \{R_{k_{th}}\}$ then
    $Cluster^{C}_{\{-R_{k_{th}}\}} \xrightarrow{IDS_{agent}} v_i$   /* $v_i$ is elected as agent node */
    $v_i \xrightarrow{Confirm} Cluster^{C}_{\{-R_{k_{th}}\}}$   /* $v_i$ sends acknowledgment */
else
    $v_i \xrightarrow{IDS_{agent}} \{R_{k_{th}}\}$
    $\{R_{k_{th}}\} \xrightarrow{Confirm} v_i$
end
```

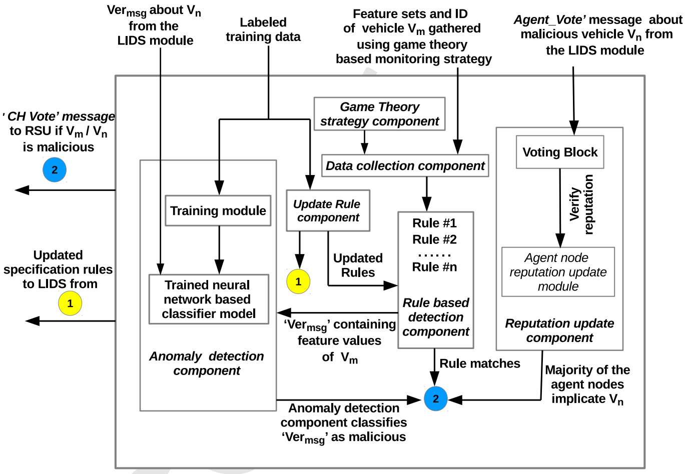
CIDS module is shown in Fig. 5. As shown in the figure, the CIDS module comprises four different components namely, the *Rule based detection component*, the neural network based *Anomaly detection component*, the *Update Rule component* and the agent node *Reputation update component*. A detailed description about each of these components are provided in the subsequent sub-sections.

3.4.1. *Rule based detection component*

This component uses a set of specification rules based on RSSI, PDR, PFR and DPR values (Feature sets) to detect malicious vehicles in the cluster. It uses the same set of specification rules as used by the agent node's LIDS module (Algorithm 2) to identify the malicious vehicles. When there are no matching specification rules, a verification message (Ver_{msg}) containing the Feature sets, reputation value and ID of the vehicle being monitored is sent to the '*Anomaly detection component*' for further analysis. The specification rules of the CH's CIDS module are updated more frequently compared to that of the agent node's LIDS module and therefore, the CIDS module contains more updated version of the specification rules. When a malicious vehicle is detected by the CIDS module, it informs both the RSU and the agent nodes in its cluster.

3.4.2. *Update rule component*

This component provides the updated specification rules for both the LIDS and CIDS modules of the agent node and the CH, respectively. However, the *Rule based detection component* of CIDS module is updated more frequently compared to that of the LIDS module. This component uses the mean and the standard deviation values of the PDR, RSSI, PFR and DPR observed during the normal training period to derive new specification rules.



Ver_{msg} about V_n from the LIDS module
Labeled training data
Feature sets and ID of vehicle V_m gathered using game theory based monitoring strategy
Agent_Vote' message about malicious vehicle V_n from the LIDS module
' *CH Vote' message* to RSU if V_m / V_n is malicious
2
Game Theory strategy component
Data collection component
Voting Block
Verify reputation
Training module
Update Rule component
Rule #1
Rule #2
......
Rule #n
Updated specification rules to LIDS from
1
Updated Rules
1
Trained neural network based classifier model
Agent node reputation update module
Rule based detection component
'Ver_{msg}' containing feature values of V_m
Reputation update component
Anomaly detection component
Rule matches
Majority of the agent nodes implicate V_n
2
Anomaly detection component classifies 'Ver_{msg}' as malicious

Figure 5: **CH's CIDS module**

3.4.3. *Neural network based anomaly detection component*

This component takes its input from the agent node's LIDS module and the CIDS's '*Rule based detection component*' in the form of verification messages (Ver_{msg}) and classify them as either normal or one of the attack types (malicious). The rationale behind adopting a neural network based classifier model is because of their ability to produce better classification models compared to other machine learning algorithms. The anomaly detection component is initially trained with a labeled dataset comprising feature sets and reputation values of the vehicles. Additional features like packet size, source IP, destination IP, hop counts, packet sequence number, velocity and region ID are also used as parameters for training the classifier. Since, the normal feature values of vehicles in VANET vary over time due to change in several network parameters like topology, traffic conditions, congestion etc., therefore, the anomaly detection component is re-trained periodically to incorporate these changes into the classifier model.

3.4.4. *Agent node reputation update component*

This component of the CIDS module maintains a reputation list of all the agent nodes in its cluster. It updates the agent nodes reputation using the rule described in Algorithm 5, where $\{Agent\}$ denotes the set of all the k agent nodes in the cluster and $\{Agent^*\} \subseteq \{Agent\}$ denotes the set of k^* ($k^* \leq k$) agent nodes that reported the vehicle v_m as malicious. The CH computes the average reputation R^k_{agg} and $R^{k^*}_{agg}$ of the agent nodes in the set $\{Agent\}$ and $\{Agent^*\}$, respectively. If $R^{k^*}_{agg}$ is greater than or equal to (R^k_{agg}), then the reputation of the agent nodes that reported v_m as malicious are incremented by one-fourth of their current reputation values. However, if $R^{k^*}_{agg}$ is less than R^k_{agg} but greater than one half of R^k_{agg}, then v_m is considered to be suspicious by the CH and a game theory based probabilistic monitoring strategy is adopted by the CH to monitor v_m. Finally, if $R^{k^*}_{agg}$ is less than one-half of R^{k}_{agg}, then the reputation of the agent nodes that reported v_m as malicious are decremented by one-forth of their current values. This mechanism ensures that the malicious agent nodes can not collude together to falsely implicate a normal vehicle as malicious. Moreover, when the reputation of any agent node falls below the threshold value ($0.3R^{k}_{agt}$), it is replaced with a new agent node.

Algorithm 5 : Agent node reputation update mechanism

```
Input : 1) Cluster C with cluster head (CH)
2) $\{Agent\} = \{agt_1, ...., agt_k\}$    // Set of k agent nodes in C
3) $\{Agent^*\} = \{agt_1, ...., agt_{k^*}\}$    // Set of $k^*$ ($k^* \leq k$) agent nodes that reported vehicle $v_m$ as malicious
Output: Updated reputation values of agent nodes in $\{Agent^*\}$
$R^{k}_{agg} = \sum_{i=1}^{k} \frac{R^{CH}_{agt_i}}{k}$    // Aggregated reputation of the 'k' agent nodes in $\{Agent\}$, where $R^{CH}_{agt_i}$ is the reputation of $agt_i$ in the CH's reputation list
$R^{k^*}_{agg} = \sum_{j=1}^{k^*} \frac{R^{CH}_{agt_j}}{k^*}$    // Aggregated reputation of the $k^*$ agent nodes in $\{Agent^*\}$
if ($R^{k^*}_{agg} \geq R^{k}_{agg}$ ) then
    $v_m$ is malicious
    $R^{CH}_{agt_j} = R^{CH}_{agt_j} + 0.25*R^{CH}_{agt_j}\ \forall\ agt_j \in \{Agent^*\}$
else if ($0.5*R^{k}_{agg} < R^{k^*}_{agg} < R^{k}_{agg}$) then
    Monitor $v_m$ with monitoring probability determined by the 'Game Theory strategy component'
else
    $R^{CH}_{agt_j} = R^{CH}_{agt_j}$ - $0.25*R^{CH}_{agt_j}\ \forall\ agt_j \in \{Agent^*\}$
end
if ($R^{CH}_{agt_j} < 0.3R^{k}_{agg}$) then
    remove $agt_j$ from $\{Agent\}$
```

3.4.5. *Game theory based strategy component*

This CIDS component devices probabilistic monitoring strategies for the CH based on various parameters like CH's detection rate, false alarm rate and monitoring cost. Persistent monitoring by the CH produces a significant volume of intrusion detection related traffic, which can cause congestion in a bandwidth constrained vehicular network. To address this issue, a game theory based probabilistic monitoring strategy is adopted by the CH to monitor the malicious vehicles reported by the agent nodes. The interaction between the malicious vehicle and the CH is formulated as a two player non-cooperative game between the attacker and the defender. Without loss of generality, we make an assumption that both the CH and the malicious vehicle are rational players and their actions are based upon intelligent consideration of the possible consequences of their chosen strategy. In the said game, the malicious vehicle has two pure strategies : *Attack* and *Wait*. Similarly, the CH has two pure strategies : *Monitor* and *Not Monitor*. Each player choses a strategy that maximizes its overall payoffs. To develop a payoff matrix corresponding to the interaction between the CH and the malicious vehicle, we introduce the following terminologies:

- Let α, β and γ denote the detection rate, the false positive rate and the monitoring cost of the CH, respectively.
- Let δ be the average number of vehicles in the cluster accepting and forwarding information from a malicious vehicle.

Table 1: Strategic form of the game between malicious vehicle (attacker) and CH (defender)

	Attack	**Wait**
Monitor	$(2\alpha - \gamma + 1)$, $(1 + \delta - 2\alpha)$	$-(\beta + \delta), \beta$
Not Monitor	$-(1 - \alpha), (1 - \alpha + \delta)$	0, 0

Table 1 shows the strategic form of the non-cooperative game between the CH and the malicious vehicle. The strategy space of the CH and the malicious vehicle are S_D = {*Monitor, Not Monitor*} and S_A = {*Attack, Wait*}, respectively. A pure Nash Equilibrium (NE) of this non-cooperative game corresponds to the strategy pair (S^*_d, S^*_a) of the CH and the malicious vehicle that satisfies the following conditions:

$$U_A(S^*_d, S^*_a) \geq U_A(S^*_d, S_a) \quad \forall S_i \in S_A$$
$$U_D(S^*_d, S^*_a) \geq U_D(S_d, S^*_a) \quad \forall S_j \in S_D$$

where, $U_A(S^*_d$,$S^*_a)$ and $U_D(S^*_d$,$S^*_a)$ are the payoff utilities of the malicious vehicle and the CH when they choose their strategies S^*_a and S^*_d, respectively. Any unilateral deviation by either the CH or the malicious vehicle from their chosen NE strategy results in a reduced payoff for the deviating player. Clearly, there does not exist any pure strategy NE for this non-cooperative game. Therefore, we derive a mixed strategy NE. Let p and q denote the probabilities of the malicious vehicle and the CH to play their pure strategies *Attack* and *Monitor*, respectively. Therefore, when the CH plays its strategy *Monitor* with probability q, the payoff utility of the malicious vehicle if it plays its pure strategies *Attack* and *Wait*, respectively are:

$$\begin{aligned} U_A(Attack) = &\ (1 + \delta - 2\alpha)q + (1 - \alpha + \delta)(1 - q) \\ U_A(Wait) = &\ \beta q \end{aligned}$$

Similarly, when the malicious vehicle plays its strategy *Attack* with probability p, the payoff utility of the CH if it plays its pure strategies *Monitor* and *Not Monitor*, respectively are:

$$\begin{aligned} U_D(Monitor) = &\ (2\alpha - \gamma + 1)p - (\beta + \delta)(1 - p) \\ U_D(Not\ \ monitor) = &\ -(1 - \alpha)p \end{aligned}$$

The malicious vehicle chooses to play its strategy *Attack* when U_A*(Attack)* > U_A*(Wait)*, i.e., when the monitoring probability of the CH (q) < $\frac{(1-\alpha-\delta)}{\alpha+\beta}$. Similarly, the CH choses to play its strategy *(Monitor)* when U_D*(Monitor)* > U_D*(Not monitor)*, i.e., when the malicious vehicle's attack probability (p) > $\frac{(\beta+\delta)}{(2+\alpha+\beta+\delta-\gamma)}$. Therefore, the mixed strategy NE of the game corresponds to the strategy combination (p^*, q^*), where $p^* = \frac{(\beta+\delta)}{(2+\alpha+\beta+\delta-\gamma)}$ and $q^* = \frac{(1-\alpha-\delta)}{\alpha+\beta}$ are the probabilities of the malicious vehicle and the CH to play their strategy *Attack* and *Monitor*, respectively. It can be observed that both the attacking and the monitoring probabilities of the malicious vehicle and the CH are inversely proportional to the detection rate of the CH (α) i.e., $p^* \propto \frac{1}{\alpha}$ and $q^* \propto \frac{1}{\alpha}$. Therefore, a high value of α decreases both the attacking and monitoring probabilities at the NE. The probabilistic game theory based IDS monitoring strategy developed by modeling the interaction between the IDS and the malicious vehicle as two player non-cooperative game significantly reduces the volume of IDS traffic in the vehicular network, without compromising the overall performance (detection rate and accuracy) of the IDS framework.

3.5. *RSU's Global Decision System (GDS) module*

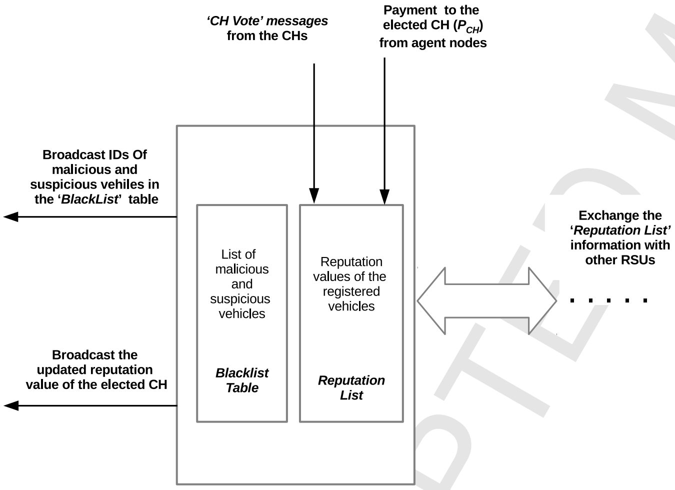

'CH Vote' messages from the CHs
Payment to the elected CH (P_{CH}) from agent nodes
Broadcast IDs Of malicious and suspicious vehiles in the 'BlackList' table
Broadcast the updated reputation value of the elected CH
List of malicious and suspicious vehicles
Blacklist Table
Reputation values of the registered vehicles
Reputation List
Exchange the 'Reputation List' information with other RSUs
. . . . .

Figure 6: **RSU's GDS module**

At the highest level of the proposed framework, the RSU maintains a blacklist of all the malicious vehicles being reported by the CHs. The CH uses the *CH Vote Message* to report the malicious vehicles to the RSU. We make an implicit assumption that RSUs are interconnected through secured connections and powerful firewalls, which prevent them from being compromised. Multiple CHs are associated with a given RSU. The i^{th} RSU (RSU^i) computes the aggregated reputation of the vehicle v_m being reported by the CHs using the following rule:

$$Agg_{v_m}^{RSU^i} = \frac{\sum_{k=1}^{n'} R_{CH_k}^{RSU^i}/n'}{\sum_{j=1}^{n} R_{CH_j}^{RSU^i}/n} \tag{6}$$

where n' is the number of CHs that reported v_m as malicious and n is the total number of CHs within the radio range of RSU^i, respectively ($n' \subseteq n$). $R_{CH_j}^{RSU^i}$ is the reputation of the j^{th} CH in the RSU^i's reputation list. The reputation of the CHs are evaluated and updated using procedures described in Sections 3.2 and 3.4.4. After computing the aggregated reputation of v_m, the RSUs exchange their aggregated reputation values of v_m. Let l be the number of RSUs through which v_m has passed. The global aggregated reputation value of v_m is then calculated using the following rule:

$$Glb_{v_m}^{RSU} = \frac{\sum_{i=1}^{l} Agg_{v_m}^{RSU^i}}{l} \tag{7}$$

Finally, v_m is categorized into one of the category class based on the following rules:

$$\begin{cases} Glb_{v_m}^{RSU} \leq 0.25, & v_m \ \ is \ \ normal \\ 0.25 < Glb_{v_m}^{RSU} \leq 0.6, & v_m \ \ is \ \ suspicious \\ 0.6 < Glb_{v_m}^{RSU} \leq 1, & v_m \ \ is \ \ malicious \end{cases}$$

The overall architecture of the RSU's GDS module is shown in Fig. 6. As shown in the figure, the RSU stores the identity of suspicious and malicious vehicles in its *Blacklist* table. It periodically broadcasts the identity of these vehicles to prevent other normal vehicles in its radio range from communicating with these malicious and suspicious vehicles. All the post crash notification and congestions messages received from malicious vehicles are ignored and discarded by the normal vehicles. On the other hand, the suspicious vehicles are debarred from participating in the CH and the agent nodes election processes. Therefore, the proposed intrusion detection framework ensures that only the trustworthy vehicles in the network have the capability to act as the CH and the agent nodes. Additionally, the *Reputation List* table of the RSU receives the payment value (P_{CH}) to be made to the elected CH from the agent nodes. The RSU increments the reputation value of the elected CH in its *Reputation List* by P_{CH} value and broadcasts a message asking all the vehicles in its radio range to update the reputation value of the elected CH.

Fig. 7 shows the overall interaction between various modules of the proposed IDS framework namely, the agent node's LIDS module, the CH's CIDS module and the RSU's GDS module. As shown in the figure, the agent node's LIDS module communicates with the CH's CIDS module using the '*Agent_Vote*' and '*Verification*' messages. The CH uses these messages from the agent nodes to detect malicious vehicles in its cluster. Additionally, the CH uses a combination of specification rules and a neural network based anomaly detection module to detect malicious vehicles and agent nodes in its neighborhood. Finally, the CH's CIDS module communicates with the RSU's GDS module using '*CH Vote*' messages. The RSU uses the vote messages received from its CHs to identify the malicious vehicles and agent nodes in its radio range. These malicious vehicles and agent nodes are then included in the RSU's Blacklist table. The CH then broadcasts the identities of the malicious vehicles in its Blacklist table to prevent other normal vehicles in the network from communicating with them.

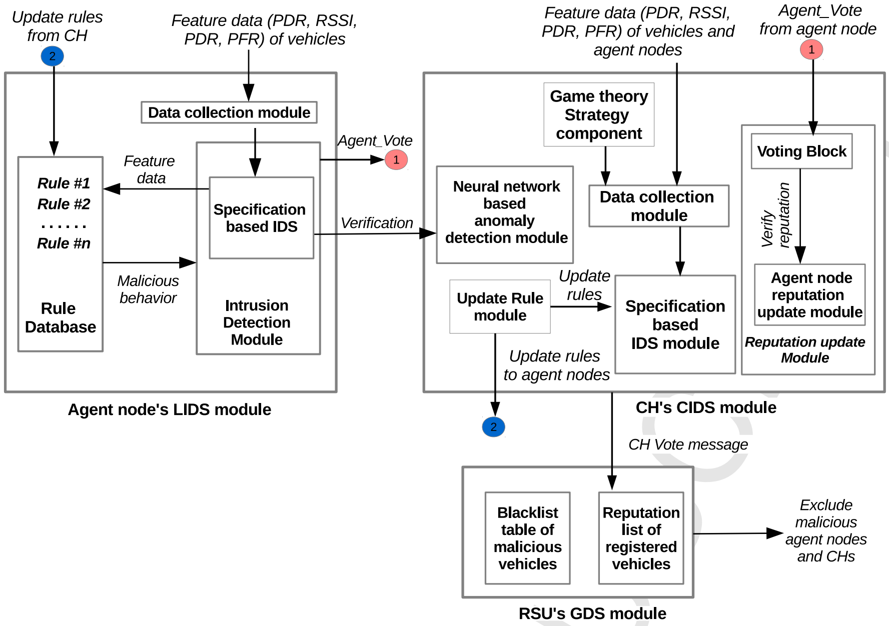

Update rules from CH
2
Feature data (PDR, RSSI, PDR, PFR) of vehicles
Data collection module
Agent_Vote
1
Feature data
Rule #1
Rule #2
......
Rule #n
Rule Database
Specification based IDS
Verification
Malicious behavior
Intrusion Detection Module
Agent node's LIDS module
Feature data (PDR, RSSI, PDR, PFR) of vehicles and agent nodes
Agent_Vote from agent node
1
Game theory Strategy component
Neural network based anomaly detection module
Data collection module
Voting Block
Verify reputation
Agent node reputation update module
Reputation update Module
Update Rule module
Update rules
Specification based IDS module
Update rules to agent nodes
2
CH's CIDS module
CH Vote message
Blacklist table of malicious vehicles
Reputation list of registered vehicles
Exclude malicious agent nodes and CHs
RSU's GDS module

Figure 7: **Overall interaction between different modules of the proposed IDS framework**

4. Experimental Results

We have classified the experimental result section into two sub-sections namely, the *simulated vehicular network traffic* and the *real time vehicular network traffic*. The experimental setup and the results obtained on the simulated and the real time vehicular network traffic are provided in the sub-sequent subsections.

Following parameters were used to analyze the performance of different IDS frameworks: 1) Detection rate 2) False alarm rate 3) IDS traffic volume and 4) Average cluster membership duration of vehicles. We define the following terminologies prior to defining the detection rate and the false alarm rate of the IDS. *True positives (TPs)*: These are cases wherein the IDS correctly identifes the the attacks. *False positives (FPs)*: These are cases in which normal data traffic is incorrectly classified as attacks by the IDS. *False negatives (FNs)*: These are cases wherein the IDS fails to detect the attacks.

- **Detection Rate (DR)**: It is defined as the ratio of the actual number of attacks detected by the IDS to the total number of attacks in the network.

$$DR = \frac{TP}{TP + FN} \tag{8}$$

- **False Alarm Rate (FAR)**: It is defined as the ratio of number of normal data incorrectly classified as attacks to the total number of attacks detected by the IDS.

$$FAR = \frac{FP}{FP + TP} \tag{9}$$

- **IDS Traffic Volume (ITV)**: It is defined as the ratio of volume of the intrusion detection related traffic to the total volume of traffic in the network (IDS and non IDS traffic) at any given instance of time.

$$ITV = \frac{IDS\ traffic}{IDS\ traffic + non - IDS\ traffic} \tag{10}$$

- **Average cluster membership duration (ACMD)**: It is defined as the average period for which the vehicle remains associated with a cluster after it has been assigned to a particular cluster by the clustering algorithm.

4.1. Simulated vehicular network traffic

To evaluate the proposed IDS framework, simulations were performed in the NS3 [34] simulator with the realistic mobility of the vehicles generated by an open-source traffic simulator, Simulation of Urban Mobility (SUMO) [35]. A coordination mechanism was built to combine the traffic simulation capabilities of SUMO with the network simulation capabilities of NS3. As shown in Fig. 8, a square grid road topology of 2000 × 2000 m consisting of a two-lane roads and four intersection points in SUMO was considered for network traffic simulation. Each grid is identified by a unique ID (G1 through G25). The vehicles were injected into the road according to a Poisson process with rate equal to two vehicles per second. The total simulation time was 500 seconds. The clustering process started at the 60^{th} second when all the vehicles had entered the road. All the performance metrics were evaluated for the remaining 440 seconds. Two classes of vehicles with different maximum speed ranges were used in the simulation to create a realistic scenario with different types of vehicles on the road, such as passenger cars, buses, and trucks. The first class of vehicles had a maximum speed of 10 m/s, whereas the maximum speed of the second class of vehicles were varied between 10 m/s to 35 m/s.

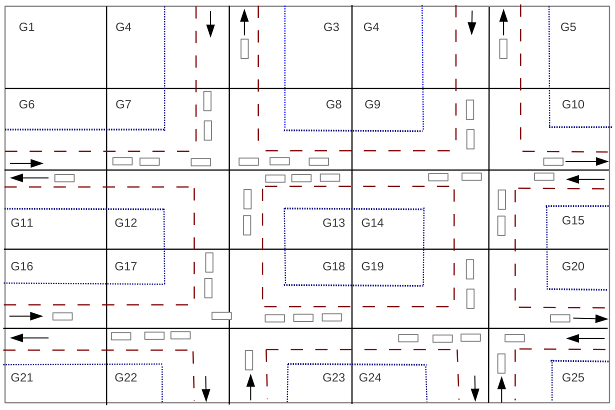


Figure 8: **Simulation traffic scenario**

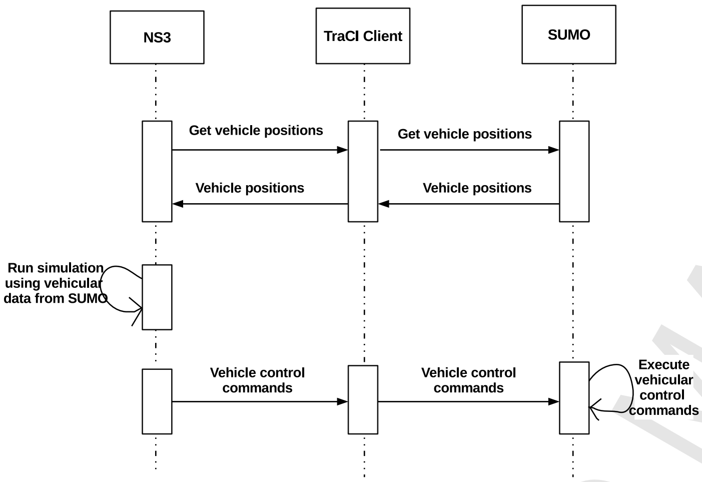


Figure 9: **Interaction between NS3 and SUMO via TraCI client**

Table 2: Simulation Parameters

Simulation Time	500 s
Simulation Area	$5 \times 5km^2$
Mobility	Car-following model
Propagation Model	Two-Ray Ground
No. of vehicle per cluster	15-20
No. of IDS agents per cluster	20-30%
Protocol Stack	IEEE 1609 WAVE [36]
Routing Protocol	AODV
Radio range	200m
$T_{pdr_{sf}}$	60-65 %
$T_{pdr_{bh}}$	90-95 %
$T_{pdr_{wh}}$	80-85 %
$T_{rssi_{syb}}$	-40 - (-45) dBm
$T_{rssi_{bh}}$	-35 - (-40) dBm
$T_{rssi_{wh}}$	-50 - (-55) dBm %
$T_{dpr_{dos}}$	80-85 %
$T_{pfr_{dos}}$	90-95 %
C_m, C_a	0.15
Transmit power	30 dBm
CH's DR (α)	0.956
CH's FP rate (β)	0.085

We have used the IEEE 1609 Wireless Access in Vehicular Environments (WAVE) protocol stack [36] that builds on IEEE 802.11p WLAN standard and operates on seven reserved channels in the 5.9 GHz frequency band for our analysis. The vehicles use 802.11p WiFi with continuous access to a 10 MHz Control Channel (CCH) to transmit 300 byte safety message 12 times per second at 3 Mbps using WAVE Share Message Protocol (WSMP) packets. In addition, all vehicles attempt to randomly send 256 byte IP packets at an application rate of 6 Mbps using the Service Channels (SCHs) channels. Our measurements are based on averaging the results obtained from 10 simulations. We vary the number of malicious vehicles to be between 10% to 30% of the overall vehicles in the network. The key parameters used for simulation are shown in Table 2. PDR, PFR, DPR and RSSI values are calculated every 10 seconds. The set of specification rules used by the LIDS and the CIDS modules are updated every 50 and 30 seconds, respectively.

To make the SUMO and NS3 work together and to change traffic lights dynamically, a client was introduced. In order to get the meaningful data, SUMO generated the realistic road traffic with different kinds of vehicles and intelligent traffic lights. SUMO and NS3 worked in parallel using Traffic Control Interface (TraCI) client, which is a generic interface that interlinks the road traffic in SUMO with network simulation of NS3. TraCI client made it possible to control a running road traffic simulation in SUMO through commands from NS3. TraCI uses a TCP-based client/server architecture, wherein SUMO acts as a server and the external NS3 script (the "controller") acts as a client. It helps simulate the real streets designed with lanes, traffic lights, turns and other traffic entities. When any application in NS3 wanted to change vehicles' state in SUMO, it sent a message to the Traci client interface, which in turn generated commands according to applications and then sent them to SUMO for execution followed by the retrieval of data back from the SUMO. Fig. 9 shows the interaction between NS3 and SUMO via the TraCI client.

Both the CH and the malicious vehicle adopt the game theory based strategies discussed in Section 3.4.5 to maximize their overall payoff utilities. Fig. 10 shows the payoff utilities of the CH and the malicious vehicle under the Nash Equilibrium (NE) and the non NE strategies. The payoff utilities are calculated every three seconds into the simulation. It can be observed from the figure that if the player (CH or malicious vehicle) deviates from its NE strategy, while the opponent player continues to play the NE strategy then the payoff utility of the deviating player decreases. Therefore, the players do not have any profitable incentive to deviate from their NE strategy.

The performance of the proposed framework was evaluated against the frameworks proposed in [7], [24] and [37]. The reason for choosing these frameworks for comparison is because of the similarity of the attack types considered in these frameworks with the attack types considered in the proposed IDS framework. It allows us to analyze and compare the per-

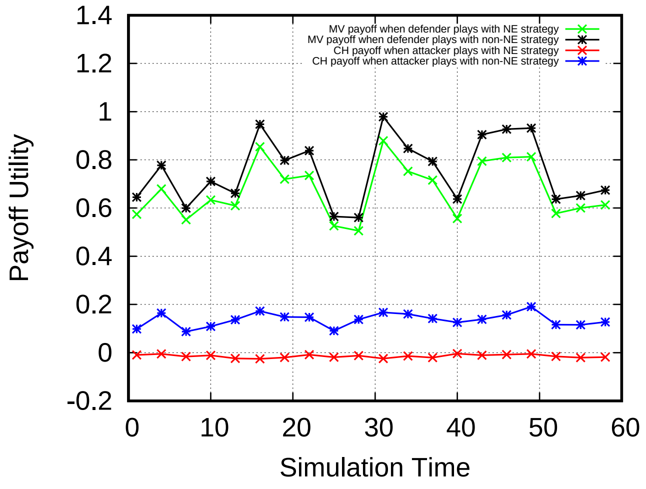


Figure 10: **Payoff utility of the CH and the Malicious Vehicle (MV) under NE and non-NE strategies**

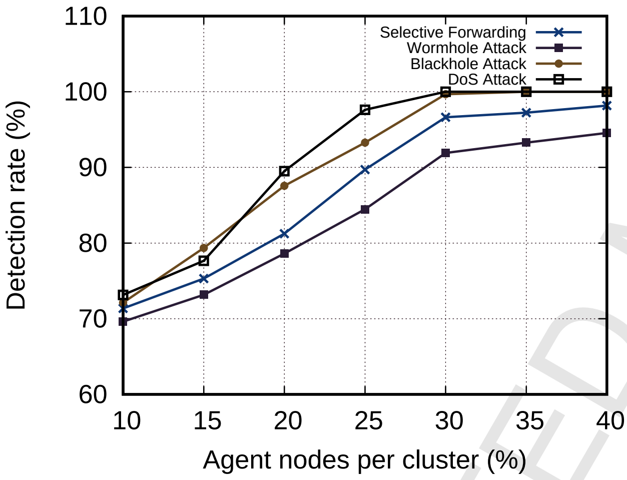


Figure 11: **Detection rate of the proposed framework with varying percentage of agent nodes**

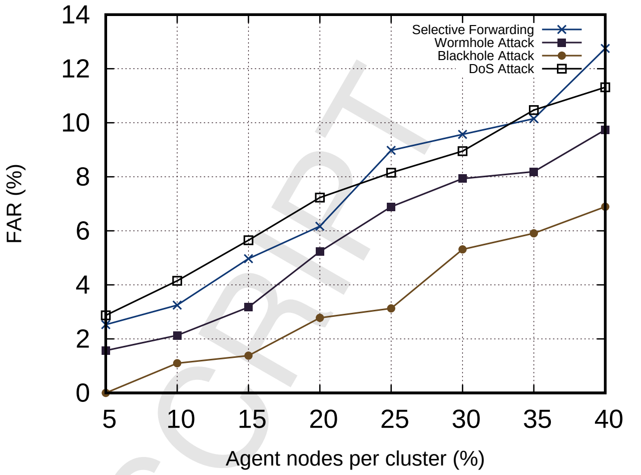


Figure 12: **False alarm rate of the proposed framework with varying percentage of agent nodes**

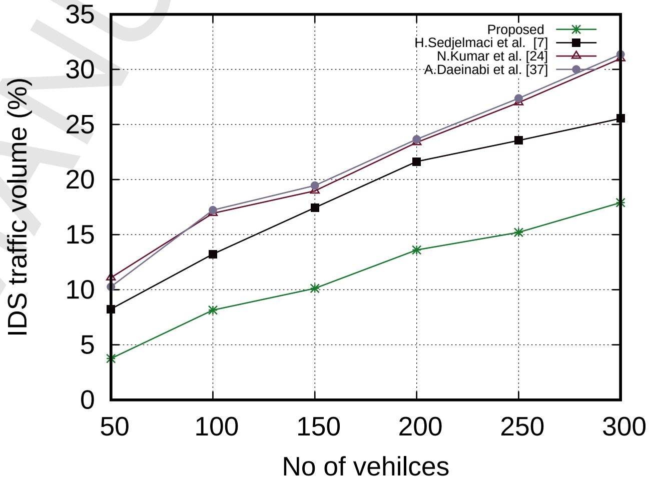


Figure 13: **Volume of IDS traffic generated by different frameworks**

formance of the proposed framework with other IDS frameworks against similar type of attacks. In addition, the framework described in [38] also proposes a clustering algorithm for VANETs, which allows us to compare it with the proposed clustering algorithm. All these factors make them ideal candidates for comparison with the proposed framework.

Figure 11 shows the *DR* of the proposed framework against different types of attacks. It can be observed from the figure that the *DR* of the framework increases with the increase in the number of agent nodes per cluster. This can be attributed to the fact that as the number of agent nodes increase, more number of malicious vehicles are detected and reported to the CH. Fig. 12 shows the *FAR* rate of the proposed framework against various type of attacks. It can be observed from the figure that the *FAR* of the proposed framework increases with the increase in the number of agent nodes. This is because as the number of agent nodes increase, some of the malicious vehicles get elected as the agent nodes, which in turn provide false reports to the CHs.

Therefore, it can be deduced from Figures 11 and 12 that the best trade-off between high *DR* and low *FAR* is obtained when 25% to 30% of the vehicles in the cluster are elected as the agent nodes.

Fig. 13 shows the volume of IDS traffic introduced into the vehicular network by various IDS frameworks ([7], [24] and [37]). It can be observed from the figure that the volume of IDS traffic increases with the increase in vehicle density for all the frameworks. However, the proposed framework introduces the least volume of IDS traffic compared to other frameworks. This can be attributed to the fact the proposed framework minimizes the amount of information exchanged between the agent node's LIDS module and the CH's CIDS module by electing optimum number of agent nodes for performing the monitoring task. In addition, the CH's CIDS module employs a game theory based probabilistic monitoring strategy, which further reduces the volume of IDS traffic. On the other hand, the frameworks pro-

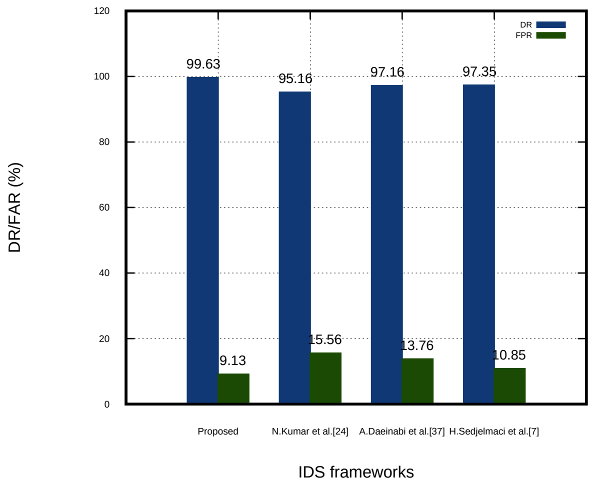


Figure 14: **Detection rate and false alarm rate of different frameworks**

posed in [7] [24] [37] require all the vehicles in the network to continuously perform the monitoring operation. This results in introduction of high volume of IDS traffic into the vehicular network, as more number of vehicles join the network.

Fig. 14 shows the *DR* and the *FAR* of various IDS frameworks against the black hole, worm hole, selective forwarding, DoS and sybil attacks. It can be observed from the figure that the proposed framework achieves the highest *DR* and lowest *FAR* amongst all the frameworks against these attacks. The high *DR* of the proposed framework can be attributed to the fact that it uses a combination of specification rules and a lightweight neural network based classifier model to detect malicious vehicles, which greatly enhances its detection capabilities. On the other hand, the proposed framework minimizes the *FAR* by electing an appropriate number of agent nodes for performing the monitoring operation.

Fig. 15 shows the average cluster membership duration of vehicles for various IDS frameworks. It can be observed from the figure that the proposed framework provides the highest cluster stability amongst all the frameworks by providing high cluster membership duration to vehicles in its clusters. Its performance is comparable to that of the framework proposed in [37], since both the frameworks use novel clustering algorithms to enhance the stability of the clusters and reduce the frequency of cluster formation process. On the other hand, the average cluster membership duration of the vehicles in [7] [24] are small, even at low vehicular densities, since they do not implement any mechanism to enhance the cluster stability. As a result the vehicular clusters in these frameworks are unstable.

4.2. *Real time vehicular network traffic*

In this subsection, we analyze the effectiveness of the proposed IDS framework on the real time road network of the German city Eichstätt obtained using the OpenStreetMap [38]. The road network was imported from the OpenStreetMap (OSM) to SUMO using an application called the NETCONVERT [35].

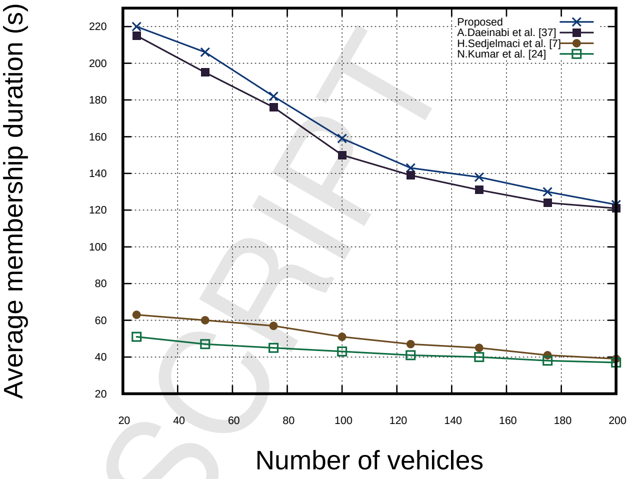


Figure 15: **Average cluster membership duration of vehicles for various frameworks**

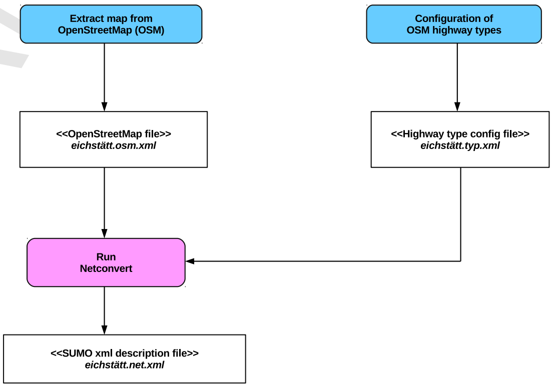

Extract map from OpenStreetMap (OSM)
Configuration of OSM highway types
<<OpenStreetMap file>> eichstätt.osm.xml
<<Highway type config file>> eichstätt.typ.xml
Run Netconvert
<<SUMO xml description file>> eichstätt.net.xml

Figure 16: **Overview of the steps involved in importing the traffic map of the German city Eichstätt from the OpenStreetMap into SUMO**

Figure 17: **Map of German city Eichstätt obtained from OpenStreetMap**

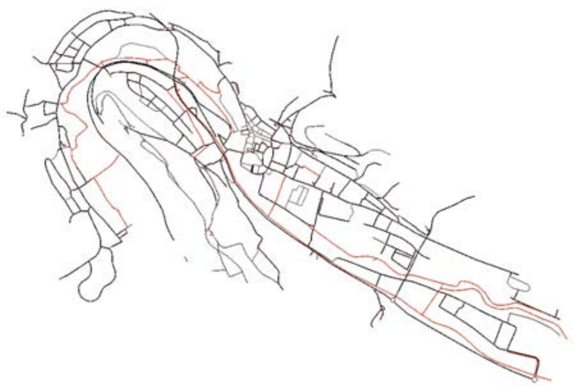


Figure 18: **SUMO network file corresponding to OpenStreetMap map of German city Eichstätt**

Figure 16 shows the procedure involved in obtaining the road network of German city Eichstätt from the OSM into SUMO using NETCONVERT. The imported SUMO road network file was provided with default values of the road attributes like, speed limit, number of lanes, priority, one-way street and allowed vehicle classes depending on the highway types, using *SUMO edge type files* described in (http://sumo.dlr.de/wiki/SUMO_edge_type_file). Figures 17 and 18 show the OSM file and the corresponding SUMO network file of the Eichstätt city. Several types of the vehicles (cars, buses and emergency vehicles) with different priorities and maximum speeds were simulated in the road traffic. Different vehicles routes were set in the road traffic in SUMO. The total simulation time was 300 seconds. The results were obtained by averaging the output of 10 round of simulations.

Fig. 19 shows the volume of IDS traffic generated by different IDS frameworks on the Eichstätt road traffic network. It can be observed from the figure that the proposed framework introduces the least volume of the IDS traffic compared to other frameworks. Fig. 20 shows the *DR* and the *FAR* of various IDS frameworks on the Eichstätt road network traffic data against the black hole, worm hole, selective forwarding, DoS and sybil attacks. Again it can be observed that the proposed framework achieves the highest *DR* with least *FAR* amongst all the frameworks. These results vindicate that the proposed IDS framework significantly reduces the volume of IDS traffic in the vehicular network, while at the same time maintains a high *DR* and low *FAR*.

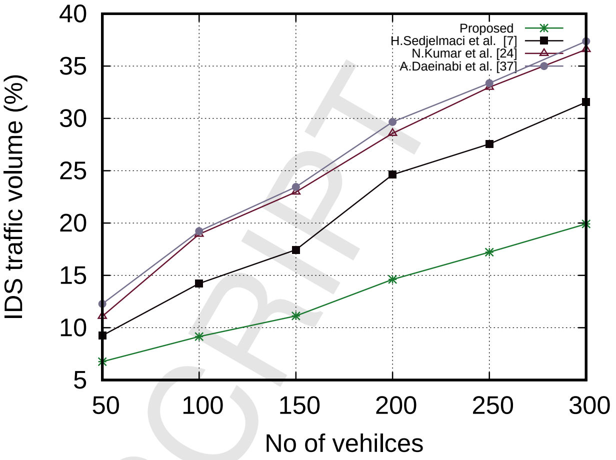


Figure 19: **Volume of IDS traffic generated by different frameworks on the Eichstätt road traffic network**

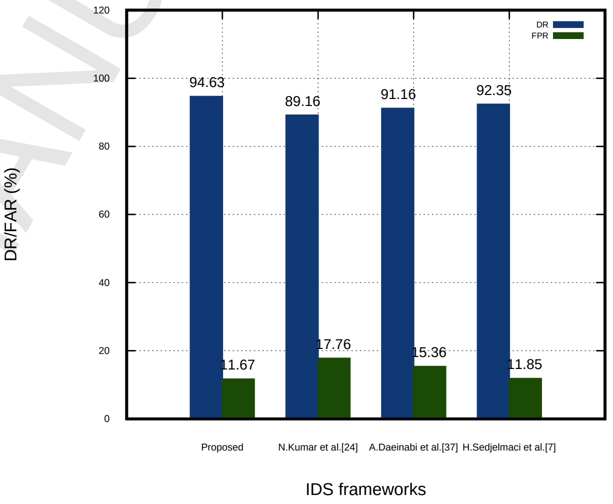


Figure 20: **Detection rate and false alarm rate of different frameworks on the Eichstätt road traffic network**

5. Conclusion and future work

In this paper, a novel clustering algorithm, a CH election algorithm and a game theory based IDS framework for VANETs have been proposed. The proposed clustering algorithm ensures the stability of the IDS framework by generating stable vehicular clusters with enhanced connectivity among member vehicles. CH and agent nodes election algorithms are then executed to elect the CH and a set of agent nodes for each cluster. The proposed IDS framework uses the the agent nodes, the CHs and the RSUs operating at three different levels of the vehicular network to carry out the intrusion detection operation in a distributed manner. The framework uses a set of specification rules based on the Packet Drop Rate (PDR), Packet Forwarding Rate (PFR), Receive Signal Strength Indicator (RSSI) and Duplicate Packet Rate (DPR) values of the vehicles, along with a lightweight neural network based classifier module for detecting malicious vehicles. In addition, the proposed framework minimizes the volume of IDS traffic introduced into the vehicular network by modeling the interaction between the IDS and the malicious vehicle as a two player non-cooperative game, and by adopting a probabilistic IDS monitoring strategy based on the Nash Equilibrium of the game.

The clustering algorithm proposed in the paper only considers the vehicles that stopped at the traffic signal and vehicles approaching the road intersection point with relatively low speed. For our future work, we envisage to formulate a dynamic clustering algorithm, which takes into account the vehicles in motion for generating stable vehicular cluster. We also aim to improve the *DR* and minimize the *FAR* rate of the proposed IDS framework by analyzing the performances of various other classifiers using Support Vector Machine (SVM), Decision Tree, Logistic Regression, Multilayer Perceptron (MLP) etc. Additionally, we also aim to extend and implement the proposed IDS framework to various other networks like Software Defined Network (SDN), Delay Tolerant Network etc., in future.

References

[1] Y. Toor, P. Muhlethaler, A. Laouiti, Vehicle Ad Hoc Networks: Applications and Related Technical Issues, IEEE Communications Surveys & Tutorials 10 (3) (2008) 74–88.

[2] D. Jiang, L. Delgrossi, Ieee 802.11p: Towards an international standard for wireless access in vehicular environments, in: VTC Spring 2008 - IEEE Vehicular Technology Conference, 2008, pp. 2036–2040.

[3] M. N. Mejri, J. Ben-Othman, M. Hamdi, Survey on VANET security challenges and possible cryptographic solutions, Vehicular Communications 1 (2) (2014) 53 – 66.

[4] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, J. P. Hubaux, Eviction of Misbehaving and Faulty Nodes in Vehicular Networks, IEEE Journal on Selected Areas in Communications 25 (8) (2007) 1557–1568.

[5] S. Gillani, F. Shahzad, A. Qayyum, R. Mehmood, A Survey on Security in Vehicular Ad Hoc Networks, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 59–74.

[6] P. Papadimitratos, V. Gligor, J.-P. Hubaux, Securing Vehicular Communications - Assumptions,Requirements, and Principles, in: Workshop on Embedded Security in Cars (ESCAR), 2006, pp. 5–14.

[7] H. Sedjelmaci, S. M. Senouci, An accurate and efficient collaborative intrusion detection framework to secure vehicular networks, Computers & Electrical Engineering 43 (2015) 33 – 47.

[8] S. Huda, J. Abawajy, M. Alazab, M. Abdollalihian, R. Islam, J. Yearwood, Hybrids of support vector machine wrapper and filter based framework for malware detection, Future Generation Computer Systems 55 (2016) 376 – 390.

[9] F. A. Khan, M. Imran, H. Abbas, M. H. Durad, A detection and prevention system against collaborative attacks in mobile ad hoc networks, Future Generation Computer Systems 68 (2017) 416 – 427.

[10] Q. Guo, X. Li, G. Xu, Z. Feng, MP-MID: Multi-Protocol Oriented Middleware-level Intrusion Detection method for wireless sensor networks, Future Generation Computer Systems 70 (2017) 42 – 47.

[11] Y. Xie, D. Feng, Z. Tan, J. Zhou, Unifying intrusion detection and forensic analysis via provenance awareness, Future Generation Computer Systems 61 (2016) 26 – 36.

[12] Y. Cho, G. Qu, Y. Wu, Insider Threats against Trust Mechanism with Watchdog and Defending Approaches in Wireless Sensor Networks, in: IEEE Symposium on Security and Privacy Workshops, 2012, pp. 134–141.

[13] F. Liu, X. Cheng, D. Chen, Insider Attacker Detection in Wireless Sensor Networks, in: IEEE INFOCOM - 26th IEEE International Conference on Computer Communications, 2007, pp. 1937–1945.

[14] H. Ehsan, F. A. Khan, Malicious AODV: Implementation and Analysis of Routing Attacks in MANETs, in: IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, 2012, pp. 1181–1187.

[15] K. R. Abirami, M. G. Sumithra, J. Rajasekaran, An enhanced intrusion detection system for routing attacks in MANET, in: International Conference on Advanced Computing and Communication Systems, 2013, pp. 1–6.

[16] S. Dietzel, J. Grtler, R. van der Heijden, F. Kargl, Redundancy-based statistical analysis for insider attack detection in VANET aggregation schemes, in: 2014 IEEE Vehicular Networking Conference, 2014, pp. 135–142.

[17] S. Dietzel, J. Petit, G. Heijenk, F. Kargl, Graph-Based Metrics for Insider Attack Detection in VANET Multihop Data Dissemination Protocols, IEEE Transactions on Vehicular Technology 62 (4) (2013) 1505–1518.

[18] J. M. de Fuentes, L. Gonzlez-Manzano, A. I. Gonzlez-Tablas, J. Blasco, Security Models in Vehicular Ad-hoc Networks: A Survey, IETE Technical Review 31 (1) (2014) 47–64.

[19] E. S. Coronado, S. Cherkaoui, An AAA Study for Service Provisioning in Vehicul Networks, in: Proceedings of 32nd Annual IEEE Conference on Local Computer Networks (LCN 2007), 15-18 October 2007, Clontarf Castle, Dublin, Ireland,, 2007, pp. 669–676.

[20] H. Moustafa, G. Bourdon, Y. Gourhant, AAA in Vehicular Communication on Highways with Ad Hoc Networking Support: A Proposed Architecture, in: Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks, ACM, New York, NY, USA, 2005, pp. 79–80.

[21] H. Zhu, R. Lu, X. Shen, X. Lin, Security in Service-oriented Vehicular Networks, Wireless Communications 16 (4) (2009) 16–22.

[22] S. M. Safi, A. Movaghar, M. Mohammadizadeh, A Novel Approach for Avoiding Wormhole Attacks in VANET, in: Proceedings of the 2009 Second International Workshop on Computer Science and Engineering - Volume 02, IEEE Computer Society, 2009, pp. 160–165.

[23] A. Rehan, K. Turgay, HEAP: A packet authentication scheme for mobile ad hoc networks, Ad Hoc Networks 6 (7) (2008) 1134 – 1150.

[24] N. Kumar, N. Chilamkurti, Collaborative trust aware intelligent intrusion detection in VANETs, Computers & Electrical Engineering 40 (6) (2014) 1981 – 1996.

[25] A. Tomandl, K. P. Fuchs, H. Federrath, REST-Net: A dynamic rule-based IDS for VANETs, in: 7th IFIP Wireless and Mobile Networking Conference (WMNC), 2014, pp. 1–8.

[26] H. Sedjelmaci and 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 1 (6) (2014) 570–577.

[27] Y. Kim, I. Kim, C. Y. Shim, A taxonomy for DOS attacks in VANET, in: 14th International Symposium on Communications and Information Technologies (ISCIT), 2014, pp. 26–27.

[28] Y. Guo, S. Schildt, L. Wolf, Detecting blackhole and greyhole attacks in vehicular Delay Tolerant Networks, in: Fifth International Conference on Communication Systems and Networks (COMSNETS), 2013, pp. 1–7.

[29] M. Dighriri, A. S. D. Alfoudi, G. M. Lee, T. Baker, R. Pereira, Comparison Data Traffic Scheduling Techniques for Classifying QoS over 5G Mobile Networks, in: 2017 31st International Conference on Advanced Information Networking and Applications Workshops, 2017, pp. 492–497.

[30] M. Dighriri, G. M. Lee, T. Baker, Measurement and Classification of Smart Systems Data Traffic Over 5G Mobile Networks, Springer International Publishing, 2018, pp. 195–217.

[31] M. Dighriri, A. S. D. Alfoudi, G. M. Lee, T. Baker, Data Traffic Model in Machine to Machine Communications over 5G Network Slicing, in: 2016 9th International Conference on Developments in eSystems Engineering, 2016, pp. 239–244.

[32] N. Mohammed, H. Otrok, L. Wang, M. Debbabi, P. Bhattacharya, Mechanism design-based secure leader election model for MANET, IEEE Transactions on Dependable and Secure Computing 8 (1) (2011) 89–103.

[33] B. Subba, S. Biswas, S. Karmakar, Intrusion detection in mobile ad-hoc networks: Bayesian game formulation, Engineering Science and Technology, an International Journal 19 (2) (2016) 782 – 799.

[34] G. F. Riley, T. R. Henderson, The NS-3 Network Simulator Modeling and Tools for Network Simulation, in: Modeling and Tools for Network Simulation, 2010, pp. 15–34.

[35] K. Daniel, E. Jakob, B. Michael, B. Laura, Recent Development and Applications of SUMO- Simulation of Urban Mobility, International Journal On Advances in Systems and Measurements 5 (3&4) (2012) 128–138.

[36] R. A. Uzcátegui, G. Acosta-Marum, Wave: A tutorial, IEEE Communications Magazine 47 (5) (2009) 126–133.

[37] A. Daeinabi, A. G. P. Rahbar, A. Khademzadeh, VWCA: An efficient clustering algorithm in vehicular ad hoc networks, Journal of Network and Computer Applications 34 (1) (2011) 207 – 222.

[38] M. M. Haklay, P. Weber, OpenStreetMap: User-Generated Street Maps, IEEE Pervasive Computing 7 (4) (2008) 12–18.

Basant Subba is a Ph.D research scholar at the Indian Institute of Technology, Guwahati. He received his bachelors in Engineering (BE) degree from Visvesvaraya Technological University, Belgaum, India in 2009 and Masters degree (M.Tech) from National Institute of Technology, Durgapur, India in 2012. His research interests are designing game theory based intrusion detection frameworks for wired networks, Mobile Ad-hoc Networks (MANETs), Vehicular Ad-hoc Networks (VANETs) and Wireless Sensor Networks (WSNs).

Santosh Biswas received B.E degree from NIT, Durgapur, India, in 2001. He completed his M.S. and Ph.D from IIT Kharagpur, India, in the year of 2004 and 2008, respectively. He works as an Associate Professor at the Department of Computer Science and Engineering, IIT Guwahati. His research interests include network security, VLSI testing and discrete event systems.

Sushanta Karmakar received his B.E and M.E degrees from Jadavpur University, India, in 2001 and 2004, respectively. He obtained his Ph.D from IIT Kharagpur, India, in the year 2009. He works as an Associate Professor at the Department of Computer Science and Engineering, IIT Guwahati. His research interest include Distributed algorithms, fault-tolerance, distributed algorithms for ad hoc and sensor networks

Basant Subba

Santosh Biswas

Sushanta Karmakar

A game theory based multi layered intrusion detection framework for VANET

Highlights

- A multi-layered game theory based VANET intrusion detection framework is proposed.
- A novel clustering and CH election algorithm for VANET is proposed.
- A payment structure based on VCG mechanism is proposed for the CH election.
- Proposed framework achieves high detection rate and accuracy across wide range of attacks.