International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2022/212336 Volume 9, Issue 2, March – April (2022)

RESEARCH ARTICLE

Analysis of Machine Learning Classifiers to Detect Malicious Node in Vehicular Cloud Computing

A. Sheela Rini
Department of Computer Science, Avinashilingam Institute for Home Science & Higher Education for Women,
Coimbatore, Tamil Nadu, India
sheelarini.a@gmail.com

C. Meena
Department of Computer Science, Avinashilingam Institute for Home Science & Higher Education for Women,
Coimbatore, Tamil Nadu, India
cccmeena@gmail.com

Received: 31 January 2022 / Revised: 26 March 2022 / Accepted: 12 April 2022 / Published: 30 April 2022

Abstract – VANET or Vehicular networks are created using the principles of MANETS and are used by intelligent transport systems to offer efficient communication between the domains of vehicles. Increasing the number of vehicles requires communication between vehicles to be fast and secure, where cloud computing with VANET is more prominent. To provide a secure VANET communication environment, this paper proposes a malicious or hacked vehicle identification system. Malicious vehicles are identified using four steps. The first step uses a clustering algorithm for similar group vehicles. In the Second step, cluster heads are identified and elected. In the next step, Multiple Point Relays are selected. Finally, classifiers are used to identify hacked vehicles. However, the existing system performance degrades as soon as the number of vehicles increases, resulting in increased cost during Cluster head election, inability to produce stable clusters, and the need for accurate and fast classification in high traffic scenarios. This work improves clustering algorithms and examines several classification algorithms to solve these issues. The classifiers analyzed are Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbour (KNN) and Naïve-Bayes (NB). A Hybrid classifier that combines SVM and KNN classifiers is also analyzed for its effectiveness to detect malicious vehicles. From the experimental results, it could be observed that the detection accuracy is high while using the hybrid classifier.

Index Terms – VANET, Malicious Node, SVM, Decision Tree, Naïve-Bayes, KNN.

1. INTRODUCTION

The increased number of vehicles on the roads requires effective methods to improve road safety, driver & passenger comfort and efficient transportation. One breakthrough in road safety technology is real-time communication using VANET (Vehicular Ad hoc NETwork), where vehicles exchange information [1,2]. It is a part of an intelligent transport system and combines the concepts of MANETs with cloud computing to address various issues involving vehicular applications involving safety and comfort application [3].

During communication, VANETs face several challenges related to low process capability, mainly due to limited resource availability like memory, computation power, and bandwidth. However, the increase in the number of vehicles demands a communication environment that can handle fast and secure message sharing and high storage capacity[4]. This necessity can be handled efficiently using Vehicular cloud computing or VCC, which combines VANETs with cloud computing [5]. The VCC uses the cloud's advantages that allow sharing of available resources among the neighbouring vehicles within a transmission range. The basic components of VCC include RSUs (Road-Side Units), moving vehicles, computer-controlled devices, radio transceivers for exchanging messages, sensors for sensing surrounding environments, GPS (Global Positioning System) and cloud servers. VCC provides various advantages like driver/traveller safety efficiency in managing safety and comfort applications.

The VCC has three types of communication models: the cluster-based model, Road Side Unit model and Vehicle-to-Vehicle (V2V) model. The scope of this work is a cluster-based communication model, as this is the latest and most advanced architecture used. Cluster-based communication groups similar nodes based on different characteristics of VCCs. Examples of such characteristics include velocity, direction and density. In each cluster one vehicle is designated as Cluster Head (CH), and the rest are so-called Cluster Members (CMs). This model's working is very similar to the client/server architecture, where the CMs act as the clients and the CHs act as the server. All the CHs are interconnected with one another, and the communication only occurs through CHs. The cluster-based communication model in VCC offers several advantages to improve road safety and traffic. It helps to improve real-time sharing of traffic and road conditions efficiently. One important factor that must be considered during such real-time communication is the safety measure that must be implemented to consider all aspects of security during message transmission.

The Ad-Hoc Routing Protocols transmits the data between the nodes by using the link information in the VANET. It is broadly classified into three types as Table-driven or Proactive Protocols, Source on demand or Reactive Protocols and Hybrid Routing Protocols. Proactive protocols are generally based on algorithms associated with shortest routes, saves all the data about the connected nodes in predefined tables. If the network topology changes, then the same is updated in the routing table by the concerned node. Since the control messages are often sent to all the nodes in the network, it present high control message overload. Reactive Protocols do not update its routing table often and it will find the route only when a node wants to send the messages to some other node by using the flooding technique which leads to the path discovery and minimise the traffic in the network.

1.1. Problem Statement

A hacker/attacker compromises vehicular Cloud Computing communication is compromised by a hacker/attacker who aims to corrupt the message communication. A hacked node may tamper with information sent by creating bogus/false alerts or suppressing legitimate messages. Both types of message tampering are serious and hence, it is important to detect CHs, corrupted by hackers to ensure driver/vehicle privacy and welfare.

1.2. Motivation

Active research is being conducted to improve the identification of such corrupted malicious CHs. The various methods proposed can be grouped into two categories, namely, credit-based methods and reputation-based methods [6]. In credit-based approaches, the nodes need to pay to get served and get paid to serve the further nodes. On the other hand, reputation-based approaches monitor the nodes and propagate the mischievous ones to separate them. This work proposes a hybrid clustering and classification algorithm to improve the performance when the number of nodes increases and provide secure communication among the nodes, i.e. both V2V and V2I.

1.3. Objective

This research focuses on proposing reputation-based approaches, where the usage of machine learning algorithms is more prominent. VANET-QoS-OLSR[7] is proposed to detect malicious CHs. This work improved the conventional QoS-OLSR to work in VANET scenarios. This work is further modified to work with VCC, explores the applicability of several classifiers, and analyses their performance on detecting malicious CHs.

1.4. Organization of the Paper

The current section has briefed the introduction of VCC, problem statement, motivation and objective of the research. State-of-the-art is discussed in Section 2. Section 3 provides the methodology behind the design of a malicious CH detection system. Four different classifiers, namely, Support Vector Machine (SVM), Decision Trees (DT), Naïve Bayes (NB) and K Nearest Neighbour (KNN) Classifiers, are analyzed. A hybrid classifier that combines SVM and KNN is also proposed based on the results. Section 4 analyses the performance of the existing classifiers and proposes an efficient hybrid classifier to detect normal and malicious CHs. Section 5 concludes the work with future research directions.

2. RELATED WORK

Because of significant study and technology improvement in wireless communication, conventional ITs have to bother with interaction among vehicles [8]. In recent times, the quantities of automobiles have increased due to transferring enormous amounts of persons from one area to another area. This results with the growth in the amount of vehicles which results in heavy traffic and road accidents [9]. Nowadays, this is a primary complication in our lives. The common term to denote Vehicular networking is VANET [10]. VANET comprises of Vehicle-to-Roadside (V2R) and Vehicle-to-Vehicle (V2V) [11]. Interactions to handover the data of automobiles. The communication in VANETS is determined by the Road Side Units (RSU) to assist Wireless Access in Vehicular Environments (WAVE). Also the Roadside Units (RSU) turns as wireless access points' that supports interaction among the automobiles within its boundary [12]. The integrated vehicular system design, related to the mobile interaction designs, will activate the facilities offered by mobile communication.

Due to this there is a need to unite vehicular networks with the data centres and there is a need to interchange data, IoV permits to access the Internet between on-road automobiles. The notable IoV application is to increase the attributes of VANETs to decrease numerous disputes in metropolitan transportation and accident situations [13].

Trust-Aware Support Vector Machine based Intrusion Detection System (TS-IDS) proposed by [14] utilizes the adjusted unbridled approach with SVM to develop an exact trust esteem table for interruption location and its counteraction in VANET.

IoV permits the vehicular road links to interrelate using wireless network tools, such as Wi-Fi and 4G/LTE for V2I, IEEE WAVE for V2V and V2R, MOST/Wi-Fi for V2S. It is beneficial to offer a wide-ranging demonstration to ML models in IoV and elucidate the regions that could add to these networks' improvement [15]. The rising requirements to familiarize AI concepts in IoV applications are experiencing certain challenges. These experiments are connected to building specific conclusions and estimating various characteristics of IoV, like monitoring and controlling transportation, High dimensional data handling, energy and supply controlling, and intellectual communication with users to offer first-class facilities. Some researches were accompanied on AI mechanisms like machine learning to improve resolutions to most of these trials [16].

AI is highly associated with the layer answerable to demonstrate its characteristics in IoV layered design. A word called "simulated cloud structure" can designate this layer and it is accountable for saving, handling, investigating the data obtained from the IoV set-up. A proper conclusion could be attained on such investigations. In IoV, the calculation and study are delivered through Big Data Analysis (BDA) and Vehicular Cloud Computing (VCC) [17] schemes that are considered to be data handling centres. As per the applications of IoV, several facilities can be offered by the IoT cloud set up, necessitating intellectual service administration. Smart servers of the cloud could offer several smart facilities; some of them are security, transportation management, which are the basis of sophistication in IoV.

Multi-cluster head IDS way [18] is proposed to deal with recognizing vindictive hubs. According to this methodology, first and foremost, many vehicles are shaped, and a group head is allocated. Then, at that point, the enhanced SVM calculation is conveyed on the bunch head to identify vindictive hubs from the particular bunch. After eliminating vindictive hubs from the bunch, other group heads are chosen utilizing Hybrid Fuzzy Multi-Criteria Decision Making (HF-MCDM) strategy. The cloud servers grounded on AI allow the process and progress of AI in Real-Time (RT) high dimensional data traffic to offer an intelligent result for intellectual customer facilities. The Vehicular Cyber-Physical System (VCPS) [19] is accepted as a vehicular link pattern that addresses data distribution through the next-gen Internet. AI influence VCPS to offer smart handling in high dimensional data traffic by exploiting fog and cloud computing for secured presentations.

In IoV systems, edge computing and data storage difficulties are important deliberated experiments demanding a clever optimization technique [20]. These trials are associated with many features, like the stability of a channel, vibrant communication topology and distribution of resources. AI in IoV offers an intellectual method to crack most of these tests. Exploiting ML offers an interaction platform to the IoV situation & facilitates the establishment of an intelligent agent which acquires stimulating features to improve the whole IoV network exploitation [21]. Q-learning and deep neural networks are ML procedures to derive conclusions as per IoV resource activities. In the IoV set-up design, the demonstration of AI in a detached level is accountable for simulated cloud structure. AI layer turns as a data controlling parameter [22]. The AI layer in IoT design comprises voluminous analysis of data, cloud computing, and smart systems.

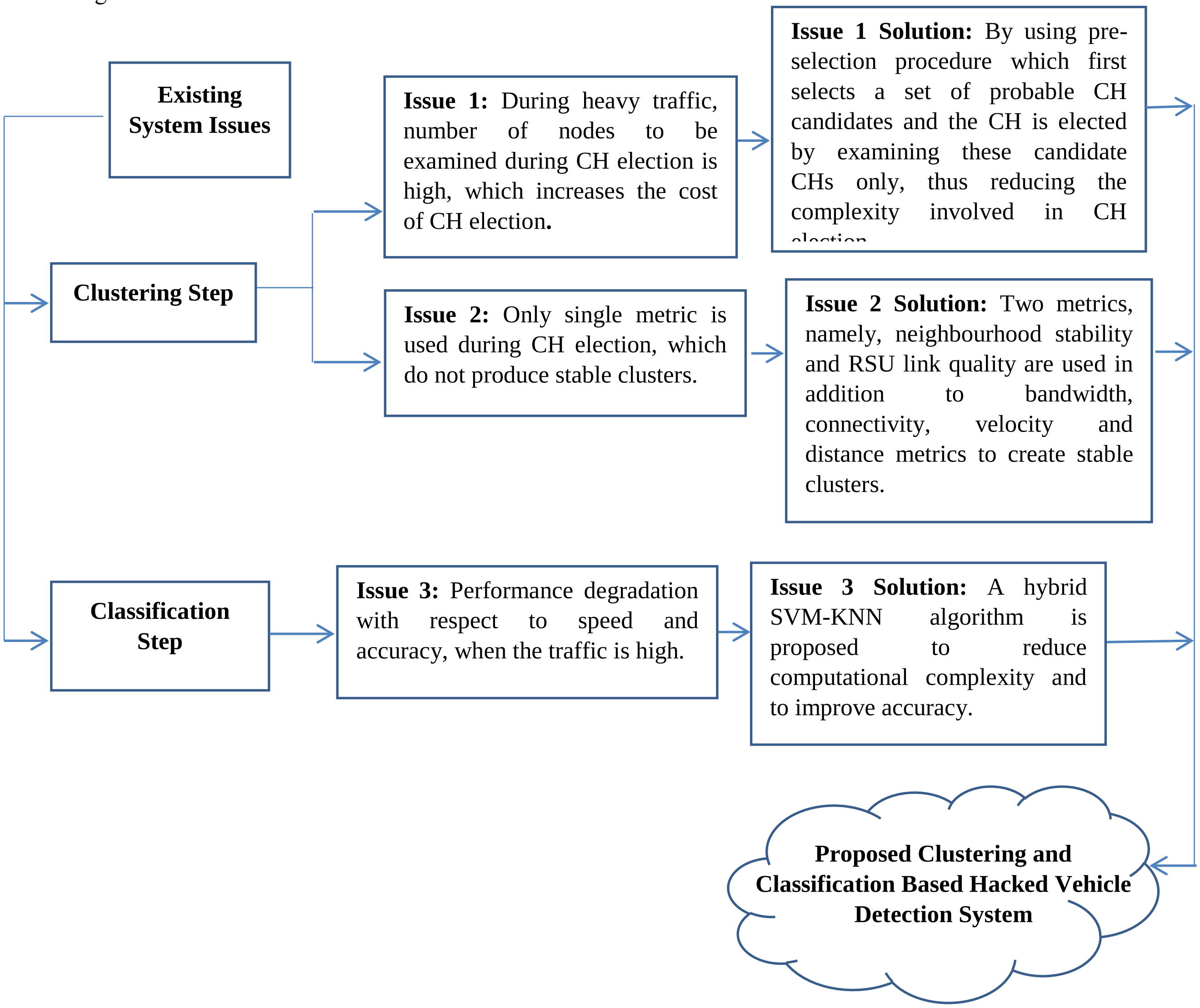
Bio-Inspired Algorithms can be partitioned into two classes, in particular, Swarm based Algorithms and Evolutionary Algorithms roused by the normal development and cumulative conduct in creatures, respectively. Swarm Intelligence shows up in natural multitudes of specific bug species. It brings about complex and regularly insightful conduct through the complex connection of thousands of independent individuals. Cooperation depends on crude impulses with no oversight. The outcome is the achievement of exceptionally complex types of social conduct and satisfaction of various improvements and errands. The fundamental guideline behind these connections is called stigmergy. Ant colony optimization algorithm [23] is a metaheuristic calculation which is inspired by the searching conduct of ants. A model is a pheromone laying on trails followed by ants in an ant colony optimization algorithm. Pheromone is an intense chemical that subterranean ants can detect as they travel along trails. It draws in ants, and accordingly, subterranean ants will often follow trails with high pheromone fixations. Ants pulled in by the pheromone will lay more pheromone on a similar path, causing more subterranean ants to be drawn in.

Though, energy consumption, capacity requirement, temporary storage availability, and message passing over IoVs might negotiate the threats of severe data transmission [24]. AI grounded on autonomous automobiles boosts some categories of applications with several advantages of intellect. Particularly for the rise in the volume of data and density, which the procedures will be processing, it is accurate and proficient for upcoming scenarios. As a next step, the rise in the volume of data in IoVs mandates a smart function, tracked to proficiently observe and accomplish the claim for intellectual IT tools [25]. Optimization plays a significant role in routing to minimize the routing overhead [26]–[31].

Due to the existing improvements in AI, particularly using ML methods to make smart resolutions in numerous IoV applications, it is beneficial to offer a wide-ranging demonstration to learn more about VANET and ML classifiers.

3. MALICIOUS CLUSTER HEAD (CH) DETECTION SYSTEM

Security of VANETs has been distinguished as one of the significant tasks. VANET application supports ongoing correspondence and manages life's basic information. One of the significant difficulties in such a manner is getting misbehaving vehicles. Without assurance, the network is powerless against assaults that influence the driver's and vehicle's security and compromise traffic well-being and may lead to death.


Existing System Issues

Clustering Step

Classification Step

Issue 1: During heavy traffic, number of nodes to be examined during CH election is high, which increases the cost of CH election**.**

Issue 2: Only single metric is used during CH election, which do not produce stable clusters.

Issue 3: Performance degradation with respect to speed and accuracy, when the traffic is high.

Issue 1 Solution: By using pre-selection procedure which first selects a set of probable CH candidates and the CH is elected by examining these candidate CHs only, thus reducing the complexity involved in CH election

Issue 2 Solution: Two metrics, namely, neighbourhood stability and RSU link quality are used in addition to bandwidth, connectivity, velocity and distance metrics to create stable clusters.

Issue 3 Solution: A hybrid SVM-KNN algorithm is proposed to reduce computational complexity and to improve accuracy.

Proposed Clustering and Classification Based Hacked Vehicle Detection System

Figure 1 Steps Taken to Enhance Hacked Vehicle Identification System

Malicious nodes might create wrong traffic warning messages and increase them to different network vehicles [17]. These active attackers might compel real drivers to change their driving conduct. For example, genuine vehicles might dial back or take backup ways to go on the off chance that phoney messages concerning unsafe occasions are circulated in the organization. Subsequently, pernicious hubs prevail in disturbance of typical driving conduct of vehicles. Attackers may likewise stifle legitimate basic well-being data messages by dropping or catching them. A malicious node may produce fake messages instead of forwarding valid messages. To prevent fake messages, there is a need to detect malicious nodes.

A Malicious CH Detection system consists of four steps, as listed below:

Step 1: Group vehicles using a clustering algorithm

Step 2: Election of CHs

Step 3: Select Multiple Point Relay (MPRs) using CHs. // MPRs are responsible for transmitting messages. The main aim of MPRs is to decrease the flooding of broadcast packets by decreasing the amount of duplicate packet retransmission in the same range.

Step 4: Identify MPRs which are hacked using machine learning classifiers

Step 5: Once identified, stop all communications to and from those identified vehicles

The two important components of the above Steps are clustering algorithm and classification algorithm. However, the existing systems [5,7] face the following difficulties during misbehaving node detection and also the present work solves the above issues to produce an enhanced hacked vehicle detection system which are described in Figure 1.

3.1. Clustering Vehicles

The proposed system Combines both Static and Dynamic Clustering. In Static Clustering, communication occurs between the RSU and vehicles within its range. RSU acts like cluster heads, and vehicles act as cluster members. In dynamic clustering, the vehicles are clustered, and one node with maximum capability is selected as cluster head. Communication takes place between vehicles through the selected cluster head vehicle. Its more flexible but performance is said to be degrade while heavy traffic.

The important steps involved during clustering are:

(i) Feature extraction and construction of clustering metric

(ii) Construct candidate CH set

(iii) Elect CHs

3.1.1. Feature Extraction and Construction of Clustering Metric

In general, the features used for clustering vehicles should consider two important types of metrics. The first is Quality of Service (QoS) metrics like bandwidth, connection degree and link quality. These metrics help to improve reliability during communication and increase coverage of CHs. In contrast, the second type, mobility metrics (like position, speed, and residual distance), help ensure VCC stability. In this work, both QoS and mobility metrics are used, and the combined metric is called QM_Metric and is constructed using Eq.(1).

$$Q\_M(i) = \xi * (BCPD + Dn) + \lambda * RLQ \qquad (1)$$

The terms in the above equation are described below:

1. ‘i’ denotes ith vehicle
2. ξ and λ are dynamic factors that vary with the vehicle density in the road segment. The λ factor is estimated using Eq.(2).

$$\lambda = \max\left\{\lambda_{\max} - \frac{Avg\_D_n}{100}, 0\right\} \qquad (2)$$

Here, D_n, Avg_D_nAvg_D$_n$ is the average neighborhood degree of vehicles, λ_{max} is the maximum impact of link quality metric defined by the local authority. In this research λ_{max} is set to 0.3 after empirical evaluation. Epsilon is estimated as α - λ, where α is the weighing factor of QoS and is set to 0.25 for safety applications, 0.75 for internet applications and 0.5 for traffic regulation applications. Again these values were set after several empirical evaluations.

3. Bandwidth-Connectivity and Proportional Distance (BCPD) (Eq.3) is a quality metric that provides details about Bandwidth (BW), Velocity Ratio (VR) and Distance Ratio (DR).

$$\text{BCPD (i)} = BW(i) * N(i) * \frac{DR(i)}{VR(i)} \qquad (3)$$

4. RLQ is the RSU link quality used to group vehicles with maximum robustness during communication. This metric, QRN(i), provides the connection quality between RSU and a vehicle (Eq. 4) using the power of membership message in dB received by vehicle i from RSU.

$$QRN(i) = P(i) – S(i) \qquad (4)$$

where P is the power of membership message in dB received by vehicle i from RSU and S is the speed of the vehicle i. Using this information, the RLQ of a RSU is determined using Eq.(5).

$$RLQ = \min\left\{\frac{Q_{RN}}{T_{RLQ}}, 1\right\} \qquad (5)$$

TRLQ is the link quality threshold, estimated as the difference among the values of the maximum and minimum power of the messages received. This threshold value is used to analyze the maximum and minimum data rates maintained by the wireless interface of the node in the VCC. If the value of RLQ~1, then it can be understood that

i) i and RSU can communicate with each other at the maximum rate

ii) i will have maximum ability to maintain the connectivity link with Road Side Unit and

iii) the impact of vehicular motion can be handled effectively by i.

3.1.2. Election of CHs

The main objective of a clustering algorithm is to form stable clusters that need minimum re-clustering and which can be used to improve the speed of communication. To achieve this objective, the clustering algorithm should select CHs of high quality and time efficient. To fasten the speed of CH selection, this work proposes the use of an initial step that can select a group of candidate vehicles. The CH selection algorithm examines only these candidate vehicles, thus saving time. The procedure used by the initial step for selecting the candidate step is discussed below.

The CH election procedure helps elect a set of nodes as CH and helps group nodes into clusters. The algorithm begins by broadcasting the HELLO message having the QoS values 2-hops away. The next step then performs voting to decide which neighboring candidate nodes have local maximum QoS metric value, which is broadcasted using a special kind of HELLO message, called Election message. After the election procedure, the elected nodes send an Ack message, acknowledging its role as CH. The Ack message contains a public key and is again sent 2-hops away. The CHs use this message to acknowledge serving its voters and also to propagate its public key that is later used to prevent cheating.

The CHs thus elected have the role of selecting MPR (Multiple Point Relay) nodes and should also broadcast the TC (Topology Control) messages having details regarding the electors. The main goal of MPRs is to reduce the flooding of broadcast packets by decreasing the amount of duplicate packet retransmission in the same range. The HELLO message is modified to include a H flag which signals that the node is a CH, and a H_NEIGH flag which denotes that a neighbour is elected as CH.

3.1.2.1. The Construction of Candidate CH Set Includes the Following Steps:

Step 1: Calculate the relative velocity of node $i,$ (RVi) with respect to its neighbours (Ne) using Eq.(6)

$$\mathrm{RVi(j)} = |\mathrm{vi} - \mathrm{vj}| \tag{6}$$

Here, vi and vj denotes the velocity of nodes i and j respectively and j ∈ Ne of i.

Step 2: Calculate the nodes Aggregate Relative Velocity (ARV) using Eq.(7).

$$\mathrm{ARVi} = \mathrm{var(RVi(jNeN))} \tag{7}$$

Step 3: Estimate the difference between the ARV of node i

and its degree of connectivity (DC) using Eq. (8). Let this be denoted as δ.

$$\delta\mathrm{i} = \mathrm{DCi} - \mathrm{ARVi} \tag{8}$$

Step 4: If δ is greater than the Selection Threshold (ST) then it is considered as a candidate CH else, it is considered as a CM.

The ST is dynamically estimated as the average of δ of all nodes that have initiated the candidate selection process. The general goal of the election messages is to transmit the votes during the CH election. The nodes use them to indicate the neighbours for which node this neighbour has voted for. The steps described to elect CHs are described below.

3.1.2.2. Steps Involved During CH Election:

```
Repeat for all nodes
Step 1: Broadcast HELLO message with QoS(i) and which is 2-hop away
Step 2: Let CCH ∈ 2HN of i ∪ i be such that
Step 3: QoS(CCH) = max(QoS(j) | j ∈ 2HN of i ∪ i
Step 4: Vote for CCH using Election messages
Step 5: $\mathrm{MPRSet(i)} = \mathrm{MPRSet(i)} \cup \mathrm{CCH}$
Step 6: Repeat for all elected CHs ∈ N do
Step 7: Broadcast Ack message 2HN away
end repeat
```

3.1.3. Select MPRs

In general, cluster communication is performed using routing protocols. In recent years, the usage of routing protocol based on Ant Colony Optimization (ACO) is effective. This work also uses the same to select MPRs. Detailed description can be found in [23].

3.1.4. Detection of Hacked MPRs

The compromised MPRs are detected using an algorithm having two main steps.

i) Data Collection – The CMs continuously monitor and analyze MPR nodes and collect data (features) regarding their behavior. The method used for data collection is same as the one used by [6].

ii) Data Analysis - Uses the data collected to train the classifier to classify a node as normal or malicious.

The first step, data collection, uses two features to obtain an initial idea of normal or hacked MPRs. They are the number of forwarded messages (N1) and the number of packets forwarded (N2). If for a vehicle i, N1 < N2, then i is hacked, it is considered normal. The second step uses the classifiers to analyze the collected features and classify the MPRs as either malicious or normal. For this purpose, five different classifiers, namely, Naïve-Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbour (KNN) and Decision Tree (DT)- are used. A hybrid classifier that combines SVM and KNN classifiers is also analyzed for its effectiveness to detect malicious vehicles. Extension to a typical SVM with KNN is to more cautiously set the points that fall adjacent to the separating border as it is where the false negatives are expected to occur. The K-Nearest Neighbours (KNN) procedure can be designated as follows: a fresh sample is categorized according to a function of the categorizations of its k-most adjoining neighbours. This function can be, for example, a mainstream balloting or a consensus criterion. The foremost factor of the procedure is the number of neighbors to k that should be tested to choose a classification. The proposed hybrid SVM_KNN classifier consists of the following steps.

Step 1: Compute distances of the input image message to all training examples and pick the nearest K neighbors (Here, k is taken as 2).

Step 2: If the K neighbors have all the same labels, the query is labelled and exit; else, compute the pairwise distances between the K neighbours.

Step 3: Convert the distance matrix to a kernel matrix and apply SVM.

Step 4: Use the resultant classier to label the test data.

This combined approach of SVM_KNN has the two main benefits of each algorithm: the simplification effectiveness of SVM, particularly for data elements that are remote to hyperplane and the intricate border that the k-NN procedure creates adjacent to the separating hyperplane where the misclassifications, specifically false negatives, are more likely.

4. RESULTS AND DISCUSSIONS

The proposed SVM_KNN is evaluated using NS2 against SVM, DT, NB and KNN. The system was simulated using ns2. The Simulation Factors are mentioned in the following Table 1.

Table 1 Simulation Factors

Factors	Value
Quantity of Vehicles	500
Data Rate	3 Mbps
Simulation Time	300 Sec
MAC protocol	MAC 802.11
Sensitivity	-92 dBm
Simulation Area	600 x 450 m^2
Vehicular Speed	0-50 Km/hr
Propagation type	Two Ray Ground
Radio Frequency	2.47 GHz
Channel Bandwidth	3 Mbps

The performance of proposed SVM_KNN is computed using parameters like accuracy rate, attack detection rate, false positive rate and packet delivery ratio which are defined as:

Accuracy Rate: It is calculated by using Eq.(9):

Accuracy Rate = (Total Number of correctly classified process / Total Number of Processes) * 100% (9)

Attack Detection Rate: It is described in Eq.(10):

Attack Detection Rate = (Total Number of attacks / Total Number of detected attacks) * 100% (10)

False Positive Rate: It may be explained in Eq.(11):

False Positive Rate = (Total Number of misclassified process / Total Number of normal processes) * 100% (11)

Packet delivery ratio: It is defined in Eq. (12) as:

Packet Delivery Ratio = Total Number of received packets / Total Number of sent packets (12)

4.1. Accuracy Analysis

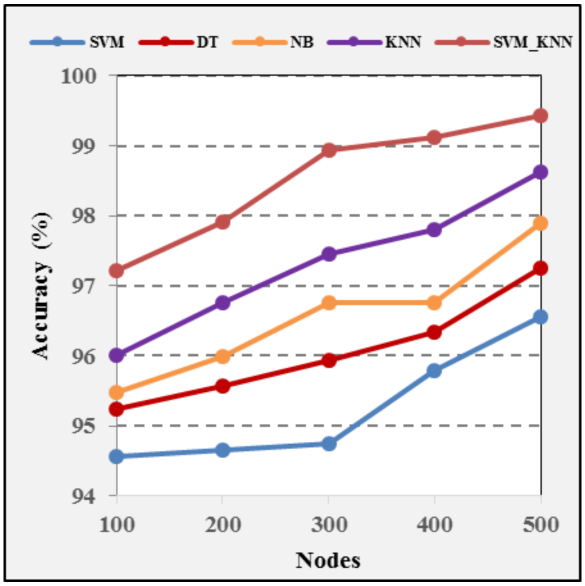


Figure 2 Accuracy (%) Vs. Network Density

In Figure 2, the x-axis indicates the network density, and the y-axis is marked with accuracy measured in percentage. In Figure 3, the x-axis indicates the number of malicious nodes, and the y-axis indicates the accuracy measured in percentage. From Figure 2 it is clear that the proposed SVM_KNN classifier achieves maximum efficiency than that of the existing classifiers namely, SVM, DT, NB and KNN in terms of accuracy while varying the number of vehicles. This is because, when the network has high number of vehicles, more number of evidences can be collected, which improved the accuracy of malicious node detection. In Figure 3, While varying the percentage of malicious vehicles, the performance decreased slightly when the number of malicious vehicles was high. However, the proposed SVM_KNN classifier still produced maximum accuracy. The values for Figure 2 and Figure 3 are provided in Table 2 and 3 respectively.

Table 2 Accuracy (%) Vs. Network Density

Nodes Classifiers	100	200	300	400	500
SVM	94.56	94.65	94.75	95.78	96.56
DT	95.23	95.56	95.94	96.34	97.26
NB	95.47	95.98	96.75	96.75	97.89
KNN	96	96.75	97.45	97.8	98.62
SVM_KNN	97.22	97.92	98.94	99.12	99.43

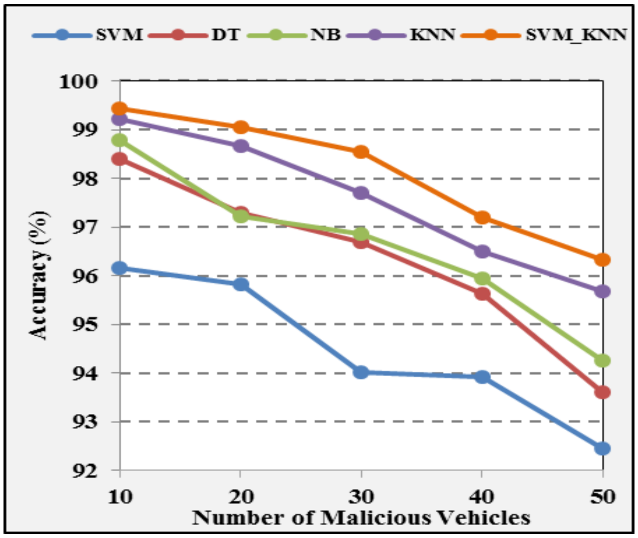


Figure 3 Accuracy (%) Vs. Number of Malicious Vehicles

Table 3 Accuracy (%) Vs. Number of Malicious Vehicles

Nodes Classifiers	10	20	30	40	50
SVM	96.17	95.82	94.01	93.93	92.46
DT	98.40	97.30	96.70	95.63	93.62
NB	98.78	97.23	96.85	95.95	94.25
KNN	99.21	98.67	97.70	96.5	95.67
SVM_KNN	99.44	99.05	98.54	97.19	96.33

4.2. Attack Detection Rate

In Figure 4, the x-axis indicates the network density, and the y-axis is marked with attack detection rate measured in percentage. In Figure 5, the x-axis indicates the number of malicious nodes, and the y-axis indicates the attack detection rate measured in percentage. From Figure 4, its clear that the attack detection capacity of the proposed SVM_KNN classifier has improved in a tremendous fashion when compared with the conventional SVM, DT, NB and KNN classifiers. From Figure 5, its observed that the Proposed SVM_KNN classifier offers better Attack Detection Rate compared to SVM, DT, NB and KNN, respectively, concerning Percentage of Malicious Vehicles. With both selected scenarios, the detection capacity decreased with increase network density and number of malicious nodes present. The values for Figure 4 and Figure 5 are provided in Table 4 and 5 respectively.

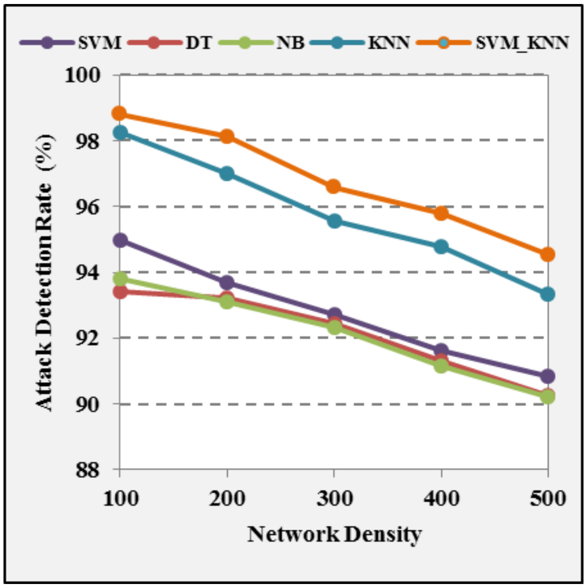


Figure 4 Attack Detection Rate Vs Network Density

Table 4 Attack Detection Rate Vs Network Density

Nodes Classifiers	100	200	300	400	500
SVM	94.99	93.67	92.73	91.61	90.85
DT	93.40	93.21	92.45	91.32	90.24
NB	93.8	93.12	92.32	91.14	90.21
KNN	98.25	97	95.55	94.77	93.33
SVM_KNN	98.81	98.13	96.59	95.78	94.54

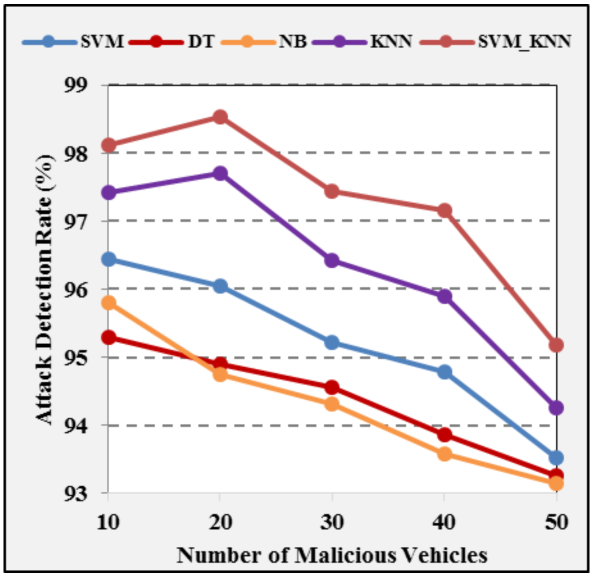


Figure 5 Attack Detection Rate Vs. % of Malicious Vehicles

Table 5 Attack Detection Rate Vs. % of Malicious Vehicles

Nodes Classifiers	10	20	30	40	50
SVM	96.45	96.04	95.21	94.78	93.52
DT	95.30	94.89	94.56	93.87	93.25
NB	95.8	94.75	94.32	93.57	93.14
KNN	97.43	97.7	96.43	95.9	94.26
SVM_KNN	98.12	98.54	97.44	97.16	95.18

4.3. False Positive Rate

In Figure 6, the x-axis indicates the network density, and the y-axis is marked with false positive rate measured in percentage. In Figure 7, the x-axis indicates the number of malicious nodes, and the y-axis indicates the false positive rate measured in percentage. From Figure 6, its clear that the proposed hybrid classifier SVM_KNN offers better False Positive Rate when compared to SVM, DT, NB and KNN with respect to Network Density.

Varying the number of vehicles shows a decreasing trend while the VANET has an increasing trend when the number of malicious nodes becomes high. This behaviour is because when the network density is high, more evidence can be collected, which increases the Accuracy, thus reducing the false positives.

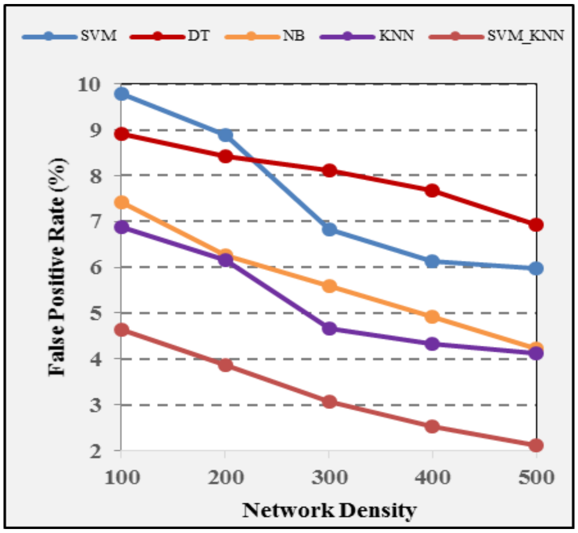


Figure 6 False Positive Rate Vs. Network Density

Table 6 False Positive Rate Vs. Network Density

Nodes Classifiers	100	200	300	400	500
SVM	9.79	8.88	6.82	6.14	5.97
DT	8.92	8.43	8.12	7.67	6.94
NB	7.43	6.25	5.58	4.91	4.22
KNN	6.87	6.15	4.67	4.32	4.13
SVM_KNN	4.63	3.87	3.06	2.54	2.11

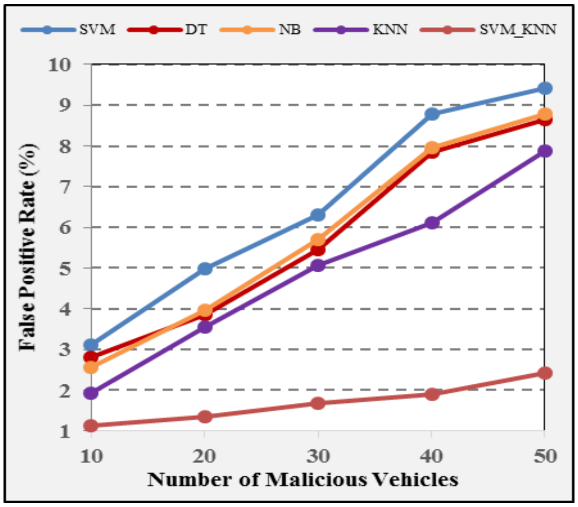


Figure 7 False Positive Rate Vs. Percentage of Malicious Vehicles

From Figure 7, its clear that the proposed hybrid classifier SVM_KNN offers better False Positive rates when compared to SVM, DT, NB and KNN concerning Percentage of Malicious Vehicles. On the other hand, the false positives also increase when the percentage of misbehaving vehicles increases. But the improvement showed by the proposed classifier shows that it has maximum discriminating power to differentiate the two target classes (normal and malicious) as it uses quality improved features. The values for Figure 6 and Figure 7 are provided in Table 6 and 7 respectively.

Table 7 False Positive Rate Vs. Percentage of Malicious Vehicles

Malicious Nodes / Classifiers	10	20	30	40	50
SVM	3.11	4.99	6.32	8.78	9.43
DT	2.80	3.85	5.45	7.84	8.65
NB	2.56	3.96	5.69	7.96	8.78
KNN	1.93	3.55	5.06	6.11	7.87
SVM_KNN	1.12	1.34	1.67	1.9	2.42

4.4. Packet Delivery Ratio

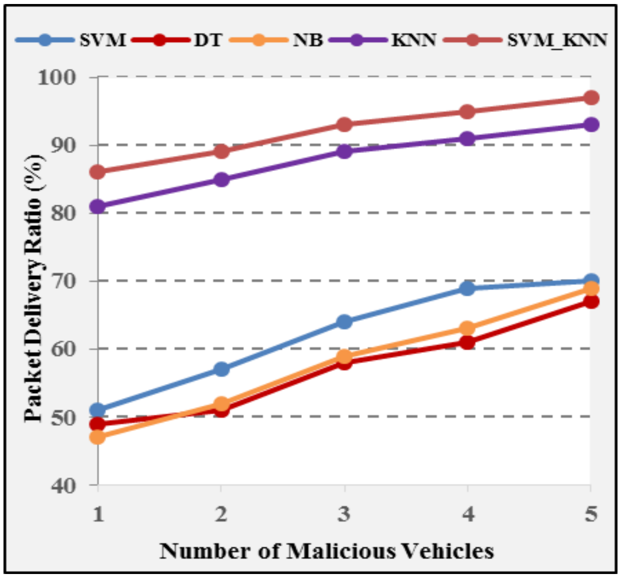


Figure 8 Packet Delivery Ratio Vs. Number of Malicious Vehicles

In Figure 8, the x-axis indicates the number of malicious nodes, and the y-axis indicates the packet delivery ratio measured in percentage. From Figure 8 it is clear that the hybrid SVM_KNN classifier offers better Packet Delivery Ratio when compared to SVM, DT, NB and KNN, respectively with respect to Percentage of Malicious Vehicles. From the results portrayed, it is evident that the enhanced SVM_KNN based malicious vehicle detection has high packet delivery ratio when compared to the conventional system. This is because, the proposed system has minimized the effect of misbehaving vehicles by accurately detecting them (as indicated by the high Accuracy and low false positives obtained), which in turn has improved the communication and routing process. The values for Figure 8 is provided in Table 8.

Table 8 Packet Delivery Ratio Vs. Number of Malicious Vehicles

Malicious Nodes / Classifiers	1	2	3	4	5
SVM	51	57	64	69	70
DT	49	51	58	61	67
NB	47	52	59	63	69
KNN	81	85	89	91	93
SVM_KNN	86	89	93	95	97

5. CONCLUSION

In VCC, the problem of identifying misbehaving hacked vehicles is crucial to forming a secure communication network. This paper presented a clustering-based model, which identified hacked vehicles using machine learning classifiers. Four classifiers, namely, SVM, DT, NB and KNN, were analyzed, and a hybrid SVM_KNN classifier was also proposed. The performance of these classifiers on their effectiveness of identifying hacked vehicles was evaluated using various parameters like Accuracy, attack discovery rate, false-positive rate and packet delivery ratio. Experimental results showed that the hybrid classifier is more effective during hacked vehicle detection. Future research is planned to improve the classification step further and include cryptographic algorithms further to improve the security of the VCC during communication.

REFERENCES

[1] Ramakrishnan, B., Rajesh, R. S., & Shaji, R. S. (2010). Performance analysis of 802.11 and 802.11 p in cluster based simple highway model. International Journal of Computer Science and Information Technologies, 1(5), 420-426.

[2] E. S. A. Ahmed and R. A. Saeed, "A survey of big data cloud computing security, "International Journal of Computer Science and Software Engineering (IJCSSE), vol. 3, no. 1, pp. 78–85, 2014.

[3] Katuka, Jatau Isaac, and Muhammad Shafie Abd Latiff. "Vanets and Its Related Issues: An Extensive Survey." Journal of Theoretical & Applied Information Technology 66.1, 2014.

[4] H. Wu, "Developing vehicular data cloud services in the IoT environment, "IEEE Transactions on Industrial Informatics, vol. 10, no. 2, pp. 1587–1595, 2014.

[5] A. Eltahir, R. A. Saeed, A. Mukherjee, and M. K. Hasan, "Evaluation and analysis of an enhanced hybrid wireless mesh protocol for vehicular ad-hoc network," EURASIP Journal on Wireless Communications and Networking, vol. 1, pp. 1–11, 2016.

[6] M. K. Hasan, A. F. Ismail, A.-H. Abdalla, H. A. M. Ramli, W. Hashim, and S. Islam, "Throughput maximization for the cross-tier interference in heterogeneous network," Advanced Science Letters, vol. 22, no. 10, pp. 2785–2789, 2016.

[7] Wahab OA, Otrok H and Mourad A. "VANET QoSOLSR: QoS-based clustering protocol for vehicular ad hoc networks", Computer Communication; 36: 1422–1435, 2013.

[8] Joe, M. Milton & Ramakrishnan, Balasundaram. (2016). Review of vehicular ad hoc network communication models including WVANET (Web VANET) model and WVANET future research directions. Wireless Networks. 22. 10.1007/s11276-015-1104-z.

[9] Y. Mao, "A survey on mobile edge computing: the communication perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, 2017.

[10] Z. K. A. Mohammed and E. S. A. Ahmed, "Internet of things applications, challenges and related future technologies," WSN, vol. 67, no. 2, pp. 126–148, 2017.

[11] J. Xu, "Joint service caching and task offloading for mobile edge computing in dense networks," in Proceedings of the IEEE Conference on Computer Communications, Honolulu, HI, USA, 2018.

[12] Nobahary, Sanaz, and ShahramBabaie. "A Credit-based Method to Selfish Node Detection in Mobile Ad-hoc Network." Appl. Comput. Syst. 23.2: 118-127, 2018.

[13] Shams EA, Rizaner A, Ulusoy AH (2018), "Trust aware support vector machine intrusion detection and prevention system in vehicular ad hoc networks" Computers & Security 78:245– 254.

[14] A. H. Sodhro, Z. Luo, G. H. Sodhro, M. Muzamal, J. J. P. C. Rodrigues, and V. H. C. de Albuquerque, "Artificial Intelligence based QoS optimization for multimedia communication in IoV systems," Future Generation Computer Systems, vol. 95, pp. 667–680, 2019.

[15] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial intelligence-enabled intelligent 6G networks," 2019, https://arxiv.org/abs/1912.05744.

[16] W. Tong, A. Hussain, W. X. Bo, and S. Maharjan, "Artificial intelligence for vehicle-to-everything: a survey," IEEE Access, vol. 7, pp. 10823–10843, 2019.

[17] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, "Artificial intelligence empowered edge computing and caching for internet of vehicles," IEEE Wireless Communications, vol. 26, no. 3, pp. 12–18, 2019.

[18] Nascimento, Douglas & Iano, Yuzo & Loschi, Hermes Jose & Razmjooy, Navid & Sroufe, Robert & Oliveira, Vlademir & Pajuelo Castro, Diego Arturo & Montagner, Matheus,. Sustainable Adoption of Connected Vehicles in the Brazilian Landscape: Policies, Technical Specifications and Challenges. Transactions on Environment and Electrical Engineering. 3. 44-62, 2019.

[19] H. Ji, "Artificial intelligence-empowered edge of vehicles: architecture, enabling technologies, and applications," IEEE Acces, vol. 8, pp. 61020–61034, 2020.

[20] M. B. Hassan, E. S. Ali, R. A. Mokhtar, R. A. Saeed, and B. S. Chaudhari, "NB-IoT: concepts, applications, and deployment challenges, book chapter (ch 6)," in LPWAN Technologies for IoT and M2MApplications, B. S. Chaudhari and M. Zennaro, Eds., Elsevier, Berlin, Germany, 2020.

[21] Z. E. Ahmed, M. K. Hasan, R. A. Saeed et al., "Optimizing energy consumption for cloud internet of things," Frontiers of Physics, vol. 8, p. 358, 2020.

[22] Bibi, Rozi, et al. "Edge AI-based automated detection and classification of road anomalies in VANET using deep learning." Computational intelligence and neuroscience, 2021.

[23] Mohammadnia, A., Alguliyev, R., Yusifov, F., &Jamali, S., "Routing algorithm for vehicular Ad Hoc network based on dynamic Ant Colony optimization", International Journal of Electronics and Electrical Engineering, 4, 79–83, 2016.

[24] L. Song, G. Sun, H. Yu, X. Du, and M. Guizani, "FBIA: a fog-based identity authentication scheme for privacy preservation in internet of vehicles," IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5403–5415, 2020.

[25] Z. Meng, "Security enhanced internet of vehicles with Cloud‑-Fog‑Dew computing," ZTE Communications, vol. 15, no. S2, 2017

[26] J. Ramkumar and R. Vadivel, "Intelligent Fish Swarm Inspired Protocol (IFSIP) For Dynamic Ideal Routing in Cognitive Radio Ad-Hoc Networks," Int. J. Comput. Digit. Syst., vol. 10, no. 1, pp. 1063–1074, 2020, doi: http://dx.doi.org/10.12785/ijcds/100196.

[27] J. Ramkumar and R. Vadivel, "Performance Modeling of Bio-Inspired Routing Protocols in Cognitive Radio Ad Hoc Network to Reduce End-to-End Delay," Int. J. Intell. Eng. Syst., vol. 12, no. 1, pp. 221–231, 2019, doi: 10.22266/ijies2019.0228.22.

[28] J. Ramkumar and R. Vadivel, "Improved frog leap inspired protocol (IFLIP) – for routing in cognitive radio ad hoc networks (CRAHN)," World J. Eng., vol. 15, no. 2, pp. 306–311, 2018, doi: 10.1108/WJE-08-2017-0260.

[29] J. Ramkumar and R. Vadivel, "Multi-Adaptive Routing Protocol for Internet of Things based Ad-hoc Networks," Wirel. Pers. Commun., pp. 1–23, Apr. 2021, doi: 10.1007/s11277-021-08495-z.

[30] J. Ramkumar and R. Vadivel, "Meticulous elephant herding optimization based protocol for detecting intrusions in cognitive radio ad hoc networks," Int. J. Emerg. Trends Eng. Res., vol. 8, no. 8, pp. 4549–4554, 2020, doi: 10.30534/ijeter/2020/82882020.

[31] M. Lingaraj, T. N. Sugumar, C. Stanly Felix, and J. Ramkumar, "Query Aware Routing Protocol for Mobility Enabled Wireless Sensor Network," Int. J. Comput. Networks Appl., vol. 8, no. 3, p. 258, Jun. 2021, doi: 10.22247/IJCNA/2021/209192.

Authors

A. Sheela Rini has completed M.Sc. and M.Phil in Computer Science and pursuing her Ph.D. in Avinashlingam Institute for Home Science and Higher Education for Women, Coimbatore. She has qualified UGC NET for Assistant Professor in Computer Science. She is working as an Assistant Professor in Department of Computer Science (PG) of PSGR Krishnammal College for Women. Her research area includes Networking, Cloud Computing and Cyber Security.

Dr. C. Meena presently designated as Incharge of Computer Center in Avanishalingam Institute for Home Science and Higher Education for Women, Coimbatore. She has 25 years of rich & extensive experience in teaching, System Analysis, Technical Information and Programming Skills. She is expertise in Project Personals and Budget Estimation. She has presented papers in many countries like Malaysia, USA etc. She has published more than 45 papers in National and International Journals. She has undertaken many UGC sponsored projects.

RESEARCH ARTICLE

How to cite this article:

A. Sheela Rini, C. Meena, “Analysis of Machine Learning Classifiers to Detect Malicious Node in Vehicular Cloud Computing”, International Journal of Computer Networks and Applications (IJCNA), 9(2), PP: 202-213, 2022, DOI: 10.22247/ijcna/2022/212336.