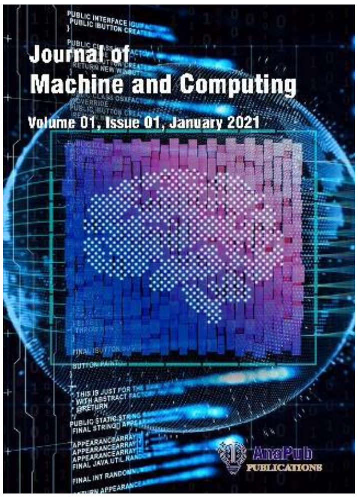
Journal Pre-proof

Sybil Attack Detection in VANET using CNN Enhanced with Chaotic Maps and Elephant Herding Optimization for Secure Data Transmission



Suganyadevi K, Swaminathan A, Baskar Kasi and Anju M A

DOI: 10.53759/7669/jmc202505097

Reference: JMC202505097

Journal: Journal of Machine and Computing.

Received 08 October 2024

Revised form 23 January 2025

Accepted 25 March 2025

Please cite this article as: Suganyadevi K, Swaminathan A, Baskar Kasi and Anju M A, "Sybil Attack Detection in VANET using CNN Enhanced with Chaotic Maps and Elephant Herding Optimization for Secure Data Transmission", Journal of Machine and Computing. (2025). Doi: https://doi.org/10.53759/7669/jmc202505097.

Sybil Attack Detection in VANET using CNN Enhanced with Chaotic Maps and Elephant Herding Optimization for Secure Data Transmission

[1]K. Suganyadevi*,[2]A. Swaminathan,[3]Baskar Kasi, [4]Anju M A

[1]Department of Electronics and Communication Engineering, Sri Eshwar College of Engineering Coimbatore, TamilNadu, India

[2]Department of Computer Science and Business systems, Panimalar Engineering College, Poonamallee, India

[3]Institute of Computer Science and Engineering, SIMATS Engineering, SIMATS, Chennai, India

[4]Department of Electronics and Communication Engineering, Nehru Institute of Engineering and Technology, Coimbatore, India

sugan.er.sd@gmail.com, swamisivam19@gmail.com, baskar_ka@yahoo.com, nietanju@nehrucolleges.com

*Corresponding Author: K. Suganyadevi

Abstractt

The Vehicular Ad-Hoc Network (VANET) model stands out as a cost-effective and easily deployable solution for traffic management and accident prevention. Within VANET, nodes employ broadcast protocols for disseminating safety information rather than relying on routing protocols. Nonetheless, there exists a vulnerability to malicious activities, such as targeted attacks where a vehicle may intentionally transmit harmful packets to cause harm. Among these, the Sybil attack (SA) poses the most severe threat, wherein the attacker creates multiple identities to impersonate distinct nodes. Detecting and defending against such attacks, particularly when perpetrators operate under genuine identities, presents significant challenges. To mitigate this issue, a deep learning-based intrusion detection system (IDS) has been proposed for effectively identifying SA in VANET. The system employs a clustering algorithm known as Glow Worm Swarm Optimization (Gon SO)-based K-harmonic means (GSOKHM) for vehicle clustering. Subsequently, it utilizes the Floyd-Warshall algorithm (FWA) to designate Cluster Heads (CH) from these clusters. Following CH selection, our advanced CMEHA-CNN algorithm utilizes a combination of Convolutional Neural Network (CNN) and chaotic maps to detect any malicious CH . This entails extracting pertinent features from the CH. Upon confirming the legitimacy of the CH, its information is firmly transmitted to the cloud by means of SHA2-ECC, a fusion of Secure Hashing Algorithm and Elliptic Curve Cryptography. The simulation (NS-2.35) outcomes of our proposed methodology achieves an impressive accuracy rate of 98.9% and ensures a high level of security at 99%, surpassing existing methodologies.

Keywords: Intrusion detection system, Vehicular adhoc networks (VANET), Optimisation, Sybil attack detection and Deep Learning Safe hash algorithm (SHA), Elephant Herding Algorithm (EHA).

1 INTRODUCTION

Vehicular Ad hoc Networks (VANETs), a subset of Mobile Ad-hoc Networks (MANETs), facilitate communication among vehicles on the road and with roadside infrastructure, as documented in references [1-3]. In a VANET, vehicles operate as nodes within a self-organizing mobile network, where the presence of other nodes is neither pre-determined nor monitored. These networks consist of two primary node types: On-board Units (OBUs) and Roadside Units (RSUs). OBUs, also known as mobile radio units, play a pivotal role in VANET communication. In contrast, fixed-site units (RSUs) are the network's backbone and are installed along roadways. RSUs are the centers through which all vehicular traffic must travel. OBUs connect automobiles to RSUs via radio apparatus for Dedicated Short-Range Communication (DSRC) [4]. Figure 1 displays VANET's structure. Every vehicle is accoutered with both an AU and an OBU. OBU enables bi-directional communication amongst vehicles to between vehicles and RSUs. RSUs with a communication range of approximately one kilometer along the highway are installed. Each RSU communicates with its network peers and contributes to weather forecasts and traffic updates. Since the vehicles in a VANET communicate wirelessly and the network topology is constantly changing as vehicles enter and leave, security is an ongoing concern.

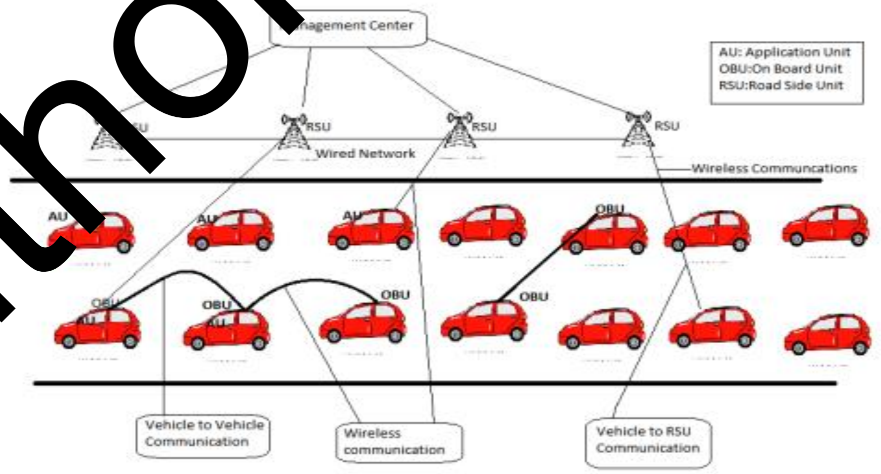


Figure 1: VANET architecture

The emergence of VANETs represents one of the most captivating advancements in mobile technology in recent years. They are deemed essential for the implementation of wireless mobile technology and are seen as a burgeoning approach. Moreover, VANETs can be integrated into Intelligent Transportation Systems (ITS) deployment strategies. Notably, VANETs relatively differ from MANETs due to factors such as heightened mobility, scalability, ever-changing and geographically constrained topologies, stringent deadlines, slower deployment processes, unreliable channel moment, intermittent node property, frequent network decomposition, and considerations of driver behavior [1-2, 5]. The primary objective of VANETs is to enable efficient vehicle-to-vehicle communication, necessitating the presence of radio interfaces for effective node communication. Moreover, the deployment of VANET technology requires the allocation of a dedicated spectrum range for data transmission. To effectively participate in VANET technology and communicate seamlessly, a node must possess a set of attributes enabling it to gather information, share data with fellow nodes, and make informed decisions. These attributes typically comprise omnidirectional antennas, cameras, sensors, Global Positioning System (GPS) receivers, onboard computers, and Event Data Recorders [6]. Fewer traffic accidents, an enhanced driving and traveling experience, and more straightforward methods of paying for tariffs, petrol, and parking are some advantages of VANET technology. Road users rely on various applications for navigation, traffic monitoring, alerts, song sharing, amusement, climate control, and even online gaming [7]. These applications necessitate a constant flow of data and information regarding traffic issues, emergency message delivery, and road condition notifications designed to keep drivers safe and productive. This highlights the need for reliable data transmission between nodes. Any malicious user who modifies the messages could significantly impact driver behavior, which could disrupt the network's topology and thereby compromise security [8].

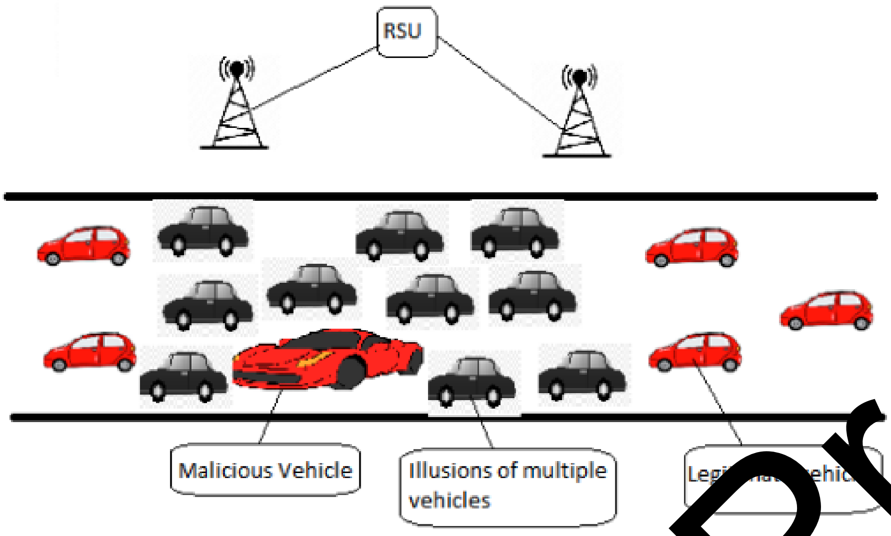


Figure 2. Sybil attack

The well-known attacks purpose at interrupt network communication is the SA, where an intruder fabricates various vehicle identities. Techniques to detect and prevent SA include attack detection, reaction, and prevention. Previous SA detection methods have focused solely on received signal strength indicators (RSSI), which have limitations in terms of robustness and detection range. An IDS is becoming increasingly important for network security construction, but detecting complex security breaches remains challenging shown in the Figure 2. While it is possible to identify abnormal network hazards using techniques such as assist vector machines with artificial neural networks, it is still challenging to do so accurately and securely. Effective SA detection in VANET requires enhancements. Using chaotic maps, the Elephant Herding Optimisation Algorithm-CNN, and the Secure Hash Algorithm for Enhanced Cryptographic Communications (SHA2-ECC), a method for identifying SA in the VANET environment has been presented as a solution to this problem. Here is what the intended work will accomplish:

- To cluster vehicles, we use the GSOKHM algorithm, which is based on GSO and K-harmonic means.
- The Floyd-Warshalls algorithm (FWA) determines which node in a cluster will be the cluster head.
- A deep learning model called Chaotic Maps Elephant Herding Optimization Algorithm-CNN (CMEHA-CNN) is utilized to identify Malevolent CH and extract relevant features of the cluster head.

- After identification, if the CH is deemed usual, the data it contains is safely uploaded to a remote server(cloud) using the Secure Hashing Algorithm (SHA2) - Elliptic curves cryptography (ECC), commonly known as SHA2-ECC.

The article is organized into several sections. Section 2 discusses on SA detection systems in VANET. Section 3 presents a new approach for predicting privacy-preserving and SA detection algorithms in VANET. Section 4 render the simulation's outcome, parameter analysis. Finally, Section 5 ends with future enhancement work.

2 RELATED WORKS

In this segment, we have discussed previous research on intrusion detection in VANET and provided a review of some of the work done.

Ma et al. [11] have presented an algorithm for encryption based on attribute that hold the roadside unit (RSU). It uses the vehicle to encrypt data and performs the computation model. The two nodes of the roadside unit, storage, and computing capabilities are utilized in this method. The decrypted message is examined for traceability and audits.

Buda et al. [12] present a dispersed clustering technique for isolating the network's peripheral link from transactional data. When performing an edge selection, the vehicle's quality is determined by averaging its typical and edge velocities. Analyses of sensor data are used to generate and validate blocks. It takes a certain number of transactions to transfer wholly decentralized data. Utilizing this method, expression difficulties are resolved. For this reason, high-performance computing nodes must be targeted. Horng et al. [13], The technique of cipher text-policy attribute-based encryption (CP-ABE) uses the same key for encryption and decryption. The policy's characteristics are disclosed after the data have been deciphered. Using encryption and decryption, the nodes and roadside units are calculated. The data is updated whenever new user input is received, or an attribute is removed. This demonstrates that the experiments successfully developed a system with robust security, scalable performance, and granular control. Additionally, the communication delay is measured and analyzed in a real-world setting. Rathee et al. [14] ABM and PBM were designed to transmit and store information in real-time. The subject matter provided by the data transffering object is used to evaluate the simulations' precision. We assess the SITO optimizer based on energy consumption, network link count, and throughput. The conversations are discreet, fruitful, and calming. Identifying reliable devices is crucial for network analysis in a dynamic environment.

Meshcheryakov et al. [15] To reconstruct the distributed ledger required for operating such devices, a Byzantine Fault Tolerance (PBFT) consensus method were created. The efficacy of the Blockchain has been evaluated to comprehend its operation better. Individual specifications limit the processing capacity and data transfer rates of IoT devices. The latency is determined using block size, generation time, and data payload size. The efficacy of up to 70 nodes is enhanced. Consequently, actions within the network environment must be performed using restricted devices. Subba et al. (2018) [16] proposed VANET as a multi-level IDS. This study suggests a original clustering technique for VANETs and a game-theory-based intrusion detection system as solutions to these issues. The simulation outcome exhibits that the planned framework can detect intrusions swiftly and precisely against a wide variety of threats while having minimal impact on the underlying vehicular network.

Shu et al. (2020) [17] developed a collaborative intrusion detection solution for VANETs utilizing a distributed software-defined network (SDN) protocol and deep learning. This innovative approach enabled multiple controllers to train a global intrusion detection framework without the need for sharing sub-network data flows. Suganyadevi et al.(2024) [18] .The system demonstrated efficacy in both IID and non-IID scenarios and underwent evaluation using real-world datasets.

Nitha C. Velayudhan et al (2021) [20] developed a deep learning-based IDS system, also suggested a CH election algorithm, and clustering algorithm to increase stability and connectedness among vehicles in a VANET. The proposed methods outperformed existing techniques regarding specificity, accuracy, recall, precision, and the F-measure. An earlier deep learning model for detecting DDoS attacks lacked load balancing and scalability. To address these issues, new algorithms, CMEHA-CNN and SHA2-ECC, have been proposed for attack detection and secure data transmission. The paper presents a framework called CMEHA-DNN that uses deep learning and a clustering algorithm called GSOKHM for intrusion detection in VANET. Optimization techniques like the elephant herd algorithm (EHA) can also be used to improve detection and accuracy. To ensure data security, a new encryption method (SHA2-ECC) is implemented as part of the suggested approach. The architecture of the recommended method is illustrated in Figure 3.

A. Formation of Cluster using Glow Worm Swarm Optimization (GSO) based K-harmonic means (GSOKHM) clustering algorithm

To optimize the performance of the VANET area, the GSOKHM clustering algorithm was used to break it down into smaller clusters. This resulted in a decrease in propagation delay and an increase in delivery ratio. The algorithm was selected for its ability to handle large datasets. Clustering simplifies important functions such as bandwidth allocation, routing, and channel access. A CH is selected for each cluster using the FWA procedure. The CH may be a vehicle with adequate data storage and retrieval capabilities. Each CH has access to the descriptions of all services.

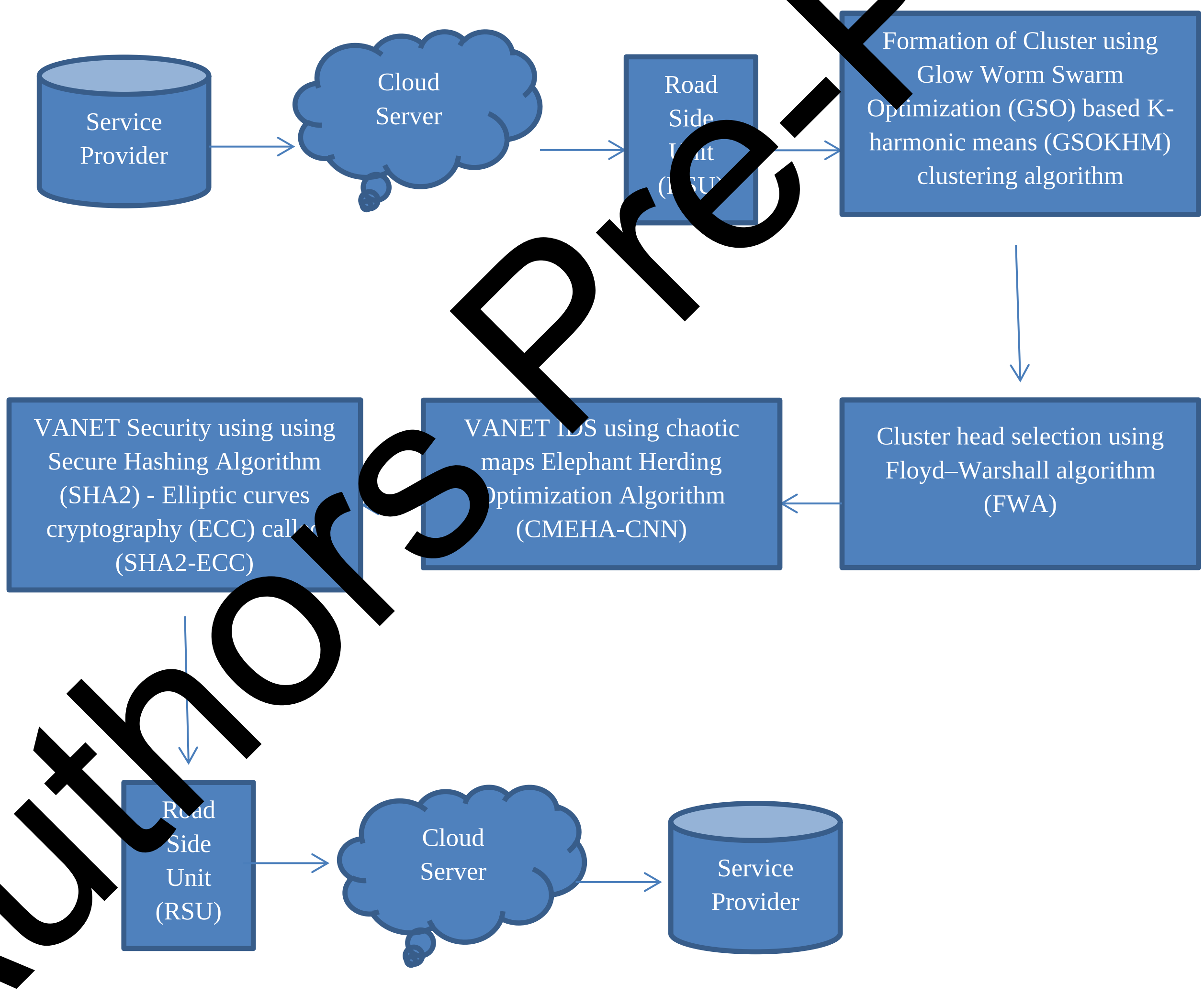


Figure 3: Proposed Flow Diagram

Glowworm Swarm Optimization (GSO)

The algorithm is truly captivating in how it imitates the flashing behavior of glowworms. It's interesting to observe how people are naturally drawn to the brightest glowworms. The algorithm consists of six crucial phases: initializing the glowworms, updating the luciferin, selecting the neighborhood, computing movement probability, moving the glowworms, and updating the decision radius.

Glowworms' initialization: In this phase, glowworms are considered picture characteristics that are randomly distributed over the specified fitness value. The aforementioned quantity of Lucifer is recovered in glowworms. Additionally, the actual iteration is set at 1. Here, categorization accuracy is treated as a fitness value.

Luciferin-update phase: Luciferin is updated based on the fitness value (accuracy) in addition to the prior luciferin value, and the rule is described by equation (1).

$$l_i(ti+1) = (1-\rho)l_i(ti) + \gamma Fitnessx_i(ti+1) \quad (1)$$

Where, $l_i(t)$ denotes glowworm luciferin i (feature) at time ti, ρ intend luciferin constant decay $(0 < \rho < 1)$ and γ denotes luciferin improvement constant, $x_i(ti+1)\epsilon R^M$ signifies glowworm (feature) location at time in addition $Fitnessx_i(ti+1)$ shows fitness value at location of the glowworm at span of $ti+1$.

Neighborhood-Select Phase: $N_{is}(t)$ glowworm neighbors and (feature) i at time incorporate of luminous and given by equation (2),

$$N_{is}(ti) = \{j : d_{ij}(ti) < r_d^i(ti);\ l_i(ti) < l_j(ti)\} \quad (2)$$

$r_d^i(ti)$ gives uncertain local-decision, $d_{i,j}(ti)$ intend Euclidean distance in features i in add-on to j at ti time.

Moving Probability-Computer Phase: It uses the probability conception to go towards another glowworms with greater luciferin levels. Equation (3) quantifies the chance of a glowworm (feature) migrating towards its neighbour (j1).

$$p_{ij}(t) = \frac{l_{j1}(t) - l_{i1}(t)}{\sum_{k \in N_i(t)} l_k(t) - l_{i1}(t)} \quad (3)$$

Movement Phase: Assume glowworm (feature)i chooses glowworm (feature)j. Equation (4) describes the individual-time simulation of glowworm (feature) i mobility

$$x_i(t+1)= x_i(t) + s(t)\left(\frac{x_j(t)-x_i(t)}{\left\|x_j(t)-x_i(t)\right\|}\right) \tag{4}$$

Here, s denotes size of the step and ||.|| represents Euclidean norm function

Decision Radius Update Phase: Equation (5) specifies the decision radius for each update as trails.

$$r_d^j(t+1)=\min\{r_s, max\{0, r_d^j(t)+\beta(n_t-|N_i(t)|)\}\} \tag{5}$$

Here, β denotes constant, r_s represents glowworm (feature) i sensory radius, and$n_t$signifies factor for neighbor numeric control.

KHM algorithm

To address the clustering problem, the KHM data technique was created. KHM is an alternative to KM that considers the harmonic mean as opposed to the negligible distance amomst data points and the cluster center. The KHM algorithm includes clustering data, cluster centers, a membership function to assign a weight to each data point based on its degree of cluster membership, and a weight function to determine how much weight to assign to each data point when recalculating cluster center parameters.

Here are the steps to follow when using the KHM algorithm:

1. Initially, the algorithm is configured with initial estimates for the centres (C). The beginning points should be randomly selected.
2. Use the following formula to obtain the objective function's value:

$$KHM(X,C) = \sum_{i=1}^{n}\frac{k}{\sum_{j=1}^{k}\frac{1}{\left\|x_i-c_j\right\|^p}} \tag{6}$$

where the input parameter p should be greater than or equal to 2.

1. For every data point x_i, calculate its membership in $m(c_j \,|x_i\,)$ each centre using equation (7).

$$m(c_j \,|x_i\,) = \frac{\|x_i - c_j\|^{-p-2}}{\sum_{j=1}^{k}\|x_i - c_j\|^{-p-2}}, \;\; m(c_j \,|x_i\,) \in [0,1] \tag{7}$$

2. For every data point x_i, calculate its weight $w(x_i)$ in accordance with the equation (8).

$$w(x_i) = \frac{\sum_{j=1}^{k}\|x_i - c_j\|^{-p-2}}{\left(\sum_{j=1}^{k}\|x_i - c_j\|^{-p-2}\right)^2} \tag{8}$$

3. Recompute the location of each center c_j using equation (9) and all the data points x_i memberships and weights:

$$c_j = \frac{\sum_{j=1}^{n} m(c_j \,|x_i\,).w(x_i).x_i}{\sum_{j=1}^{n} m(c_j \,|x_i\,).w(x_i)} \tag{9}$$

4. Repeat steps 2–5 til KHM(X, C) shifts infrequently or until a predetermined scores of iterations have elapsed.
5. Assign every data point x_i to the cluster with the largest $m(cj \,|xi)$.

When using the KHM method, the objective function rests on the conditional likelihood of cluster centre regarding data points, and the corresponding weights of data points are dynamically adjusted throughout each iteration.The KHM algorithm proves particularly adept in scenarios where cluster boundaries are nebulous and indistinct, attributed to its utilization of the membership function.The KHM algorithm addresses the KM algorithm's vulnerability to initial values, but it may still reach a local optimum.

The Proposed Algorithm for Cluster formation

In the conventional KHM algorithm, the distance metric is utilized to calculate the distance between two nodes or two vertices. This measurement only considers their relative positions. However, when creating clusters in VANET, it's essential to factor in the mobility of vehicles, determined by both their positions and velocities. Consequently, a weighted distance metric has been devised to account for these elements.

Introducing a novel clustering algorithm named GSOKHM, which integrates the GSO and KHM algorithms. This hybrid approach preserves the advantages of both GSO and KHM while addressing their convergence and sensitivity challenges. GSO can segment the data points effectively without anterior knowledge, while KHM can derive effective initialization from GSO, enhancing its convergence. The GSO method utilizes single-dimensional arrangement to represent cluster centers as discrete material, with each material or potential result depicted by a d k-cell array indicating the coordinates of all cluster centers. KHM-GSO aims to optimize the partitioning of k-dense, well-separated clusters. In this proposed method, particles execute only one type of motion at a time across two distinct phases. The initial step is to eliminate unfavorable regions of the search space and escape from local optimums. The second phase is convergence to the global optimal solution. These two processes are repeated until a predetermined endpoint is reached (such as when the maximum number of iterations has been reached or when no changes have been made in a certain number of iterations). KHM-GSO uses the mathematical relation value of the glowworm's present position to calculate the glowworm's luminescence, or luciferin. Glowworm employs local-decision areas to identify its neighbors and a probabilistic approach to travel toward a neighbor with a higher luciferin value than its own [22]. The entire search algorithm has been matched with the GSO technique. The location of the glowworm and the objective function value determine the luciferin concentration. The optimally positioned glowworm has a higher luciferin value than the others due to its superior lamination Kumaraguruparan et al.(2024) [23]. Each glowworm examines its immediate surroundings within its tiny decision region and then moves towards its luminous companion. The objective purpose of the KHM-GSO clustering algorithm is the fitness value.

Luciferin is updated based on the fitness value (accuracy) in addition to the previous luciferin value, and its rule is given by an equation.

$$l_i(ti+1) = (1-\rho)l_i(ti) + \gamma Fitness x_i(ti+1) \tag{10}$$

Where, $l_i(t)$ denotes glowworm luciferin (feature) i at time ti, ρ intends constant luciferin decay $(0 < \rho < 1)$, γ denotes luciferin improvement constant, $x_i(ti+1) \in R^M$ signifies glowworm (feature) location i at timein addition $Fitness x_i(t+1)$conception fitness value at glowworm i's position at time $ti+1$.

B. Cluster head selection using Floyd–Warshall algorithm (FWA)

Following the completion of CF, the assortment of the Cluster Head (CH) is undertaken. To guarantee a stable CH, the paper utilizes the Floyd-Warshall Algorithm (FWA). This particular FWA is favored due to its absence of negative cycles, resulting in faster CH selection.

The algorithm is typically employed to compute the shortest paths between all pairs of vertices in a graph. In this context, the emphasis is on highway vehicles, where every link react to a vehicle, and the edges stand for the distances among them. The distance equation can be applied to quantify the distance among couple of vehicles.

$$d = \sqrt{(a_2 - a_1)^2 + (b_2 - b_1)^2} \quad (11)$$

The calculation of the distance between each vehicle's coordinate points is achieved using the previously mentioned formula. The coordinates of the vehicles are denoted by 'a' and 'b', while the distance between them is denoted by 'd' in Equation (6). After computing the distance between each vehicle or node, the Fuzzy Weighted Average (FWA) algorithm is implemented to initiate the Cluster Head (CH) selection process. To select the CH, the FWA algorithm operates on each node within the cluster. Every vehicle within the FWA is called an intermediate node, and that node is used to figure out how far it is from other vehicles. Comparison of the direct distance is used to choose the minimal distance, while bypass distances are computed in the FWA using Equation (12).

$$D_{xy}^{(t)} = min\left(D_{xy}^{(t-1)}, D_{xy}^{(t-1)} + D_{xy}^{(t-1)}\right) \quad (12)$$

The process of Floyd iteration comes to a close once all the vehicles have been selected as intermediate vehicles. The CH is observed by conniving the average distance of each vehicle throughout the entire execution of FWA. The minimum value obtained from this calculation is chosen as the CH.

C. VANET IDS using chaotic maps Elephant Herding Optimization Algorithm (CMEHA-CNN)

A proficient IDS with MCMEHA-CNN has been developed to identify and differentiate malevolent nodes from cooperative ones. Each CH node examines its neighboring node for malicious behavior using MCMEHA-CNN, which is the primary deep learning method that provides high-quality results. By utilizing training and testing data sets, MCMEHA-CNN classifies the tested node as normal or malevolent. The training set selection is critical to the IDS detection accuracy as the weight vectors are proportional to the dataset. The intrusion detection process may experience difficulties due to false correlation traits. To enhance classification, feature selection through principal components analysis is employed to eliminate unnecessary features and select only the most valuable ones. The chosen feature subset is then used to train the IDS to detect SA.

The selected cluster heads are used as inputs for CNNs. CNNs, which are composed of several hidden layers that use convolutions and subsampling allows for the extraction of high-level and low-level characteristics from input data., are one of the most promising DLTs. These networks typically contain three layers namely convolution, sub-sampling or pooling, and complete connection layers. These network's inputs were quad histograms where the networks used input, output, and intermediate layers. The input layers considered features as inputs, outputs sent trained outputs to the system using intermediate layers (hidden layers) as presented in Figure 4. Weight values of the features are optimized to attain precise results in the proposed Convolutional Neural Network (CNN).

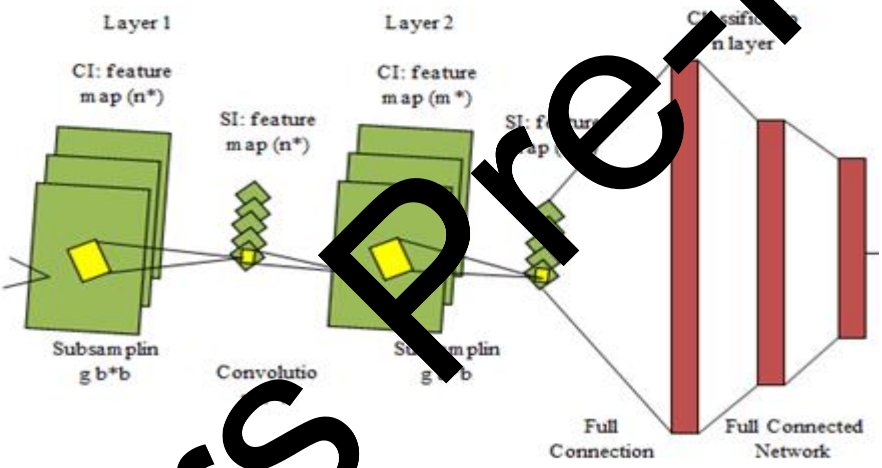


Figure 4: Convolutional Neural Network

Convolution Layer: The process of convolution involves using a kernel to convolve with each block of the input matrix, resulting in a pixel. When an input image and kernel are convolved, a set of n output image features is generated, each consisting of i*i dimensional feature maps. CNNs often incorporate multiple convolution layers, where the feature vector serves as input and output for each layer. The scope of each map's generated feature convolution layer, obtained by convolving with the input, is determined by the no of filters (n) used in the process of convolution. Hence, each filter map represents a distinct attribute at a specific area in the original image. Applying the following formula to the lth convolution layer produces the resulting output as feature maps:

$$C_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{a_i^{(l-1)}} K_{i,j}^{(l-1)} * C_j^{(l)} \quad (13)$$

The bias matrix ($B_i^{(l)}$)and convolution filter ($K_{i,j}^{(l-1)}$) of size a*a connect the feature map (j[th]) in layer (l-1) with the i[th] feature map on same layer. The result $C_i^{(l)}$ layer comprises of multiple feature maps. The primary convolutional layer $C_i^{(l-1)}$ represents the input space $C_i^{(0)} = X_i$ in equation (14). The kernel create the feature map through convolution. Nonlinear transmutation of the Convolutional layer's output can be achieved by using an activation function afterward,

$$Y_i^{(l)} = Y(C_i^{(l)}) \quad (14)$$

$Y_i^{(l)}$ is the response on application of the stimulation purpose to the input $C_i^{(l)}$. Sigmoid, tanh, and rectified linear units (ReLU) are the frequently used energizing functions. This study employs ReLUs; their notation is $Y_i^{(l)} = max\ (0, Y_i^{(l)})$. This function is commonly integrated into deep learning models owing to its effectiveness in mitigating interactions and nonlinear personal effects. When the input signal is negative, ReLU sets the output signal to 0, while it retains the same value for positive inputs. ReLU surpasses other activation functions because the error derivative in the saturation zone is exceedingly small, facilitating substantially faster training. This phenomenon, known as the "problem of vanishing gradient," is characterized by adjustments to the practical nn weights.

Two-Layer Sampling: The main intention of this layer is to cut down the spatiality of the feature maps produced by the preceding convolutional layer. A mask of dimension b is selected, and the subsampling technique is employed between the front and the feature maps to attain the desired outcome. Subsequently, thanks to the sub-sampling layer, the convolutional layer becomes more robust against image transformations. In the proposed approach, the harmonic mean of the feature weights is utilized to adjust the optimal weights. The calculation is outlined as follows:

$$\text{Weighted Harmonic mean } w_H = \frac{N}{\sum_{i=1}^{N} wx_i} \quad (15)$$

where N- Scores of features, w- feature Weight , and x_i- Features.

Fully Connected layer: The output layer employs softmax activation function. Softmax activation function is used to evaluate the model's reliability. It is computed as follows,

$$Y_i^{(l)} = f(z_i^{(l)}), \text{ where } z_i^{(l)} = \sum_{i=1}^{m_i^{(l-1)}} w_H y_i^{(l-1)} \tag{16}$$

In this context, w_H represents the features weighted harmonic mean of that the fully connected layer must adjust to construct the representation of each class comprehensively. The function f denotes the activation function that introduces nonlinearity. Within the proposed system, the input image undergoes classification into three classes: background, crop, and weed. The CNN algorithm can be outlined here:

Step 1: Input the image dataset and process the training set's images using the specified filter size, creating the data matrix image X.

Step 2: Set the weight values $w^{(l)}{}_{i,j}$ and bias bi, also utilize the TensorFlow kernel purpose $K_{i,j}^{(l-1)}$ to initiate parallel operations.

Step 3: Apply the Conv2d function to perform a two-dimensional convolution operation, resulting in the generation of the initial convolutional feature matrix $X^{(1)}$.

Step 4: Use the pooling sheet to perform a pooling operation on the initial convolutional feature matrix X(1) and acquire the feature matrix $X^{(2)}$.

Step 5: Utilize the CMEHA optimizer to compute the learning rate , Adjust the weight w_i and bias b_i using TensorFlow weights and the update-bias interface to acquire the feature matrix $X^{(3)}$. It is crucial to note that this process does not involve the usage of an AI-powered assistant.

Step 6: Repeat Steps 3, 4, and 5 to generate the second convolution and obtain the feature matrix $X^{(4)}$.

Step 7: Convert the feature matrix X(4) to a columnar vector, then multiply the weight matrix by the bias at the neurone in the layer that is fully connected. Use the Leaky ReLU function for activation to get the resulting eigenvector b1. Use equation (15) to get the harmonic mean weighted (wH).

Step 8: To use the dropout layer, input the fully attached layer and compute the neuron's output measure using equation (10).

Step 9**:** To achieve the results, use the input and the Softmax classifier output.

i. **Chaotic maps Elephant Herding Optimization Algorithm (CMEHA) to update the weight**

A technique called EHO takes inspiration from how elephants behave in groups to solve optimization problems. This method involves organizing elephants into clans, which are then combined to create the overall population. Some male elephants are designated to separate from their clans and move away from the main group during each formation. The elephants are supported within their clans by a matriarch.

The EHA algorithm is a population-focused, modern method inspired by elephants' herding behavior. The operators comprising this technique are called "clan updating operators" and "separating operators," respectively. In various search space configurations, the EHA has demonstrated its ability to locate the optimal solution. Initially, two distinct chaotic maps were introduced to the EHA to enhance search quality and the system's performance in certain circumstances. The updated version is known as CMEHA. The EHA is based on the idea that numerous lineages of elephants coexist under the leadership of a matriarch, with a specific number of elephants in each clan. Each family is assumed to contain the same number of elephants for this model. Using an updating operator, the relative positions of elephants within a clan are modified to reflect their relationship with the matriarch. A percentage of males from each generation of elephants will abandon their families and live alone. In the EHO's procedure of updating, a separation operator is implemented. In most elephant households, the matriarch is the oldest surviving female elephant; she is also regarded as the most competent member of the population when it comes to modeling and solving optimization problems.

Step 1: The elephant population should be initialised with j clans. Every clan 'c' member 'a' moves, as directed by matriarch $'S_m'$ with the highest fitness level in the generation, which can be technically represented as

$$W_{new,S_m,a} = W_{S_m,a} + \lambda(W_{best,S_m} - W_{S_m,a}) \times R \tag{17}$$

where $W_{new,S_m,a}$ represents the new location of an inside c, $W_{S_m,a}$ represents the previous location, and W_{best,S_m} represents the optimal solution to the equation $E_m, \lambda \in [0,1]$. R stands for the random number used to increase the population's variety and, the parameter of the algorithm that determines the matriarch's effect.

Step 2: Update the position of the finest elephant on clan W_{best,s_m} employing the Eq. (18)

$$W_{new,s_m} = \chi \times W_{Center,s_m} \tag{18}$$

Here, $\chi \in [0,1]$ inferred the 2nd parametric quantity of the algorithm that defines the consequence of W_{Center,s_m} which is represented as

$$W_{Center,s_m,d} = \frac{1}{a_{s_m}} \times \sum_{j=1}^{a_{F_m}} b_{s_m}, j, d \tag{19}$$

where a_(s_m) represents the total definite quantity of elephants in clan s_m and $1 \leq d \leq$ D represents the dth dimension of space.

Step 3: The ME who abandon their clan are engaged for exploration design. Within each clan \(c\), certain elephants exhibiting inferior fitness values are assigned subordinate roles.

$$W_{worst,s_m} = W_{min} + (W_{max} - W_{min} + 1) \times R \tag{20}$$

where P_(min) represents the smallest SS, P_(max) represents the largest SS, and R [0,1] represents uniformly distributed random integers.

Step 4: The proposed method relies on arbitrary sequences centred on chaos mapping instead of random integers. Given that CM generates numbers with non-repetition and ergodicity, the enhanced search is to be expected. Using the proposed CM, a chaotic numerical sequence was generated. In this instance, '2' distinct kinds of one-dimensional maps are compared: (i) the circular map and (ii) the sinusoidal map. The following equation describes a circular-shaped map:

$$W_{new+1,s_m} = \left[W_{new,s_m} + p - \frac{q}{2\pi}\sin\left(2\pi W_{new,s_m}\right)\right] mod\ 1 \tag{21}$$

When p = 0.2 and q = 0.5 , the result of chaotic sequence is between 0 and 1. The following is the expression for the sinusoidal map:

$$W_{new+1,s_m} = q(W_{new,s_m})^2 \sin\left(\pi W_{new,s_m}\right) \tag{22}$$

consequently, the following reduced form functions well for q = 2.3:

$$W_{new+1,s_m} = sin\left(\pi W_{new,s_m}\right) \tag{23}$$

Swapping the arbitrary numbers in the EHA with the numbers from the chaos sequences is a viable option. If any malicious activity is detected, the respective CH is informed for a final decision. Otherwise, the data from regular nodes is encrypted and securely stored in the cloud to safeguard the information of the nodes.

D. VANET Security using using Secure Hashing Algorithm (SHA2) - Elliptic curves cryptography (ECC) called (SHA2-ECC)

Elliptic curves cryptography (ECC) is a cryptographic technique that is both computationally efficient and highly secure, providing superior protection to algorithms that rely on larger key sizes and more intricate mathematical proofs. The ECC algorithm is more secure than other cryptographic methods, but its complexity and implementation difficulty make it less secure overall. In response, SHA2-ECC has been proposed as a replacement for ECC. SHA-2 can be subdivided into variants that generate hashes of varying lengths, whereas MD5 can only generate 128-bit hashes making it a more reliable and secure encryption algorithm.

ECC generates public and private keys but SHA2-ECC creates hashes of varying lengths to enhance the system's security.

Step 1: Considered a curve's origin Bp . Generate the public key A by using Equation (20).

$$A = (K \times B_p) \qquad (24)$$

where K is a private key selected in the range of (1, n 1) inclusive.

Step 2: To produce a new secret key by including the compound property to the public key and then applying the SHA2 hash function to the public key. In this instance, padding is employed to guarantee that the input message can be stored in "n" consecutive 512-bit blocks.

An SHA-2 cipher computation is fundamentally comprised of two halves.

- The input message is padded or divided into blocks of a specific size, and the block count is transmitted to subsequent components. Each block contains 16 message words, with a message word size of 32 bits for SHA-224/SHA-256 and 64 bits for the other four algorithms.
- The digest function iteratively determines the hash values. By requiring a loop-carried dependence, this method precludes this section from attaining II=1.

The generateMsgSchedule module is responsible for constructing the sequential message word stream. In contrast, the dup_strm module is used to duplicate the number of block streams.

$$S_k = SHA2(A \parallel S_d) \tag{25}$$

wherein S_d indicates that the salt value is determined at random.

Step 3: Encrypting the data using both the public key (a curve point) and the private key (a secret). The encryption formula, which incorporates the key, is an integral component of the SHA2-ECC algorithm. The encrypted data comprises of two ciphertexts, which can be represented as follows:

$$E_1 = (R * B_p) + S_k \tag{26}$$

$$E_2 = D + (R * A) + S_k \tag{27}$$

In this encryption process, E1 refers to the first encrypted text, while E2 refers to the second encrypted text. R represents a random number within the range of [1, n-1], and D represents the data. Decryption is the process of obtaining the original information.

Step 4: To decrypt data, you must employ the same method used to encrypt it. The process of decryption can be mathematically described as the subtraction of the confidential key from the decryption equation.

$$D = ((C_2 - K) * C_1) - N_k \tag{28}$$

4 RESULT AND DISCUSSION

Detecting and preventing attacks in VANET holds equal significance. To accomplish this goal, a methodology has been devised and implemented utilizing Network Simulators-2 (NS-2). Multiple experiments have been conducted to assess the efficacy of the proposed Sybil Attack (SA) detection technique. The proposed methodologies, namely CMEHA-CNN and MD5-ECC, have been evaluated against existing approaches based on specific quality metrics. The execution of the projected methodologies has been calibrated using a publicly accessible dataset. 80% of the dataset has been allocated for training, while the remaining 20% has been reserved for testing purposes.

4.1: Analyzing the Performance of CMEHA-CNN.

A comparison was made between the performance of CMEHA-DNN and other conventional classifiers such as DNN, ANN, SVM, and K nearest neighbor (KNN) to evaluate its accuracy and efficiency. The assessment of its performance was based on seven quality performance metrics including precision, accuracy, specificity, F-measure, recall, false negatives rates (FNR), and the false positives rate (FPR).

Precision: Precision is the fraction of relevant matches among the retrieved matches.

$$\boldsymbol{Precision} = \frac{TP}{(TP+FP)} \tag{29}$$

Specificity: Specificity is defined as the likelihood of negative matches, assuming they are actually negative.

$$\boldsymbol{Specificity} = \frac{TN}{(TN+FP)} \tag{30}$$

F-measure: The harmonic mean of recall and precision is F-measure.

$$\boldsymbol{F-measure} = \frac{2*Precision*Recall}{Precision+Recall} \tag{31}$$

False Positive Rate (FPR): The false-positive rate (FPR) is calculated by splitting the entire amount of negative cases misclassified as positive by the entire amount of negative instances.

$$\boldsymbol{FPR} = \frac{FP}{(FP+TN)} \tag{32}$$

False Negative Rate (FNR): When a test falsely indicates the absence of a condition when one is present, the FNR is determined.

$$\boldsymbol{FNR} = \frac{FN}{(FN+TP)} \tag{33}$$

Accuracy: The proportion of accurately predicted class labels to total class labels.

$$\boldsymbol{Accuracy} = \frac{TP+TN}{(TP+TN+FP+FN)} \tag{34}$$

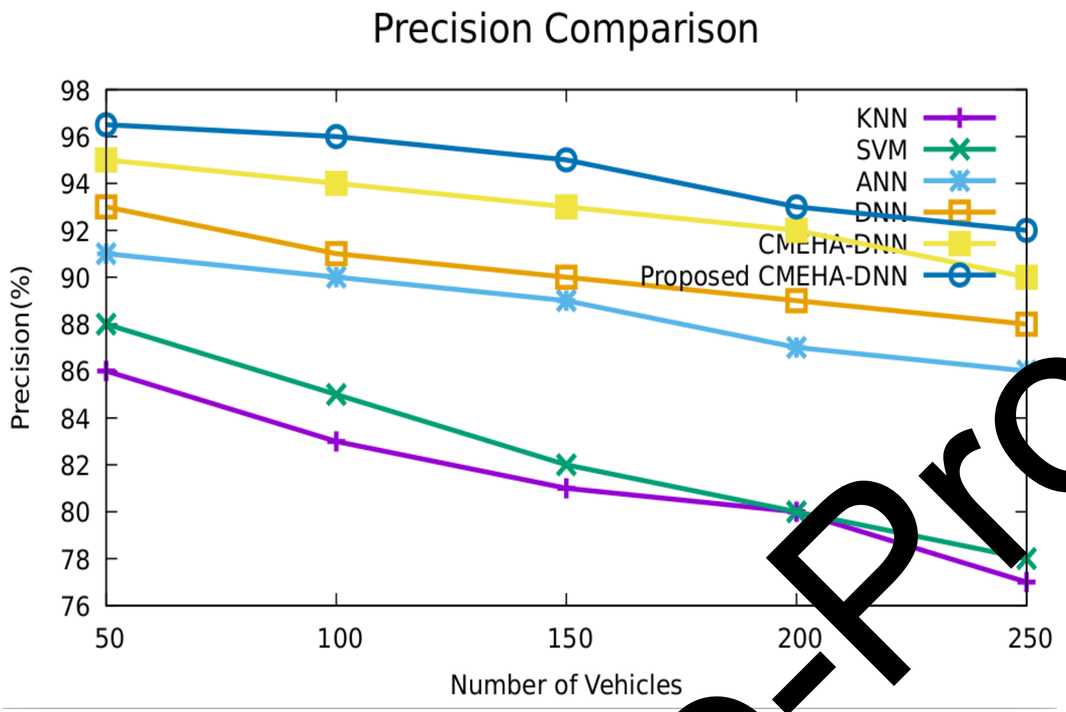


Figure 5: Precision Comparison

Figure. 5 shows a comparison of the CMEHA-CNN with five other classifiers in terms of precision.The proposed classifier surpasses existing classifiers, as evidenced by graphical analysis, across a vehicle count range of 50 to 250. For vehicle counts within this range, the CMEHA-CNN achieves precision and recall rates of 98%, 97.59%, 96.92%, 94.99%, 93.29%, and 92.05%, respectively. In contrast, precision and recall rates of existing classifiers fall below those of the proposed classifier. Furthermore, the proposed classifier exhibits higher accuracy in detecting Sybil Attacks compared to active classifiers.

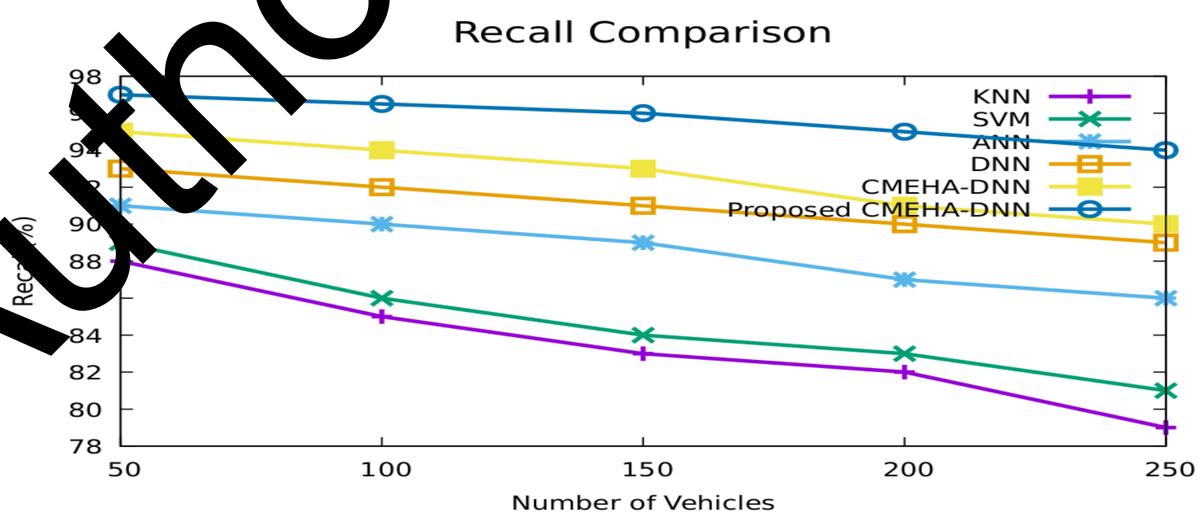


Figure 6: Recall Comparison

Figure 6 shows the recall examination of the proposed and existing approach. It is shown that, the proposed CMEHA-CNN provides better recall value when compared with the other existing approaches like CMEHA-DNN, DNN, ANN, SVM and KNN.

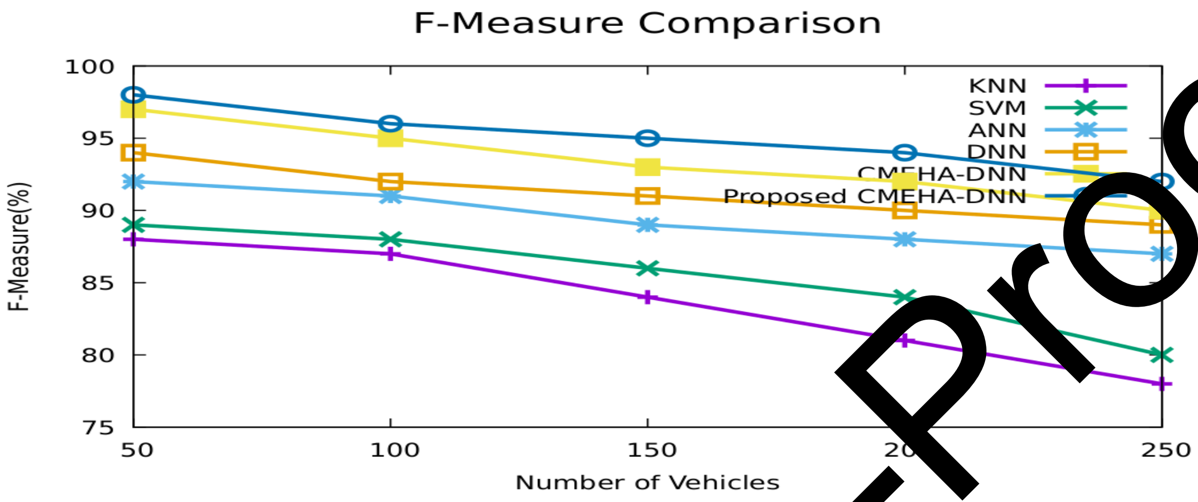


Figure 7: F-measure Comparison

The comparison of f-measure of the proposed and existing approach is shown in the Figure 7. It is shown that, the proposed CMEHA-CNN provides better recall value when compared with the other existing approaches like CMEHA-DNN, DNN, ANN, SVM and KNN.

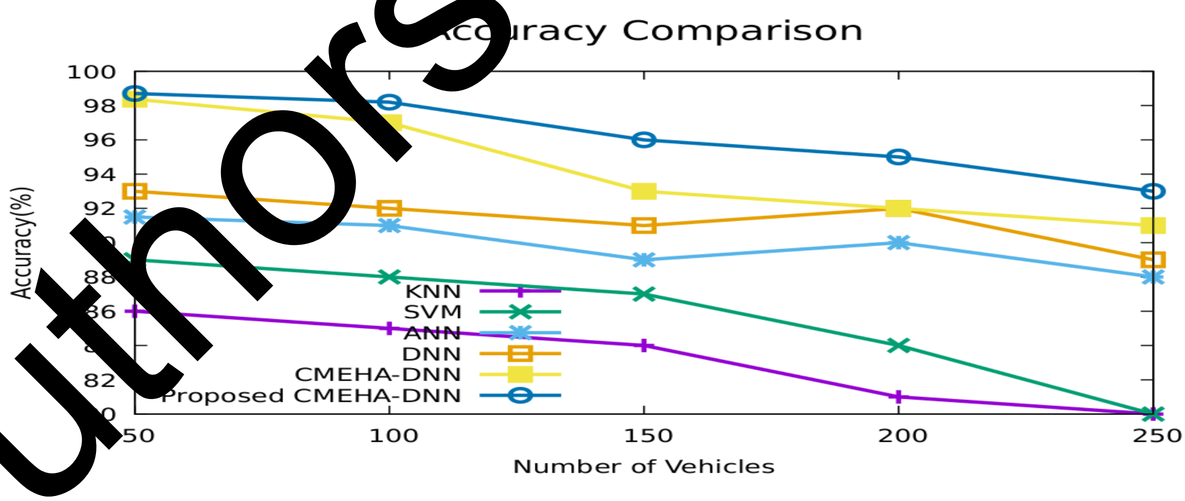


Figure 8: Accuracy Comparison

Figure 8 depicts the accuracy comparison between the projected approach and existing methods. The outcome demonstrate that the projected CMEHA-CNN surpasses other existing approaches, including CMEHA-DNN, DNN, ANN, SVM, and KNN, achieving an accuracy level of 98.7%.The proposed approach's accuracy is 1%, 2.26%, 3.71%, 9.56%, and 11.15% when compared to CMEHA-DNN, DNN, ANN, SVM, and KNN. The CMEHA-CNN technique provides superior SA recognition in VANET compared to the existing classifier. These findings provide support for the efficacy and efficiency of the proposed CMEHA-CNN method for detecting SAs in VANET.

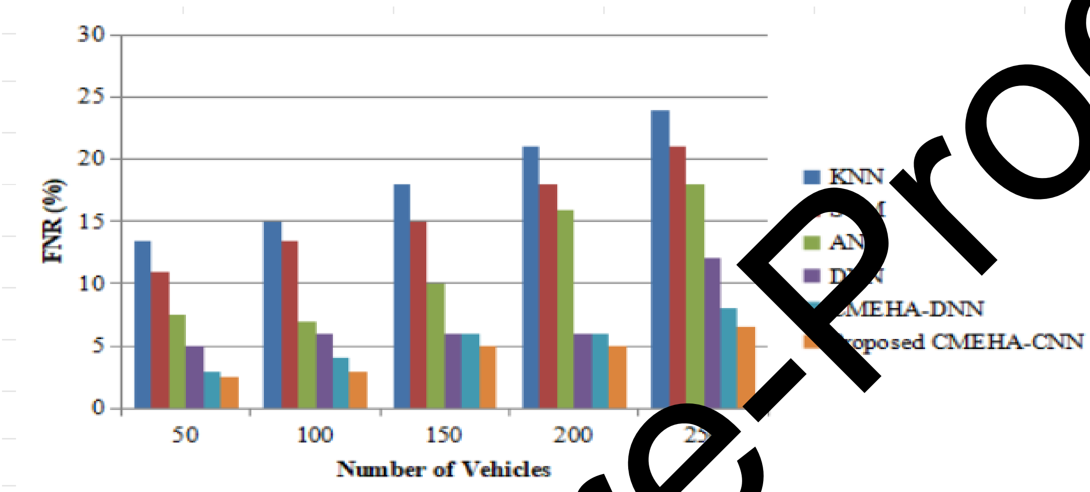


Figure 9: FNR Comparison

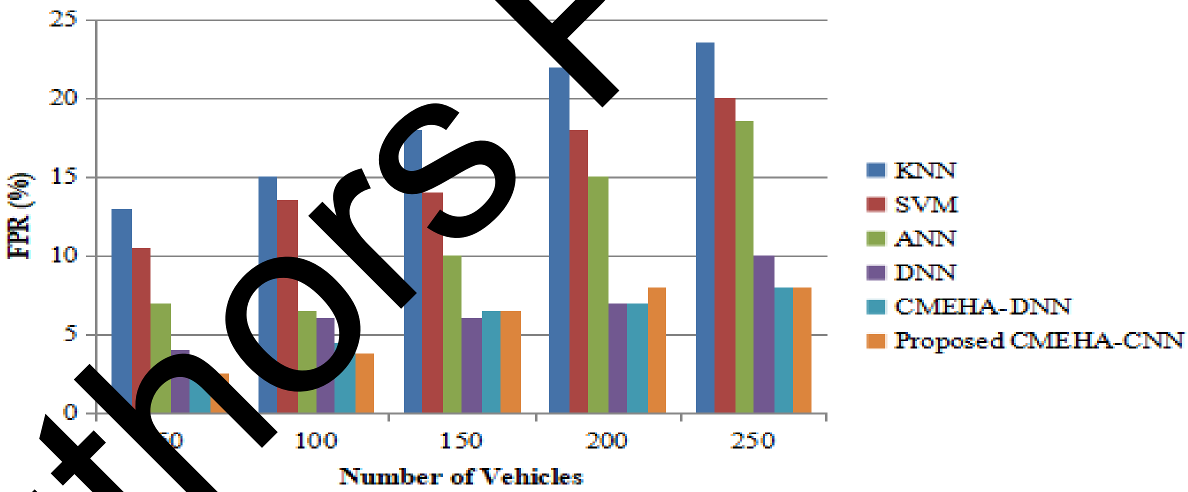


Figure 10: FPR Comparison

In Figure 9 and Figure 10, we can see the FNR and FPR values for both proposed and existing classifiers. The computation of these values was performed by varying the number of vehicles from 50 to 250. The proposed classifier had a low FNR of only 3%, whereas the existing techniques had a much higher FNR value for 50 vehicles, this means that the existing classifiers mistakenly predicted attacked nodes as normal ones, while the proposed method had a low FNR value. For 250 vehicles, the CMEHA-CNN classifier had an FNR value of only

8%, which is also low compared to other methods. Therefore, we can conclude that the CMEHA-CNN classifier is more accurate than existing classifiers in detecting attacks.

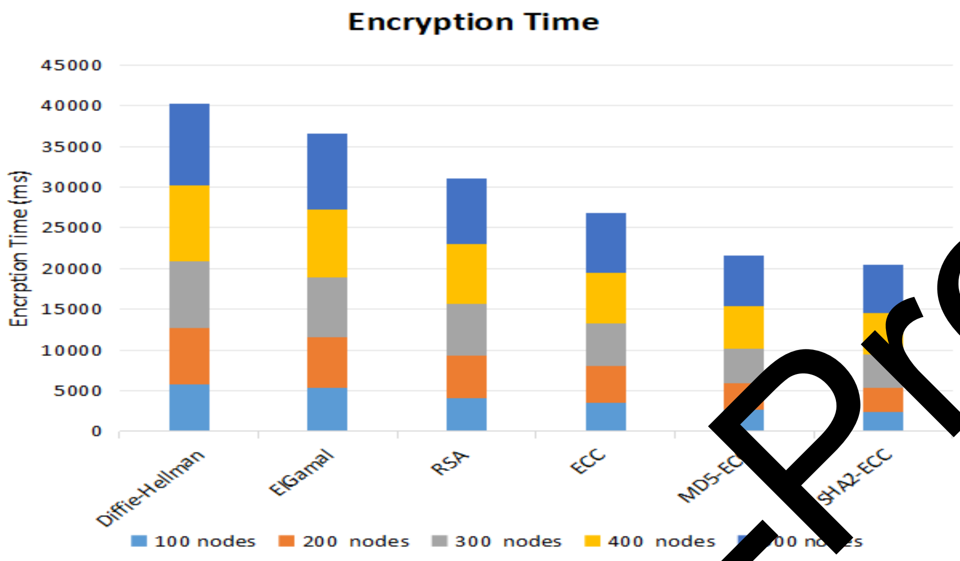


Figure 11: Encryption Time

Figure 11 and Figure 12 display the results of a comparison between the execution of the cryptographic algorithm MD5-ECC and that of other algorithms, such as Rivest-Shamir-Adleman (RSA), Diffie-Hellman, ECC, and ElGamal, to ensure the security of the proposed method.

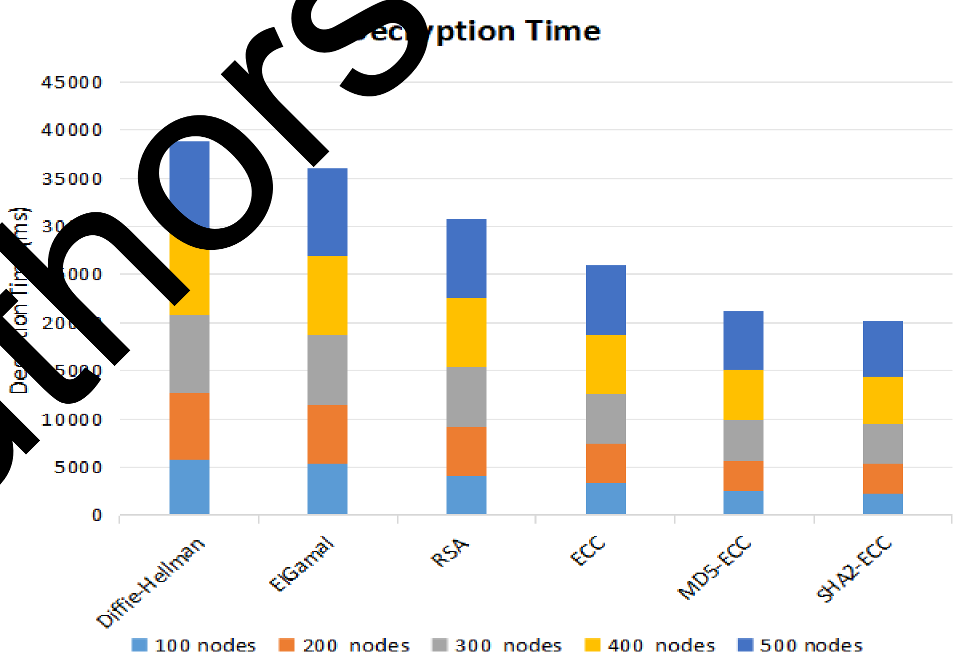


Figure 12: Decryption Time

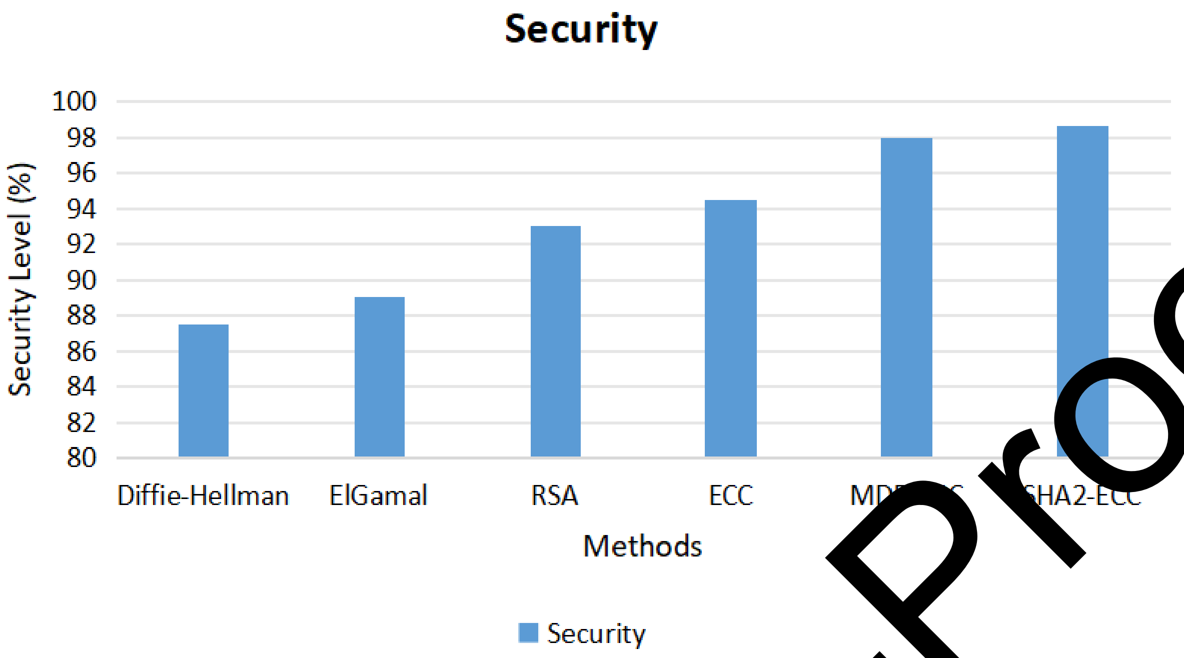


Figure 13: Security Level

Figure 13 compares the level of security achieved by the proposed SHA2-ECC to the other five existing approaches. Five existing algorithms obtain security levels of 92.2%, 94.32%, 92.98%, 90.01%, and 87%, respectively; the proposed SHA2-ECC algorithm achieves a level of 94%. Existing methods provide a higher level of security, whereas the proposed procedure offers an additional 4%. This demonstrates that the proposed SHA2-ECC is highly secure, as it prevents the VANET from accessing the SA.

5 CONCLUSIONS

VANET is proposed with a original clustering algorithm, an IDS framework based on deep learning, and a CH election mechanism. The steadiness of the IDS framework is secured by the proposed clustering, which generates stabilized vehicular communities with reinforced connection between member vehicles. The system's efficacy is evaluated by drawing parallels between the proposed classifiers, cryptographic algorithms, and several prevalent methods. By altering the number of vehicles, specificity, recall, accuracy, F-measure, precision are determined for the proposed CMEHA-CNN. Encrypting and decrypting durations for the proposed SHA2-ECC are defined over a broad spectrum of node counts. The results of the experimentation demonstrated that the novel strategies accomplish the old ones. CMEHA-DNN has attained an accuracy of 98.7%. In addition, the proposed SHA2-ECC obtains a security level of 98.5%. These outcomes confirmed that the proposed system effectively identifies and categorizes SA, thereby protecting the VANET environment from intrusion by malignant actors.

Conflicts of Interest: The authors declare no conflict of interest

Data Availability Statements: The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Funding: Not applicable

Consent to publish: All the authors gave permission to Consent to publish.

REFERENCES

1. M. H. Alwan, K. N. Ramli, Y. A. Al-Jawher, A. Z. Sameen, and H. F. Mahdi, “Performance comparison between 802.11 and 802.11p for high speed vehicle in VANET,” International Journal of Electrical and Computer Engineering (IJECE), vol. 9, no. 5, pp. 3687–3694, Oct. 2019, doi: 10.11591/ijece.v9i5.pp3687-3694.
2. M. N. Mejri, J. Ben-Othman, and M. Hamdi, “Survey on VANET security challenges and possible cryptographic solutions,” Vehicular Communications, vol. 1, no. 2, pp. 53–66, Apr. 2014, doi: 10.1016/j.vehcom.2014.05.001.
3. N. K. Chaubey and D. Yadav, “A taxonomy of sybil attacks in vehicular ad-hoc network (VANET),” in IoT and Cloud Computing Advancements in Vehicular Ad-Hoc Networks, 2020, pp. 174–190.
4. F. Perry, K. Raboy, E. Leslie, Z. Huang, and D. Van Duren, “dedicated short range communications roadside unit specifications," National operations center of excellence. 2017, FHWA-JPO-17-589, Accessed: Dec, 2020. [Online]. Available: https://transportationops.org/publications/dedicated-short-range-communications-roadside-unit-specifications
5. M. Rahbari and M. A. Jabreil Jamali, “Efficient detection of sybil attack based on cryptography in VANET,” International Journal of Network Security & Its Applications, vol. 3, no. 6, pp. 185–195, Nov. 2011, doi: 10.5121/ijnsa.2011.3614.
6. S. Pal, A. Mukhopadhyay, and P. Bhattacharya, “Defending mechanisms against Sybil attack in next generation mobile ad hoc networks,” IETE Technical Review, vol. 25, no. 4, pp. 209–215, 2008, doi: 10.4103/0256-4602.42813.
7. N. K. Chaubey, “Security analysis of vehicular ad hoc networks (VANETs): A comprehensive study,” International Journal of Security and Its Applications, vol. 10, no. 5, pp. 261–274, May 2016, doi: 10.14257/ijsia.2016.10.5.25.

8. S. Tarapiah, K. Aziz, and S. Atalla, “Analysis the performance of vehicles ad hoc network,” Procedia Computer Science, vol. 124, pp. 682–690, 2017, doi: 10.1016/j.procs.2017.12.205.
9. Hamdan, S., R. Al-Qassas, and S. Tedmori, Comparative Study on Sybil Attack Detection Scheme
10. Gao Y, Wu H, Song B, Jin Y, Luo X, Zeng X (2019) A distrib-
uted network intrusion detection system for distributed denial of service attacks in vehicular Ad Hoc network. IEEE Access 7:154560–154571.
11. Jianfeng Ma et al., “Attribute-Based Secure Announcement Sharing Among Vehicles Using Blockchain,” IEEE Internet of Things Journal, vol. 8, no. 12, pp. 10873-10883, 2021.
12. S. Dhanasekaran, S. Ramalingam, K. Baskaran & P. Vivek Karthick (2023) Efficient Distance and Connectivity Based Traffic Density Stable Routing Protocol for Vehicular Ad Hoc Networks, IETE Journal of Research, DOI: 10.1080/03772063.2023.2252385
13. Shi-Jinn Horng, Cheng-Chung Lu, and Wanlei Zhou, “An Identity-Based and Revocable Data-Sharing Scheme in VANETs,” IEEE Transactions on Vehicular Technology, vol. 69, no. 12, pp.15933-15946, 2020.
14. Geetanjali Rathee et al., “Trusted Computation Using ABM and PBM decision models for ITS,” IEEE Access, vol. 8, pp. 195788-195798, 2020.
15. Yaroslav Meshcheryakov et al., “On Performance of PBFT Blockchain Consensus Algorithm for IoT-Applications with Constrained Devices,” IEEE Access, vol. 9, pp. 80559-80570, 2021.
16. Subba B, Biswas S, Karmakar S (2018) A game theory based multi layered intrusion detection framework for VANET. Future Gener Comput Syst 82:12–28.
17. Shu J, Zhou L, Zhang W, Du X, Guizani M (2020) Collaborative intrusion detection for VANETs: a deep learning-based distributed SDN approach. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/tits.2020.3027390
18. Suganyadevi, K., & Nandalal, V. (2023). Swarm Intelligence-Inspired Meta-Heuristics Hybrid Optimization for Multi-Constraint Routing in Vehicular Adhoc Networks. IETE Journal of Research, 70(1), 95–115. https://doi.org/10.1080/03772063.2023.2253763
19. Velayudhan, N. C., Anitha, A., & Madanan, M. (2021). Sybil attack detection and secure data transmission in VANET using CMEHA-DNN and MD5-ECC. Journal of Ambient Intelligence and Humanized Computing, 1-13.
20. B. Zhang, M. Hsu and U. Dayal, *K*-harmonic means – a spatial clustering algorithm with boosting, in temporal, spatial, and spatio-temporal data mining, in: *Temporal, Spatial, and*

Spatio-temporal Data Mining, J. Roddick and K. Hornsby, eds., pp. 31–45, Springer, Berlin, 2001.

21. Zainal N, Zain AM, Radzi NHM, Othman MR (2016) Glowworm swarm optimization (GSO) for optimization of machining parameters. J Intell Manuf 27(4):797–804.
22. Kumaragurubaran, S., & Vijayakumar, N. A novel swarm intelligence-based fuzzy logic in efficient connectivity of vehicles. International Journal of Communication Systems, e579[illegible].