IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 November 2022, accepted 8 December 2022, date of publication 15 December 2022,
date of current version 21 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3229623

RESEARCH ARTICLE

An Efficient Localization and Avoidance Method of Jammers in Vehicular Ad Hoc Networks

IMAN ALMOMANI[1,2], (Senior Member, IEEE), MOHANNED AHMED[1], DIMITRIOS KOSMANOS[3], AALA ALKHAYER[1], AND LEANDROS MAGLARAS[1,4], (Senior Member, IEEE)

[1] Security Engineering Laboratory, Computer Science Department, Prince Sultan University, Riyadh 11586, Saudi Arabia
[2] Computer Science Department, King Abdullah II School of Information Technology, The University of Jordan, Amman 11942, Jordan
[3] Department of Electrical & Computer Engineering, University of Thessaly, 38334 Volos, Greece
[4] School of Computing, Edinburgh Napier University, EH14 1DJ Edinburgh, U.K.

Corresponding author: Iman Almomani (imomani@psu.edu.sa)

This work was supported by the Prince Sultan University, Saudi Arabia, under Grant SEED-CCIS-2022-115.

ABSTRACT Jamming is a terrifying attack that could harm 802.11p-based vehicular communications by occupying the communication channels by overwhelming the network with jamming packets, especially for self-driving cars, as it is essential to send/receive messages without any interruptions to control the vehicles remotely. In wireless vehicular ad hoc networks (VANET), the attacker's mission is more accessible due to the network's open nature, way of communication, and lack of security measures. Most of the existing studies have focused on jamming detection approaches. However, few of them have addressed the jammer localization challenge. Moreover, even in these limited studies, the solutions' assumptions, the proposed countermeasures, and their complexity were also missing. Therefore, this paper introduces a new approach to detecting, localizing, and avoiding jamming attacks in VANETs with high efficiency in terms of accuracy, implementation and complexity. The proposed approach uses the signal strength of the jammer for estimating only the distance between jammer and receiver, while then a less complex algorithm is proposed for localizing the jammer and then redirecting the vehicles away from the roads the attacker is using. This approach was simulated using real-life maps and specialized network environments. Additionally, the performance of the new approach was evaluated using different metrics. These evaluation metrics include (1) the estimated position of the jammer, (2) the handling of the jammer by announcing its location to normal vehicles (3) the avoidance of the jammed routes by increasing their weight, which forces the cars to reroute and evade the jamming area. The high localization accuracy, measured by the Euclidean distance, and the successful communication of the attacker's position and its avoidance have highly increased the packet delivery ratio (PDR) and the signal-to-interference-plus-noise ratio (SINR). This was noticed significantly before and after avoiding the jamming area when for example, the PDR increased from 0% to 100% before and after bypassing the jammer's routes.

INDEX TERMS Ad hoc network, avoidance, detection, euclidean distance, jamming, localization, OMNeT++, pdr, routing, SUMO, smart cities, RSU, SINR, VANET, Veins.

I. INTRODUCTION

Transportation is reshaped by the advancement of autonomous vehicles. Autonomous vehicles employ embedded systems to interact with their external environment and perform the corresponding smart decisions [1]. The phrase vehicular ad hoc networks(VANET) refers to vehicle nodes that provide various services such as accident avoidance, traffic management, and intelligent parking. Thus, mobile or static vehicles serve as a transportation messenger and are investigated in several research areas, including the Internet of vehicles and vehicular ad hoc networks [2]. In autonomous

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Sookhak.

vehicles, there are two types of communication that can be implemented to provide the drivers with the required convenience [3]. Firstly, vehicle-to-vehicle (V2V) communication utilizes intelligent appliances and wireless networks (e.g., cellular) to allow drivers to share critical information while driving, such as hazardous conditions on the road. Secondly, vehicle-to-infrastructure (V2I), in which autonomous features are implemented in the vehicles by deploying various sensors. The innovative developments in autonomous vehicles have converted traditional vehicles into intelligent devices that provide convenience and safety.

However, the evolved dependency on wireless communication makes autonomous vehicles susceptible to various security threats that can have disastrous consequences [4], [5], [6]. Nevertheless, due to the high traffic and the data complexity of the autonomous vehicle networks, implementing security solutions is challenging [7]. Wireless connections are especially vulnerable to radio frequency (RF) jamming attacks [8]. In a jamming attack, the attacker interrupts the legitimate radio signal by emitting malicious signals to the communication. Thus, the radio jamming attack can be considered the most threatening attack among other security threats for the following reasons. Firstly, the ease of launching an RF jamming attack [9], [10]. An attacker can easily perform a jamming attack using a USB or a software-defined radio (SDR) device as a jamming emitter [8]. Secondly, despite the considerable enhancement of wireless communications, anti-jamming solutions are still limited. Due to the lack of anti-jamming mechanisms, most recent wireless communication networks, such as Wi-Fi and cellular networks, can be easily exploited by jamming attacks [8], [10]. The reasons above raise the urgent necessity for designing and implementing anti-jamming systems.

Due to the emergence and significance of this field, various solutions have been put forward to prevent RF jamming attacks [11], [12], [13]. An efficient detection algorithm is essential in anti-jamming schemes to ensure secure wireless communication. After detecting the jamming attack, it is essential to allocate the attacker to perform the required countermeasures. Therefore, the main contributions of this research are summarized as follows:

- Providing a comprehensive comparative analysis of recent works related to VANET jamming detection and localization.
- Proposing a new approach for detecting, localizing, and avoiding different types of VANET jammers, including constant and reactive jammers.
- Localizing VANET jammers with high accuracy and less complexity & assumptions. E.g., the proposed approach succeeded in calculating the position of the jammer in the case of only two receivers of the jamming signals.
- Avoiding the jamming area by sharing the location of the jammers through the roadside unit and applying a rerouting mechanism to evade the routes the jammers are using.
- Implementing and evaluating the proposed approach with all its services and functionalities using specialized simulation environments such as OMNeT++ 5.6.2, SUMO 1.8.0, and Veins 5.2.
- Proving the efficiency of the proposed approach in accurately locating the jammer and successfully avoiding the jamming area with minimum requirements and assumptions. The evaluation metrics included Euclidean distance, PDR, and SINR.

The rest of this paper is organized as follows. Section II. presents the related works in detecting and locating jamming attackers. Section III describes the proposed work in detecting, localizing, and avoiding VANETs' Jammers. Section IV shows the simulation environments and implementations. Section V. presents the evaluation results and discussions. Finally, Section VI concludes the work and draws possible future directions.

II. LITERATURE REVIEW

This section highlights recent studies focusing on jamming detection and localizing wireless jammers.

A. JAMMING DETECTION

Kosmanos et al. proposed an algorithm to understand the jammer's behavior by estimating the relative speed between the receiver and the jammer [14]. Furthermore, the jammer's relative direction was also captured along with its speed (u) by utilizing the physical metric of Δ u from radio frequency (RF) communication between the transmitter and the receiver. Subsequently, the authors have deployed the signal-to-interference-plus-noise ratio (SINR) in the simulation software to mimic real-life scenarios. The evaluation results showed the effectiveness of the proposed approach. An intrusion detection system was developed in [15] to detect spoofing attacks on a dynamic wireless charging system. The proposed IDS also utilized the relative speed value to calculate the distance between two communicated vehicles and verify the position of the jammer as a detection metric. The deployment of the new metric increased the accuracy by 6%.

Kasturi et al. proposed a scheme to classify the jamming attack type by employing packet delivery ratio (PDR) and received signal strength (RSS) to train several machine learning algorithms, including Gradient Boosting, k-nearest neighbors (KNN), decision tree (DT), and random forest (RaFo) [16]. Mainly, the jamming attack was classified into three categories: constant, reactive, and random. The experiment results demonstrated that gradient boosting was superior to other algorithms. The authors of [17] have proposed another supervised model in which they deployed Random Forest (RaFo) and K-Nearest Neighbor (KNN) algorithms to develop a model using the variations of relative speed (VRS) between the attacker and the victim as a feature set.

Authors in [18], proposed a protocol called DeepWiFi to help mitigate out-of-network interference and jamming attacks. It uses deep learning to classify various waveforms and whither are affected by jammers.

Another work employed a neural networks model to secure ZigBee communications against constant jamming. [19].

Some studies discussed applying frequency hopping (FH) to overcome jamming. Many of these studies reported the overhead due to the pre-sharing of the key of channel selection and the high throughput required to apply FH. In addition to the un-adequacy of FH to protect 802.11 networks with current spectrum allocations. Also, FH will not be advantageous in the presence of a wide-band jammer that can concurrently jam many bands (and, in the worst scenario, all possible bands) [20], [21].

Some anti-jamming techniques might be used to monitor the interaction status between the vehicles and the roadside unit (RSU). Rajesh et al. utilized unmanned aerial vehicles (UAVs) in observing the communication status by implementing a specific relay strategy; Q-learning analysis [22]. However, if an intelligent jammer interfered with the communication between the RSU and the onboard unit (OBU), the proposed relay strategy assigns the OBU to another RSU. Pirayesh et al. developed a scheme called JammingBird, which consists of two modules for jamming suppressor and jamming-resistant synchronizer to identify, synchronize, and retrieve signals affected by a jamming attack [23]. The JammingBird supports vehicle-to-anything (V2X) routing, a single transmitter connected to two receivers. Initially, the jamming-resistant synchronizer classifies and locates the received signal duration, which contains the legitimate packets. After that, the jamming suppressor filters the received single to remove the jamming effect. Similarly, Santhi et al. developed an Index-based Voting technique (IBVA) against hybrid jamming attacks to secure short-range vehicle communication [24]. The centric focus of the proposed algorithm is the detection of nodes' misbehavior. Furthermore, the authors have proposed a novel recursive candidate elimination algorithm to enforce node authenticity.

B. LOCALIZATION

Estimating the jammer location could be considered a core step in implementing a countermeasure solution against the jammer attack [25]. Furthermore, implementing an accurate localization and cost-efficient algorithm is critical in vehicular networks [26]. In related works, localization in VANETs is mainly achieved via two approaches [26]. The first approach deploys the GPS technique to verify the node position based on the reliable positions of its neighbor [27]. Each node calculates its position using GPS data and broadcasts it. Subsequently, its neighbors can build a positioning table of the neighboring nodes [28]. The second approach depends on the physical parameters such as received signal strength (RSS) [29], angle of arrival (AoA) [30], and time of arrival (ToA) [31] in obtaining the jammer location. These RSS-based solutions achieve quite a low accuracy in estimating the jammer's position.

The boundary nodes of the jammer are utilized to locate the jammer position. Wang et al. proposed the geometrical jammer localization(GJL) algorithm, which uses the jamming strength in a geometric-covering method to localize the jammer [32]. The jammer's location is determined by calculating the minimum covering circle of the boundary nodes. Even though the proposed method achieved low power consumption, it was designed for a stationary network in which the nodes' locations are unchanged until the information on the network topology is obtained. The authors of [33] have also employed the boundary node in tracking the jammer position. After computing the maximum distance between two jammed nodes, a boundary node selection threshold (BNST) was estimated by finding the maximum distance between them and their unjammed neighbors. Another proposed solution by [34] deployed both RSS and packet error rate (PER) in defining both the jammer localization and the amount of energy consumption. However, the proposed solution was designed for ground-to-air communication in aerial vehicles.

In [35], the authors located the jammed nodes in broadcast networks by developing an algorithm named the number of jammed slots (NJS), which collects the MAC-layer statuses of the jammed nodes at the software level. The NJS increases the number of jammed slots in each node if the node is receiving a meaningless signal from an illegitimate wireless device. Thus, the presence of a jammer can be detected from the non-zero NJS value. Furthermore, by calculating the NJS values of neighbor nodes, the approximate location of the jammers can be estimated. The authors of [36] put forward an antenna identification and localization of the jammer (AILJ) algorithm, which deploys the geometry of the jammed nodes and boundary nodes to localize the jammer. The required information about the jammed and boundary nodes was initially obtained using the received jamming signal strengths (RJSSs). Subsequently, the type of jammer antenna is derived based on the boundary nodes classification. At last, the jammer position is localized by calculating the mean value of the intersections point between circumcircles.

Table 1 summarizes and compares the recent work related to jamming detection and localization in the context of VANET. As can be observed from this comparison, state of the art differs in (a) their main aim & approaches, (b) communication type, (c) the simulation environment they have used, (d) type of jamming considered, (e) and their evaluation measures and values. Additionally, many of the above comparison metrics were not stated in many of them. Moreover, the current approaches did not highlight the complexity of their solutions & their assumptions or requirements, and whether they have considered moving jammers. Another remarkable comment is the limited existing approaches that managed to localize the jammers in VANETs. Therefore, this paper presents a new approach to detecting different types of jammers and calculating their locations with high efficiency in terms of assumptions, implementation, accuracy, and complexity. Specifically, the proposed approach proposes a passive estimation of the distance between the jammer and receiver using the propagation model of the wireless V2V communication between normal nodes under the interference of the jammer. After that, a low-complexity algorithm is used, which accepts as input the estimated distance between the jammer-receiver for the effective positioning of the jammer.

III. PROPOSED WORK

This section proposes a new, efficient approach that provides many essential services related to jammers' implementation, detection, localization, and avoidance. These services are detailed in the following sections.

A. JAMMING

There are two types of jammers implemented in this research: constant jammer (Algorithm 1) and reactive jammer (Algorithm 2). For the constant jammer, once the simulation starts, the jammer keeps sending packets every interval (*interval* parameter in algorithms 1, 2), affecting all the nodes within the jammer's range until a number of repetitions is reached (*repetitions* parameter in algorithms 1, 2). The interval and the repetitions are decided based on the amount of interference planned by the attacker. The interval indicates every when the jammer sends the packets, whereas the repetitions are the total number of jamming packets to be sent. For the reactive one, the jammer starts sending packets once it senses that there is a specific number of packets being sent between the communicating cars (according to a threshold, jammerThreshold in Algorithm 2).

B. HANDLING JAMMERS

To handle jamming attackers, three stages need to be followed as elaborated in Figure 1.

1) Detection: the possibly jammed car counts the number of packets received from similar distances, which are calculated from the physical layer using the propagation model, and stores them in a map data structure (*DistanceCountMap*) in Algorithm 3. This structure maps the estimated distance with its counter. The similarity between two distances is determined by checking if their absolute difference is less than EPSILON (a value that needs to be tuned). Once one distance value count reaches a specific value (*e.g., the threshold in Algorithm 3*), the car sends to the RSU its position information(x, y) along with the estimated distance from the suspected jammer, the method of estimation of which is described in the following paragraph (*Algorithm 4: Detection part*).
2) Estimating the distance between Jammer - Receiver: Before starting the analysis of the core of the localization algorithm, we need to introduce some basic assumptions for the estimation of elements related to the jammer (e.g., the distance between the jammer and the jammed car), basics of passive V2V wireless communication between a normal transmitter and a receiver (e.g., the jammed car). This passive estimation is done because the jammer does not want to betray

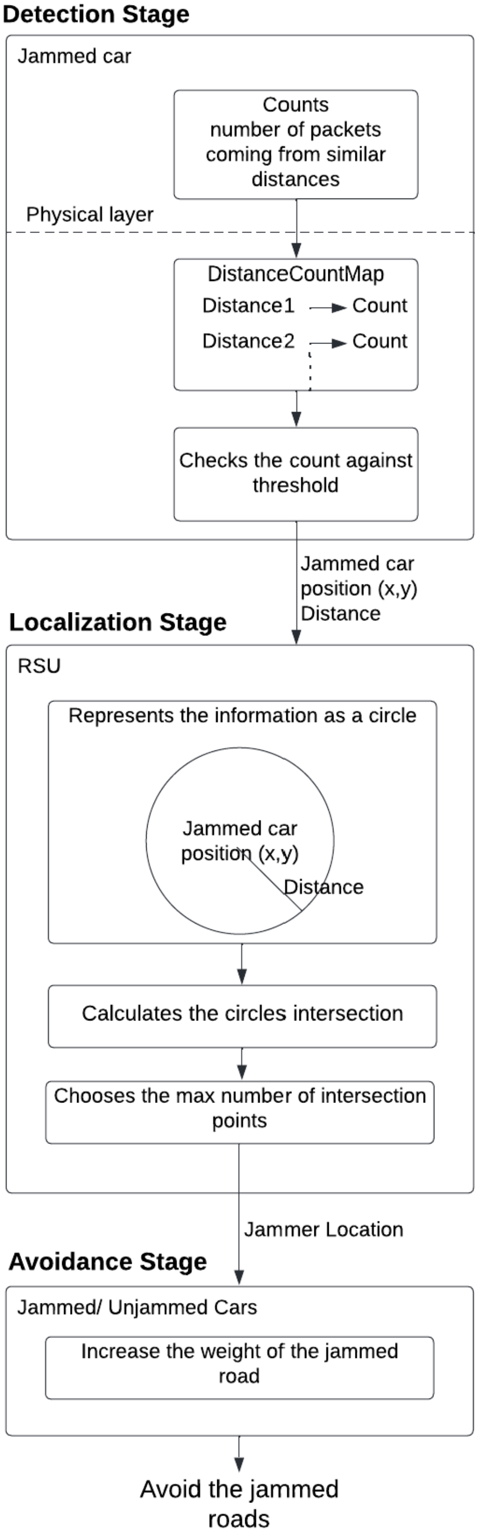

Detection Stage
Jammed car
Counts number of packets coming from similar distances
Physical layer
DistanceCountMap
Distance1 → Count
Distance2 → Count
Checks the count against threshold
Jammed car position (x,y) Distance
Localization Stage
RSU
Represents the information as a circle
Jammed car position (x,y)
Distance
Calculates the circles intersection
Chooses the max number of intersection points
Jammer Location
Avoidance Stage
Jammed/ Unjammed Cars
Increase the weight of the jammed road
Avoid the jammed roads

FIGURE 1. The proposed stages of handling VANET's jammers.

TABLE 1. Summary of current works related to jamming attacks detection and localization.

	Related Work	The Aim	Approach	Communication Type	Simulation SW	Type of Jamming	Performance Evaluation
Jamming detection related works	[14]	To estimate the relative speed of a mobile vehicle that interferes with RF communication of a transmitter-receiver pair.	Radio frequency (RF), angle of projection (AOP)	Unicast V2V communication	Veins-Sumo simulator	Jamming: Denial of service (DoS) attack	SINR, MAE, relative speed estimation
	[15]	To develop an Intrusion Detection System to detect spoofing attacks by utilizing Machine Learning techniques to protect electric vehicles	K-NN, RaFo	Unicast V2V beacon-based routing	SUMO, OMNET++/VEINS, GEMV	Spoofing attack	Busy Time, Average Travel Time, MAC Layer Delay, Average Waiting Time, ROC curve
	[16]	To classify the jamming attack type by employing Delivery Ratio (PDR) and Received Signal Strength (RSS) to train several machine learning algorithms.	Gradient Boosting, KNN,RaFo, DT	V2V	Network simulator NS-3	constant, reactive, and random jamming attacks	Accuracy (for changing distance, transmission power)
	[17]	To deploy supervised learning (KNN, RaFo) in developing a detection scheme by utilizing the variations of relative speed	K-NN, RaFo	Unicast V2V	SUMO, OMNET++/VEINS, GEMV	Jamming attack	Signal-to-interference-and-noise ratio (SINR), Confusion matrix
	[22]	To utilize unmanned aerial vehicles (UAVs) in observing the communication status by implementing Q-learning analysis.	Q-learning analysis	Unicast V2V	Not mentioned	Jamming attack	SINR, bit-error-rate (BER)
	[23]	To develop two modules for jamming suppressor and jamming-resistant synchronizer in order to identify, synchronize, and retrieve signals affected by a jamming attack.	signal processing	V2X communication	Not applicable	Jamming attack	Three outdoor scenarios
	[24]	To secure vehicles communication by developing an index-based voting technique against hybrid jamming attacks	IBVA	V2I	Network simulator NS-2	Hybrid jamming	Average End-to-End delay, PDR, throughput
	[37]	To propose a a hideaway anti-jamming strategy for VANET infrastructure	Channel Surfing and Hideaway	V2I	JiST/SWANS	Static and semi-/dynamic jamming	PSR, SEG
Jamming localization related workS	[35]	To locate the jammed nodes in broadcast networks by collecting the MAC-layer statuses of the jammed nodes at the physical layer.	The number of jammed slots (NJS)	Unicast V2V communication	MIXIM 2.3 framework, omnet++	Proactive, reactive jamming	The Precision and accuracy metric, the First detection time
	[36]	To deploy the geometry of the jammed nodes and boundary nodes in order to localize the jammer.	RJSS	V2V	Not mentioned	Constant jamming	The mean localization error,
	[32]	To utilize the jamming strength in a geometric-covering method in order to localize the jammer.	GJL	V2V	Not mentioned	Constant jamming	localization error
	[33]	To select boundary nodes in a moving jammer attack scenario by computing the maximum reachable sensing range of a node.	Extended Kalman filter	V2V	Not mentioned	Constant jamming	The percentage error of the selected boundary nodes
	[25]	To propose a localization system for advanced metering infrastructure (AMI) against mobile jamming attacks	DiffLSQ, LimTrack	AMI communication	MATLAB	Jamming attack	Euclidean distance, root mean squared error
Ours		**To propose an algorithm to detect and calculate the location of the jammer with the minimum complexity using a mathematical model based on the distance derived from the propagation model.**	**mathematical model & propagation model**	**V2V, I2V, V2I**	**OMNeT++, SUMO, Veins**	**Constant & Reactive jamming**	**PDR, SINR, Euclidean Distance**

any characteristic of his movement. The only way to estimate information related to the jammer is by using the received power of the jammer to a normal node through a known propagation model. So, we used the known power adaptation techniques [38] to estimate the distance between the jammer and the receiver. Also,

Algorithm 1 Constant Jamming (Jammer)

```
procedure startConstantJamming(interval, repetitions)
    $jammingPacket \leftarrow createJammingPacket()$;
    $sendPacketToAllCars(jammingPacket).interval(interval).repetitions(repetitions)$;
end procedure
```

Algorithm 2 Reactive Jamming (Jammer)

```
procedure startReactiveJamming(jammerThreshold, interval, repetitions)
  ▷ PacketsCount increments at the physical layer when packets with different destination IP received by the jammer
    Global: PacketsCount
    if PacketsCount > jammerThreshold then
        $jammingPacket \leftarrow createJammingPacket()$;
        $sendPacketToAllCars(jammingPacket).interval(interval).repetitions(repetitions)$;
    end if
end procedure
```

Algorithm 3 Jamming Detection (Receiver)

```
procedure isThereJammer(packet,threshold)
    Global: DistanceCountMap, maxDistance
    $isFound \leftarrow false$;
    for possibleJammerDistance in DistanceCountMap
        if abs(possibleJammerDistance - distance) < EPSILON then
            $DistanceCountMap[possibleJammerDistance] + +$;
            $isFound \leftarrow true$;
            if $DistanceCountMap[possibleJammerDistance] \geq threshold$ then
                $maxDistance \leftarrow possibleJammerDistance$
            end if
        end if
    end for
    if !isFound then
        $distanceInt \leftarrow (int)packet.getDistance()$;
        $DistanceCountMap[distanceInt] \leftarrow 1$;
        $isFound \leftarrow false$;
    end if
    return $maxDistance > 0$;
end procedure
```

we modified these techniques to be applicable to the propagation model, which is one of the most common models used in VANETs for multipath radio propagation and is also an ideal model for many wireless applications. This model is also implemented in the VEINS that we used in our simulations. The model considers the direct path and the ground-reflected direction. So, the signal strength P_{TR} received by a receiver (R) due to a signal emanating from a transmitter (T) is expressed as follows:

$$P_{TR} = \frac{P_T G_T G_R (H_T^2 \times H_R^2)}{d_{TR}^4 L} \tag{1}$$

where the reception power, the transmitter and receiver antenna's height, and the distance between the transmitter-receiver pair are signified by P_{TR}, H_T and H_R, and d_{TR}, respectively. G_R and G_T denote the antenna's gain of receiving and transmitting, respectively, with L as system losses, that also accounts for fast fading effects. In this way, we take into account in our scenarios multi-path propagation phenomenon in urban environments for unicast communication between any nodes. All the parameters of the above relation can be assumed to be known to the receiver, such as the distance d_{TR}. Moreover, we assume that all vehicles participating in the simulation, whether normal or jammer have the same height.

From (1) we can derive the minimum SINR value at which the communication from T to R is possible while under J's interference as:

$$SINR_{MIN} = \frac{\hat{P_T} G_T (H_T^2 \times H_R^2) d_{JR}^4}{P_J G_J (H_T^2 \times H_R^2) d_{JR}^4} \tag{2}$$

Algorithm 4 Processing Received Packets (Receiver)

```
procedure processPacket(packet)
    Global: maxDistance
  ▷ Avoidance Part
    if $isFromRSU(packet)$ then
        $xi1 \leftarrow packet.getXI1();$
        $yi1 \leftarrow packet.getYI1();$
        $xi2 \leftarrow packet.getXI2();$
        $yi2 \leftarrow packet.getYI2();$
        if $xi2 < 0 \,||\, yi2 < 0$ then
            $jammerRoadId \leftarrow getRoadMapPos(xi1, yi1);$
            $changeRoadWeight(jammerRoadId, 99999.9);$
            return;
        end if
        $jammerRoadId1 \leftarrow getRoadMapPos(xi1, yi1);$
        $changeRoadWeight(jammerRoadId1, 99999.9);$
        $jammerRoadId2 \leftarrow getRoadMapPos(xi2, yi2);$
        $changeRoadWeight(jammerRoadId2, 99999.9);$
    end if
  ▷ Detection Part
    if $isThereJammer(packet)$ then
        $currentPosition \leftarrow getCurrentPosition();$
        $newPacket \leftarrow createPacket(currentPosition.getX(), currentPosition.getY(), packet.getDistance());$
        $sendPacketToRSU(newPacket);$
        $maxDistance \leftarrow -1$                                        ▷ Reset
    end if
end procedure
```

where, $\hat{P_T}$ is the minimum transmitting power required by T to deliver a message to R successfully. P_J is the transmitting power of the jammer and G_J is the antenna gain of the jammer. The transmitting power of the jammer and the antenna gain of the jammer is unknown to the receiver.

In (2), P_J can also be obtained by the node T. So, the following equation is derived from the strength of the jamming signal as seen by the transmitting node T as:

$$P_J = \frac{d_{JT}^2 \hat{P}_{JT} L}{G_J G_T (H_J^2 \times H_T^2)} \tag{3}$$

Finally, we can combine (2) and (3) such as:

$$SINR_{MIN} = G_T^2 \frac{\hat{P}_T}{\hat{P}_{JT}} \left(\frac{d_{JR}}{d_{JT}}\right)^4 \frac{(H_J^2 \times H_T^2)}{d_{TR}^4 L} \tag{4}$$

From the above equation, in the last term $\frac{(H_J^2 \times H_T^2)}{d_{TR}^4 L}$, the height of the vehicles has a specific stable value as does the distance between transmitter and receiver. The parameter of channel losses (L) can be determined through simulations of wireless transmitter-receiver communication implementing this propagation model. The other unknown variable is the received power to the transmitter T from the jammer $\hat{P}_{JT}$. This value can be determined at the receiver by subtracting the noise value from the total signal strength that T perceives when there is no legitimate packet transmission. So, we have one equation (equation (4)) with two unknowns (the distances d_{JR} and d_{JT}). We will arrive at a second corresponding equation with these two unknowns if we follow the same approach by exchanging the roles of transmitter (T) and receiver (R). Therefore, we will have two equations with two unknowns from which we can calculate both unknown values of the distances. However, the distance we need for the localization algorithm is that of the distance between jammer and receiver (d_{JR}).

3) Localization: The RSU receives information sent by the jammed cars and stores them into arrays (Algorithm 5). To get the location, we write each jammed car information as an equation of circle where x and y are the center coordinates and the distance (d_{JR}), which has been calculated using the relation (4)), is the radius. Then, we can find the location by calculating the circles' intersection by setting the equations to be equal to each other, as shown in Equation 5.

$$(x - x_1)^2 + (y - y_1)^2 - r_1^2 = \tag{5}$$

$$(x - x_2)^2 + (y - y_2)^2 - r_2^2 \tag{6}$$

Algorithm 6 shows handling the different cases of circle-circle intersections.

Algorithm 5 Processing Received Packets (RSU)

```
procedure processPacket(packet)
    Global: jammedCarsArraySize, X, Y, Distances
    if $isNewJammedCar(packet)$ then
        $x \leftarrow packet.getX()$;
        $y \leftarrow packet.getY()$;
        $distance \leftarrow packet.getDistance()$;
        $newIndex \leftarrow jammedCarsArraySize - 1$;
        $X[newIndex] \leftarrow x$;
        $Y[newIndex] \leftarrow y$;
        $Distances[newIndex] \leftarrow distance$;
        $jammedCarsArraySize++$;
        $xi1, yi1, xi2, yi2 \leftarrow getJammerLocation(X, Y, Distances, size)$
        $newPacket \leftarrow createPacket(xi1, yi1, xi2, yi2)$;
        $sendPacketToCars(newPacket)$;
    end if
end procedure
```

Algorithm 6 Circles Intersections Algorithm

```
procedure getIntersections(x1,y1,r1,x2,y2,r2)
    $d \leftarrow sqrt(pow((x2 - x1), 2) + pow((y2 - y1), 2))$
    if $d > r1 + r2$ then
        return $None$;
    end if
    if $d < abs(r1 - r2)$ then
        return $None$;
    end if
    if $d == 0$ && $r1 == r2$ then
        return $None$;
    end if
    $a \leftarrow (pow(r1, 2) - pow(r2, 2) + pow(d, 2))/(2 * d)$;
    $h \leftarrow sqrt(pow(r1, 2) - pow(a, 2))$;
    $x3 \leftarrow x1 + a * (x2 - x1)/d$;
    $y3 \leftarrow y1 + a * (y2 - y1)/d$;
    $x4 \leftarrow x3 + h * (y2 - y1)/d$;
    $y4 \leftarrow y3 - h * (x2 - x1)/d$;
    $x5 \leftarrow x3 - h * (y2 - y1)/d$;
    $y5 \leftarrow y3 + h * (x2 - x1)/d$;
    return $x3, y3, x4, y4$;
end procedure
```

If there are two jammed cars, we get two possible locations for the jammer (Figure 2-a). One of them is the jammer's correct position, which will be shared by all cars within the area.

In case there are three or more jammed cars, we will get only one location because we can calculate the circles' intersection twice for two different sets of cars, such as Car1&Car2 and Car1&Car3, and get the common point, which will be the correct position of the jammer(Figure (2-b)).

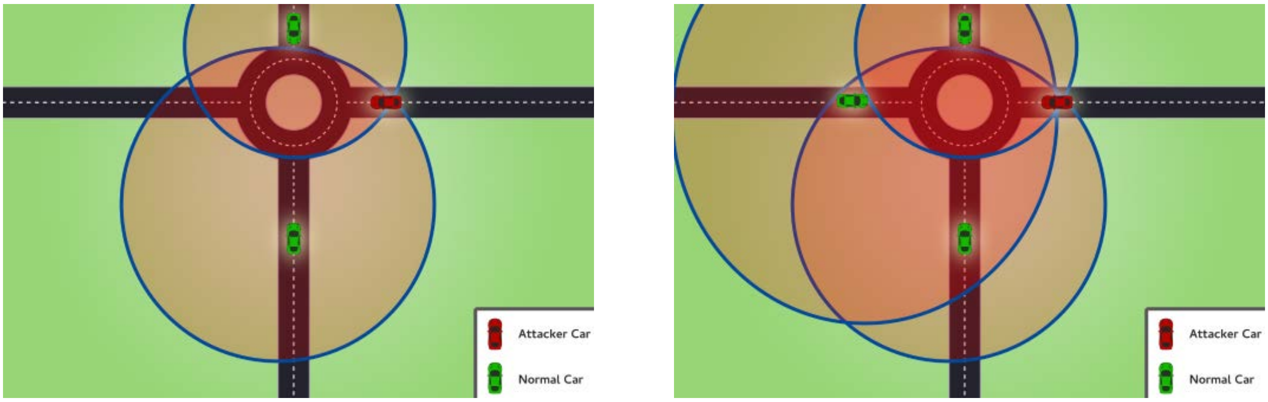
In case there are many circles and no one common intersection point (Figure 3), the circles are filtered based on a radius threshold (e.g. 50m). Then, the circle with max number of intersection points with all the other circles is selected. If there is more than one, the smallest circle (based on the radius) is selected. Finally, the intersection points are sent to all cars as the possible location(s) of the jammer(s) (Xs and Ys) to avoid jammed areas.

The detailed steps and calculations of the localization stage are presented in Algorithm 7.

4) Avoidance: once a car receives a packet containing the position (X, Y) of the jammer sent by the RSU, it avoids the location of the jammer by changing the weight of the jammed road to a high value (e.g., 99999.9) after mapping the road ids to the received Xs, Ys. Therefore, the car chooses routes with lower weight values and avoids the jammed road/area. The details of the avoidance stage are presented in (*Algorithm 4: Avoidance part*). In the case of only two jammed cars, the two intersection points will be considered, and the weight of the two roads will be increased to ensure the avoidance of the normal cars for the jammer road. But, in case of three or more jammed cars, the approach can calculate one location that will be avoided.

IV. SIMULATION ENVIRONMENTS & IMPLEMENTATION

In order to implement different types of VANET jammers and evaluate our proposed approach to detecting, localizing, and voiding them, we utilized three primary specialized simulation environments: OMNeT++ 5.6.2, SUMO 1.8.0, and Veins 5.2. OMNeT++ is a component-based discrete simulation framework written in C++ used to model networking simulations with the support of libraries and packages such as INET. SUMO is generally used to simulate traffic scenarios. To use the traffic simulation offered by SUMO in OMNeT++, Veins uses TraCI APIs to build a bridge between the two simulators to provide vehicular mobility.



(a) Two Cars

(b) Three Cars

FIGURE 2. **Illustration of the proposed localization algorithm.**

Algorithm 7 Localization Algorithm

```
procedure getJammerLocation(Xs,Ys,Ds,size,filteringThreshold)
    $xi1 \leftarrow -1$
    $yi1 \leftarrow -1$
    $xi2 \leftarrow -1$
    $yi2 \leftarrow -1$
    if $size == 2$ then
        $xi1, yi1, xi2, yi2 \leftarrow getIntersections(Xs[0], Ys[0], Ds[0], Xs[1], Ys[1], Ds[1]);$
    else if $size \geq 3$ then
        $ci1 \leftarrow getIntersections(Xs[0], Ys[0], Ds[0], Xs[1], Ys[1], Ds[1]);$
        $ci2 \leftarrow getIntersections(Xs[1], Ys[1], Ds[1], Xs[2], Ys[2], Ds[2]);$
        if $abs(ci1.getX3() - ci2.getX3()) < 0.00001$ && $abs(ci1.getX3() - ci2.getX4()) < 0.00001$ then
            $xi1 \leftarrow ci1.getX3()$
            $yi1 \leftarrow ci1.getY3()$
        else if $abs(ci1.getX4() - ci2.getX3()) < 0.00001$ && $abs(ci1.getX4() - ci2.getX4()) < 0.00001$ then
            $xi1 \leftarrow ci1.getX4()$
            $yi1 \leftarrow ci1.getY4()$
        else if $NoCommonIntersectionPoint(ci1, ci2)$ then
            $filteredCircles \leftarrow filterCircles(Xs, Ys, Ds, filteringThreshold)$
            $circlesWithMaxNumOfInt \leftarrow getMaxCircleInt(filteredCircles)$
            if $circlesWithMaxNumOfInt.length() == 1$ then
                $intAverage \leftarrow getIntAverage(circlesWithMaxNumOfInt[0])$
                $xi1 \leftarrow intAverage.getX3()$
                $yi1 \leftarrow intAverage.getY3()$
            else if $circlesWithMaxNumOfInt.length() > 1$ then
                $smallestCircleInt \leftarrow getSamllestCircleInt(circlesWithMaxNumOfInt)$
                $xi1 \leftarrow smallestCircleInt.getX3()$
                $yi1 \leftarrow smallestCircleInt.getY3()$
            end if
        end if
    end if
    return $xi1, yi1, xi2, yi2$
end procedure
```

Three main agents were developed using Veins: jammer cars agent, normal cars agent, and RSU agent. For the jammer cars agent, it was extended to simulate two varieties of jamming attacks: constant and reactive. The normal car agent has modules for detecting and handling jamming attacks. Finally, the RSU agent helps calculate the jammer's location given

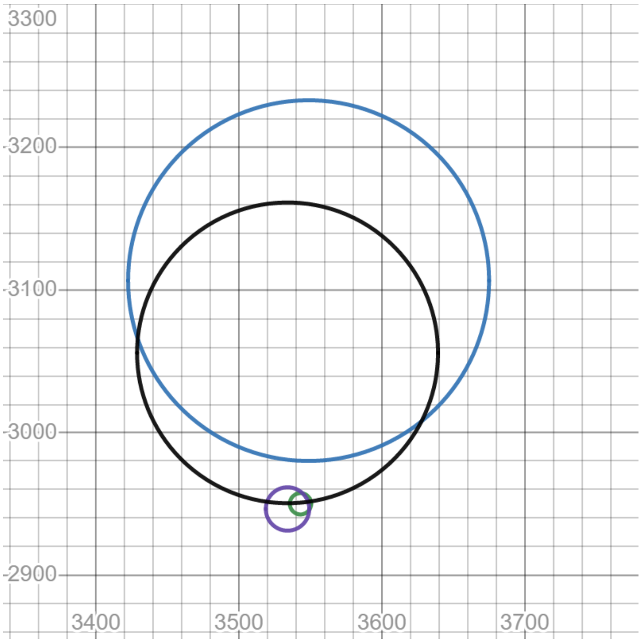


FIGURE 3. Illustration of the proposed localization algorithm in case there are more than three cars.

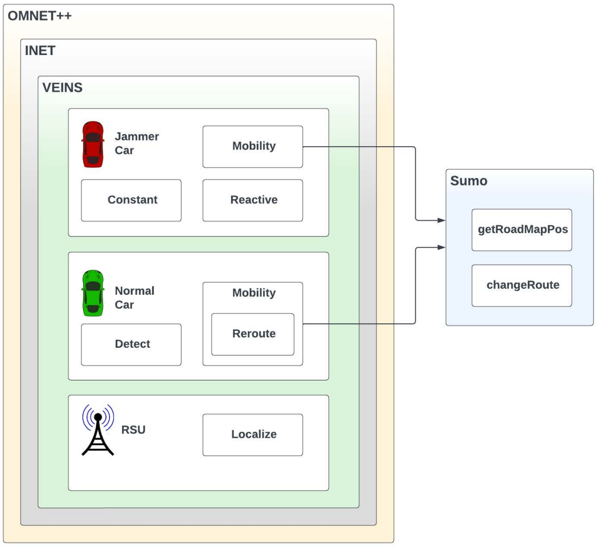


FIGURE 4. Proposed system implemented agents/modules and corresponding simulation environments.

information sent by the cars and then broadcasts it to all the vehicles to avoid the affected routes. Figure 4 illustrates the agents and modules implemented in this study with their corresponding simulation and programming environments. We have conducted several experiments with different scenarios to evaluate our proposed localization and avoidance approach. Table 2 illustrates the simulation parameters we have used in our experiments with changes mainly in the number of receiver cars and jammers.

TABLE 2. Simulation Parameters, * two types of jamming (constant and reactive).

Parameter	Value
Number of normal nodes	5
Number of jammers	2*
Number of RSUs	1
Simulation time	120s
Power of background noise	-86dBm
Transmission power of jammer	50mW
Sensitivity of jammer	-75dBm
Transmission power of normal node	50mW
Transmission power of RSU	100mW
Sensitivity of RSU	-250dBm
Path Loss Model	FreeSpacePathLoss
Jammer's interval	0.5s
Jammer's repetitions	1000
Reactive Jammer's threshold	5
Threshold of detection	3
EPSILON of detection	50
Filtering threshold of localization	50

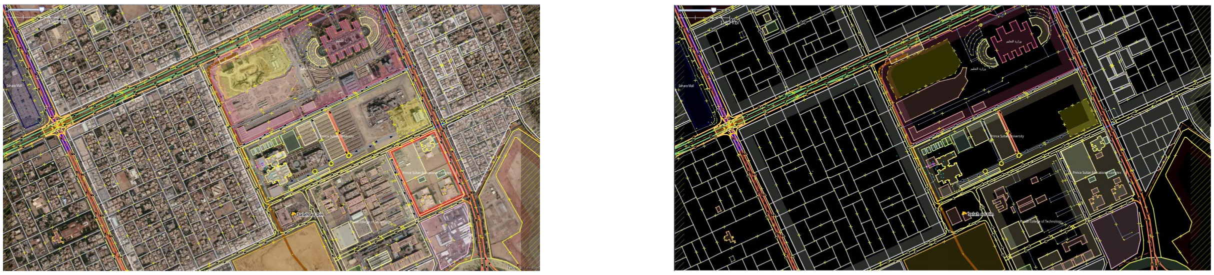
Additionally, the map we used in our experiments is a real map that covers Prince Sultan University area in Riyadh, Saudi Arabia. To extract the map, OpenStreetMap website (https://www.openstreetmap.org/) was used to export the map as an OSM file. Then, the JOSM software editing tool (https://josm.openstreetmap.de/) was used to clean the map and export it into two images (original and blueprint) as shown in Figure 5.

In this paper, to demonstrate the idea of the proposed approach, we considered the scenario with six vehicles in total: one jammer (node [0]), three jammed cars (node [1], node [3], and node [5]), and two unjammed cars (node [2], node [4]). Also, there is one RSU within the given area and at a specific time, as depicted in Figure 6. Regarding the jammer, it can be a constant or a reactive jammer. The scenario starts with a legitimate wireless V2V communication established among any of the cars (1,3 &5). At the same time, a jammer sends many packets affecting the V2V communications, especially the receiver cars within the same jamming area. First, the three vehicles calculate the distance between them and the jammer using the propagation model and the equation (4). Then, the jammed cars send their locations (x and y) to the RSU along with the distance estimated. Next, the RSU calculates the location of the jammer using the proposed Algorithm 7 and sends it to all cars, jammed or unjammed. Finally, other vehicles (in this case, cars 2&4) avoid the jamming area by increasing the weights of the jammed roads in their routing algorithm.

V. EVALUATION AND RESULTS DISCUSSION

The main simulation metrics considered in this study to evaluate the performance of the detection, localization, and rerouting services are the euclidean distance, PDR (packet delivery ratio), and SINR (signal-to-interference-plus-noise ratio), which are calculated in Eq. 7, Eq. 8 and Eq. 9; respectively.

$$Euclidean\ Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{7}$$



(a) Orginal

(b) Blueprint

FIGURE 5. A demonstration of the used map in the simulation.

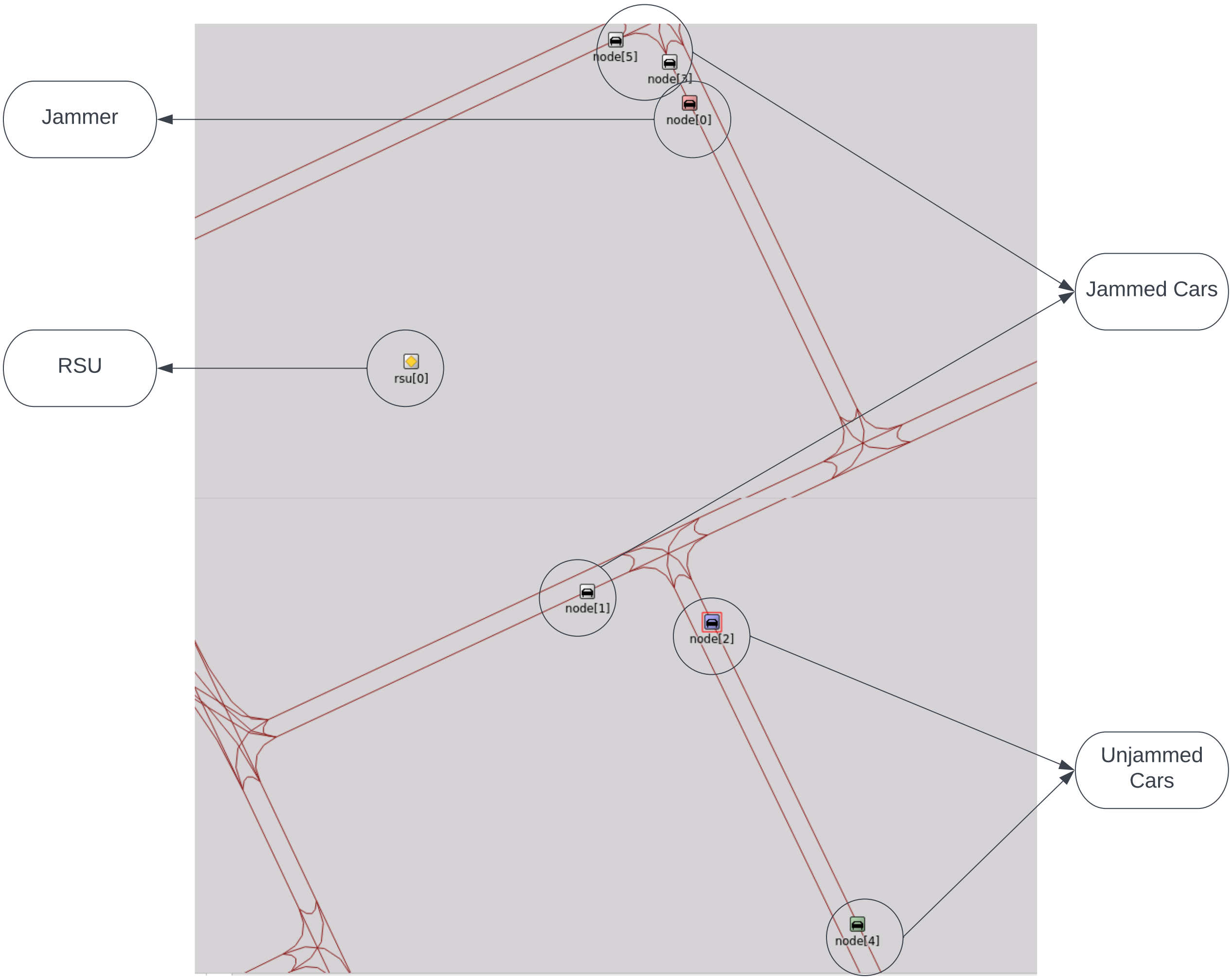


FIGURE 6. Annotated illustration of the simulated scenarios.

$$PDR = \frac{\#\ of\ \ data\ packets\ arrived}{\#\ of\ \ data\ packets\ sent} \tag{8}$$

$$SINR = \frac{Signal\ Power}{Noise\ +\ Interference\ Power} \tag{9}$$

We conducted various experiments and cases to evaluate the proposed localization algorithm, considering the min requirements that might occur during real-life scenarios. Then, the euclidean distance was calculated for each case by taking the actual and predicted jammer locations. As shown in Table 3, the proposed method achieved a meager error rate of 5m. This error is considered low as both the actual and predicted locations are on the same road. Therefore, the cars can exclude the jammed road(s) from their routes.

TABLE 3. Evaluation of the proposed localization method in different low-traffic scenarios.

Scenario	Actual Location (x,y)	Predected Location (x,y)	Localization Error (m)
Two Cars	(3546.34,2956.99)	(3546.9405,2951.7625)	5.2
Three Cars		(3544.82175,2952.11125)	5.1
More Than Three Cars		(3544.82175,2952.11125)	5.1

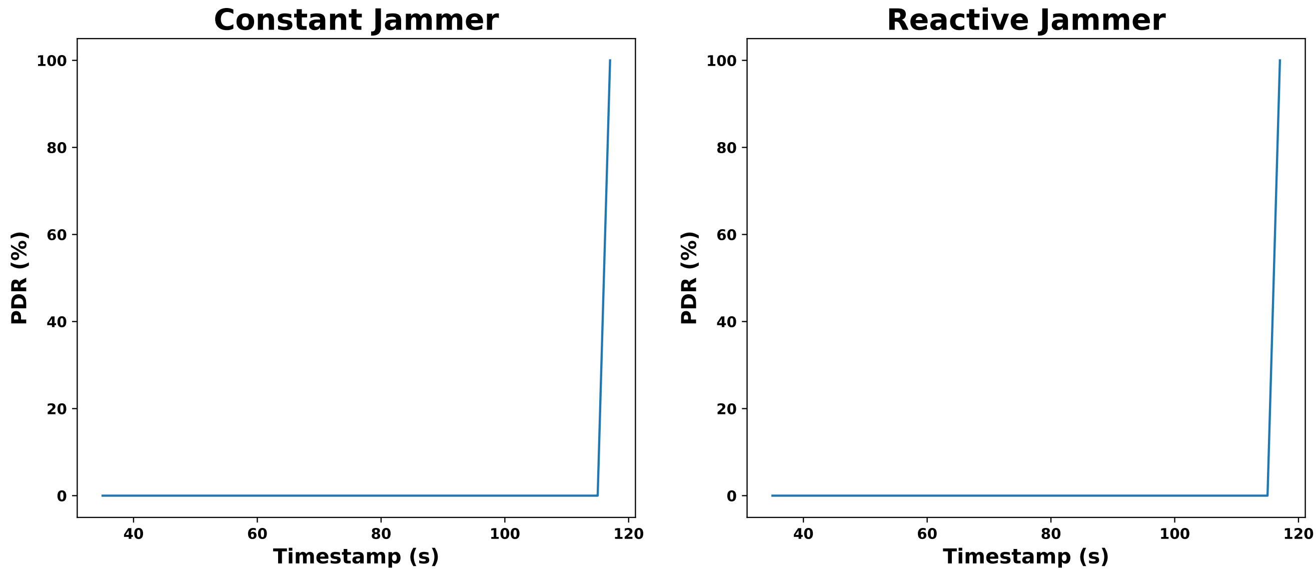


FIGURE 7. Constant vs Reactive jamming in terms of PDR during rerouting scenario.

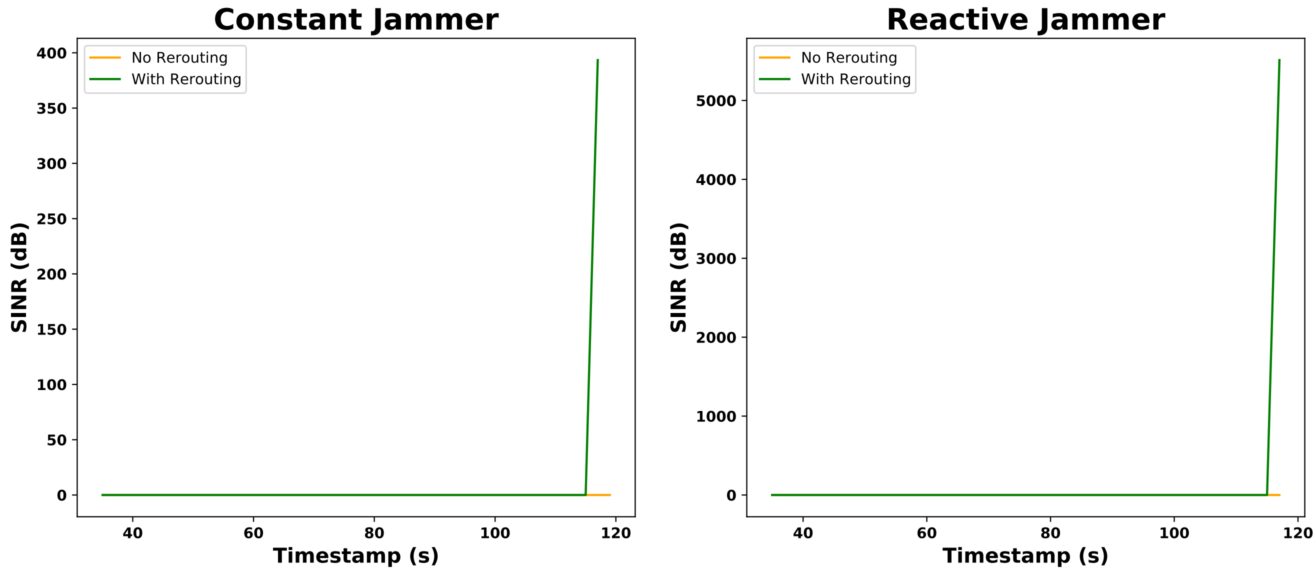


FIGURE 8. Constant vs Reactive jamming in terms of SINR during No-rerouting/rerouting scenarios.

To highlight the impact of the jamming attacks on the data delivery service in VANETs, Figure 7 shows a comparison of PDR values between constant and reactive jammers in the scenario where a car avoids the jammer location after being detected. The PDR values were 0% when the jammed car was in the range of the jammer until the jammer was avoided at the second 110s, then PDR values increased to reach 100%. Additionally, it can be observed that both types of jamming attacks have almost the same effect on the network services. This behavior is justified by the fact that the simulated vehicular network has a lot of wireless traffic, with the jammer being activated all the time.

TABLE 4. Evaluation of the proposed localization method in high-traffic vs low-traffic scenarios with different numbers of jammers.

	Two Cars			Three Cars		
# of Jammers	**Prediction (m)**	**Prediction (Closest) (m)**	**Baseline (m)**	**Prediction (m)**	**Prediction (Closest) (m)**	**Baseline (m)**
1	54.25	0.6	88.62	3.3	3.3	68.27
5	27.4	3.56	112.36	29.85	7.83	111.49
10	40.88	4.07	94.29	42.34	42.34	80.86
20	63.88	4.05	89.52	23.26	2.52	84.64

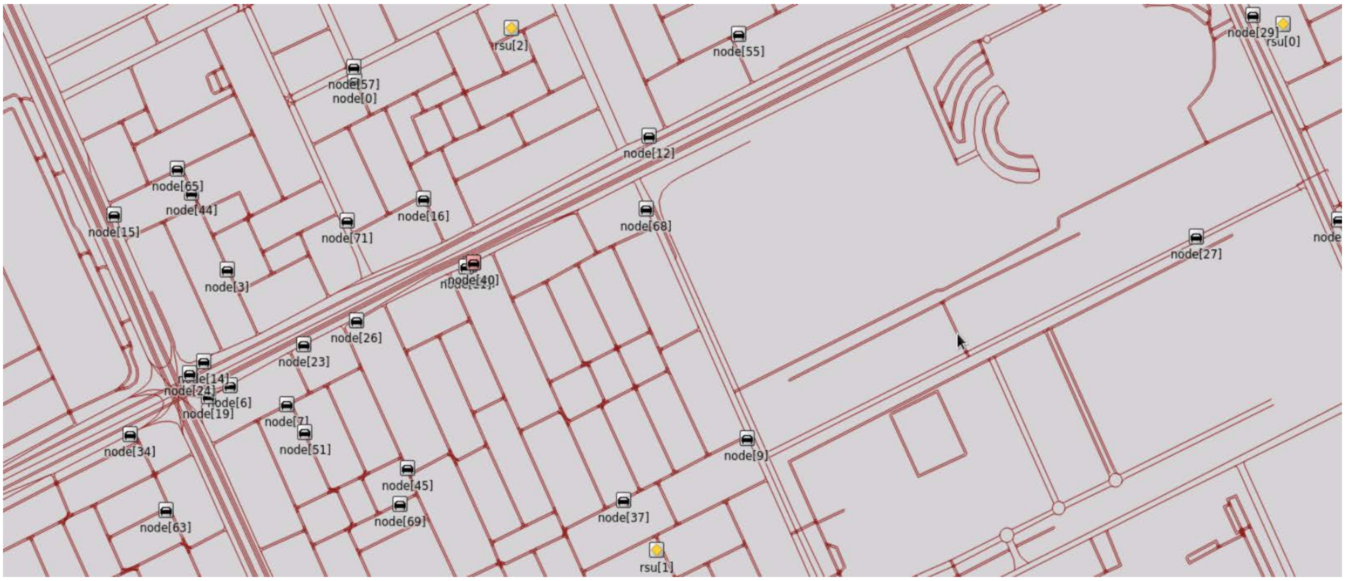


FIGURE 9. Screenshots which show the simulation environment in a low-traffic infrastructure.

Another assessment of the damage caused by VANET's jammers can be done by examining the value of the SINR. Figure 8 illustrates the impact of constant and reactive jammers in two scenarios: (a) when the car avoids the jamming area by applying rerouting to bypass the jammer on its way to the destination, and when (b) it does not avoid it. The experiments' results reveal low SINR values as long as the car does not avoid jammed roads. On the other hand, the moment the car decides to reroute its way and bypass the jammer's routes, the SINR increases regardless of the type of the jammer.

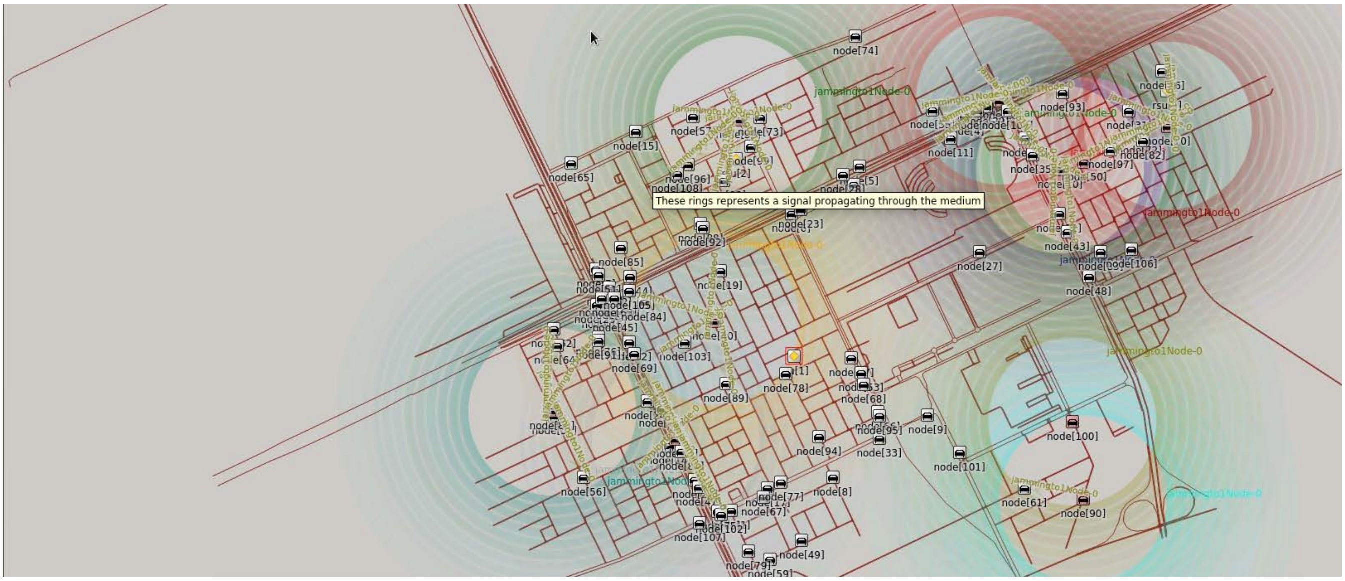
Moreover, the proposed localization algorithm was examined against high-traffic network topology. In total, 600 cars and 3 RSUs were run throughout the simulation time. Regarding the number of jammers, the proposed approach applied a different number of jammers. Table 4 summarizes the experiments' results in the case of a dense network.

Due to the absence of any other similar method (in terms of assumptions, parameters, simulation environments, etc..), we decided to compare our solution against a baseline approach. This approach takes only the average of Xs and Ys of the jammed cars' locations (as shown in equation 10) to predict the location of the jammer.

$$Jammer_X = \frac{\sum_{i=1}^{N} x_i}{N}, \; Jammer_Y = \frac{\sum_{i=1}^{N} y_i}{N} \qquad (10)$$

Additionally, the accuracy of the proposed method and the baseline approach was calculated and compared by taking the mean of the Euclidean Distance values (Prediction and Baseline values in Table 4). To test all inputs of the jammed cars, a rolling window was used with 2 and 3 sizes to test the possibility of having only two or three affected cars (as a minimum). The more affected vehicles, the more accurate location will be calculated. As shown in the table, the accuracy of the proposed method is much better than the baseline approach. Another advantage over the baseline approach is that if we consider all the predicted points and increase the weights of corresponding roads, we might avoid the same road of the jammer with high accuracy. This accuracy reached a 0.6m difference from the exact location of the jammer, as shown in the Prediction (Closest) columns in Table 4. Many experiments were conducted with random car destinations and movements. Then, an average was taken to produce the results in table 4. Figures 10 and 9 show the simulation environment in a high-traffic infrastructure.

Therefore, it can be concluded from the given results that the proposed approach successfully detected and localized the VANET's jammers with high accuracy, avoided the jamming areas, and considered other routes to preserve the network services and ensure correct delivery of the data and less signal interference. Moreover, the proposed methodology handled cases where the jammer is on the same road as the



These rings represents a signal propagating through the medium

FIGURE 10. Screenshots which show the simulation environment in a high-traffic infrastructure.

affected car(s). This proves the proposed approach's efficiency in containing the threatening DoS attacks in VANET with fewer assumptions, requirements, and complexity.

VI. CONCLUSION

The importance of VANET's applications in smart cities has attracted different types of security attackers, including jammers. Jammers interfere with media signals to prevent normal users from communicating. They send too many fake signals simultaneously to overhead the network resources and harm their running services. Many existing studies tried to detect jammers using different approaches, communication types, simulation environments, and evaluation metrics. Limited studies have focused on detecting the jammer and localizing it. Even in these limited studies, the authors did not mention their minimum requirement to localize the attacker, their assumptions, whether the network is mobile, or the complexity of their approaches. Finally, none of these studies suggest any countermeasure technique for the presence of the jammer in the area.

Therefore, this research (a) implemented different types of VANET jammers, (b) detected and localized jammers with high accuracy (c) avoided the jammers by informing the normal cars of their positions and increasing the cost of the routes the jammers are using to direct the cars to other routes and avoid the jamming area.

The proposed approach functionalities were implemented using specialized simulation environments, including OMNeT++, Veins, and SUMO. The performance of the proposed approach was evaluated by examining the accuracy of the estimated location of the jammer. Moreover, the avoidance strategy was implemented to test the impact of the jammer before and after re-routing the cars to avoid the roads the jammer is using. The evaluation results have proven the efficiency of the proposed approach in successfully locating and avoiding the jammer, especially for the cars that still have the opportunity to bypass the jammed routes with minimum requirements. For example, the proposed approach was able to locate the jammer only if two receivers (cars) received the jamming packets. The more receivers, the easier and the more accurate jammers' locations will be calculated.

Euclidean distance was used as a metric to measure the accuracy of the estimated position of the jammer. Moreover, PDR and SINR have measured the impact of the jammers on the network services and how the proposed avoidance approach contained the DoS jamming attack and protected the network resources and data.

Moreover, the proposed method has many strengths over other related works, such as offering an efficient approach that detects many jammers simultaneously in different locations. Moreover, it can localize the jammers in real-time with low computing costs. It has a very high accuracy that reached a 0.6m difference from the exact location of a jammer. Therefore, accurate localization of the jammers succeeded in avoiding them and preserving the packet delivery ratio. However, some weaknesses can be improved, such as considering a wider variety of assumptions and parameters (e.g., intelligent jammers) and testing the proposed work in a real environment.

For future work, the proposed system could consider different assumptions and parameters, including the diversity in the vehicles' heights, smarter jammers, mobility models, and transmitting power strength. Moreover, the proposed system will be tested on real testbeds. Also, other Denial of Service (DoS) attacks will be implemented, and their impact will be studied. Consequently, suggested security solutions will be offered and examined. Also, the proposed system will be optimized into a system for predicting the future position of the jammer, so that we can design a proactive rerouting scheme for the jammed cars.

ACKNOWLEDGMENT

The authors would like to thank the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication. Moreover, this research was done during the author Iman Almomani's sabbatical year 2021/2022 from the University of Jordan, Amman, Jordan.

REFERENCES

[1] Y. Wiseman, "Autonomous vehicles," in *Research Anthology on Cross-Disciplinary Designs and Applications of Automation*. Hershey, PA, USA: IGI Global, 2022, pp. 878–889.

[2] T. H. Aldhyani and H. Alkahtani, "Attacks to automatous vehicles: A deep learning algorithm for cybersecurity," *Sensors*, vol. 22, no. 1, p. 360, 2022.

[3] S. D. Farashah and N. Rahbar, "Privacy implication in autonomous vehicles: A comparative study of threats and legal requirements," *Law Quarterly*, vol. 51, no. 4, pp. 695–715, 2022.

[4] V. K. Kukkala, S. V. Thiruloga, and S. Pasricha, "Roadmap for cybersecurity in autonomous vehicles," *IEEE Consum. Electron. Mag.*, vol. 11, no. 6, pp. 13–23, Nov. 2022.

[5] I. Almomani and M. Alenezi, "Efficient denial of service attacks detection in wireless sensor networks," *J. Inf. Sci. Eng.*, vol. 34, no. 4, pp. 977–1000, 2018.

[6] N. Aung, W. Zhang, S. Dhelim, and Y. Ai, "Accident prediction system based on hidden Markov model for vehicular ad-hoc network in urban environments," *Information*, vol. 9, no. 12, p. 311, Dec. 2018. [Online]. Available: https://www.mdpi.com/2078-2489/9/12/311

[7] G. Bathla, K. Bhadane, R. K. Singh, R. Kumar, R. Aluvalu, R. Krishnamurthi, A. Kumar, R. N. Thakur, and S. Basheer, "Autonomous vehicles and intelligent automation: Applications, challenges, and opportunities," *Mobile Inf. Syst.*, vol. 2022, pp. 1–36, Jun. 2022.

[8] H. Pirayesh and H. Zeng, "Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 767–809, 2nd Quart., 2022.

[9] R. Valiente, L. Montero, C. Ballesteros, and L. Jofre, "Directional analysis of jamming attack for connected vehicular platoons," in *Proc. 16th Eur. Conf. Antennas Propag. (EuCAP)*, Mar. 2022, pp. 1–5.

[10] G. Dubosarskii and S. Primak, "Jamming and anti-jamming strategies of mobile vehicles," *Electronics*, vol. 10, no. 22, p. 2772, Nov. 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/22/2772

[11] V.-L. Dao, L.-N. Hoang, S. Girs, and E. Uhlemann, "Defeating jamming using outage performance aware joint power allocation and access point placement in uplink pairwise NOMA," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1957–1979, 2021.

[12] B. Kihei, H. Wilson, and M. Fall, "Experimental results of detecting primitive jamming attacks using machine learning in vehicle-to-everything communication networks," in *Proc. IEEE 7th World Forum Internet Things (WF-IoT)*, Jun. 2021, pp. 530–535.

[13] H. Bangui, M. Ge, B. Buhnova, and L. H. Trang, "Towards faster big data analytics for anti-jamming applications in vehicular ad-hoc network," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 10, p. e4280, Oct. 2021.

[14] D. Kosmanos, A. Argyriou, and L. Maglaras, "Estimating the relative speed of RF jammers in VANETs," *Secur. Commun. Netw.*, vol. 2019, pp. 1–18, Nov. 2019.

[15] D. Kosmanos, A. Pappas, L. Maglaras, S. Moschoyiannis, F. J. Aparicio-Navarro, A. Argyriou, and H. Janicke, "A novel intrusion detection system against spoofing attacks in connected electric vehicles," *Array*, vol. 5, Mar. 2020, Art. no. 100013.

[16] G. Kasturi, A. Jain, and J. Singh, "Detection and classification of radio frequency jamming attacks using machine learning," *J. Wireless Mob. Netw. Ubiquitous Comput. Dependable Appl.*, vol. 11, no. 4, pp. 49–62, 2020.

[17] D. Kosmanos, D. Karagiannis, A. Argyriou, S. Lalis, and L. Maglaras, "RF jamming classification using relative speed estimation in vehicular wireless networks," *Secur. Commun. Netw.*, vol. 2021, pp. 1–16, Aug. 2021.

[18] K. Davaslioglu, S. Soltani, T. Erpek, and Y. E. Sagduyu, "DeepWiFi: Cognitive WiFi with deep learning," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 429–444, Feb. 2021.

[19] H. Pirayesh, P. K. Sangdeh, and H. Zeng, "Securing ZigBee communications against constant jamming attack using neural network," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4957–4968, Mar. 2021.

[20] K. Pelechrinis, C. Koufogiannakis, and S. V. Krishnamurthy, "Gaming the jammer: Is frequency hopping effective?" in *Proc. 7th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wireless Netw.*, Jun. 2009, pp. 1–10.

[21] B. Gopalakrishnan and M. A. Bhagyaveni, "Anti-jamming communication for body area network using chaotic frequency hopping," *Healthcare Technol. Lett.*, vol. 4, no. 6, pp. 233–237, Dec. 2017.

[22] L. Rajesh, K. B. Bagan, P. T. Sankar, and V. Suchitra, "Performance analysis of UAV relay in VANETs against smart jamming with Q-learning techniques," in *Emerging Technologies in Data Mining and Information Security*. Berlin, Germany: Springer, 2021, pp. 771–783.

[23] H. Pirayesh, P. K. Sangdeh, S. Zhang, Q. Yan, and H. Zeng, "JammingBird: Jamming-resilient communications for vehicular ad hoc networks," in *Proc. 18th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jul. 2021, pp. 1–9.

[24] G. B. Santhi and D. Sheela, "Reliability refinement in VANET with hybrid jamming attacks using novel index based voting algorithm," *Peer-Peer Netw. Appl.*, vol. 13, no. 6, pp. 2145–2154, Nov. 2020.

[25] T. Zhang, X. Ji, Z. Zhuang, and W. Xu, "JamCatcher: A mobile jammer localization scheme for advanced metering infrastructure in smart grid," *Sensors*, vol. 19, no. 4, p. 909, Feb. 2019.

[26] F. B. Gunay, E. Özturk, T. Çavdar, Y. S. Hanay, and A. U. R. Khan, "Vehicular ad hoc network (VANET) localization techniques: A survey," *Arch. Comput. Methods Eng.*, vol. 28, no. 4, pp. 3001–3033, Jun. 2021.

[27] J. Liu and G. Guo, "Vehicle localization during GPS outages with extended Kalman filter and deep learning," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.

[28] N. Piperigkos, A. S. Lalos, and K. Berberidis, "Multi-modal cooperative awareness of connected and automated vehicles in smart cities," in *Proc. IEEE Int. Conf. Smart Internet Things (SmartIoT)*, Aug. 2021, pp. 377–382.

[29] D. Biswas, S. Barai, and B. Sau, "Improved RSSI based vehicle localization using base station," in *Proc. Int. Conf. Innov. Trends Inf. Technol. (ICITIIT)*, Feb. 2021, pp. 1–6.

[30] M. A. G. Al-Sadoon, M. N. Patwary, M. A. Rahman, and R. A. Abd-Alhameed, "Efficient small angle-of-arrival array sensor for intelligent localisation and tracking systems," in *Proc. IEEE 4th 5G World Forum (5GWF)*, Oct. 2021, pp. 475–480.

[31] R. Halili, N. BniLam, M. Yusuf, E. Tanghe, W. Joseph, M. Weyn, and R. Berkvens, "Vehicle localization using Doppler shift and time of arrival measurements in a tunnel environment," *Sensors*, vol. 22, no. 3, p. 847, Jan. 2022.

[32] S. Wang and C. Chu, "Geometry-covering jammer localization based on distance comprehension in wireless sensor networks," 2015, *arXiv:1512.06468*.

[33] W. Aldosari and M. Zohdy, "Tracking a jammer in wireless sensor networks and selecting boundary nodes by estimating signal-to-noise ratios and using an extended Kalman filter," *J. Sensor Actuator Netw.*, vol. 7, no. 4, p. 48, Nov. 2018.

[34] B. Duan, D. Yin, Y. Cong, H. Zhou, X. Xiang, and L. Shen, "Anti-jamming path planning for unmanned aerial vehicles with imperfect jammer information," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2018, pp. 729–735.

[35] M. Abdollahi, K. Malekinasab, W. Tu, and M. Bag-Mohammadi, "An efficient metric for physical-layer jammer detection in Internet of Things networks," in *Proc. IEEE 46th Conf. Local Comput. Netw. (LCN)*, Oct. 2021, pp. 209–216.

[36] J. Fan, T. Liang, T. Wang, and J. Liu, "Identification and localization of the jammer in wireless sensor networks," *Comput. J.*, vol. 62, no. 10, pp. 1515–1527, Sep. 2019.

[37] I. K. Azogu, M. T. Ferreira, J. A. Larcom, and H. Liu, "A new anti-jamming strategy for VANET metrics-directed security defense," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2013, pp. 1344–1349.

[38] Y. S. Kim, F. Mokaya, E. Chen, and P. Tague, "All your jammers belong to us—Localization of wireless sensors under jamming attack," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 949–954.

IMAN ALMOMANI (Senior Member, IEEE) received the bachelor's and master's degrees in computer science from United Arab Emirates and Jordan, in 2000 and 2002, respectively, and the Ph.D. degree in wireless network security from De Montfort University, U.K., in 2007. She is currently an Associate Professor in cybersecurity. She is also the Associate Director of the Research and Initiatives Centre (RIC), the Associate Director of the Innovation Center (IC), and also the Leader of the Security Engineering Laboratory (SEL), Prince Sultan University (PSU), Riyadh, Saudi Arabia. Before joining PSU, she worked as an Associate Professor and the Head of the Computer Science Department, The University of Jordan, Jordan. Her research interests include wireless networks and security, mainly wireless mobile ad hoc networks (WMANETs), wireless sensor networks (WSNs), multimedia networking (VoIP), and security issues in wireless networks. She is also interested in the area of electronic learning (e-learning) and mobile learning (m-learning). She has several publications in the above areas in a number of reputable international and local journals and conferences. She is in the organizing and technical committees for a number of local and international conferences. Also, she serves as a reviewer and a member of the editorial board in a number of international journals. She is also a Senior Member of IEEE WIE.

MOHANNED AHMED received the bachelor's degree in software engineering from Prince Sultan University, Riyadh, Saudi Arabia, in 2020. He is currently a Research Engineer at the Security Engineering Laboratory (SEL), Prince Sultan University. He has published papers in the mentioned areas of interests in well-known ISI/Scopus-indexed journals and international conferences. His research interests include VANET, the IoT, software security, cybersecurity maturity models, malware analysis, and data science.

DIMITRIOS KOSMANOS received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Thessaly, Greece, in 2020. He is currently working as an Adjunct Lecturer with the University of Thessaly. He was collaborating as a Fellow Researcher with the CERTH Department of Volos and a Research Assistant with De Montfort University, U.K. He has authored or coauthored technical papers in international conferences and ISI/Scopus-indexed journals. His current research interests include techniques the cross-layer detection and suppressing denial of service attacks in vehicular ad hoc networks, physical layer error detection, and correction techniques. He has been a TCP member in international conferences.

AALA ALKHAYER received the Bachelor of Engineering degree in information technology engineering from SVU University, Damascus, in 2017, and the bachelor's degree in software engineering from Prince Sultan University (PSU), Riyadh, Saudi Arabia, in 2018. She is currently a Research Engineer at the Security Engineering Laboratory (SEL), PSU. Her research interests include software engineering, networks security, malware analysis, multimedia networking, and computer vision.

LEANDROS MAGLARAS (Senior Member, IEEE) is currently a Professor in cybersecurity with the School of Computing, Edinburgh Napier University, and a Research Fellow at the Security Engineering Laboratory (SEL), Prince Sultan University (PSU), Riyadh, Saudi Arabia. From September 2017 to November 2019, he was the Director of the National Cyber Security Authority of Greece. He is featured in Stanford University's list of the world's Top 2% scientists. He is the author of more than 200 papers in scientific magazines and conferences.