This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/OJITS.2022.1234567

Position Verification in Connected Vehicles for Cyber Resilience Using Geofencing and Fuzzy Logic

Maria Drolence Mwanje *, **Omprakash Kaiwartya** *, **Abdallah Naser** *

[1]Department of Computer Science, Nottingham Trent University, Nottingham NG11 8NS, UK

CORRESPONDING AUTHOR: Omprakash Kaiwartya (e-mail: omprakash.kaiwartya@ntu.ac.uk)

ABSTRACT Position verification is essential in connected and autonomous vehicle technology to enable secure vehicle-to-everything communication. Previous attempts to verify location information have used specific hardware, traffic parameters, and statistical model-based techniques dependent on neighbouring vehicles and roadside infrastructure and whose judgements can be influenced by untrustworthy entities. Considering the back-and-forth communications during verification, these techniques are also unsuitable in the dynamic vehicular networking environment. In this context, this paper proposes a self-reliant trust-based position verification technique using dynamic geofencing, neural network, and Mamdani fuzzy logic controller. The method uses vehicular dynamics, such as distance between the sender and receiver vehicles, magnitude of the speed difference, and direction, to verify the trustworthiness of vehicle positions. An experimental analysis of a dataset of simulated driving scenarios in MATLAB demonstrates that the feedforward neural network records the highest direction classification performance at 99.8% in conjunction with the centroid defuzzification method. Subsequently, further quantitative analysis, including the Receiver Operating Characteristic curve with Area Under Curve and trust level distribution histograms, indicates that the suggested classification model outperforms a random classifier and effectively identifies false position data from the actual during trust computation.

INDEX TERMS Defuzzification, Fuzzification, Fuzzy logic, Geofencing, Location verification, Position verification, Time of Flight, Time Difference of Arrival, Trust computation

I. INTRODUCTION

THERE is a growing dependability and demand for location-based applications and services in connected and autonomous vehicles (CAV) traffic environments. Such services include emergency incident reporting, collision avoidance systems, information geocasting, driver assistance systems, and secure vehicle-to-everything (V2X) communications. Trustworthy location information ensures explicit and timely decision-making, which improves road reliability and safety. However, untrustworthy location information can lead to collision risk, navigation errors, system safety and human life compromise. Various location-based attacks include position spoofing, replay attacks, Sybil attacks, and false position dissemination caused by perception errors and untrustworthy vehicles. However, this study focuses on false position dissemination attacks. The researchers in [1], [2] explored the effect of false position on ad hoc geographic vehicular routing. The study results indicated reduced network performance due to a drop in the packet delivery ratio with an increase in the number of untrustworthy vehicles with false positions. A similar study by [3] investigated the effect of forged positions on channel utilisation, packet delivery, and vehicle speed. The simulation results indicated that untrustworthy positions might reduce vehicle speed due to the perception of congestion on nearby roads, reduced packet delivery, and increased packet collisions.

Previous studies have suggested solutions to verify the position of vehicles in connected and autonomous vehicle environments. According to [4], these solutions are classified into hardware, traffic parameters, and statistical model-based techniques, including received signal strength, plausibility checks, distance bonding, and time of flight. A study by [5] suggested a transferable belief model that verifies location information for geocast routing in VANETS. Firstly, the researchers used the tile-based verification model to ascertain the location of the tiles within the transmission range. After the verification, the researchers calculated the belief of the presence of a vehicle as a neighbour. Simulation

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

results indicated adequate location verification in map-based realistic environments. The researchers in [6] suggested an anonymous and cooperative position verification framework for basic safety messages (BSMs). The study implemented a central location authority to verify the claimed vehicle positions based on nearby vehicle reports. Despite the promising results, the centralised verification model is a limitation due to the highly dynamic vehicular environment and depends on nearby vehicle reports. Another study [7] suggested a position verification model that implements the Time Difference of Arrival (TDoA) multilateration method. A receiver (observer) vehicle records the Time of Arrival (ToA) for each post-crash notification message from the sender three consecutive times along the trajectory transmission range and computes the sender's location using the TDoA multilateration technique. The limitation of this study is that there is an increase in false sender position estimates with an increase in distance between the receiver and the sender. The large distance gap between the sender and receiver delays position estimation, rendering it appropriate for dense or static traffic environments.

A previous study by [8] suggested a cooperative location verification model based on the distance calculated using the ToF for challenge-response messages between the sender (verifier) and next-hop neighbour (prover). The sender verifies the location of its neighbours by calculating the distance to the prover using the ToF during the challenge-response process. The prover then selects the next hop neighbour, known as the cooperator, that verifies the location of the prover using the same challenge-response architecture. The limitation of this model is that it relies on nearby vehicles during the verification and introduces a delay due to the challenge response communications. A similar study by [4] suggested using one-hop neighbour information to verify the claimed position based on various plausibility checks and the strength of the received signal in sparse environments. Despite the reasonable performance of the model as per the simulations, untrustworthy neighbours can corrupt the majority opinion. Furthermore, in the absence of neighbouring vehicles, the feasibility of the model is based on the received signal strength measurements used to calculate the distance; however, these are affected by obstacles and surroundings. Recent studies advocate the implementation of artificial intelligence in location verification for vehicular networks. The researchers in [9] presented a model that uses neural networks to verify location information based on the Time of Arrival (ToA) measurements. Despite the promising results of the above location verification models, the following limitations illustrated in Fig.1 still exist, making them infeasible in real-world vehicular environments.

1) Depend on nearby vehicles or infrastructure for external position validation, making them impractical in isolated areas and susceptible to influence from untrustworthy vehicles.

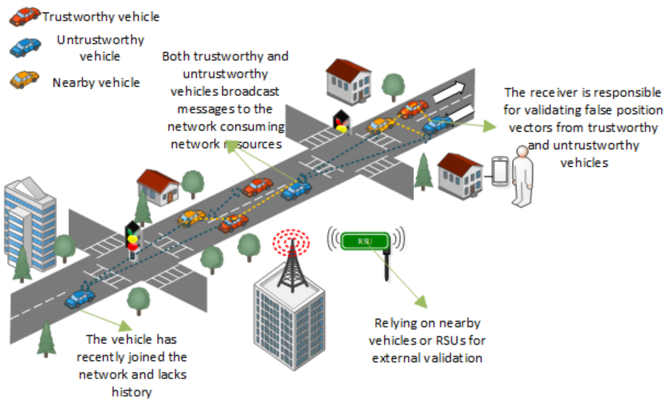

Trustworthy vehicle
Untrustworthy vehicle
Nearby vehicle
Both trustworthy and untrustworthy vehicles broadcast messages to the network consuming network resources
The receiver is responsible for validating false position vectors from trustworthy and untrustworthy vehicles
RSU
The vehicle has recently joined the network and lacks history
Relying on nearby vehicles or RSUs for external validation

FIGURE 1: A graphical representation of the current limitations within trust position verification models.

2) The receiver accepts false position data from trustworthy and untrustworthy vehicles.
3) Both trustworthy and untrustworthy vehicles send messages to the network, which consumes network resources.
4) The absence of a communication history for vehicles that have recently joined the network makes it challenging to identify the trust level based on previous communications.

Considering the recent drive towards trust computation in vehicular ad-hoc networks, researchers are implementing security solutions that evaluate the trust of vehicles [10]. These solutions are known for their ability to identify internal attackers with untrustworthy intent even after passing the identity check and require less computational resources [11]. Therefore, to address these position verification challenges, this paper proposes a novel trust-based position verification system that computes the trustworthiness of the position vectors based on the information provided by the sender and receiver. In brief, below are the key contributions of this paper:

1) Examine various machine learning classification models to pinpoint the most effective technique for identifying the direction of vehicles on the network.
2) Assess the effectiveness of various defuzzification methods to determine the desirable fuzzy logic approach for the model.
3) Implementation of a Mamdani fuzzy logic controller for assessing sender trustworthiness utilising positional characteristics.
4) Conduct experiments to identify the effectiveness and performance of the system in the presence of untrustworthy position data.

The remainder of this paper is organised as follows. Section II critically reviews related literature on geofencing and position verification. Section III presents the details of the proposed position verification model using geofencing and

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

fuzzy logic. Section IV discusses the performance evaluation considering different simulation setups and analysis of results. Section V concludes the paper with future research directions.

II. LITERATURE REVIEW

This section reviews literature subdivided into two subsections: geofencing and related work.

A. GEOFENCING

According to [12], geofencing [1] approach can be used to identify occurrences of mobile objects within the vicinity of geographical zones. The focal point of interest is the location of one vehicle within each circular boundary formed by a known radius. A vehicle is outside the Geofence if the distance between it and the focal point exceeds the circle's radius. While the vehicle transitions, the radius remains consistent, yet the point of interest, namely the vehicle's position, changes to form a dynamic geofence. Previous researchers have defined the geofencing techniques such as geofence area, proximity with point of interest and route adherence [13], [14] for tracking and tracing systems such as human activity monitoring, fleet and freight management. Studies such as [15] implement static geofences in Unmanned Aerial Vehicles where the destination is predefined based on the known point of interest and the radius. Geofencing within this study is used to identify the no-fly zones based on the static information. A recent study by [16] proposed a smart GPS geofencing system for various tracing and tracking applications. Like earlier studies, this system needs a predefined geofence area. However, given the VANET environment's dynamic nature, there is no static point of interest because vehicle positions constantly change along the network. Therefore, it is crucial to create dynamic geofences around the receiver's current location as the point of interest and assess the sender's location based on whether it falls within the geofence area.

B. RELATED WORK

Countless studies have suggested techniques to improve the accuracy and reliability of location information in vehicular environments. An earlier study classifies these techniques into position localisation [2] and position verification methods [3] Position localisation techniques estimate the positions. The standard position localisation technique used in vehicular networks is range-based localisation. It includes methods that estimate distance and angle using received signal strength, angle of (AoA) arrival, TDoA, and ToA [17]. These methods deploy computation algorithms such as trilateration, multilateration, and triangulation. Position verification validates the estimated positions. The section below reviews the position verification methods classified according to infrastructure and infrastructure-less methods.

1) INFRASTRUCTURE-BASED POSITION VERIFICATION

Early researchers in [18] introduced the concept of verifiable multilateration in position verification using roadside infrastructure. The study used four base stations (verifiers) placed at different known locations on the network to calculate the time of flight and distance from the vehicle based on the challenge-response process using the distance bounding protocol. This method is affected by the processing and network delays when generating and sending an answer to the challenge. Furthermore, real-world deployment would require more base stations set up. A recent study by [19] suggested a location verification system that relies on base stations. In the study, the researchers calculate the unknown locations of the vehicles from the nearby known base station location coordinates using the conditional probability distribution of the received signal strength measurements. Researchers in [20] suggested a position verification model that uses roadside units (RSUs) to verify node position legitimacy using speed and time information. In the study, the RSU performs a series of logical operations, such as acceptance range verification, maximum allowable speed check, and maximum density check, to verify the legitimacy of position information. To avoid broadcasting false positions from untrustworthy vehicles, researchers in [21] suggested a location verification model that uses RSUs to perform the localisation process. In the study, vehicles send localisation requests to the network picked up by at least three nearby RSUs. Each RSU calculates the distance from the vehicle, and the three distance measurements are combined using the triangulation technique to determine the vehicle's position.

2) INFRASTRUCTURE-LESS POSITION VERIFICATION

In [22], researchers proposed an infrastructure-less scheme for verifying position and velocity in vehicular networks, which does not depend on external hardware or infrastructure. In this model, the sender initiates the process by sending a ping message to the receiver, indicating its intention to broadcast a message. The receiver responds with an encrypted time token. Once the sender receives this token, it broadcasts the message. The study uses round-trip propagation time and distance comparisons to verify the accuracy of distance and velocity values. Although the simulation results are promising, the model faces significant delays due to the back-and-forth communication between the sender and receiver, posing a challenge in VANET environments where communications are ephemeral and dynamic. Previously discussed in section I is a challenge-response architecture co-

[1]It is a technology that defines a boundary around a geographic area, which enables software to trigger a notification when a device enters or exits the defined area.

[2]position localisation known as secure localisation aims to protect the accuracy and integrity of the computed node location.

[3]position verification known as location verification seeks to validate the authenticity and trustworthiness of the reported location information.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

TABLE 1: NOMENCLATURE

Notation	Description
x_s	Sender's latitude coordinate
y_s	Sender's longitude coordinate
x_r	Receiver's latitude coordinate
y_r	Receiver's longitude coordinate
D_{sr}	Distance between sender (x_s, y_s) and the receiver (x_r, y_r) vehicles
r	Geofence radius
S_s	Sender's Speed
S_r	Receiver's Speed
$\|S_{sr}\|$	Magnitude of the Speed Difference
Q_{sr}	The angle of the sender from the receiver
$\mu_{\text{WithinG}}(D_{sr})$	Within Geofence
$\mu_{\text{SlightlyAG}}(D_{sr})$	Slightly Above Geofence
$\mu_{\text{AboveG}}(D_{sr})$	Above Geofence
$\mu_{\text{WithinSL}}(\|S_{sr}\|)$	Above Speed Limit
$\mu_{\text{SlightlyASL}}(\|S_{sr}\|)$	Slightly Above Speed Limit
$\mu_{\text{AboveSL}}(\|S_{sr}\|)$	Above Speed Limit

operative location verification suggested by [8]. The verifier and cooperator send a challenge to the prover who claims to be at a particular location on the network using radio frequency. This model does not depend on infrastructure but relies on the presence of nearby vehicles. A previous study by [23] suggested a secure location verification trust model based on fuzzy logic. It consists of a plausibility check with two inputs: location verification using distance and location verification using time. The output of this check is a plausible level combined with the authentication level and experience level inputs and fed into the fuzzifier to generate an acceptable and non-acceptable trust level output. From the experimental analysis, the model performance decreases with an increase in untrustworthy vehicles.

III. METHOD-POSITION VERIFICATION USING GEOFENCING AND FUZZY LOGIC

This approach employs a fuzzy logic model that is adaptable to various scenarios to validate the sender's position information, as presented in Fig.2. We simulate the structure of the basic safety message that is periodically exchanged by vehicles in a vehicular environment, informing nearby vehicles of its status using MATLAB driving scenario. The basic safety message consists of sender ID, position coordinates, speed, heading, message ID, and timestamp. Upon arrival of the sender's basic safety message, the receiver extracts the sender's position coordinates from the message along with parameters such as distance, speed difference magnitude, and angle of the sender from the receiver values. The receiver then determines the sender's direction to filter out irrelevant notifications, avoiding unnecessary computations. If the message is relevant, the receiver uses fuzzy logic to compute the trustworthiness of the position coordinates.

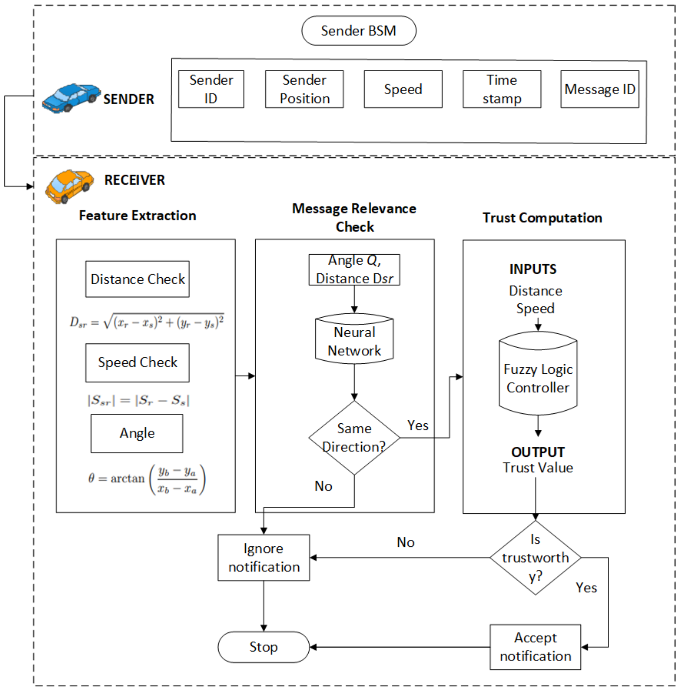


FIGURE 2: Position verification model

This computation is based on the distance and magnitude of the speed difference to assess the provided information's credibility. Finally, a threshold decision mechanism guided by the trust distribution evaluates the trust output of the fuzzy logic model to decide whether to accept or reject the notification. The proposed model is based on the following assumptions.

1) The messages are exchanged over a secure communication path.
2) The sender and receiver vehicles have a direct communication link with no obstacles, including buildings.
3) The maximum expected distance between the sender and the receiver is 200 m.
4) All receiver vehicles travel at a maximum speed of 30mph.

The symbols used in the following subsections are explained in Table 1.

A. DISTANCE AND SPEED CHECK

In this verification process, the system checks if the sender's position falls within the receiver's transmission range by calculating the change in positions of the sender (x_s, y_s) and the receiver (x_r, y_r) over time to determine the distance using Equation (1). The receiver's current position is the point of interest, with a maximum transmission range of 200 meters. We use the 200-meter range to manage the trade-off between data relevance and processing requirements. Additionally, a 200-meter range is the maximum detection range for a long-range Radar, a commonly used sensor

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

in autonomous driving applications and advanced driver assistance systems [24]. We normalise (D_{sr}) if it exceeds 200 meters according to Algorithm 1.

$$D_{sr} = \sqrt{(x_r - x_s)^2 + (y_r - y_s)^2} \tag{1}$$

Next, the system also checks whether the sender's speed (S_s) meets the specified criteria computed in Equation (2) and defined in Algorithm 1. For each position coordinate, the magnitude of the difference in speed between the sender and receiver $|S_{sr}|$ is expected to be less or equal to the maximum speed of the receiver. In this study, the maximum speed of the receiver is set to 30 mph (48km/hr), the national speed limit on all single and dual carriageways with street lights in England [25]. We normalise $|S_{sr}|$ if it exceeds 30 mph according to Algorithm 1.

$$|S_{sr}| = |S_r - S_s| \tag{2}$$

Algorithm 1: Distance and Speed Check

```
Input  : $x_s, y_s, x_r, y_r$
         $S_s$, $S_r$
Output: $D_{sr}, |S_{sr}|$
1  Initialize:
2  $D_{sr} = []$;
3  $|S_{sr}| = []$;
4  Start:
   scenario= drivingScenario();
5  while scenario do
6      pause=0.1;
7      position=vehicle.Position;
8      if $x_s$ $y_s$ is not empty == True then
9          calculate the distance between the vehciles;
10         Normalize the distance values greater than 200;
11         if $S_s$ is not empty == True then
12             calculate the magnitude of the diference in speed;
13             Normalize the magnitude values greater than 30;
14 return $D_{sr}, |S_{sr}|$
```

B. DIRECTION BASED MESSAGE RELEVANCE CHECK

This check identifies the sender's direction relative to the receiver to assess the message's relevance, preventing unnecessary trust computations and storage and allowing for quicker position verification and reduced storage needs. The receiver accepts the message for further verification if it is determined to travel in the same direction as the sender; otherwise, the receiver ignores the message. The direction

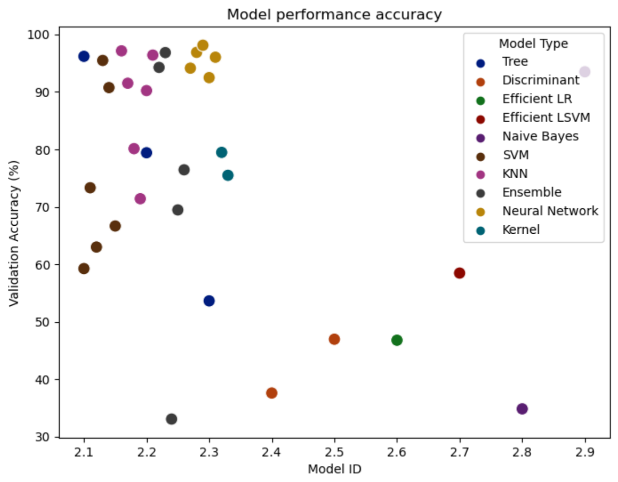


FIGURE 3: A visual presentation of the different model validation accuracy.

of the sender vehicle is estimated based on the change in the angle of the sender vehicle from the receiver vehicle as presented in Equation (3) and the change in distance between the two vehicles.

$$Q_{sr} = \arctan\left(\frac{y_b - y_a}{x_b - x_a}\right) \tag{3}$$

This study analysed different machine learning classification models, including decision trees, discriminant, linear regression, linear support vector machine, naive Bayes, support vector machines, k nearest neighbour, assemble, neural network and kernel, to determine the most accurate model for identifying the direction. From Fig.3, the classification of angle and distance features using the feedforward classification neural network [26] produces the best performance of 99.8%. The input of angle and distance is fed into the feedforward neural network to precisely determine the sender's direction (e.g. north, northeast, east, etc.) relative to the receiver, as analysed in section IV and computed using Algorithm 2. The choice of the feedforward neural network is strategic, given its ability to handle dynamic and noisy measurements, including distance and angle, thereby providing a more accurate prediction. This dual consideration also serves as a cross-verification mechanism, significantly reducing the likelihood of nodes providing false information about their positions. It reassures the integrity of the system and the accuracy of the reported distances relative to the angles and vice versa. The neural network architecture includes one fully connected layer of size 100 comprising two predictors and one response. Each input is standardised to ensure a uniform format. The Rectified Linear Unit (ReLU) activation function [27], which introduces non-linearity to the network, follows each fully connected layer with 1000 training iterations. The final fully connected layer and the

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

Algorithm 2: Identify the direction

```
Input  : $Q_{sr}$, $D_{sr}$
Output: Predicted labels
1  Preprocess:
2  $minelements \leftarrow min(Q_{sr}, D_{sr})$;
3  $Q_{sr} \leftarrow Q_{sr}$(1:minelements);
4  $D_{sr} \leftarrow D_{sr}$(1:minelements);
5  Create table:
     $carTable \leftarrow table(Q_{sr}, D_{sr})$;
6  Load the classifier:
7  $load\ trainedclassifier$;
8  Make Predictions:
9  $trainedClassifier.predictFcn\ (carTable)$;
10 $predictedLabels \leftarrow yfitcar$;
11 return  Predicted labels
```

subsequent ReLU activation function produce the network's output, namely classification scores and predicted labels.

C. TRUST COMPUTATION USING FUZZY LOGIC

Now that we have established an acceptable distance (D_{sr}), magnitude of the speed difference $|S_{sr}|$, and direction message relevance, the system's next task is to compute the trustworthiness of the position data. The Mamdani Fuzzy logic controller calculates trust by checking for inconsistencies between the sender's distance and speed values over time, as presented in Algorithm 3. In the dynamic vehicular environment, where a sender's movement may cause inexact measurements, the controller's role is crucial in maintaining the system's performance. The Mamdani fuzzy logic controller operates on the principles of fuzzy logic, which include fuzzification, fuzzy inference, and defuzzification [28]. Fuzzy logic is a technique used to make decisions about uncertain situations [29] based on human reasoning that makes non-binary decisions on a sliding scale from truth, partial truth, and partial false to false. Fuzzy logic consists of four components: the fuzzifier, knowledge base (fuzzy rules), intelligence (inference engine), and defuzzifier. At the start of the fuzzy logic operation, the fuzzifier converts numerical values into linguistic variables using membership functions that define the degree to which an object belongs to a particular fuzzy set. The controller then feeds the linguistic variables into the inference engine, which applies rules from the knowledge base and provides an output value. The value is then fed into the defuzzifier to convert the fuzzy variables to numerical values. The key advantages of fuzzy logic include the ability to relate uncertain or complex data and increased robustness, but such models require more simulation and review before actual implementation [30].

1) FUZZIFICATION

In the fuzzification verification process, the two numerical input parameters, namely, (D_{sr}) and $|S_{sr}|$ are fed into the fuzzy controller and converted into fuzzy sets. For each input value, a degree of membership is assigned based on the defined membership functions. A membership function (MF) is a graphical representation illustrating the mapping of each point within the input space to a membership value ranging between 0 and 1. This model implements the trapezoidal membership function type for speed and distance input values. Fig. 4 is a representation of the three distance

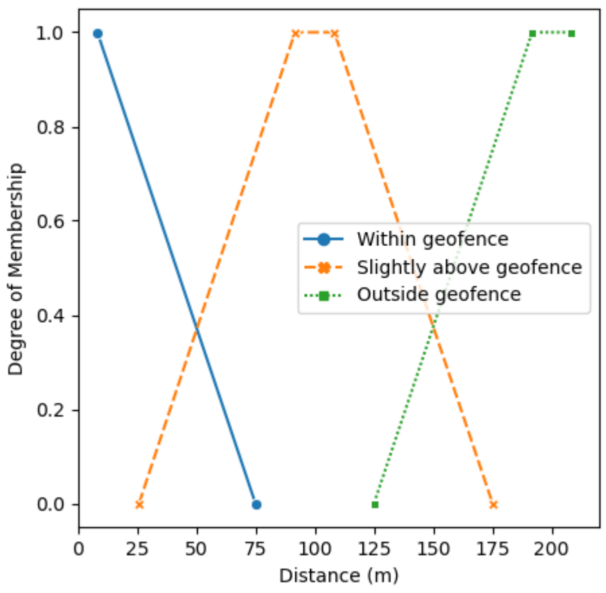


FIGURE 4: Distance membership function

membership function distributions as derived in Equations (4),(5), and (6) using the max-min inference method [31].

$$\mu_{\text{WithinG}}(D_{sr}) = \begin{cases} 0 & \text{if } (D_{sr}) \leq -75 \\ \frac{(D_{sr})+75}{-8.333+75} & \text{if } -75 < (D_{sr}) \leq -8.333 \\ 1 & \text{if } -8.333 < (D_{sr}) \leq 8.333 \\ \frac{75-(D_{sr})}{75-8.333} & \text{if } 8.333 < (D_{sr}) \leq 75 \\ 0 & \text{if } (D_{sr}) > 75 \end{cases} \tag{4}$$

$$\mu_{\text{SlightlyAG}}(D_{sr}) = \begin{cases} 0 & \text{if } (D_{sr}) \leq 25 \\ \frac{(D_{sr})-25}{91.67-25} & \text{if } 25 < (D_{sr}) \leq 91.67 \\ 1 & \text{if } 91.67 < (D_{sr}) \leq 108.3 \\ \frac{175-(D_{sr})}{175-108.3} & \text{if } 108.3 < (D_{sr}) \leq 175 \\ 0 & \text{if } (D_{sr}) > 175 \end{cases} \tag{5}$$

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

$$\mu_{\text{AboveG}}(D_{sr}) = \begin{cases} 0 & \text{if } (D_{sr}) \le 125 \\ \frac{(D_{sr})-125}{191.7-125} & \text{if } 125 < (D_{sr}) \le 191.7 \\ 1 & \text{if } 191.7 < (D_{sr}) \le 208.3 \\ \frac{275-(D_{sr})}{275-208.3} & \text{if } 208.3 < (D_{sr}) \le 275 \\ 0 & \text{if } (D_{sr}) > 275 \end{cases} \quad (6)$$

Figure 5 is a representation of the speed membership function distributions as derived in Equations (7),(8), and (9) using the max-min inference method [31].

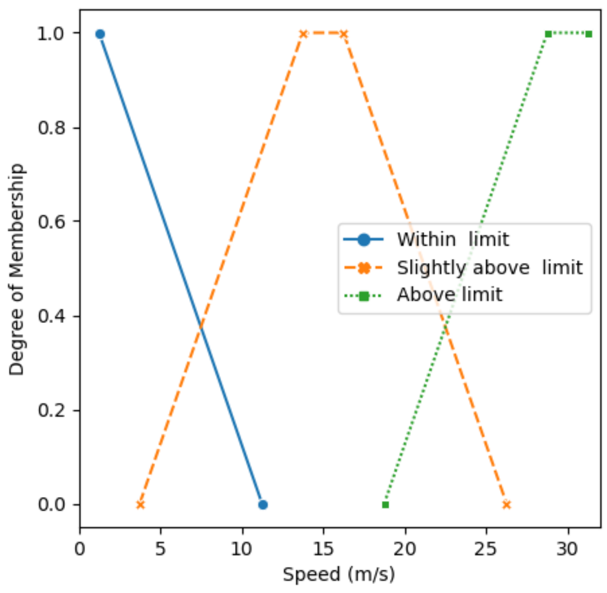


FIGURE 5: Speed membership function

$$\mu_{\text{WithinSL}} |S_{sr}| = \begin{cases} 1 & \text{if } 0 \le |S_{sr}| \le -11.25 \\ \frac{|S_{sr}|+11.25}{-1.25+11.25} & \text{if } -11.25 < |S_{sr}| \le -1.25 \\ 1 & \text{if } -1.25 < |S_{sr}| \le 1.25 \\ \frac{15-|S_{sr}|}{11.25-1.25} & \text{if } 1.25 < |S_{sr}| \le 11.25 \\ 0 & \text{if } |S_{sr}| > 11.25 \end{cases} \quad (7)$$

$$\mu_{\text{SlightlyASL}} |S_{sr}| = \begin{cases} 0 & \text{if } (|S_{sr}|) \le 3.75 \\ \frac{|S_{sr}|-3.75}{13.75-3.75} & \text{if } 3.75 < |S_{sr}| \le 13.75 \\ 1 & \text{if } 13.75 < |S_{sr}| \le 16.25 \\ \frac{26.25-|S_{sr}|}{26.25-16.25} & \text{if } 16.25 < |S_{sr}| \le 26.25 \\ 0 & \text{if } |S_{sr}| > 26.25 \end{cases} \quad (8)$$

$$\mu_{\text{AboveSL}} |S_{sr}| = \begin{cases} 0 & \text{if } |S_{sr}| \le 18.75 \\ \frac{|S_{sr}|-18.75}{28.75-18.75} & \text{if } 18.75 < |S_{sr}| \le 28.75 \\ 1 & \text{if } 28.75 < |S_{sr}| \le 31.25 \\ \frac{41.25-|S_{sr}|}{41.25-31.25} & \text{if } 31.25 < |S_{sr}| \le 41.25 \\ 0 & \text{if } |S_{sr}| > 41.25 \end{cases} \quad (9)$$

2) INFERENCE ENGINE

Next, the inference engine (decision-making model) combines membership functions with fuzzy rules derived from experience and system knowledge to attain a fuzzy output. The defined nine fuzzy rules take the form presented in Table 2.

3) DEFUZZIFICATION

Finally, the output from the fuzzy inference engine is converted back to numerical values through a process known as defuzzification [23]. In this study, the two standard defuzzification techniques, including the centroid and maxima, are deployed and evaluated to identify the most suitable

TABLE 2: Fuzzy Inference Rules

Name	Weight	Rule
rule1	1	If (D_sr) is within Geofence AND $(\|S_{sr}\|)$ is within the acceptable range, THEN Position Trust Level is High
rule2	1	If (D_sr) is within Geofence AND $(\|S_{sr}\|)$ is slightly above the acceptable range, THEN Position Trust Level is Medium
rule3	1	If (D_sr) is within Geofence AND $(\|S_{sr}\|)$ is above the acceptable range, THEN Position Trust Level is Low
rule4	1	If (D_sr) is slightly above Geofence AND $(\|S_{sr}\|)$ is within the acceptable range, THEN Position Trust Level is High
rule5	1	If (D_sr) is slightly above Geofence AND $(\|S_{sr}\|)$ is slightly above the acceptable range, THEN Position Trust Level is Average
rule6	1	If (D_sr) is slightly above Geofence AND $(\|S_{sr}\|)$ is above the acceptable range, THEN Position Trust Level is Low
rule7	1	If (D_sr) is outside Geofence AND $(\|S_{sr}\|)$ is within the acceptable range, THEN Position Trust Level is Low
rule8	1	If (D_sr) is outside Geofence AND $(\|S_{sr}\|)$ is slightly above the acceptable range, THEN Position Trust Level is Low
rule9	1	If (D_sr) is outside Geofence AND $(\|S_{sr}\|)$ is above the acceptable range, THEN Position Trust Level is Low

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

Algorithm 3: Trust Computation

```
Input  : $Predicted\ labels,\ D_{sr},\ |S_{sr}|$
Output: Trustworthy or Untrustworthy

1  Initialize:
     $trustvalue = []$;
2  Load:
     $predictedlabels$;
3  if $isthesame == True$ then
4      Load the fuzzy controller:
         $myFIS = fuzzy\ controller$;
5      $trustvalue \leftarrow evalfis(myFIS\ (D_{sr},\ |S_{sr}|))$;
6      if $trustvalue == 0.5$ then
7          $accept\ notification$ sender $is\ moving\ in\ the\ same\ direction\ and\ is\ trustworthy$;
8          Store:
             $predicted\ label$;
9          $trustvalue$;
10     else
11         $reject\ notification\ the\ sender\ is\ not\ trustworthy$
12 else
13     $clear\ trust\ value$;
14     $Ignore\ notification\ sender\ is\ not\ moving\ in\ the\ same\ direction$;
```

method for the trust computation model in VANET. The defuzzification algorithm takes the fuzzy area's centre of gravity (COG) in the centroid method. It is a point at which the fuzzy set balances along the x-axis. The maxima defuzzification method accounts for the element with the highest degree of membership. It is further categorised into mean of maxima (MOM), first of maxima (FOM), last of maxima (LOM), and the highest methods. All these methods consider elements with the maximum degree of membership but differ in the computation of the final output value. The mean of maxima takes the average of the elements with maximum values; the first of maxima considers the value of the first element with the highest output value of all elements with the highest degree of membership, and the last of maxima considers the least element in the aggregated output of all elements with the highest degree of membership. The trust output membership function's minimum and maximum values are determined by the trust values used within the vehicular trust framework, as indicated in [30]. A triangular membership function [29] establishes the boundaries within the 0 to 1 output range. The trust values begin at zero because the verification model operates independently of recommendations or neighbouring nodes, thus avoiding the cold start problem when no neighbours or trust history is available.

D. COMPLEXITY ANALYSIS

The storage and processing time requirements are two components that define the complexity of Algorithms 1, 2 and 3. Considering the dynamic nature of the vehicular environment that demands instant connections excluding delays, it is essential to evaluate the processing time and minimise the storage to ensure effective operation in dense environments and in the long run. This analysis is conducted differently for each algorithm. Let (N_s) be the number of vehicles that broadcast a message notification and (N_r) the receiver. The time complexity in Algorithms 1 and 2 is O(N_s) because the $|S_{sr}|$, D_{sr}, Q_{sr}, and predicted labels depend on the number of sender vehicles simulated within the single scenario. Let the number of vehicles travelling in the same direction be (N_{sr}). The trust algorithm computes the trust values for only the sender vehicles travelling in the same direction as the receiver. Therefore, the time complexity for algorithm 3 is O((N_{sr})). Regarding space complexity analysis in Algorithm 3, the receiver only stores the trust values and predicted labels in arrays for the vehicles moving in the same direction and whose trust value is above a certain threshold expressed as O($(T_s >$ threshold) where (T_s) represents the trust values of sender's position data. It eases the storage requirements in the long run.

IV. PERFORMANCE EVALUATION

This section conducts experiments to analyse various methods and parameters for better model development. This section is divided into three parts. Firstly, considering that road structures are often not straight, we analysed the angular changes between the receiver and sender across various road structures to identify the most accurate labelling for the classification model and study the change in angles of the sender from the receiver as the distance between the sender and receiver increases or decreases. Secondly, we performed an analytical evaluation of the defuzzification methods to identify the most suitable method. Finally, the third part is an analysis of the effectiveness of the suggested model.

A. ANALYSIS OF THE CHANGE IN Qsr ALONG DIFFERENT ROAD STRUCTURES

Firstly, in Fig.6a, Car_A, Ego_Car, and Car_B are moving along a horizontal straight road with Ego_Car moving from left to right. The Q_{sr} for Car_A and Ego_Car are at 90° throughout the simulation. Subsequently, the Q_{sr} for Car_B is observed to increase from 90° minimum value towards 270°. It indicates that Car_A and Ego_Car are moving in the same direction towards the east while Car_B is moving towards the west.

Secondly, in Fig.6b Car_A, Ego_Car, and Car_B are moving along a horizontal straight road with Ego_Car moving from right to left. The Q_{sr} for Car_A and Ego_Car is observed at 270° slightly iterating between 268° and 270° throughout the simulation. In contrast, the Car_B Q_{sr} is observed to increase initially from 270° towards the minimum

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

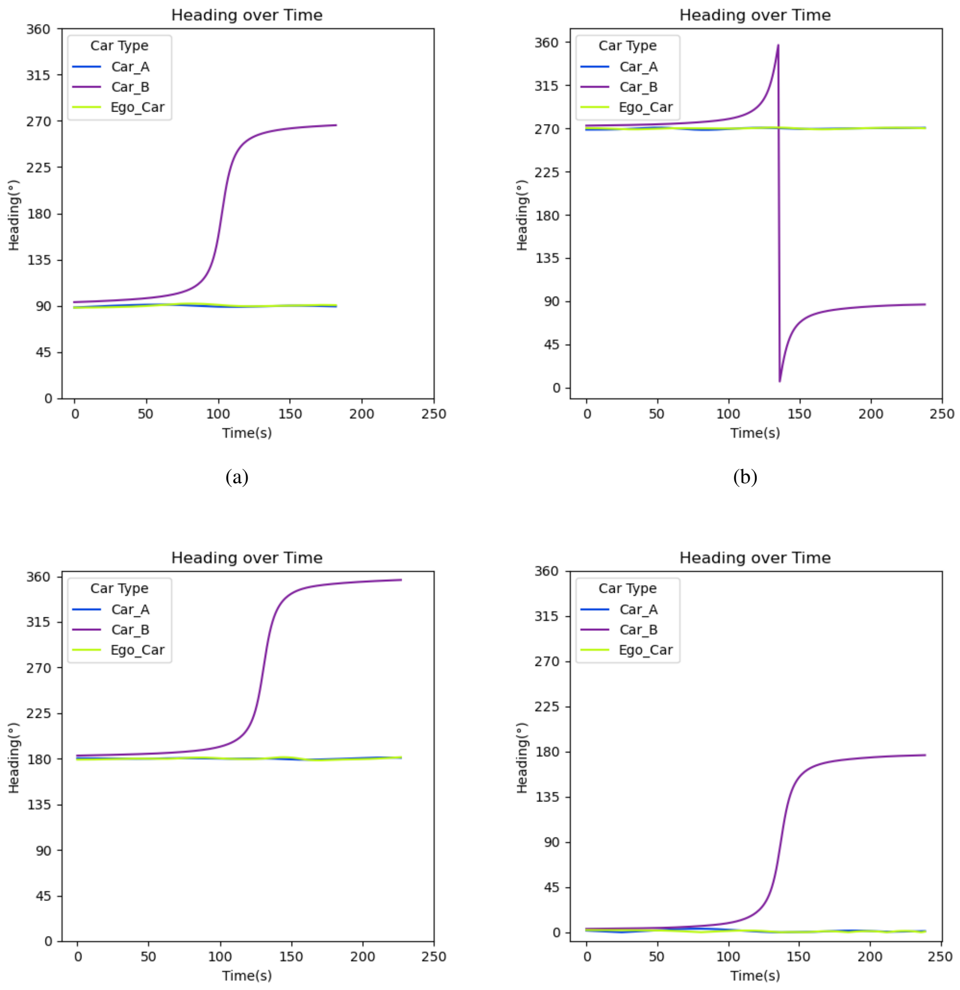


FIGURE 6: change in the angles along: (a)Horizontal Straight road Car_A and Ego_Car = east, Car_B = west (b)Horizontal Straight road Car_A and Ego_Car = west, Car_B = east (c) Vertical Straight road Car_A and Ego_Car = south, Car_B = north (d) Vertical Straight road Car_A and Ego_Car = north, Car_B = south

angle $0°$ then increases towards $90°$. This analysis indicates that Car_A and Ego_Car are moving in the same direction towards the west, and Car_B is moving in the opposite direction towards the east. Thirdly, in Fig.6c, Car_A, Ego_Car, and Car_B are moving along a vertical straight road with Ego_Car moving from top to bottom. The Q_{sr} for Car_A and Ego_Car is observed at $180°$ slightly iterating between $179°$ and $180°$ throughout the simulation. At the same time, Car_B Q_{sr} increases from the minimum value $180°$ towards the maximum $360°$. It indicates that Car_A and Ego_Car are moving in the same direction towards the south, and Car_B is moving in the opposite direction towards the north. Lastly, in Fig.6d, Car_A, Ego_Car, and Car_B are moving along a vertical straight road with Ego_Car moving from bottom to top. The Q_{sr} for Car_A and Ego_Car is observed at $0°$ iterating slightly between $180°$ and $180°$ throughout the simulation. Subsequently, the Car_B Q_{sr} increases from the minimum angle $0°$ towards a maximum value of $180°$. It indicates that Car_A and Car_Ego are moving in the same direction towards the north, and Car_B is moving in the opposite direction towards the south.

Fig.7 presents the sender and receivers' angle variation along the left and right diagonal roads. Firstly, in Fig.7a, Car_A, Ego_Car and Car_B are moving along a left diagonal road with the Ego_Car moving from the left to the right. The Q_{sr} for Car_A and Ego_Car is observed to be at $225°$ slightly iterating between $219°$ and $225°$ throughout the simulation. Subsequently, the Q_{sr} for Car_B is observed to increase from $225°$ towards the maximum value $360°$. After that, it decreases to the minimum value $0°$ and starts increasing again towards $45°$. It indicates that Car_A and Ego_Car are moving in the same direction towards the

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

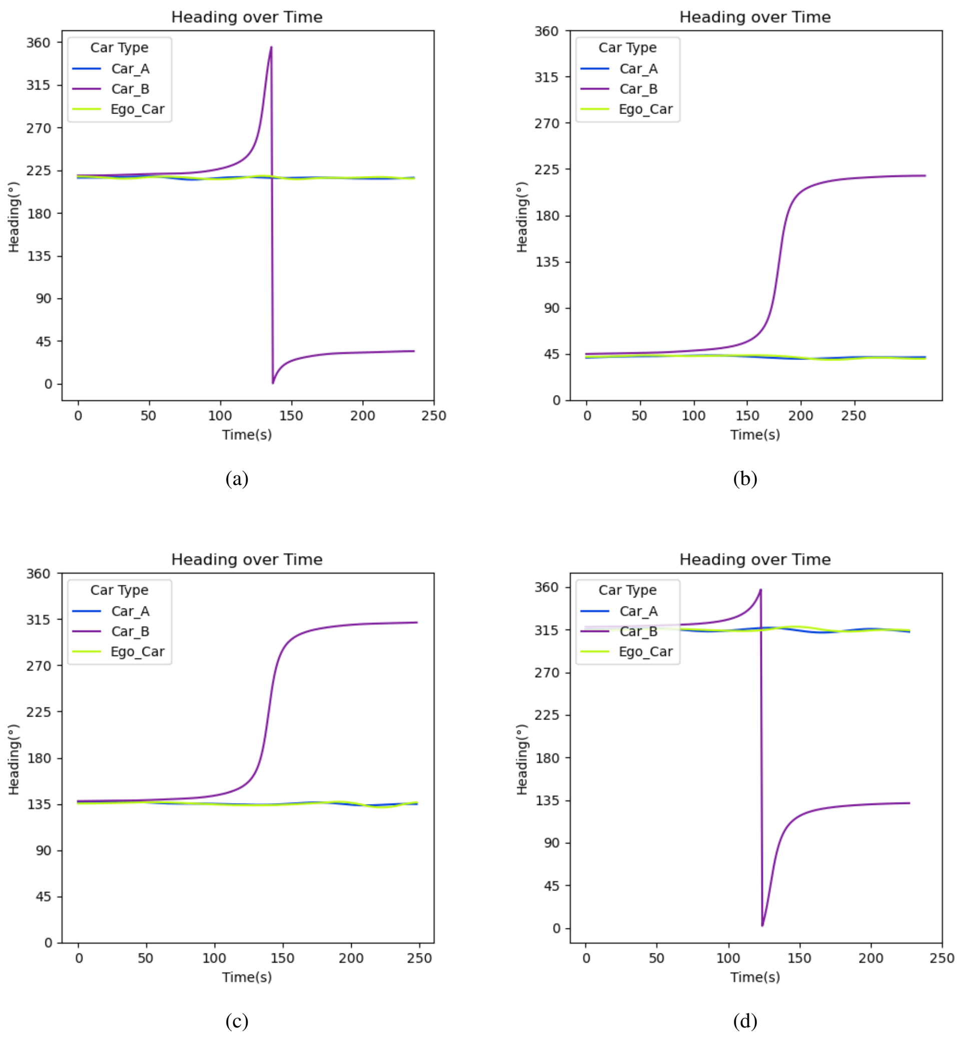


FIGURE 7: Change in the angles along:(a)Left Diagonal road Car_A and Ego_Car = southwest and Car_B = northeast (b) Left Diagonal road Car_A Car_A and Ego_Car = northeast and Car_B =southwest (c)Right Diagonal road Car_A and Ego_Car = southeast and Car_B =northwest (d) Right Diagonal road Car_A and Ego_Car = northwest and Car_B =southeast

southwest while Car_B is moving towards the northeast. Secondly, in Fig.7b, Car_A, Ego_Car, and Car_B are moving along a left diagonal road, with the Ego_Car moving from the right to the left. The Q_{sr} for Car_A and Ego_Car Q_{sr} is observed at $45°$, slightly iterating between $40°$ and $45°$ throughout the simulation. In contrast, the Car_B Q_{sr} is observed to increase initially from $45°$towards the maximum angle value $225°$. This analysis indicates that Car_A and Ego_Car are moving in the same direction towards the northeast, and Car_B is moving in the opposite direction towards the southwest. Thirdly, in Fig.7c, Car_A, Ego_Car, and Car_B are moving along a right diagonal road with Ego_Car moving from the right to the left. The Q_{sr} for Car_A and Ego_Car is observed at $135°$ slightly iterating between $130°$ and $135°$ throughout the simulation. While Car_B Q_{sr} is observed to increase from the minimum value $135°$ towards the maximum $315°$. It indicates that Car_A and Ego_Car are moving in the same direction towards the southeast, and Car_B is moving in the opposite direction towards the northwest. Lastly, in Fig.7d, Car_A, Ego_Car, and Car_B are moving along a right diagonal road with Ego_Car moving from the left to the right. The Q_{sr} for Car_A and Ego_Car is observed at a constant value of $315°$ iterating slightly between $314°$ and $315°$ throughout the simulation. Subsequently, Car_B Q_{sr} is observed to increase from $315°$ towards a maximum value of $360°$, after which there is a sharp decrement in the Q_{sr} towards $0°$, the minimum value, followed by an increase towards $135°$. It indicates that Car_A and Car_Ego are moving in the same direction towards the northwest, and Car_B is moving in the opposite direction towards the southeast.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

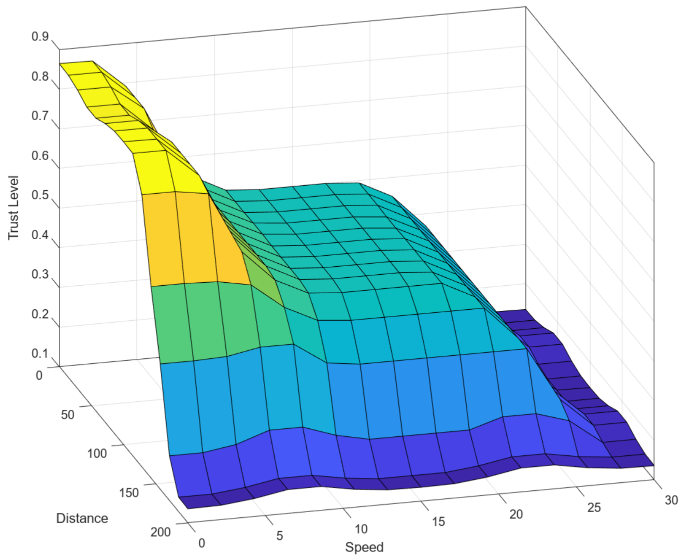


FIGURE 8: COG Fuzzy logic control surface plot.

B. ANALYSIS OF THE DIFFERENT DEFUZZIFICATION METHODS

Table 3 summarises the performance of the fuzzy logic model based on the different defuzzification techniques at specific distance and speed inputs. According to Fig.8, the impact of distance and speed on the position trust level using the centroid method reaches its maximum point of 0.864409 when D_{sr} and $|S_{sr}|$ is at the lowest value of 0. It is attributed to the vehicle being close to the receiver and travelling at the same speed as the receiver. Second, the trust value decreases further to a minimum of 0.135591 at four noticeable points with a sharp increase in $|S_{sr}|$ to a maximum of 30 and a decrement in D_{sr} to a minimum of 0 and vice versa.

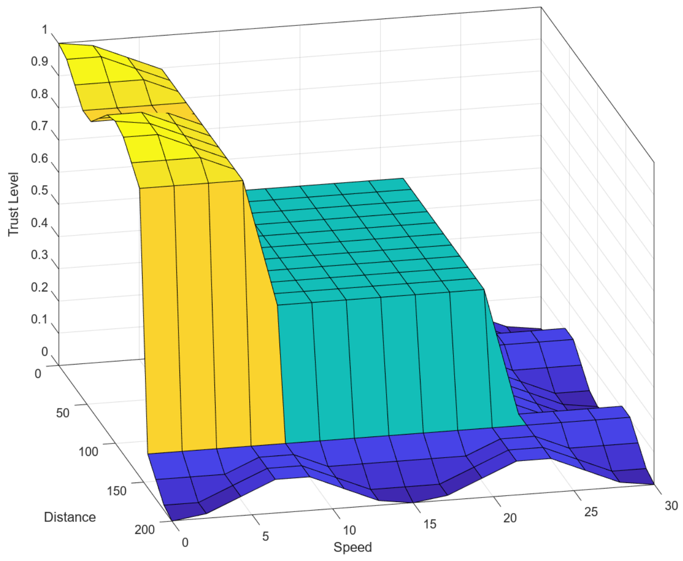


FIGURE 9: MOM Fuzzy logic control surface plot.

Fig.9 illustrates an evaluation of speed and distance on the position trust level using the mean of the maxima defuzzification method. The impact of distance and speed increases to a maximum value 1 at two points when there is a significant decrement in $|S_{sr}|$ and D_{sr} to the lowest value 0 and when there is relatively average increase in D_{sr} to 100 at a minimum $|S_{sr}|$ of 0. Secondly, it is notable that the position trust level decreases to a minimum of 0 at five points at a maximum $|S_{sr}|$ and D_{sr}, maximum distance at a relatively low speed, maximum speed with a decrement in distance, average speed with a sharp increment in distance to a maximum value and at a maximum speed with a relatively average distance.

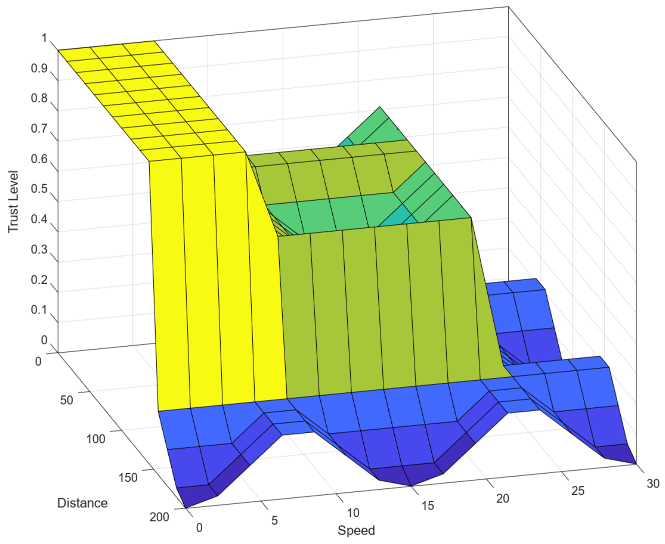


FIGURE 10: LOM Fuzzy logic control surface plot.

Finally, Fig.10 is an analytical evaluation of the impact of distance and speed on position trust level using the last of maxima defuzzification. Firstly, the trust level reaches its maximum value of 1 at various points within the figure's top rectangular shape. Secondly, a sharp increment in the D_{sr} to a maximum value of 200 at low and average $|S_{sr}|$ results in a decrement of the trust value to a minimum of 0. Furthermore, a sharp increment in the speed to a maximum value with a minimum (0) and average D_{sr} (100) results in a minimum trust value of 0.

The following observations are made from the experiment results. Firstly, Table 3 shows that the centroid method is superior due to its detailed trust values. While the Mean of Maxima and last of maxima methods produce more absolute trust values (such as 0, 0.5 or 1), indicating either complete trust or no trust, the centroid method offers a range of trust values from 0.135591 to 0.864409. It allows the centroid method to represent varying degrees of trust, which is more reflective of real-world driving scenarios where inputs are not always entirely trustworthy or untrustworthy. Secondly, the flexibility and precision of the centroid method enhance the system's ability to handle ambiguity and partial

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

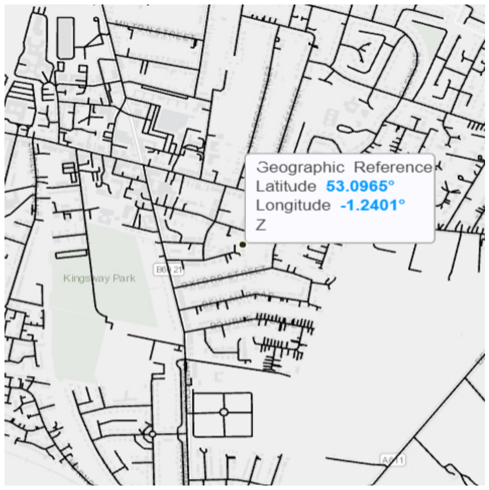
reliability, leading to more accurate and refined decision-making. Consequently, the centroid method is better suited for trust computation as it provides varying degrees of trust levels that improve the performance and reliability of the model. Secondly heading analysis has informed our labelling process, delineating cardinal directions as follows: the north cardinal point spans from 337.5 to 22.5 degrees, northeast spans from 22.5 to 67.5 degrees, east spans from 67.5 to 112.5 degrees, southeast spans from 112.5 to 137.5 degrees, south spans from 137.5 to 202.5 degrees, southwest spans from 202.5 to 247.5 degrees, west spans from 247.5 to 292.5 degrees, and northwest spans from 292.5 to 337.5 degrees. Furthermore, when vehicles are moving in the same direction at constant speeds, the change in angle and distance between them is approximately constant, while When vehicles move in the opposite direction, the angle between them increases continuously while the distance reduces and increases continuously.

TABLE 3: A comparison to evaluate the performance of the different defuzzification methods.

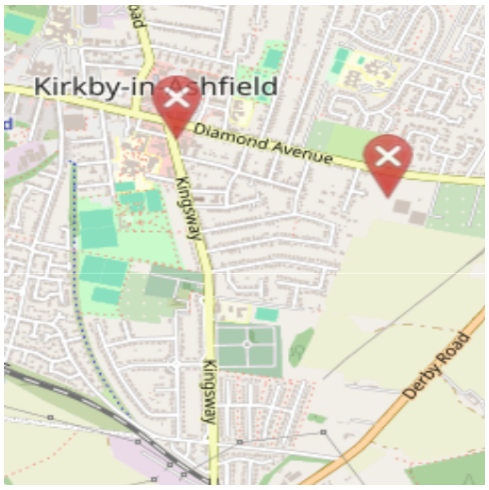
MOM				
Distance	0	200	200	0
Speed	0	30	0	30
Trust	1	0	0	0
LOM				
Distance	0	200	200	0
Speed	0	30	0	30
Trust	1	0	0	0
COG				
Distance	0	200	200	0
Speed	0	30	0	30
Trust	0.864409	0.135591	0.135591	0.135591

C. SIMULATION BASED CASE STUDY

This section has created two sets of conditions to evaluate the effectiveness and performance of the direction and fuzzy logic models that make up the position verification model. The first setup defined in Table 4 aims to achieve real-world road dynamics by importing Kirby in Ashfield real-world road map data dynamics from the open street map in Fig.11 into the MATLAB driving scenario to evaluate the practicability of the classification model in section IV subsection A based on real-world data. The MATLAB driving scenario designer object is used because it supports the import of realistic road map dynamics from OpenStreetMap (OSM). It also supports the simulation of custom road layouts with different actors, such as cars, trucks, bicycles, and pedestrians, as well as sensor simulations, including LiDAR, radar, cameras, and ultrasonic sensors used in autonomous driving systems. Waypoints and speed describe the trajectory of actors within the designer to define the route throughout the simulation. Considering the position object within the designer, it is



(a)



(b)

FIGURE 11: Kirby in Ashfield (a) open street map view (b) google map view

TABLE 4: Driving Scenario Properties

Parameter	Value
Type of actors	cars
Number of actors	9
Number of roads	3
Number of actors on each road	3
Maximum road length	300m
Speed Range	15-30 m/hr
Sample Time	0.01 ms

possible to attain the position coordinates of the actors defined by the trajectory returned as a 1 * 3 matrix (x, y, z). The dynamic vehicular characteristics, such as speed, are extracted from these position coordinates to mimic the content of the basic safety messages in V2X communication. Furthermore, the driving scenario designer also supports the

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

evaluation of the position vectors of neighbouring vehicles using the ego vehicle, which can be used as a reference point to calculate and identify the change in distance and heading vectors between vehicles. Despite the enormous features of the driving scenario designer, limitations within OpenStreetMap import and sensor import or export are barriers to accurate and effective vehicular environment simulation. For example, OSM files with large road networks take a long load; therefore, only the desired area of interest with specific selected roads is imported to overcome this limitation. Appropriate data extraction techniques are applied within the exported driving scenario function to extract only vehicle object dynamics and prevent the report of unclustered data.

TABLE 5: Simulation Parameters

Normal samples	False Samples	Distance Random Range	Speed Random Range
100	100	151-300	23-40
100	100	161-400	33-50
100	100	171-500	43-60

The second experimental setup is designed to simulate false position data to assess the efficiency of the proposed model. Specifically, this study concentrates on random data attacks as a preliminary benchmark for evaluation. In the context of these attacks, the adversary transmits random angle, distance, and speed values that fall within a predefined false range and incorporate varying noise levels, as described in Table 5.

1) IDENTIFIED DIRECTION USING REAL WORLD ROAD DYNAMICS

In Fig.12a, the car_ego maintains a consistent predicted label of 5 throughout the simulation. In contrast, car2 initially maintains a steady predicted label of 8 for approximately 200 seconds, then a transition to 1 for the next 100 seconds. Subsequently, it fluctuates between 3 and 4 for 50 seconds before stabilising at label 5. Similarly, the predicted label for car1 remains constant at 1 for the initial 120 seconds, shifts to 5 for the following 130 seconds, exhibits fluctuation between 4, 3, and 2 for the subsequent 50 seconds, and ultimately stabilises back at 1. It implies that car_ego is travelling in the south, car1 is heading to the north, and car2 is heading to the south. In Fig.12b, the predicted label for car3_ego maintains a constant value of 1 throughout the simulation, as opposed to car4 and car5. During the initial 100 seconds, the predicted label for the car5 fluctuates between 1, 2, and 3, then remains constant at 3 for the subsequent 150 seconds. Then, it fluctuates between 4, 5, and 6 for 50 seconds before settling at 7. Similarly, during the first 100 seconds, the predicted label for car4 initially varies between 1, 2, and 3, then remains constant for the next 200 seconds at 3, and finally fluctuates between 4 and 5 before stabilising at 6 as the predicted label. It indicates that car3_ego is heading to the north, car4 is heading to the southwest, and car five is heading to the west. Finally, in Fig.12c, car6_ego, car7, and car8 maintain constant predicted labels of 3, 1, and 5 throughout the simulation. It indicates that car6_ego is heading east, car7 is heading north, and car8 is heading south.

2) MESSAGE RELEVANCE CHECK ANALYSIS

Fig.13 shows the performance of the message relevance check in the presence of increasing random noisy Q_{sr} and D_{sr} values. The performance is better than random classification at all the different noisy levels since the False Positive Rate does not equal the True Positive Rate at any point indicated by the AUC values at the bottom right corner. From Fig.13a, the Q_{sr} and D_{sr} values deviate from the actual values by 10, and the message check verification is observed above 0.99. When the noise level is increased to 20 in Fig.13b, where the Q_{sr} and D_{sr} values deviate further away, the message check verification is observed above 0.85. In Fig.13c, performance is above 0.78, with the noise level at 30. Therefore, model performance is better at the different random or untrustworthy values than random classification.

3) TRUST COMPUTATION ANALYSIS

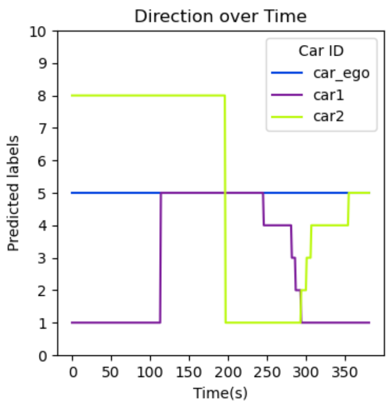
Fig.14 presents the effectiveness of the trust computation algorithm in the presence of trustworthy and untrustworthy data at varying noise levels, summarised in Table 5. From Fig.14a, the false data values deviate from the actual values by 10, and we can see that the false data has lower trust levels concentrated towards the left, closer to 0. The normal data has higher trust levels concentrated towards the right, closer to 1. Like the previous figure, Fig.14b has false data values higher to the left towards 0 and normal to the right. We can observe a distinction in the trust values of the normal and false data with less overlap at a deviation of 20. Finally, in Fig.14c, the false data deviates from normal by 30. We still observe the trust values for the false data concentrated towards 0 while the normal data trust values are towards 1 with much less overlap. Therefore, the trust computation model can identify false data, and its effectiveness improves with an increase in deviation from the actual values.

V. CONCLUSION AND FUTURE WORKS

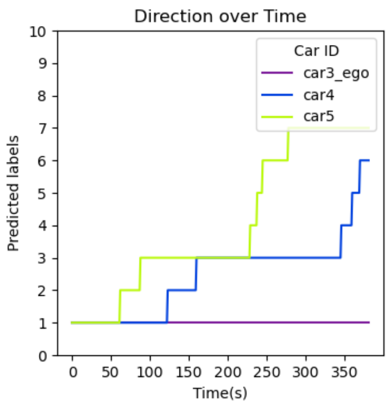
In this study, a position verification model that implements geofencing and fuzzy logic for decision-making is presented. The system analyses vehicle dynamic data, including position, speed, distance, and direction, to assess the reliability and integrity of received position data. Instead of depending on recommendations, voting, or trust history values from nearby vehicles, this model eliminates trust calculation bias by directly assessing vehicle notifications' reliability within the geofenced area and adhering to the defined speed limit. In addition, it improves storage efficiency by computing trust

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

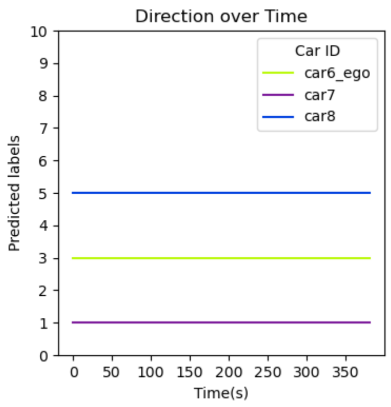
This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666



(a)

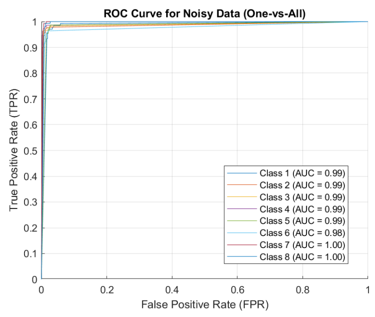


(b)

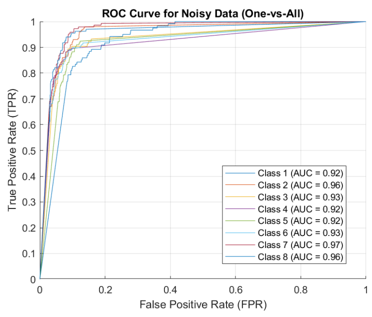


(c)

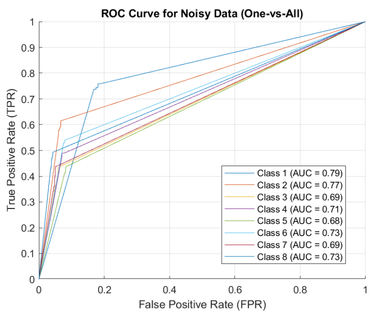
FIGURE 12: Visualisation of the predicted labels using real-world road dynamics (a) car_ego and car2 = south, car1 = north (b) car3_ego = north car4 = southwest, car5 = west (c) car6_ego = east, car7 = north, car8 = south



(a)



(b)



(c)

FIGURE 13: ROC Curves with AUC at: (a)Deviation from normal data by 10, (b)Deviation from normal data by 20, (c)Deviation from normal data by 30

values exclusively for notifications from vehicles moving in the same direction. This approach prevents unnecessary computations, leading to maximised storage utilisation and reduced computing time. The simulation results demonstrate the effectiveness of the classification model due to its ability to outperform random classification in the presence of false data. Furthermore, a clear distinction between the trust values of false and actual data indicates the model's ability to identify false positions. Future research will focus on evaluating the model's effectiveness using publicly available position attack datasets and implementing the model in a simulator with real-time V2X communications to identify false position attacks.

ACKNOWLEDGMENT

Data Availability: Data will be made available via individual request to the corresponding author.

Funding: The research is funded by the fully funded Ph.D. Scholarship by Nottingham Trent University, UK.

Conflict of Interest: The authors declare no conflict of interest regarding the research and publication of this research.

REFERENCES

[1] T. Leinmüller, E. Schoch, F. Kargl, and C. Maihöfer, "Influence of falsified position data on geographic ad-hoc routing," in *Security and Privacy in Ad-hoc and Sensor Networks: Second European Workshop, ESAS 2005, Visegrad, Hungary, July 13-14, 2005. Revised Selected Papers 2*, pp. 102–112, Springer, 2005.

[2] T. Leinmüller and E. Schoch, "Greedy routing in highway scenarios: The impact of position faking nodes," in *Proceedings of Workshop On Intelligent Transportation (WIT 2006)(Mar. 2006)*, 2006.

[3] J. Grover, M. S. Gaur, and V. Laxmi, "Position forging attacks in vehicular ad hoc networks: implementation, impact and detection," in *2011 7th International Wireless Communications and Mobile Computing Conference*, pp. 701–706, IEEE, 2011.

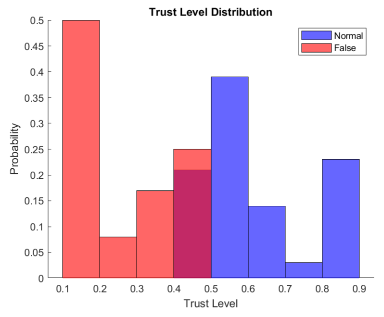
[4] M. Abu-Elkheir, S. A. Hamid, H. S. Hassanein, I. B. M. Elhenawy, and S. Elmougy, "Position verification for vehicular networks via analyzing two-hop neighbors information," in *2011 IEEE 36th Conference on Local Computer Networks*, pp. 805–812, IEEE, 2011.

[5] D. K. Sheet, O. Kaiwartya, A. H. Abdullah, and A. N. Hassan, "Location information verification cum security using tbm in geocast routing," *Procedia Computer Science*, vol. 70, pp. 219–225, 2015.

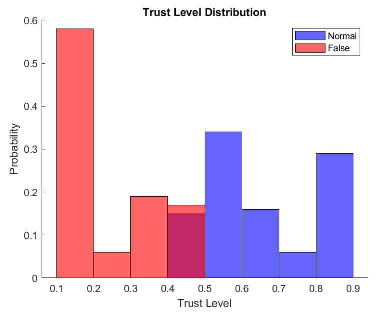
[6] F. Malandrino, C. Borgiattino, C. Casetti, C.-F. Chiasserini, M. Fiore, and R. Sadao, "Verification and inference of positions in vehicular net-

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

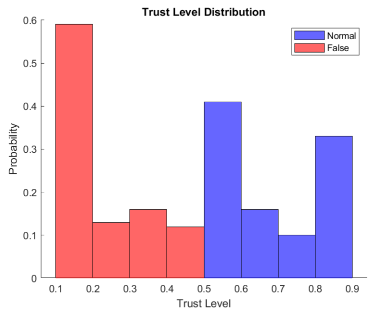
This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666



(a)



(b)



(c)

FIGURE 14: Trust Level Distribution with False Distance and Speed Data Range (a)151-300 & 23-30, (b)161-400 &33-40, (c)171-500 & 43-50

works through anonymous beaconing," *IEEE transactions on mobile computing*, vol. 13, no. 10, pp. 2415–2428, 2014.

[7] S. Das and M. Saha, "Position verification in vehicular communications," in *2013 Fifth International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–2, IEEE, 2013.

[8] P. Zhang, Z. Zhang, and A. Boukerche, "Cooperative location verification for vehicular ad-hoc networks," in *2012 IEEE International Conference on Communications (ICC)*, pp. 37–41, IEEE, 2012.

[9] U. Ihsan, Z. Wang, R. Malaney, A. Dempster, and S. Yan, "Artificial intelligence and location verification in vehicular networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.

[10] X. Sun, F. R. Yu, and P. Zhang, "A survey on cyber-security of connected and autonomous vehicles (cavs)," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6240–6259, 2021.

[11] M. M. Mehdi, I. Raza, and S. A. Hussain, "A game theory based trust model for vehicular ad hoc networks (vanets)," *Computer Networks*, vol. 121, pp. 152–172, 2017.

[12] F. Reclus, "Geofencing," *Geopositioning and Mobility*, pp. 127–154, 2013.

[13] F. Reclus and K. Drouard, "Geofencing for fleet & freight management," in *2009 9th International Conference on Intelligent Transport Systems Telecommunications,(ITST)*, pp. 353–356, IEEE, 2009.

[14] S. Rodriguez Garzon and B. Deva, "Geofencing 2.0: taking location-based notifications to the next level," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 921–932, 2014.

[15] M. Maiouak and T. Taleb, "Dynamic maps for automated driving and uav geofencing," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 54–59, 2019.

[16] S. W. Rahate and M. Shaikh, "Geo-fencing infrastructure: Location based service," *International Research Journal of Engineering and Technology*, vol. 3, no. 11, pp. 1095–1098, 2016.

[17] N. A. Alrajeh, M. Bashir, and B. Shams, "Localization techniques in wireless sensor networks," *International journal of distributed sensor networks*, vol. 9, no. 6, p. 304628, 2013.

[18] J.-P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security & Privacy*, vol. 2, no. 3, pp. 49–55, 2004.

[19] W. Jabbar, R. Malaney, and S. Yan, "Location information verification in future vehicular networks," in *2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)*, pp. 1–5, IEEE, 2020.

[20] J. Grover, M. S. Gaur, V. Laxmi, and R. K. Tiwari, "Detection of incorrect position information using speed and time span verification in vanet," in *Proceedings of the Fifth International Conference on Security of Information and Networks*, pp. 53–59, 2012.

[21] R. Purkait, S. Tripathi, and S. Patnaik, "Location verification and stability with trust management for avoiding malicious vehicles in geographic routing for vanet," in *2021 Smart City Challenges & Outcomes for Urban Transformation (SCOUT)*, pp. 243–248, IEEE, 2021.

[22] H. D. Weerasinghe, R. Tackett, and H. Fu, "Verifying position and velocity for vehicular ad-hoc networks," *Security and Communication Networks*, vol. 4, no. 7, pp. 785–791, 2011.

[23] S. A. Soleymani, A. H. Abdullah, M. Zareei, M. H. Anisi, C. Vargas-Rosales, M. K. Khan, and S. Goudarzi, "A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing," *IEEE Access*, vol. 5, pp. 15619–15629, 2017.

[24] S. Mukherjee, A. M. Wallace, and S. Wang, "Predicting vehicle behavior using automotive radar and recurrent neural networks," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 254–268, 2021.

[25] U. Government, "Speed limits," Mar 17, 2024.

[26] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *Ieee Potentials*, vol. 13, no. 4, pp. 27–31, 1994.

[27] C. Banerjee, T. Mukherjee, and E. Pasiliao Jr, "An empirical study on generalizations of the relu activation function," in *Proceedings of the 2019 ACM Southeast Conference*, pp. 164–167, 2019.

[28] A. Coppola, L. Di Costanzo, L. Pariota, and G. N. Bifulco, "Fuzzy-based variable speed limits system under connected vehicle environment: A simulation-based case study in the city of naples," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, pp. 267–278, 2023.

[29] L. A. Zadeh, "Fuzzy logic," *Computer*, vol. 21, no. 4, pp. 83–93, 1988.

[30] F. M. McNeill and E. Thro, *Fuzzy logic: a practical approach*. Academic Press, 2014.

[31] I. Iancu, "A mamdani type fuzzy logic controller," *Fuzzy logic-controls, concepts, theories and applications*, vol. 15, no. 2, pp. 325–350, 2012.

MARIA DROLENCE MWANJE is currently pursuing a Ph.D. at Nottingham Trent University within the School of Science and Technology. Prior to this, she completed her M.Sc. in Information Technology Security at the same university, having been awarded the Commonwealth Shared Scholarship in 2020 for her master's program. She holds a first-class B.Sc. in Computer Science from Mbarara University of Science and Technology. In 2017, she was selected as one of the winners of the Huawei Seeds for the Future Program winners, which sponsored outstanding university students for a study trip to China. A district quota scholarship from the Ugandan government supported her undergraduate studies. Her research interests currently encompass vehicular technology, artificial intelligence, and human activity recognition.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2024.3453666

OMPRAKASH KAIWARTYA (Senior Member, IEEE)(M 14, SM 19) is currently working as a Senior Lecturer and PhD Research Coordinator at the Department of Computer Science, Nottingham Trent University (NTU), UK. Previously, He was a Research Associate at the Northumbria University, Newcastle, UK, in 2017 and a Postdoctoral Research Fellow at the Universiti Teknologi Malaysia (UTM) in 2016. He received his Ph.D. degree in Computer Science from Jawaharlal Nehru University, New Delhi, India, in 2015. He is an expert in 'Network Communication and Cyber Security' working with industries to solve emerging network challenges in Connected Vehicles, EV Charging, and the Internet of Things. He is an Internationally well-known researcher with 150+ articles including 37+ in high impact IEEE Transactions/IEEE Journals with 5663+ citations (h-index 43). He is Associate Editor/Guest Editor of reputed SCI Journals including IEEE IoT, IET Intelligent Transport Systems, EURASIP Journal on Wireless Communication and Networking, MDPI Sensors, and Electronics, Wireless Communications and Mobile Computing Hindawi. Ad-Hoc and Sensor Wireless Networks, and Transactions on Internet and Information Systems. He is leading projects as PI worth more than £600K at NTU.

ABDALLAH NASER is a Lecturer with the Department of Computer Science at Nottingham Trent University, where he is also a member of the Computational Intelligence and Applications (CIA) research group. He received his PhD degree from Nottingham Trent University, UK. He completed his B.Sc. degree in computer engineering from Cyprus International University with a High-Honour class and an M.Sc. degree in information security and biometrics with Distinction from the University of Kent. He granted a patent for inventing a new way to secure online accounts using concepts of pattern recognition from the Palestinian patent office. Besides, he was one of the winning teams in the K-Hackathon competition organised by Middle East Technical University for developing a classification IoT system to detect car accidents in addition to other academic awards. His current research interests focus on utilising privacy-preserving sensors for occupancy monitoring through image processing and machine learning approaches.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/