



OPEN A reliable score-based routing protocol using a fog-assisted intrusion detection system in vehicular ad-hoc networks

Samira Tahajomi Banafshehvaragh¹, Mani Zarei^{2✉} & Amir Masoud Rahmani³

One of the most significant challenges of vehicular ad-hoc networks (VANETs) is establishing reliable connections with the network infrastructure despite the participation concern of attacking vehicles in routing. Vehicles need a proper defense mechanism against all types of attacks, even if a reliable path-planning strategy accompanies them. This paper proposes a reliable score-based routing protocol using a fog-assisted intrusion detection system (RSR-IDS) in VANETs. First, RSR-IDS pre-processes data using minimum–maximum normalization and Pearson's correlation coefficient. The IDS is trained using three machine learning-based algorithms and a voting technique to reduce false detection. These algorithms include the decision tree, random forest, and extra trees. Deploying the IDS in the fog server solves the data diversity problem in the classifier training. Therefore, RSR-IDS detects abnormal data accurately to calculate the untrust score (US). Then, RSR-IDS selects a route with the lowest total USs and hop count compared to others for communications. RSR-IDS is evaluated based on the accuracy, F1-score, false negative rate, packet delivery ratio (PDR), packet loss ratio, end-to-end delay, and throughput criteria using OMNeT++ and the UNSW-NB15 dataset. The significant improvements in RSR-IDS include 14.1% in accuracy, 11.4% in F1-score, and 5.4% in PDR regarding various vehicle densities.

Keywords Fog server, Intrusion detection system, Routing protocol, Untrust score, Vehicular ad-hoc networks

Protecting inter-vehicle communication (IVC) from cyber threats is controversial due to its design without security requirements¹. Vehicles use vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications to manage data traffic in vehicular ad-hoc networks (VANETs)². Data propagation through unreliable V2V and V2I communications creates security challenges. These challenges may lead to the loss of precise and up-to-date information about road conditions and urban traffic³. According to statistical reports, 40 million people die and 1.35 million people are injured each year due to vehicle accidents, respectively⁴. The World Health Organization report also shows that 93% of road accident fatalities occur in low-income countries with unsafe intelligent transportation systems⁵. The data loss and delay in exchanging information endanger drivers' road safety. Informing about collisions or accidents 1–2 s earlier can prevent 60% of drivers' accidents⁵. Therefore, expanding vehicles and their communications increases cyber risks in VANETs⁶. Hackers threaten the cybersecurity of vehicles through various attacks such as denial of service (DoS), Sybil attacks, distributed DoS (DDoS), etc.⁷. These attacks affect the availability, integrity, and trust of data in VANET communications⁸. DoS, DDoS, and Black Hole attacks decrease the availability of VANET services by making the channel busy and sending/receiving data by the attacker vehicle, while message tampering attacks destroy integrity⁸. For example, hackers remotely controlled 1.4 million vehicles by infiltrating a Jeep Cherokee⁹. Also, attackers hijacked a Tesla vehicle by targeting its primary functions and components, like the steering wheel and brakes¹⁰. Despite these risks, vehicles need a proper defense mechanism to detect abnormal data traffic. End-to-end encryption/decryption is a strong security method to protect the confidentiality and integrity of sensitive data, such as identifiable information and location details of vehicles¹¹. However, generating cryptographic keys by third parties, exchanging them, and decrypting messages are threat points that may lead to key leakage, cyberattacks, and misuse¹¹. Encryption and decryption require high computational power, storage, and

¹Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.

²Department of Computer Engineering, Shahr-E-Qods Branch, Islamic Azad University, Tehran, Iran. ³Future Technology Research Center, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan. ✉email: m.zarei@srbiau.ac.ir

energy consumption, which vehicles don't have¹¹. Therefore, encryption/decryption algorithms may not be suitable security approaches due to a lack of real-time responses, high latency, and delay¹¹. Finally, intrusion detection systems (IDSs) are presented as defense mechanisms of vehicles with different performances. One of the challenges in current intrusion detection systems is scalability and mobility, especially in VANETs with many vehicles¹². The inability to cover the mobility of many high-speed vehicles is a fundamental problem in detection systems, leading to link breakages and delays in communication between vehicles¹³. Traditional IDSs developed on onboard units (OBUs) have limited problems with storage capacity, transmission range, computing, and power supply¹². Sharing information about these IDSs leads to delays and congestion in VANET connections¹². Implementing intra-vehicle IDSs in OBUs also increases the cost of VANETs. In VANET urban scenarios with high vehicle density, it is necessary to deploy a trained IDS in the fog server. The deployed IDS in the fog server, unlike the OBU's detection system, should be able to identify and manage a high volume of data. It's essential to handle processing, storage resources, networking power, and computation of vehicles/RSUs in the fog-assisted server⁵. Therefore, a critical challenge of the fog server is efficiently sharing resources between vehicles¹⁴. Fog-assisted servers are also in danger in terms of security, data, virtualization, and monitoring due to the distributed sharing of fog computing¹⁴. Distributed fog computing is suitable for utilizing time-sensitive or real-time services of VANETs due to installing its nodes at the network edge¹⁵. Despite a shorter response time than cloud computing, fog computing of VANETs still has several time issues, like delay and latency¹⁵. Some researchers solved the problem of delay and latency and reduced them using a fog computing model based on priority in urban environments¹⁵. Since such a strategy cannot detect new attacks or all of them, vehicles need an efficient IDS mechanism to detect all known/unknown attacks. Most IDSs only detect known or one special type of attack. The overfitting problem also exists in most machine learning (ML)-based IDSs. However, the lack of a suitable IDS is not the only concern of researchers in vehicular security. Many routing protocols have been proposed to find suitable routes. Physical obstacles in the urban communication environment reduce the probability of achieving a clear line of sight between the source and the destination in VANETs³. Solving this problem requires providing a suitable multi-hop protocol using roadside units (RSUs) and the fog server, considering the urban environment's challenges for message routing. The presented routing¹⁶ is a stable segment-aware protocol to solve the high vehicular mobility in VANETs. The proposed segment-aware protocol may select far-away segments with high delivery delays. Due to the non-confidential nature of information in message routing, it's also necessary to find a trusted route in VANETs³. Choosing a path with an acceptable delay and high reliability to establish communication is one of the important components of the routing protocol in VANETs¹⁷. However, in some protocols^{18–21}, the joint route reliability and the hop count (H_c) are not considered jointly in selecting the best path. The presented protocol²² is a global positioning system (GPS)-based routing that considers reliability and the minimum H_c in selecting a communication path. The proposed GPS-based protocol is also not suitable in real scenarios due to the assumption of real-time awareness of the destination GPS from the source. Therefore, vehicles need an efficient routing protocol to support urban challenges in selecting a short and reliable route for V2V and V2I communications. Regardless of reliability, an attacking vehicle can penetrate V2V/V2I communications even with a suitable IDS^{23,24}. Attacking vehicles may send false messages to VANETs, which cause accidents. Attacking vehicles may also modify or delete messages from the VANET. Participation of attacking vehicles in routing leads to improper traffic management, security risks, and even irreparable financial and life damage on highways²⁵. Despite these risks, it is necessary to provide a new score-based protocol to increase the security of vehicles and routing reliability in VANETs.

This paper presents a reliable score-based routing protocol using a fog-assisted intrusion detection system (RSR-IDS) in VANETs. RSR-IDS deploys an IDS as a scoring mechanism based on received data in the fog server. RSR-IDS identifies abnormal data and calculates the untrust score (US) through this scoring mechanism. The calculated scores are available through communications between the fog server, RSUs, and vehicles during routing. RSR-IDS selects a path using USs and H_c for VANET communications. The main contributions and innovations of this article are as follows:

- First, an IDS-based scoring mechanism is deployed in the fog server layer. Establishing the detection system of RSR-IDS in the fog server leads to data diversity in the training phase. The proposed fog-assisted detection strategy in RSR-IDS shows abnormal data detection with great accuracy.
- The classifier of this IDS is trained using three ML algorithms. These algorithms include the decision tree, random forest, and extra trees. They categorize data of each feature using the linear regression algorithm. The final detection results are determined using the voting technique. Therefore, using the voting technique is expected to reduce the false detection of data in RSR-IDS.
- After detecting data, a US is considered based on the normality/abnormality of data traffic. Calculated USs are available through communications between the fog server, RSUs, and vehicles during routing.
- In message routing, calculated USs and H_c are considered for each route. RSR-IDS selects a short and reliable route for communications in VANETs. According to the results, RSR-IDS enhances reliability by combining the IDS and the routing protocol in VANETs.
- RSR-IDS suggests a data traffic analysis model by deploying an IDS on the fog server. The classifier of RSR-IDS greatly reduces overfitting using the combination of the decision tree, random forest, and extra trees algorithms. According to the results, RSR-IDS detects abnormal data efficiency.
- RSR-IDS is provided to detect abnormal data and reliable routing. It is expected that RSR-IDS will increase vehicle safety in VANETs. Therefore, this paper evaluates RSR-IDS using appropriate criteria in the OM-NeT++ simulator. These criteria include accuracy, F1-score, false negative rate (FNR), packet delivery ratio (PDR), packet loss ratio (PLR), end-to-end (E2E) delay, and throughput.

The remaining sections are organized as follows: Section “Related work” presents related work. Section “System model” presents the system model of RSR-IDS. Section “Proposed model” presents RSR-IDS in this paper. Section “Performance evaluation” simulates and evaluates RSR-IDS. Section “Discussion” discusses RSR-IDS, and Section “Conclusion” concludes.

Related work

Automotive communications are prone to many data security challenges in the VANET ecosystem. These challenges are inherently affected by vehicular path planning, environmental security threats, communication protocols, network scalability, data handling strategies, etc.²⁶. It is necessary to provide delay-tolerant routing protocols that outperform security concerns, even if the network suffers from weak communications. Automotive routing protocols have recently been planned to handle secure data propagation over unsecured networks. Due to the time-variant vehicular mobility models and the variety of propagated data types in VANETs, identifying hackers’ attack strategies is quite challenging²⁶. The wireless communication nature of VANETs always makes these networks susceptible to security threats. Hackers always threaten vehicle security with various new attacks in VANETs. Considering the financial and life damages of these attacks for consumers, appropriate defense mechanisms for these risks are always necessary. So far, researchers have proposed various IDS techniques to detect cyberattacks in VANETs^{23,24,27}. Due to the participation of attacking vehicles in the routing messaging, it is necessary to have a reliable routing protocol. The scalability of secured-routing protocols for real-time responses to highly dynamic and time-variant vehicular requirements is another security challenge of VANETs²⁶. Researchers have proposed various routing protocols in VANETs^{18–21}. This section reviews previously published research addressing the IDS models and IDS-aware routing protocol solutions.

IDS

IDSs play an essential role in improving vehicular security by detecting abnormal data. Researchers present various IDS models with different efficiencies. This subsection examines some IDS models in VANETs.

Bengui et al.²³ presented a two-phase IDS to increase accuracy and reduce false alarms. They mentioned a challenge that limited resources and data lead to novel attacks not being detected in ML models. They were also going to decrease detection time by lowering the computing process and deploying a lightweight IDS. First, the classifier of this IDS was trained by the random forest algorithm and attack signatures. The random forest algorithm aimed to process a large amount of data, decrease computational time, and detect well-known attacks. After classifying, the IDS planned to generate an alert message when facing an attack. Otherwise, the trained classifier reclassified normal data with a clustering algorithm based on coresets. The aim of using the coresets cluster-based algorithm is to detect unknown attacks and decrease the false positive rates. After reclassifying, an alert message was generated if an attack took place by received data. The proposed model showed better accuracy and efficiency than classical ML algorithms. According to the results, the suggested IDS reduced processing time while maintaining the quality of big data analysis. It is reasonable that the computation time of the proposed IDS was higher than the random forest and coresets algorithms. The authors intended to reduce the false alarm using the proposed IDS, but didn’t statistically examine FNR.

A hybrid IDS model was presented²⁴ to detect known/unknown attacks on IVCs and intra-vehicle communications. Traditional ML algorithms preprocessed data and selected essential features to increase data quality for accurate detection. The signature-based IDS was trained using four supervised ML algorithms, including the decision tree, random forest, extreme gradient boosting (XGBoost), and extra trees. These ML algorithms detected known attacks on IVCs and intra-vehicle communications. Bayesian optimization with the tree Parzen estimator method was also used to optimize supervised ML algorithms. The anomaly-based IDS was also trained using an unsupervised k-means clustering algorithm and biased classifiers to detect zero-day attacks. A Bayesian optimization with the Gaussian process (BO-GP) method was used to optimize unsupervised ML algorithms. The advantages of the proposed model include efficiency in handling high volumes of data and the ability to run on vehicles in real time. Moreover, using optimization algorithms and the low computational-time complexity are other advantages. The low processing time led to the real-time implementation of the proposed model in modern vehicles. The proposed IDS can’t effectively detect most fuzzy attacks on intra-vehicle networks. The proposed model is also less effective for detecting unknown attacks than designed mechanisms for a type of attack, especially in external vehicle networks.

Alsarhan et al.²⁷ presented a support vector machine (SVM)-based IDS in VANETs. They mentioned that the SVM algorithm can manage a small sample or extensive data. Using SVM also leads to irrelevance between the algorithm’s complexity and the sample’s dimension. A penalty function was used in the proposed IDS to regularize and control the complexity of the classification. The IDS was optimized using the genetic algorithm, particle swarm optimization (PSO), and ant colony optimization algorithms. The utilized algorithms solved the optimum local problem. The suggested IDS showed the best performance when using the genetic algorithm. Several issues, such as the connection of algorithm complexity and sample dimension, classifying a large amount of data, and managing a small sample, were solved in this system. It seems that the proposed IDS has a high mathematical complexity.

Karthiga et al.²⁸ presented an Intelligent hybrid IDS based on ML and deep learning algorithms to detect known/unknown attacks in VANETs. Due to the large size of VANET’s traffic, they preprocessed data to remove duplicate items. In the training phase, patterns were obtained from the pre-processed traffic dataset using the signature module and the adaptive neuro-fuzzy inference system (ANFIS) classifier. Known IDS (KIDS) and unknown IDS (UIDS) modules used these patterns to detect attacks. In the testing phase, real-time traffic data were pre-processed to remove redundant data. Then, known and unknown attacks were detected by ANFIS-based KIDS and convolutional neural networks (CNN)-based UIDS modules, respectively. The modified LeNet, as a simple architecture of conventional deep learning, was used to detect the type of unknown attacks. This IDS

accurately detected external attacks such as Botnet, PortScan, DoS, and Brute Force in VANETs. The proposed IDS had a lower detection rate for brute-force attacks than others on the i-VANET dataset. The presented IDS also had more detection time for Botnet attacks in VANETs. One of the advantages of their proposed model is detecting unknown attacks using soft computing techniques. The suggested IDS improves the detection rate compared to other works.

Amaouche et al.²⁹ presented an enhanced IDS based on the random forest method to improve precision and accuracy in VANETs. They believed that the high speed of vehicles is an important issue, leading to unstable communications and the complexity of security measures in VANETs. First, non-numeric data were converted into numerical values using the one-hot encoding technique in the proposed model. The synthetic minority over-sampling technique (SMOTE) was also used to solve the class imbalance and overfitting problems in the data pre-processing step. In the feature selection step, the mutual information filtering method was used to optimize the data quality and decrease resource consumption. The minimum–maximum normalization (MinMaxScaler) method also normalized feature values. In the classification step, the trained classifier identified data using the random forest algorithm and 15 features. Their suggested model achieved high accuracy, recall, precision, and F1-score compared to others. One of the advantages of this model is the reduction of resource consumption. The proposed IDS is limited to the fast mobility of vehicles in VANETs.

Arya et al.³⁰ presented an IDS based on ensemble federated learning and CNN while saving time and resources in VANETs. Vehicles were local devices in the proposed IDS. They mentioned that the high mobility of vehicles and the lack of security approaches are important challenges in smart cities. In the first step, heterogeneous data were pre-processed for individual clients to solve the class imbalance. Then, the classifier of client models was trained using a 1D CNN as an information extractor and heterogeneous local data. The goal of using CNN was to select assistant features in spatial data. The trained classifier of local models retained data and shared it upon request to reduce communication overhead with neighbors. The trained heterogeneous client classifiers were ensemble into a centralized federated CNN model in the third step. The weights of trained models were aggregated to update the centralized model weight. Due to the high communication cost and data heterogeneity problems of traditional federated learning, the proposed method used the PSO algorithm to optimize the weight of local models. The centralized model weight was tuned and optimized using a nature-inspired PSO algorithm. The detection accuracy improved using PSO optimization and the ensemble technique. Abnormal data were detected by the federated ensemble CNN model. Using federated learning aimed to protect the privacy of local data in edge devices. The proposed model also generated an alarm if it identified a problem or harmful data in VANETs. The advantages of the proposed system are saving time and resources, solving the overfitting problem, and managing a large volume of data in the urban environment. The presented IDS had a higher false positive rate (FPR) than the K-nearest neighbor (K-NN) algorithm.

Ajjaj et al.³¹ presented a multivariate statistical detection scheme (MVSDS) using real-time traffic monitoring to identify routing attacks like the black hole in VANETs. They mentioned that the restriction of delay in the delivery of messages is a challenge for VANETs. In the first step, real-time traffic data were monitored based on throughput, dropped packets ratio, and overhead traffic ratio metrics. The collected data were normalized using z-score, MinMaxScaler, and decimal scaling techniques in the rescaling step. The obtained data were modeled as detection input by a multivariate dataset sampled at different times. Multivariable datasets were transformed into univariate samples using the Rao–Ali test for each time window. Ryan–Joiner test calculated the correlation coefficient for the obtained univariate samples. In the attack detection step, abnormal behavior was detected if the calculated correlation coefficient was less than the threshold value. Indeed, the proposed model detected normal behavior from abnormal by confirming multiple network characteristics with data samples. A notification was generated when an abnormal behavior was detected. Based on the results, the proposed scheme performed better using the MinMaxScaler technique. The proposed scheme could be applied without changes to routing protocols like ad-hoc on-demand distance vector (AODV) in VANETs. The presented scheme was only designed for routing attacks like the black hole.

The studied IDSs have different strengths and weaknesses in identifying abnormal data. Table 1 summarizes studied IDSs regarding challenges, innovations, advantages, disadvantages, and future works. Fog computing-based methods were presented for VANETs using encryption methodology to identify malicious nodes, but had high processing delay and overhead^{32,33}. However, to solve network overhead, a customized fog-based IDS was presented for a specific type of attack³³. Some IDSs²³ were also designed to reduce the false alarm, but were not studied statistically. The FNR of RSR-IDS is studied statistically in section “Performance evaluation” to show how to solve resource constraints by deploying the proposed IDS in the fog server. Some IDSs^{24,28} cannot effectively detect one type of attack. Most IDSs can detect known attacks or a special type in VANETs³¹. Some IDSs can detect both known/unknown attacks. The proposed RSR-IDS is presented to identify any kind of known/unknown abnormal data. Unlike some other IDSs²⁹, the proposed RSR-IDS covers the mobility of vehicles in urban environments and solves the overfitting of ML algorithms in some classifiers. RSR-IDS, like most detection systems, can also be efficient with a high volume of data. However, the studied IDSs try to maintain high accuracy in detecting attacks, but none consider routing reliability in VANET communications. To resolve this challenge, RSR-IDS is proposed to enhance the reliability of communications using a fog-assisted IDS and a US-based routing protocol.

Routing protocol

Routing protocols are proposed to select suitable message routes in VANETs. This subsection presents several routing protocols in the Internet of Vehicles (IoV) and VANETs.

Sharma et al.²¹ presented an extended on-demand multicast routing protocol (ODMRP) to control the communication in VANETs. They referred to rapid changes, huge scale, and link disconnections as VANET’s challenges. This protocol was secured using a security algorithm based on a cryptographic method. This

Ref.	Challenge	Innovation	Advantage	Disadvantage	Future work
23	Detecting known/unknown attacks with high accuracy	Random forest and coresets-based clustering algorithms	<ul style="list-style-type: none"> ✓ Higher accuracy than classical ML ✓ Reducing the processing time while maintaining the quality of big data analysis 	<ul style="list-style-type: none"> ✓ Higher computation time than basic algorithms ✓ The Lack of statistically examining FNR 	-
24	Detecting known/unknown attacks on IVCs	Signature/anomaly-based IDSs using ML and the k-means clustering	<ul style="list-style-type: none"> ✓ Efficiency in the high volume of data ✓ Running on vehicles in real-time ✓ Using optimization algorithms ✓ Low computational-time complexity 	<ul style="list-style-type: none"> ✓ Failure to detect fuzzy attacks effectively ✓ Less efficient in detecting unknown attacks than IDSs designed for one attack 	Using unsupervised and online learning methods
27	Detecting types of attacks while solving algorithm complexity and sample dimension problems	SVM-based IDS using optimization algorithms in VANETs	<ul style="list-style-type: none"> ✓ Classifying a large amount of data ✓ Managing a small sample ✓ Solving optimum local problems ✓ Controlling the complexity of the classification using a penalty function 	High mathematical complexity	Using deep learning and real big datasets for training SVM classification
28	Detecting known/unknown attacks such as Botnet, PortScan, DoS, and Brute Force	ANFIS-based KIDS and CNN-based UIDS modules	<ul style="list-style-type: none"> ✓ Improving the detection rate ✓ Detecting unknown attacks using soft computing techniques 	<ul style="list-style-type: none"> ✓ High detection time for Botnet attacks ✓ Lower detection rate for Brute Force attacks than others 	<ul style="list-style-type: none"> ✓ Using Deep Learning to Improve Security ✓ Optimizing performance using the proposed methods
29	Detecting various attacks by selecting features and handling class imbalance	Random forest and pre-processing methods	<ul style="list-style-type: none"> ✓ Reduction of resource consumption ✓ Higher performance than others 	Limitations in the fast mobility of vehicles	Developing adaptive IDSs to solve the problem of fast mobility
30	Detecting intrusion while saving time and resources and reducing communication overhead	Centralized IDS containing weighted local models using ensemble federated learning and CNN	<ul style="list-style-type: none"> ✓ Improving accuracy using PSO ✓ Solving the overfitting problem ✓ Managing a large volume of data 	Higher FPR than the K-NN algorithm	<ul style="list-style-type: none"> ✓ Deploying a model with higher performance in Apache Spark and Kafka ✓ Using nature-inspired optimization methods to reduce dimensionality
31	Detecting routing attacks like the black hole	Proposing MVSDS using real-time traffic monitoring and pre-processing methods	Applying without changes to the routing protocols	Lack of statistical analysis of the proposed scheme with other detection models	<ul style="list-style-type: none"> ✓ Integrating performance metrics with the proposed IDS ✓ Implementing a reaction scheme for routing attacks

Table 1. A summary of studied IDSs.

security algorithm combined the advanced encryption standard (AES) and the revest Shamir Adelman (RSA). The combined key generation algorithm generated a message and sent it to the base station to announce the responsible RSU for authentication. After initializing the configuration and running the ODMRP routing, RSA created public/private keys to generate the asymmetric key by AES. Finally, a secret shared key was generated to send to the receiver RSU. The receiver RSU used the private key for the decryption of the secret shared key. The AES algorithm had a better encryption time than the combined security algorithm. The proposed protocol increased throughput and PDR, reducing the PLR and packet delivery time. The suggested model used the encryption technique to keep the message secure. The encryption ratio is limited to reduce the proposed protocol's complexity. A weakness of this protocol is the participation of attacking vehicles in routing.

Magaia et al.²⁰ presented a cluster-based routing protocol to support IoV's reliable communications. They mentioned that high vehicle speeds and unstable and unreliable communication are essential issues in IoV. The proposed protocol used a clustering algorithm at the network's edge and the group'n route (GnR) approach in routing. Nodes were clustered based on Euclidean distance and k-means clustering in this protocol. Indeed, the proposed protocol clustered vehicles with the same connections using social relationships. Cluster heads were selected based on two metrics by the edge. After clustering, routing was done based on specific metrics and forwarding policies. The proposed protocol supported stable communication through graphs of relationships and ML in the IoV. Using social relationships in clustering routing covered the mobility and patterns of vehicles. Despite controlling the number of message copies, the proposed protocol has acceptable PDR and overhead. The proposed protocol maintained its proper efficiency by increasing the number of nodes. The suggested routing has worse results in overhead and PDR than the enhanced privacy-preserving opportunistic (ePRIVO) protocol. In the proposed protocol, routing reliability was not considered for V2V communications.

An extended AODV routing protocol was presented in VANETs by Kumar et al.¹⁹ to detect black hole attacks. They referred to the router role of vehicles and the probability of sharing spoofed routing tables in VANETs. The route request (RREQ) and the route reply (RREP) packets were modified to secure routing in the proposed protocol. Before forwarding the packet, the destination vehicle confirmed the source through the packet sequence number. A cryptographic function was used for encryption/decryption to increase routing security. The proposed protocol removed malicious vehicles from the network by cutting off their communications. Based on the results, the proposed protocol has a high network performance compared to the default AODV, but the disadvantage is low efficiency with high overhead in real scenarios.

A secure routing protocol was presented by Divya et al.¹⁸ using the clustered vehicle location protocol (CVLP) and the hybrid krill herd-bat optimization (HKH-BO) technique to enhance secure communications in VANETs. CVLP reduced energy consumption and packet delay on the network. HKH-BO identified black hole and wormhole attacks based on packet delivery time. Black hole and wormhole attacks were mentioned as destroyers of data broadcasting regions in VANETs. Based on the results, the proposed protocol improved data transmission by minimizing overload and delay while increasing the sent packets. The suggested algorithm is not scalable, and if the vehicle density increases, the network throughput decreases significantly. A disadvantage

of the proposed protocol is its focus on the security of cluster heads, and the security of cluster members was not considered in the proposed protocol.

The studied routing protocols have different performances regarding PDR, PLR, E2E-delay, throughput, etc. Table 2 summarizes the protocols studied regarding challenges, innovations, advantages, disadvantages, and future works. Many routing protocols are secured against only one type of attack¹⁹. Therefore, these protocols are vulnerable to other attacks. RSR-IDS is designed to detect all known/unknown abnormal data. Several protocols are empowered by encryption and decryption methods using security algorithms^{19,21}. These methods may have a limited encryption ratio¹⁹. RSR-IDS chooses a safe and short path without encrypting/decrypting messages. Increasing nodes decreases the network throughput significantly in many protocols¹⁸. Some routing protocols have high overhead in real scenarios¹⁹. Many clustering protocols are only focused on the security of cluster heads¹⁸. RSR-IDS considers the safety of all vehicles by identifying data and choosing a reliable route in VANETs. The studied protocols try to have a high network performance without considering routing reliability. Therefore, attacking vehicles may participate in the routing of VANETs.

System model

RSR-IDS is an IDS-adaptive routing protocol wherein the IDS of Yang et al.²⁴ inspires the detection system of RSR-IDS. Figure 1 shows the diagram of the RSR-IDS system model. First, the input data is entered into the pre-processing step. Data traffic is pre-processed in RSR-IDS by two techniques, including MinMaxScaler and Pearson’s correlation coefficient (PCC) techniques. Data traffic is categorized by the three ML classifier algorithms of Yang’s IDS using linear regression. The routing of RSR-IDS is modeled from the AODV protocol. This section examines data pre-processing techniques, linear regression, basic IDS, and routing protocol. Table 3 shows notations used in the equations of the system model to facilitate understanding.

Data pre-processing techniques

Various techniques are presented to pre-process data in IDSs. RSR-IDS uses MinMaxScaler and PCC techniques for data pre-processing. These two techniques are examined in the following:

MinMaxScaler technique: MinMaxScaler is one of the scaling techniques in data pre-processing to normalize data in a fixed range³⁴. In addition to equating the scale of data, the limits of their change should also be within a fixed range. This rescaling is obtained through Eq. (1)³⁴:

$$X_{norm} = \frac{X_{old} - X_{min}}{X_{max} - X_{min}} \tag{1}$$

where X_{min} represents the minimum value, X_{max} represents the maximum value and X_{old} is the original data.

PCC technique: PCC is a data pre-processing technique to measure the dependency between features in data without labels³⁵. By determining the correlation of features, it is possible to estimate their efficiency in identifying normal and abnormal data³⁵. Suppose a dataset has N samples with f_1, f_2, \dots, f_d features. The number of classes in this dataset is equal to w . Also, f_1 has x_i values, and f_2 has y_i values. According to these, the PCC between f_1 and f_2 features is calculated through Eq. (2)³⁵:

$$PCC(f_1, f_2) = \frac{cov(f_1, f_2)}{\sigma_{f_1} \cdot \sigma_{f_2}} = \frac{\sum_{i=1}^N (x_i - M_{f_1})(y_i - M_{f_2})}{\sqrt{\sum_{i=1}^N (x_i - M_{f_1})^2} \cdot \sqrt{\sum_{i=1}^N (y_i - M_{f_2})^2}} \tag{2}$$

Ref	Challenge	Innovation	Advantage	Disadvantage	Future work
21	Evaluating the efficiency of the proposed ODMRP and improving the security in VANETs	Proposing ODMRP using an encryption algorithm	✓ Reducing the protocol’s complexity by encryption ratio ✓ Increasing the throughput and PDR ✓ Reducing PLR and the packet delivery time	✓ Participation of attacking vehicles in routing ✓ Higher encryption time than AES	Using ODMRP in security-sensitive applications such as government and police agencies
20	Low computational and bandwidth overheads in routing while covering node mobility	Cluster-based routing protocol using GnR	✓ Supporting stable communication ✓ Maintaining efficiency by increasing nodes	✓ Ignoring the routing reliability in V2V communications ✓ Worse overhead and PDR ePRIVO	✓ Using clustering algorithms to estimate the number of clusters ✓ Finding the optimal location of RSUs
19	Routing while detecting the black hole attack	AODV used a cryptographic function	✓ Removing malicious vehicles from the network ✓ Higher performance than the default AODV	✓ Low efficiency with high overhead in real scenarios ✓ Failure to detect other attacks	Using the proposed approach to detect end-to-end rounding overhead attacks
18	Routing while detecting black hole and wormhole attacks	CVLP and HKH-BO	✓ Reducing energy consumption and packet delay ✓ Improving the overload, delay, and sent packets	✓ Ignoring the security of members in the cluster ✓ Decreasing throughput while increasing vehicles ✓ Failure to detect other attacks	-

Table 2. A summary of studied protocols.

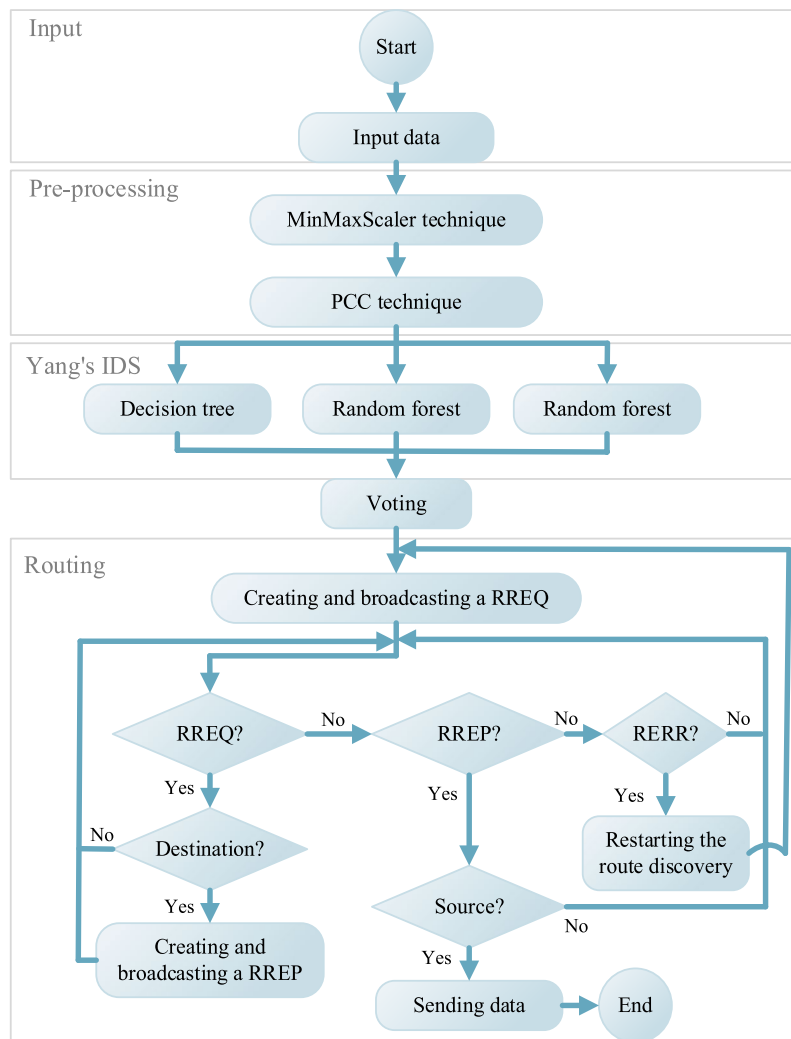


Fig. 1. The diagram of the RSR-IDS system model.

Notations	Phrase
X_{norm}	Normalized data using MinMaxScaler
X_{old}	Original data in MinMaxScaler
X_{min}	Minimum value of data in MinMaxScaler
X_{max}	Maximum value of data in MinMaxScaler
N	Number of data samples
f_1, f_2, \dots, f_d	Features of data in PCC
w	Number of data classes in PCC
x_i	Values of f_1 feature in PCC
y_i	Values of f_2 feature in PCC
$cov()$	Covariance
σ	Standard deviation
x	Input data value in linear regression
y	Class of data in linear regression

Table 3. Notations of equations in the system model section.

where $cov()$ is the covariance and σ is the standard deviation. Also, M_{f_1} and M_{f_2} are calculated through Eqs. (3–4)³⁵:

$$M_{f_1} = \frac{\sum_i^N x_i}{N} \quad (3)$$

$$M_{f_2} = N \sum_i^N y_i \quad (4)$$

The PCC output is a number between -1 and 1. Values close to -1 or 1 indicate a strong correlation of features. Values close to 0 also indicate no correlation between features³⁵.

Linear regression

In VANETs, not all data traffic is binary numbers. Tree-based ML algorithms need a technique to categorize feature values for drawing their trees. RSR-IDS uses linear regression to draw trees and categorize data. This technique looks for a linear function among feature values, as shown in Eq. (5)³⁶:

$$y = ax + b \quad (5)$$

where a is the correlation coefficient, and b is the constant parameter. The correlation coefficient and constant parameter are calculated from Eq. (6–7)³⁶:

$$Coefficient = \frac{N \sum xy - \sum x \sum y}{N \sum x^2 - (\sum x)^2} \quad (6)$$

$$ConstantTerm = \frac{\sum y \sum x^2 - \sum x \sum xy}{N \sum x^2 - (\sum x)^2} \quad (7)$$

where x represents the input data value and y represents the data class. N indicates the number of samples.

Baseline IDS

RSR-IDS is inspired by the detection system of Yang et al.²⁴. Their suggested IDS detects abnormal data for IVC and external networks. The proposed IDS in²⁴ uses the K-means clustering, the SMOTE, and the z-score techniques in the pre-processing. Moreover, their proposed detection system uses the information gain method and the fast correlation-based filter (FCBF) algorithm in the feature engineering step. The classifier is trained for signature-based and anomaly-based detection steps using four tree-based ML algorithms and K-means cluster labeling, respectively. These algorithms include the decision tree, random forest, extra trees, and XGboost ensemble as a model. Signature-based and anomaly-based detection systems are optimized using two Bayesian algorithms. Yang's IDS can detect normal data, unknown/known attacks, and their type. We modified Yang's detection system to match the RSR-IDS protocol. The pre-processing of RSR-IDS is completely different from Yang's system. RSR-IDS includes the decision tree, random forest, and extra trees for classification. The detection system of RSR-IDS obtains final outputs through the voting technique.

Routing requirements

This paper models RSR-IDS from the AODV protocol. Routes are discovered in the AODV protocol only when needed. AODV protocol suits wide area networks with high dynamics. This protocol selects a path with the least H_c between the source and destination nodes. Table 4 shows abbreviations used in the AODV routing protocol to facilitate understanding. Control messages of this protocol include RREQ, RREP, and route error (RERR) messages³⁷. The RREQ message includes the route request ID (RREQ_ID), IP addresses (SRC_IP, DST_IP), sequence numbers (SRC_SEQ, DST_SEQ), H_c and request flags (F_q) fields³⁸. The RREP message contains SRC_IP, DST_IP, DST_SEQ, lifetime (T_l), response flags (F_r), prefix size (P_s), and H_c fields. To start routing discovery, a RREQ message is generated and broadcasted by the source node. If the neighbor node is the destination, it returns a RREP message to the source. Receiving the RERR message means link failure. This message is sent by a node adjacent to the damaged. The RERR message starts the path discovery to remove the lost routes. This message contains the unreachable destination count (UNR_DST_CNT), unreachable IP Addresses (UNR_DST_IP), and unreachable destination sequence number (UNR_DST_SEQ) fields. We modify the AODV protocol to be compatible with RSR-IDS.

Proposed model

The proposed RSR-IDS establishes reliable routing based on the classified data and uses a detection system as the scoring mechanism deployed on the fog server. RSR-IDS selects the message path in the routing using extracted USs and the investigated H_c from the source to the destination.

RSR-IDS is described in four subsections, including the network architecture and RSR-IDS protocol. Section “Network architecture” describes the RSR-IDS network architecture and shows the intelligent transportation network strategy in the proposed data-based routing. Section “RSR-IDS” also explains the RSR-IDS routing protocol in this paper.

Abbreviations	Phrase
RREQ	Route request
RREP	Route reply
RERR	Route error
RREQ_ID	Route request ID
SRC_IP	Source IP addresses
DST_IP	Destination IP addresses
SRC_SEQ	Source sequence numbers
DST_SEQ	Destination sequence numbers
H_c	Hop count
F_q	Request flags
T_l	Lifetime
F_r	Response flags
P_s	Prefix size
UNR_DST_CNT	Unreachable destination count
UNR_DST_IP	Unreachable IP Addresses
UNR_DST_SEQ	Unreachable destination sequence number

Table 4. Abbreviations of the AODV routing protocol.

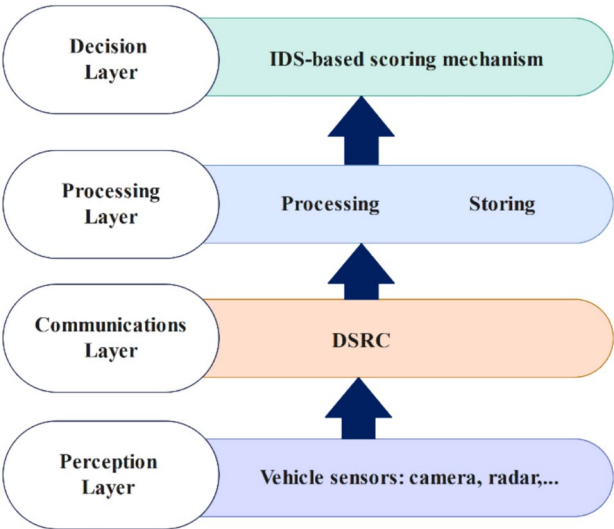


Fig. 2. Layers of the VANET network in RSR-IDS.

Network architecture

As shown in Fig. 2, the RSR-IDS network architecture consists of four layers, including the perception layer, communication layer, processing layer, and decision layer. The perception layer comprises vehicles with different internal/external sensors to collect information³⁹. RSR-IDS supports V2V and V2I communications via dedicated short-range communication (DSRC)^{39–41}. The processing layer includes gateways, storage, and fog servers for easier data access and management. As depicted in Fig. 2, RSR-IDS deploys a 4-step mechanism in the processing layer before the results are sent to the decision layer. The decision layer processes and manages the collected data to deploy the scoring mechanism of RSR-IDS.

Deployed RSUs in RSR-IDS communicate to the fog server directly to send the collected data from surrounding vehicles. The utilized one-hop communication between RSUs and fog servers makes a real-time location-based detection strategy in the proposed RSR-IDS. By collecting and processing data, RSR-IDS tries to enhance the security and reliability of VANET communications through the proposed fog-enabled decision-making system. Using the fog-assisted computing layer, RSR-IDS can reduce the latency of data transmission and increase the performance of the IDS even in such dynamic networks. Customized location-dependent real-time data analysis through a deployed fog computing layer leads to the quick identification and response to anomalies.

Figure 3 shows the intelligent transportation network for the RSR-IDS protocol. According to Fig. 3, each vehicle sends data to a connected RSU. Then, the collected data are sent to the fog server by RSUs for the classification and scoring process of the centralized IDS. During the RSR-IDS routing, calculated USs are

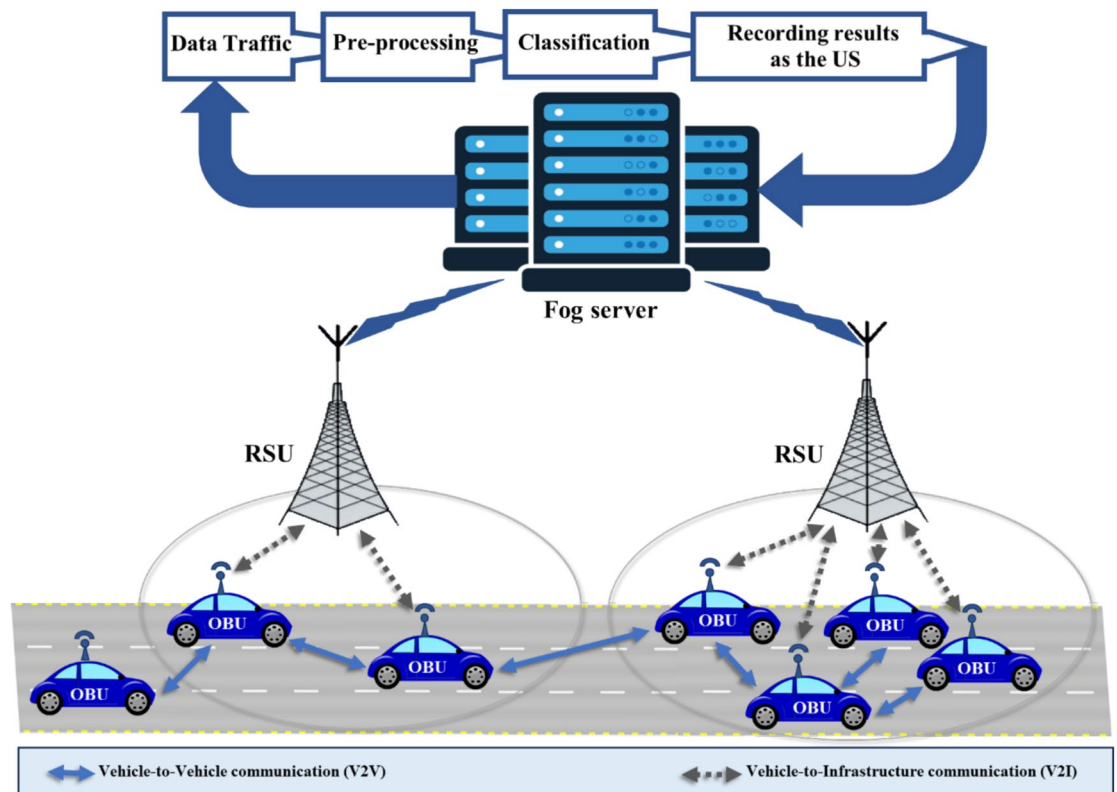


Fig. 3. The intelligent transportation network in RSR-IDS.

provided to vehicles if needed. Vehicles receiving the RREQ message with a lower US send the RREP message to the source. Upon receiving the first RREQ message by the destination, the main route with the least total USs and H_c is selected.

RSR-IDS

RSR-IDS is a reliable routing protocol organized by a centralized scoring mechanism deployed in the fog server. In the path discovery operation of RSR-IDS, the message route is selected considering USs and H_c from the source to the destination. Figure 4 shows the flowchart of the RSR-IDS protocol. The RSR-IDS protocol is completely studied in the “Data-based scoring mechanism” and “US-based routing protocol” subsections to address the proposed scoring strategy and routing protocol, respectively. Section “Data-based scoring mechanism” explains the proposed data-based scoring mechanism in five steps. The steps include data pre-processing and classification, checking the data and vehicle status, and the output of the scoring mechanism. Section “US-based routing protocol” explains the proposed US-based routing protocol in six steps. The steps include checking the received message, receiving the RREQ and checking the vehicle status, receiving the RREP and checking the status of the vehicle, and receiving the RERR message.

Data-based scoring mechanism

In the RSR-IDS protocol, the data-based scoring mechanism is a detection system deployed in the fog server. This mechanism calculates the US for each vehicle based on data classification. Algorithm 1 shows the data-based scoring mechanism of RSR-IDS. This mechanism is described in the following five steps:

Step 1: Data pre-processing

RSR-IDS pre-processes data traffic using two techniques. Data traffic is normalized in RSR-IDS using the MinMaxScaler technique. The correlation of data is measured in RSR-IDS using the PCC technique. Section “Data pre-processing techniques” describes these techniques.

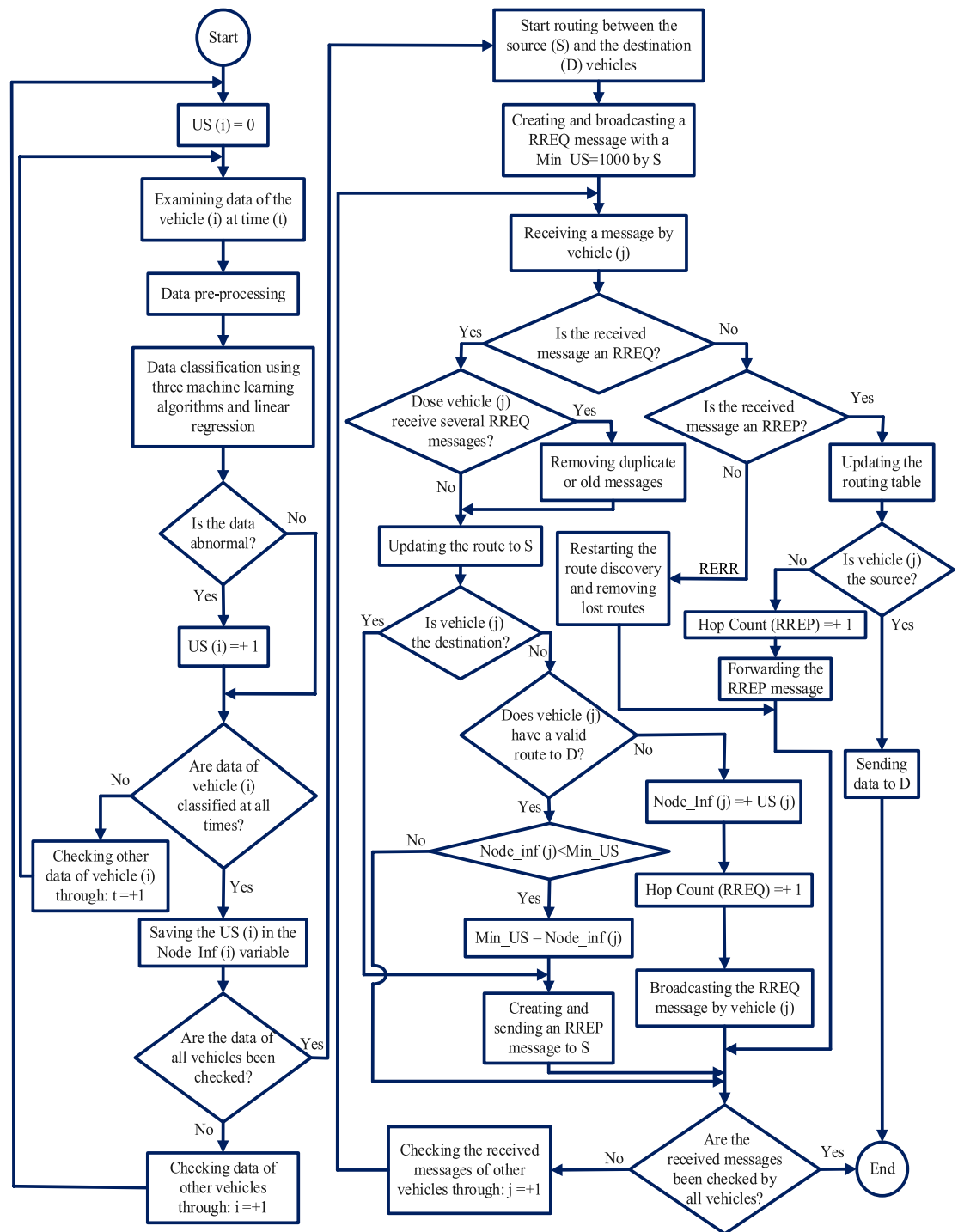


Fig. 4. The flowchart of the proposed RSR-IDS.

```

➤ Input: Data traffic of vehicles;
➤ Output: USs of vehicles;
➤ Initialize:
     $n \leftarrow$  number of vehicles;
     $m \leftarrow$  number of data with the same identity;
     $US() \leftarrow 0$ ;
     $Node\_Inf() \leftarrow$  variables for recording  $US()$ ;
➤ Begin
1: Step1: for ( $D = data\_traffic()$ ) do
2:   Pre-process ( $D$ );
3:   Split  $data\_traffic()$  to  $train\_data()$  and  $test\_data()$ ;
4: end for
5: Step2: Train the classifier by giving  $train\_data()$  to the decision tree,
6:   random forest and extra trees algorithms using regression;
7:   while ( $D = test\_data()$ ) do
8:     Classify  $D$  using tree algorithms of step2;
9:     Determine the result of the normality/abnormality using the
10:    voting method;
11: Step3:   for ( $i \leftarrow 1$  to  $n$ ) do
12: Step4:   for ( $t \leftarrow 1$  to  $m$ ) do
13:     if  $D$  is abnormal then
14:        $US(i) \leftarrow US(i) + 1$ ;
15:       Dropping the abnormal data;
16:     end if
17:      $t \leftarrow t + 1$ ;
18:   end for
19: Step5:   Recording  $US(i)$  in  $Node\_Inf()$ ;
20:      $i \leftarrow i + 1$ ;
21:   end for
22: end while
➤ End

```

Algorithm 1. The data-based scoring mechanism of RSR-IDS.

Step 2: Data classification

The trained classifier classifies data as normal or abnormal after the pre-processing step of RSR-IDS. This classifier is trained using the labeled dataset and three ML algorithms. These three algorithms include the decision tree, random forest, and extra trees. These algorithms categorize feature values using linear regression. The final outputs of this classifier are calculated based on the voting technique in the testing phase.

Step 3: Checking the data status

After the data classification into normal or abnormal categories, one of the following two situations occurs in this step:

1. If the input data of the i th vehicle (V_i) at the moment T_1 are identified as abnormal, the fog server drop it and update the US of V_i ($US(i)$) according to Eq. (8):

$$US_{T_1}(i) = US_{T_0}(i) + b \quad (8)$$

where b is a small value (such as 1) in the proposed IDS.

2. If the input data of V_i at the moment T_1 are identified as normal, $US(i)$ is unchanged.

Step 4: Checking the data status of vehicles

Vehicles generate time-variant data traffic, and the fog server receives and classifies all the data from all vehicles. The US of V_i is calculated based on the classification results of all data with the same ID until the desired time (T_t). This step checks whether vehicle data at all times are classified. In this step, one of the following two situations occurs:

1. If all data of V_i are not classified at all times, data of V_i are checked at different times by adding to the time counter (t).
2. If all data of V_i are classified at all times, the $US(i)$ is recorded in a variable. Data traffic of other vehicles is also checked by adding to the vehicle counter (i).

Step 5: Output of scoring mechanism

RSR-IDS deploys a detection system in the fog server to use the USs of vehicles and record the calculated scores in variables. The USs are always available, and vehicles can access them through RSUs during route discovery of RSR-IDS. Based on the above, the detection system of RSR-IDS is a data-based scoring mechanism in message routing.

US-based routing protocol

RSR-IDS is an untrusted score-based routing protocol that selects a route with a lower H_c and minimal USs compared to other routes. RSR-IDS uses the AODV protocol but proposes a different method for route discovery. The routing protocol in RSR-IDS is performed if the source and destination vehicles are not in the same RSU range. Algorithm 2 shows the US-based routing protocol of RSR-IDS.

Step 1: Checking the received message

The neighboring vehicle checks the message type after receiving a message. This message can be an RREQ, RREP, or RERR message.

Step 2: Receiving the RREQ message

The source vehicle generates and broadcasts an RREQ message at the beginning of the route discovery process. The H_c value of this message is 0. At the beginning of the route discovery, the Minimum value of US (Min_US) equals 1000. Min_US is a variable to determine the lowest US. The Min_US variable leads to selecting the route with the lowest total USs compared to others.

Step 3: Checking the vehicle status after receiving the RREQ message

After receiving an RREQ message, the vehicle checks whether it is the destination. In this step, one of the following two situations occurs:

- (A) If the vehicle is the destination, the RREQ message is stopped from being broadcast. This vehicle generates and sends an RREP message to the source. In the message, H_c and DST_SEQ are equal to 0 and the DST_SEQ of the destination, respectively.
- (B) If the vehicle is not the destination, it checks whether it knows a valid route to the destination. In this case, one of the following two situations occurs:
 1. If the vehicle has a valid route to the destination, it compares the Min_US value and its US variable (Node_Inf). If the Node_Inf value of this vehicle is smaller than the Min_US, the Min_US is updated to it. This vehicle also generates and sends an RREP message to the source. In the RREP message, H_c equals H_c from this vehicle to the destination. The DST_SEQ of the RREP message equals the DST_SEQ of this vehicle.
 2. If the vehicle has no valid route to the destination, its US is added to the Node_Inf value. The H_c of the RREQ message is also increased. After modifying the RREQ message, this vehicle broadcasts it to neighbors.

➤ **Input:** Control packets (R_{req} ; R_{rep} ; R_{err});

➤ **Output:** Unicast main route from source to destination;

➤ **Initialize:**

$S \leftarrow$ Source vehicle;	$US(i) \leftarrow$ US of the vehicle (i);
$D \leftarrow$ Destination vehicle;	$Min_US \leftarrow$ Minimum US ();
$V_i \leftarrow$ Vehicle (i);	$Node_Inf(i) \leftarrow$ Variable for US (i);
$CP \leftarrow$ Control packet;	$Dest_seq \leftarrow$ Destination sequence number;
$I() \leftarrow$ vehicles receiving CP;	$Org_seq \leftarrow$ Originator sequence number;
$R_{req} \leftarrow$ Route request;	$DOF \leftarrow$ Destination only flag;
$R_{rep} \leftarrow$ Route reply;	$H_c \leftarrow$ Hop count;
$R_{err} \leftarrow$ Route Error;	$RT \leftarrow$ Routing table;

➤ **Begin**

- 1: Step1: S create and broadcast R_{req} ;
- 2: $H_c(R_{req}) \leftarrow 0$; $Min_US \leftarrow 1000$;
- 3: **while** ($V_i = I()$) **do**
- 4: Step2: **if** ($CP = R_{req}$) **then**
- 5: **if** invalid R_{req} **then**
- 6: Discard old or duplicate R_{req} ;
- 7: **end if**
- 8: Step3: **if** ($V_i = D$) **then**
- 9: D create R_{rep} and send it to S;
- 10: $Dest_seq(R_{rep}) \leftarrow Dest_seq(D)$; $H_c(R_{rep}) \leftarrow 0$;
- 11: **else if** (V_i has an active route to D) **then**
- 12: **if** ($Node_Inf(i) < Min_US$) **then**
- 13: $Min_US \leftarrow Node_Inf(i)$;
- 14: **end if**
- 15: **if** ($DOF = FALSE$) **then**
- 16: **if** ($Node_Inf(i) < Min_US$) **then**
- 17: Create R_{rep} and send it to S for the main route;
- 18: $Dest_seq(R_{rep}) \leftarrow Dest_seq(V_i)$; $H_c(R_{rep}) \leftarrow H_c(V_i \text{ to } D)$;
- 19: **end if**
- 20: **end if**
- 21: **else**
- 22: $Node_Inf(i) \leftarrow Node_Inf(i) + US(i)$;
- 23: $RT \leftarrow Org_seq(R_{req})$; $H_c \leftarrow H_c + 1$;
- 24: Setup reverse route and rebroadcast R_{req} ;
- 25: **end if**
- 26: Step4: **else if** ($CP = R_{err}$) **then**
- 27: Remove lost routes and rediscover the route;
- 28: Step5: **else if** ($CP = R_{rep}$) **then**
- 29: Step6: **if** ($V_i = S$) **then**
- 30: Send data to D;
- 31: **else**
- 32: $H_c(R_{rep}) \leftarrow H_c(R_{rep}) + 1$; Forward R_{rep} towards S;
- 33: **end if**
- 34: **end if**
- 35: **end while**

➤ **End**

Algorithm 2. US-based routing of RSR-IDS.

- Step 4: Receiving the RREP message
After receiving an RREP message, the vehicle creates a reverse route entry to the source in its routing table.
- Step 5: Checking the vehicle status after receiving the RREP message
After receiving an RREP message, the vehicle checks whether it is the source. Therefore, one of the following two situations occurs:
- (A) If the vehicle is the source, it sends data to the destination.
 - (B) If the vehicle is not the source, the H_c of the RREP message is increased. After modifying the RREP message, this vehicle sends it again.
- Step 6: Receiving the RERR message
After receiving a RERR message, the route discovery operation is restarted to remove lost routes.

Performance evaluation

The OMNeT++ 6.0.1 simulator⁴² is used to evaluate RSR-IDS in this paper. OMNeT++ is a framework-based open-source platform for easily simulating a wide variety of modern network configurations with supported real-world communication protocols. To simulate a time-variant high-dynamic VANET, we used Veins⁴³, an event-driven open-source framework to simulate a vehicular network. Moreover, we used SUMO, a road traffic simulator on OMNeT++. In the detection strategy of VANETs, many datasets like VeReMi, CICIDS2017, and UNSW-NB15 were extracted from real scenarios or vehicles. These datasets can be chosen based on specific requirements, like attack scenarios, to cover the dynamics and unique challenges of VANETs. We used the UNSW-NB15 dataset with 49 features containing normal data and nine kinds of attack data. These attacks include fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, and worms. Therefore, a trained IDS with the UNSW-NB15 dataset is expected to detect these attacks (like DoS, worms, etc.) as an intrusion at least accurately. Due to the diversity of attack data, the UNSW-NB15 dataset is suitable for evaluating IDSs. UNSW-NB15 dataset contains 100 GB of data samples collected using the tcpdump tool. The simulation of RSR-IDS is run on a 64-bit system with Windows 11 core (TM) i7-10750H CPU, 2.60 GHz main frequency, and 16 G memory. Considering this system's processor power and memory space, a reduced dataset from UNSW-NB15 is used to speed up the training and testing phases of the RSR-IDS simulation. The reduced UNSW-NB15 was pre-processed by removing null values, converting them to numerical, and using two techniques, MinMaxScaler and PCC. Table 5 shows the summary of simulation parameters. The testing dataset contains 10,000 data samples, 14 features, and one class feature to run the simulation smoothly. Some of these selected features include sttl (source to destination lifetime), load/dload (source/destination bits per second), state (state and the related protocol), and is_sm_ips_ports (same source, destination, IP addresses, and port numbers means value one else 0), etc. Samples have two normal (0) and abnormal (1) classes. The common abnormal-to-normal data ratio in the testing dataset is 35% to 65%. In the testing dataset, samples labeled 1 and 0 equal 3500 and 6500, respectively. After trial and error, 1000 samples are considered the training dataset for fast and robust drawing of classification trees. The common ratio of abnormal to normal data in the training dataset is 70–30%. In the training dataset, samples labeled 1 and 0 are equal to 700 and 300, respectively. OMNeT++ supports integration with simulation tools and heterogeneous networks for realistic mobility in urban scenarios. The Ostad Moin area of Tehran city was selected for the RSR-IDS simulation due to the authors' better knowledge and the clarity of the paths of vehicles. Figure 5 shows the Ostad Moein urban area, which is extracted using OpenStreetMap⁴⁴. For establishing RSR-IDS routing between vehicles in different ranges, the number of RSUs is considered three units ($1 <$), which can be changed. RSUs are located 1000 m from each other based on the real scenarios of the intelligent transportation network. Due to the mentioned number of RSUs and their distance, the simulated geographic area is considered $3300 \times 3300 \text{ m}^2$. The Unit Disk Radio model is a simple wave propagation model used in RSR-IDS. In the Unit Disk Radio model, vehicles communicate within a specific range. The simulation

Parameters	Values
Simulator	OMNeT++ 6.0.1
Scenario area	Urban area
Simulation dimensions	$3300 \times 3300 \text{ m}^2$
Dataset	UNSW-NB15
Number of vehicles	10, 20, 30, 40, 50, 60
Number of training samples	1000 sample
Number of testing samples	10,000 sample
Communication protocol	IEEE 802.11p
Communication channel delay	100 ms
Simulation time	100 s

Table 5. The summary of simulation parameters.

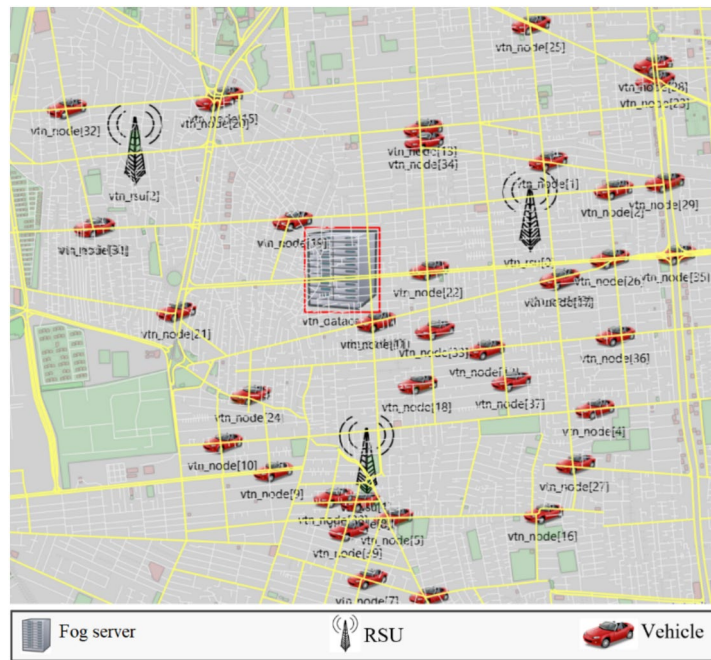


Fig. 5. The urban scenario of RSR-IDS with 40 vehicles, 3 RSUs, and one fog server in OMNET + +.

time is considered 100 s in this paper. Communication channels in this network also have a 100 ms delay, according to the default in VANETs.

Based on realistic and varied mobility behaviors of vehicles in urban environments, this network includes a special movement policy for each one (ID) with different speeds, angles (directions), and coordinates. The initial coordinates and the next movements of each vehicle are determined with coordinates (x, y) axes values. The movement speed of vehicles is uniformly set randomly between 20 and 60 units per second. The angle of vehicles' movement in streets until reaching the intersection is considered unchanged. When reaching the intersection, the angle of vehicles changes to 360° for the opposite direction, $+90^\circ$ for moving up, and -90° for moving down. The movement scripts are repeated to continue motion patterns for each vehicle.

Evaluation criteria

RSR-IDS is an IDS-based routing protocol for selecting the best route regarding shortness and reliability. This section investigates the evaluation criteria of routing and detection in the RSR-IDS simulation. The detection evaluation criteria include accuracy, F1-score, and FNR. Parameters like true positive (TP), true negative (TN), false positive (FP), and false negative (FN) are used to calculate detection criteria⁷. The number of samples correctly identified as abnormal by an IDS is called TP. The number of samples correctly identified as normal by an IDS is called the TN. The number of samples incorrectly identified as abnormal by an IDS is called FP. The number of samples incorrectly identified as normal by an IDS is called FN. Evaluation criteria of routing include PDR, PLR, E2E-delay, and throughput. The evaluation criteria of this article are expressed as a percentage, except for E2E-delay, which is expressed in milliseconds. The evaluation criteria of this article are defined below:

Accuracy: In evaluating an IDS, the detection accuracy or accuracy rate is calculated through Eq. (9)⁴⁵:

$$Accuracy(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (9)$$

F1-score: In evaluating an IDS, the F1-score or F1-measure is the average harmonic of recall and precision, which is calculated through Eq. (10)⁷:

$$F1 - score(\%) = \frac{2 * Precision * Recall}{Precision + Recall} \times 100 \quad (10)$$

where precision and recall are calculated through Eqs. (11–12)^{45,46}:

$$Precision(\%) = \frac{TP}{TP + FP} \times 100 \quad (11)$$

$$Recall(\%) = \frac{TP}{TP + FN} \times 100 \quad (12)$$

FNR: In evaluating an IDS, the FNR is calculated through Eq. (13)⁴⁷:

$$FNR(\%) = \frac{FN}{TP + FN} \times 100 \quad (13)$$

PDR: In evaluating a routing protocol, the rate of the received packet number by the destination vehicle compared to the total number of packets sent is called the PDR. This metric is calculated through Eq. (14)¹⁹:

$$PDR(\%) = \frac{\sum_{i=0}^n PacketReceived_i}{\sum_{j=0}^m PacketSent} \times 100 \quad (14)$$

PLR: In evaluating a routing protocol, the rate of the number of lost packets to the total number of sent packets is called the PLR. This metric is calculated through Eq. (15)⁴⁸:

$$PLR(\%) = \frac{\sum_{i=0}^n PacketLoss}{\sum_{j=0}^m PacketSent} \times 100 \quad (15)$$

E2E-delay: In evaluating a routing protocol, the required time to receive a message packet from the source to the destination is E2E-delay¹⁹. This metric is usually checked as an average in milliseconds (*ms*).

Throughput: Throughput is the ratio of the delivered data bits to the total simulation time⁴⁹. Therefore, the throughput percentage is calculated through Eq. (16):

$$Throughput(\%) = \frac{\sum_{i=0}^n PacketReceived_i}{TotalSimulationTime} \times \frac{PacketSize}{MaximumThroughput} \times 100 \quad (16)$$

Percentage improvement: The percentage improvement of RSR-IDS over others in terms of criteria like accuracy, F1-score, PDR, and throughput, where a higher value is better, is used through Eq. (17):

$$PercentageImprovement(\%) = \frac{Value_{RSR-IDS} - Value_{other}}{Value_{other}} \times 100 \quad (17)$$

where $Value_{RSR-IDS}$ and $Value_{other}$ show the average value of RSR-IDS and other methods for an evaluation criterion, respectively. The percentage improvement of RSR-IDS over others in terms of criteria like FNR, PLR, and E2E-delay, where a lower value is better, is used through Eq. (18):

$$PercentageImprovement(\%) = \frac{Value_{other} - Value_{RSR-IDS}}{Value_{other}} \times 100 \quad (18)$$

Simulation and results

This section shows simulation results for RSR-IDS by investigating various numbers of vehicles, including 10, 20, 30, 40, 50, and 60 *veh*. This section compares RSR-IDS with cooperative and misbehavior-aware on-demand collaborative IDSs^{50,51} regarding accuracy, F1-score, and FNR. Cooperative IDSs (CIDSs) and Misbehavior-aware on-demand collaborative IDSs (MACIDSs) are based on XGBoost, SVM, and random forest.

Table 6 compares the accuracy of RSR-IDS with CIDSs (i.e., SVM-based CIDS and XGBoost-based CIDS) and MACIDSs (i.e., SVM-based MACIDS and XGBoost-based MACIDS) considering various numbers of vehicles. The detection accuracy of RSR-IDS in the simulation scenario with 10, 20, 30, 40, and 50 *veh* equals 99.56%, 99.62%, 99.43%, 99.57%, and 99.45%, respectively. RSR-IDS has an average accuracy of 99.52%. XGBoost-based and SVM-based CIDSs have an average accuracy of 93.85% and 87.2%, respectively. The average accuracy of XGBoost-based and SVM-based MACIDSs is equal to 95.8% and 94.53%, respectively. Random forest-based CIDS and MACIDS are ensemble models with average accuracy of 86.35% and 97.54%, respectively. Therefore, RSR-IDS has a higher average accuracy than CIDSs and MACIDSs. Indeed, RSR-IDS can detect normal and abnormal data with greater accuracy and better performance than other ensemble models. The high accuracy of RSR-IDS directly impacts the accuracy of the calculated US for each vehicle and, ultimately, the reliability of routing.

Accuracy					
IDS	Number of vehicles				
	10	20	30	40	50
XGBoost-based CIDS	88.42%	94.14%	95.48%	96.26%	94.99%
SVM-based CIDS	82.05%	85.98%	87.02%	90.05%	91.04%
XGBoost-based MACIDS	94.72%	95.37%	96.88%	95.63%	96.85%
SVM-based MACIDS	94.05%	93.85%	95.03%	94.96%	94.75%
RSR-IDS	99.56%	99.62%	99.43%	99.57%	99.45%

Table 6. Accuracy evaluation of RSR-IDS compared to CIDSs and MACIDSs considering various numbers of vehicles.

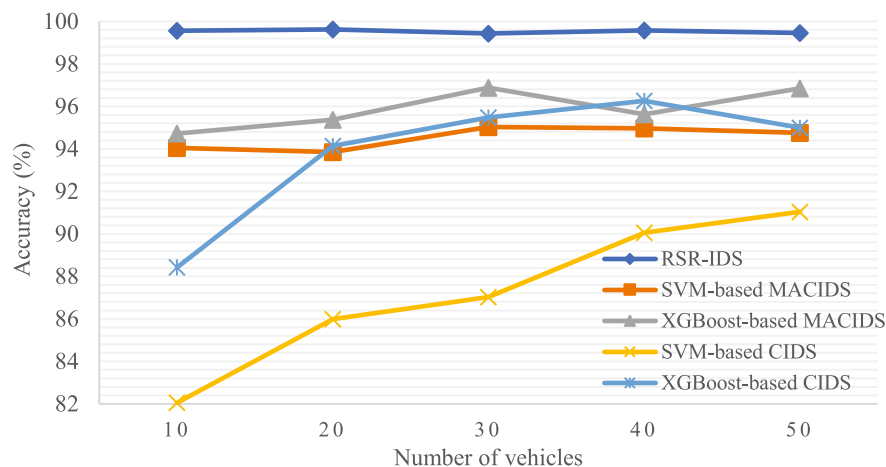


Fig. 6. Comparison of RSR-IDS, CIDSs, and MACIDSs regarding accuracy under the number of vehicles.

F1-score					
IDS	Number of vehicles				
	10	20	30	40	50
XGBoost-based CIDS	87.92%	94.41%	95.88%	96.39%	95.19%
SVM-based CIDS	84.99%	88.04%	89%	91.92%	91.85%
XGBoost-based MACIDS	94.06%	95.02%	97.05%	95.7%	97.01%
SVM-based MACIDS	94.41%	94.58%	95.29%	95.49%	95.41%
RSR-IDS	99.47%	99.39%	99.31%	99.42%	99.3%

Table 7. F1-score evaluation of RSR-IDS compared to CIDSs and MACIDSs.

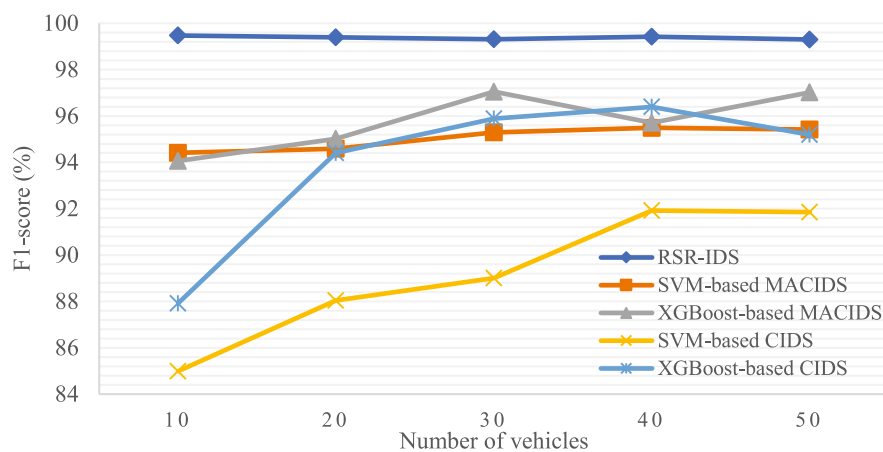


Fig. 7. Comparison of RSR-IDS, CIDSs, and MACIDSs regarding F1-score under the number of vehicles.

Figure 6 shows the simulation results for RSR-IDS, CIDSs, and MACIDSs in terms of accuracy, wherein the RSR-IDS outperforms the other detection systems under the various vehicle densities. The simulation results also show that RSR-IDS keeps the accuracy high by increasing the number of vehicles. The results depict a significant improvement in the accuracy of RSR-IDS compared to SVM-based CIDS, with an increase of 14.1%. The slight improvement in the accuracy of RSR-IDS is compared to XGBoost-based MACIDS, with an increase of 3.8%.

Table 7 compares the F1-score of RSR-IDS with CIDSs and MACIDSs under various numbers of vehicles. The F1-score of RSR-IDS in the simulation scenario with 10, 20, 30, 40, and 50 *veh* equals 99.47%, 99.39%, 99.31%, 99.42%, and 99.3%, respectively. The average F1-score of RSR-IDS is 99.37%, whereas XGBoost-based and SVM-based CIDSs have an average of 93.96% and 89.16%, respectively. Moreover, the average F1-score of XGBoost-based and SVM-based MACIDSs equals 95.77% and 95.03%, respectively. Random forest-based

IDS	Number of vehicles				
	10	20	30	40	50
XGBoost-based CIDS	11.98%	5.63%	3.63%	0.22%	4.45%
SVM-based CIDS	1.07%	0.31%	0.33%	0.24%	0.35%
XGBoost-based MACIDS	4.08%	2.59%	2.29%	4.61%	1.59%
SVM-based MACIDS	0.79%	0.41%	0.29%	0.25%	0.51%
RSR-IDS	1.04%	1.2%	1.35%	1.15%	1.38%

Table 8. FNR evaluation of RSR-IDS compared to CIDSs and MACIDSs.

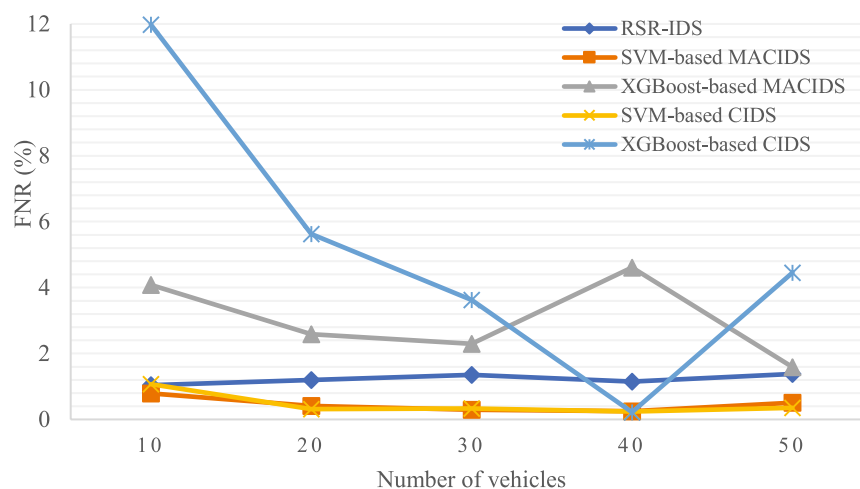


Fig. 8. Comparison of RSR-IDS, CIDSs, and MACIDSs regarding FNR under the number of vehicles.

CIDS and MACIDS also have the average F1-score of 88.07% and 97.73%, respectively. Therefore, RSR-IDS has a higher average F1-score than CIDSs and MACIDSs.

Figure 7 shows the F1-score simulation results for RSR-IDS, CIDSs, and MACIDSs under various traffic flows. RSR-IDS demonstrates a better result than other detection systems. RSR-IDS maintains the F1-score high by increasing the number of vehicles. The great improvement of RSR-IDS is compared to SVM-based CIDS, with an increase of 11.4% in the F1-score. The low F1-score improvement of RSR-IDS is compared to XGBoost-based MACIDS with a rise of 3.7%.

Table 8 investigates the FNR results for RSR-IDS, CIDSs, and MACIDSs, considering the different numbers of vehicles. The FNR results for RSR-IDS in the various simulation scenarios with 10, 20, 30, 40, and 50 *veh* equal 1.04%, 1.2%, 1.35%, 1.15%, and 1.38%, respectively. The average FNR for RSR-IDS is 1.22%, whereas the XGBoost-based and SVM-based CIDSs have an average of 5.2% and 0.46%, respectively. Moreover, the average FNR for XGBoost-based and SVM-based MACIDSs equals 3.03% and 0.45%, respectively. Random forest-based CIDS and MACIDS have an average FNR of 5.85% and 1.65%, respectively. RSR-IDS has an acceptable average FNR. In fact, RSR-IDS has a low false detection of normal data. Therefore, untrusted vehicles are less likely to participate in VANET routing messages than trusted vehicles.

Figure 8 shows the FNR of IDSs and RSR-IDS under the number of vehicles. RSR-IDS has a better result than XGBoost-based detection systems, but the SVM-based CIDS and SVM-based MACIDS outperform RSR-IDS regarding the FNR. Figure 9 shows the average performance of RSR-IDS, CIDSs, and MACIDSs in terms of accuracy, F1-score, and FNR. As depicted in Figs. 6 and 7, RSR-IDS shows better performance in terms of accuracy and F1-score, respectively. According to Fig. 8, RSR-IDS also outperforms XGBoost-based CIDS and MACIDS in terms of FNR.

Figure 10 summarizes the evaluated improvement results of the proposed IDS and depicts that in the accuracy and F1-score factors, the RSR-IDS outperforms SVM-based CIDS, XGBoost-based CIDS, Random forest-based CIDS, SVM-based MACIDS, XGBoost-based MACIDS, and Random forest-based MACIDS. The maximum accuracy improvement of RSR-IDS is 15.25% compared to Random forest-based CIDS, while the minimum is 2.03% compared to Random forest-based MACIDS. Random forest-based MACIDS also achieved the highest F1-score (12.83%), while the lowest was related to Random forest-based CIDS (1.68%).

The routing evaluation criteria of the RSR-IDS include the PDR, PLR, E2E-delay, and throughput. We considered three RSUs to communicate with 20, 40, and 60 *veh* in different traffic-centric scenarios to evaluate RSR-IDS routing. The following section compares RSR-IDS with the fuzzy assisted location aided routing using cache (FLAR-C)^{52,53}, the directional cache agent-based location-aided routing (D-CALAR)⁵⁴, and CALAR using distance and direction (CALAR-DD)⁵⁵ in an urban scenario. Figure 11 shows the PDR of RSR-IDS, FLAR-C,

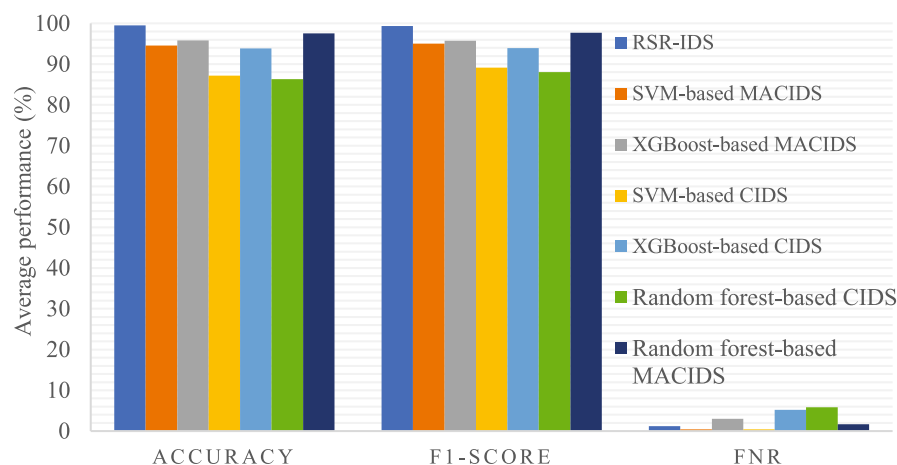


Fig. 9. Comparison of the average performance of RSR-IDS with CIDs and MACIDs.

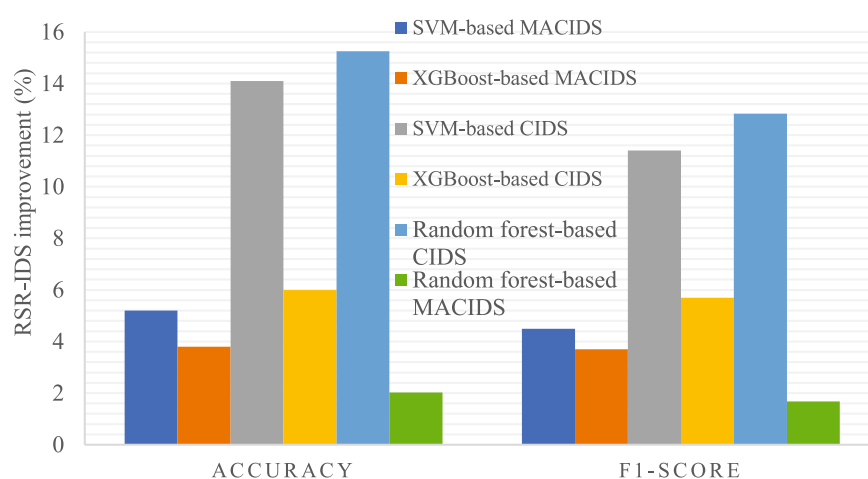


Fig. 10. Improvement percentage of RSR-IDS compared to CIDs and MACIDs.

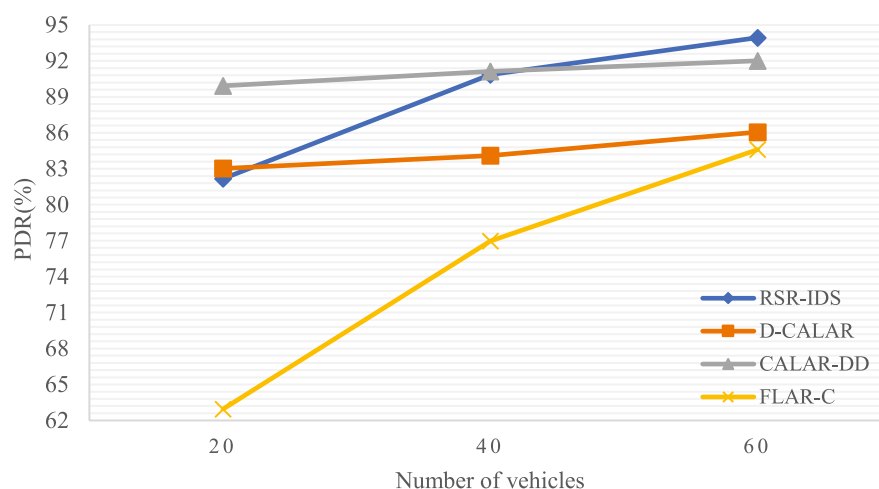


Fig. 11. Comparison of RSR-IDS, FLAR-C, D-CALAR, and CALAR-DD regarding PDR under various numbers of vehicles.

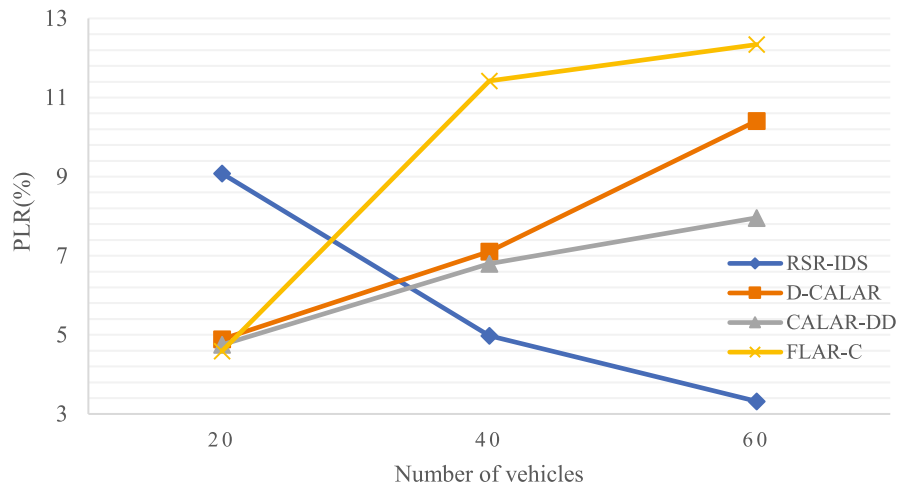


Fig. 12. PLR Comparison of RSR-IDS, FLAR-C, D-CALAR, and CALAR-DD under various numbers of vehicles.

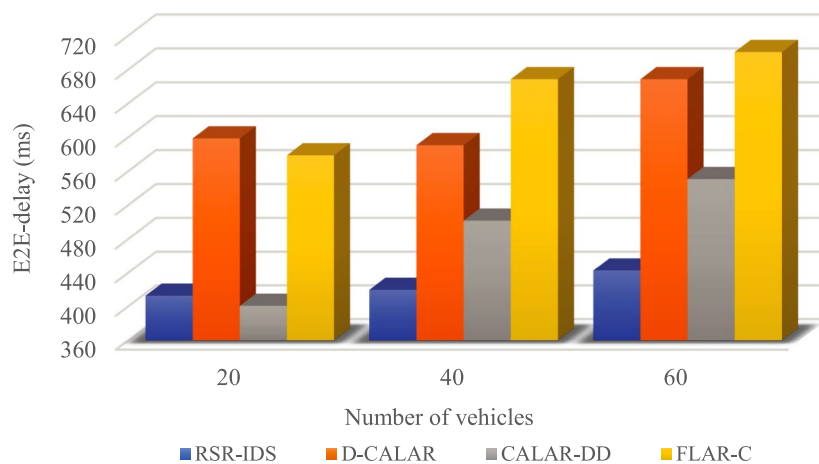


Fig. 13. Comparison of RSR-IDS, FLAR-C, D-CALAR, and CALAR-DD regarding E2E-delay under the number of vehicles.

D-CALAR, and CALAR-DD under the number of vehicles. The PDR of RSR-IDS in the simulation scenario with 20, 40, and 60 *veh* equals 82.17%, 90.86%, and 93.94%, respectively. RSR-IDS has an average PDR of 89%. The FLAR-C, D-CALAR, and CALAR-DD protocols have average PDR of 74.84%, 84.41%, and 91.02%, respectively.

Based on the results, RSR-IDS improves PDR by increasing the number of vehicles. RSR-IDS achieves the greatest PDR in the scenario with 60 *veh*, while CALAR-DD has better results in 20 and 40 *veh* scenarios. RSR-IDS has better PDR than FLAR-C and D-CALAR protocols. The PDR of RSR-IDS has an 18.9% increase compared to the FLAR-C protocol. RSR-IDS also increases PDR by 5.4% compared to the D-CALAR protocol. However, the PDR of RSR-IDS has decreased by 2.2% compared to the CALAR-DD protocol.

Figure 12 shows the PLR results for RSR-IDS, FLAR-C, D-CALAR, and CALAR-DD. The PLR of RSR-IDS in the simulation scenario with 20, 40, and 60 *veh* equals 9.08%, 4.98%, and 3.32%, respectively. RSR-IDS has an average of 5.79% for PLR, whereas FLAR-C, D-CALAR, and CALAR-DD protocols have an average PLR of 9.45%, 7.47%, and 6.5%, respectively. Thus, RSR-IDS has a much lower PLR than other protocols, especially FLAR-C and D-CALAR. The simulation results show that RSR-IDS decreases PLR by increasing the number of vehicles. RSR-IDS achieves a lower PLR in 40 and 60 *veh* scenarios, while FLAR-C has better results in the 20 *veh* scenario. The simulation results show a 10.9% decrease in the PLR of RSR-IDS compared to the CALAR-DD protocol.

Figure 13 shows the E2E-delay for various traffic flows for RSR-IDS, FLAR-C, D-CALAR, and CALAR-DD. E2E-delay of RSR-IDS in the simulation scenario with 20, 40, and 60 *veh* equals 413 ms, 420 ms, and 443 ms, respectively. Based on the results, the average E2E-delay of RSR-IDS is 425 ms, while FLAR-C, D-CALAR, and CALAR-DD protocols have an average E2E-delay of 649 ms, 619 ms, and 478 ms, respectively. RSR-IDS greatly reduces E2E-delay compared to others, especially FLAR-C and D-CALAR protocols. RSR-IDS also reduces E2E-

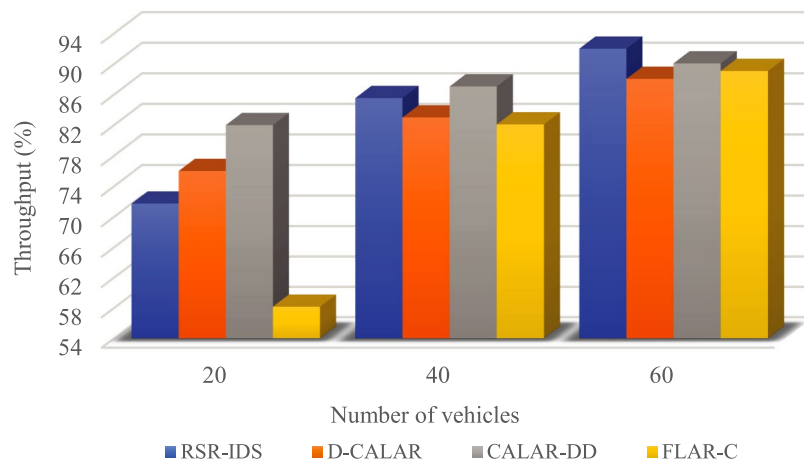


Fig. 14. Comparison of RSR-IDS, FLAR-C, D-CALAR, and CALAR-DD regarding the throughput under the number of vehicles.

delay by 11% compared to the CALAR-DD protocol. Of course, increasing the number of vehicles raises the E2E-delay of RSR-IDS.

Figure 14 shows the throughput of RSR-IDS, FLAR-C, D-CALAR, and CALAR-DD under the number of vehicles. The throughput of RSR-IDS in the simulation scenario with 20, 40, and 60 *veh* equals 71.76%, 85.57%, and 92.03%, respectively. RSR-IDS has an average throughput of 83.12%. The FLAR-C, D-CALAR, and CALAR-DD protocols have average throughputs of 76.47%, 82.37%, and 86.4%, respectively. Based on the results, RSR-IDS improves throughput by increasing the number of vehicles. RSR-IDS achieves the greatest throughput in the scenario with 60 *veh*. RSR-IDS has better throughput than FLAR-C and D-CALAR protocols. The throughput of RSR-IDS has an 8.7% increase compared to the FLAR-C protocol. RSR-IDS also increases throughput by 0.9% compared to the D-CALAR protocol. However, the throughput of RSR-IDS has decreased by 3.8% compared to the CALAR-DD protocol.

Discussion

In this paper, we suggested RSR-IDS as a routing protocol to select the suitable route based on calculated USs and H_c in VANETs. RSR-IDS classified data using the voting technique among three tree-based algorithms. Using the voting technique in tree-based classification has reduced overfitting. RSR-IDS accurately detected the normality/abnormality of data and dropped abnormal items. RSR-IDS also calculated the US for each vehicle based on the detection results. The high US indicated unreliability. RSR-IDS ensures the reliability and security of communications by selecting a route based on the lowest USs and hop count. It was possible to access USs through RSUs and the fog server. Integrating fog computing with VANETs allows for the deployment of IDS in the fog server, adapting to the scalable and dynamic nature of vehicular networks. Deploying the detection strategy of RSR-IDS on the fog server solves the data diversity problem compared to deploying it on the OBU of each vehicle. Of course, RSR-IDS requires a relatively large amount of memory. Providing the required memory for RSR-IDS is less expensive than the deployed IDS in the physical data center. Using a fog-assisted IDS increases the distributed and decentralized nature of fog computing. Despite fog computing's potential to reduce latency, achieving low-latency performance for routing protocols in dynamic VANET environments is necessary. Selecting a reliable and efficient path by the routing protocol maintains high throughput and low delay in VANETs. Ensuring real-time detection and response with high-mobility and dense scenarios is also a research challenge for VANETs. Using a lightweight IDS can improve the scalability of the urban scenario. Dense areas, especially city centers, always have many connected vehicles in their intelligent transportation system. In such high-demand environments where the low time of processing data and identification is essential, a lightweight RSR-IDS detection system can be suggested.

RSR-IDS is a score-based multi-hop protocol using the classification results of an IDS. RSR-IDS just classifies data into normal and abnormal classes without detecting their type. We added the detection results of IDS for each vehicle's data to its US. Therefore, if abnormal data was detected, RSR-IDS only added one unit to the vehicle's US detection result. This point may decrease the computing complexity and time in the urban scenario of VANET, while approaches with the ability to detect attack types are more complex. Indeed, unlike other score-based protocols, RSR-IDS didn't use complex equations to calculate the USs of vehicles. Therefore, we conclude that RSR-IDS has an acceptable computational complexity compared to score-based routing protocols. Of course, designing a multi-class classifier that can detect attack types can be examined for RSR-IDS in the future. Identifying the attack type can help assign specific weights for each attack type in the scoring mechanism of RSR-IDS. For example, black hole and Sybil attacks can be more important in disrupting message routing. Therefore, assigning high-weight values for special attacks like black hole and sybil in the US can make the RSR-IDS routing more reliable. Finally, the RSR-IDS routing mechanism has also selected vehicles with lower scores to broadcast RREQ messages by comparing USs of single-hop vehicles.

The time complexity of IDSs is also calculated based on their classification algorithms, especially in the training phase. Tree-based models (like XGBoost, Decision Tree, Random Forest, Extra Trees, etc.) are scalable and fast on large datasets because they can parallelize computation. RSR-IDS identified data based on the decision tree, random forest, and extra trees algorithms. The time complexity of training RSR-IDS is almost equal to $O(2N_t d N_s \log N_s + d N_s \log N_s)$, where N_t , d , and N_s are the number of trees, features, and training samples, respectively. The time complexity of training MACIDS and CIDS based on XGBoost is equal to $O(N_t d N_s \log N_s)$. The time complexity of training non-linear SVM is also between $O(N_s^2 d)$ and $O(N_s^3 d)$. Therefore, MACIDS and CIDS based on XGBoost are highly efficient and scalable with a logarithmic combination to handle large datasets. RSR-IDS is also efficient and scalable, particularly with logarithmic complexity, making it suitable for large datasets. However, its efficiency might be slightly less compared to XGBoost when dealing with huge datasets due to the optimized boosting process in XGBoost. MACIDS and CIDS based on SVM are typically the most computationally intensive, especially with large datasets. Therefore, SVM-based IDSs have a high time complexity and less scalability than others due to the quadratic or cubic dependency on the number of samples.

The simulator was OMNeT++, continuously updating libraries and fixing bugs for better scalability and performance in VANETs. In the detection strategy of RSR-IDS, UNSW-NB15 was also chosen as an available dataset in VANETs. Using the preprocessed UNSW-NB15 dataset, which contains 14 features and nine types of attack (like DoS, worms, etc.), should lead to detecting them accurately (99.52%). The 14 selected features were important items to train the RSR-IDS. Of course, some of these, especially time and connection features, must be effective in other attack detection, such as delay and deception attacks. However, it's necessary to use a training dataset with the related features and data to detect other attack types more accurately, such as delay and deception attacks. The simulation scenario was an urban area with different vehicle density scenarios, including 20, 40, and 60 *veh*. Based on the average results, RSR-IDS has achieved an accuracy of 99.52%, an F1-score of 99.37%, and an FNR of 1.22%. Due to fog-assisted IDS, RSR-IDS was able to keep its accuracy and F1-score high by increasing the number of vehicles. In this paper, RSR-IDS had higher accuracy and F1-score than other IDSs. RSR-IDS has improved a 3.8% accuracy and a 3.7% F1-score compared to XGBoost-based MACIDS. The RSR-IDS improvement of accuracy and F1-score to SVM-based MACIDS has achieved 5.2% and 4.5%, respectively. RSR-IDS has improved 2.03% accuracy and 1.68% F1-score compared to random forest-based MACIDS. RSR-IDS has significantly improved accuracy and F1-score compared to CIDS models. RSR-IDS had a lower value than XGBoost-based and random forest detection systems while it had a higher result of FNR than SVM-based IDSs. In security scenarios where preventing attacker penetration is particularly important, using SVM-based IDSs has a higher priority than RSR-IDS. Based on the average results, RSR-IDS has achieved a PDR of 89%, PLR of 5.79%, E2E-delay of 425 ms, and throughput of 83.12% in routing. RSR-IDS has improved PDR, PLR, and throughput while increasing the number of vehicles in the urban environment. RSR-IDS had the highest PDR in dense urban scenarios (60 *veh*), while CALAR-DD had better PDR in low-density (20 *veh*) and medium-density (40 *veh*) scenarios. RSR-IDS also achieved the greatest throughput in the dense scenario (60 *veh*). RSR-IDS has improved PDR by 5.4% compared to D-CALAR, while CALAR-DD has increased PDR by 2.2%. The throughput of RSR-IDS also had an 8.7% increase compared to FLAR-C. Therefore, the CALAR-DD protocol is more suitable than others for dense urban scenarios in terms of PDR and throughput. According to the CALAR-DD protocol, using distance and direction as important parameters in routing scenarios can help to increase packet delivery. RSR-IDS had a lower PLR in medium-density (40 *veh*) and dense (60 *veh*) urban scenarios, while FLAR-C had a better PLR in low-density (20 *veh*) scenarios than others. RSR-IDS also had better E2E-delay compared to other protocols. With the increase in the number of vehicles, the E2E-delay of RSR-IDS has also increased. To improve the E2E-delay in scale-up urban scenarios, the number of RSUs can be increased slightly in RSR-IDS to increase network coverage more widely and faster. Moreover, using unmanned aerial vehicles (UAVs) is another way to control E2E-delay in dense environments. UAVs are cost-effective mobile infrastructures that can be utilized in time-variant traffic conditions and can handle messages to guarantee RSR-IDS scalability. Therefore, we can conclude that the proposed RSR-IDS would be more scalable by using extra RSUs or UAV-assisted solutions, specifically in dense urban areas. In the urban scenario, the mobility and dynamics of many vehicles are also covered using RSR-IDS. RSR-IDS selected a route with a lower H_c and USs for communication in VANETs. The routing reliability and security of vehicles were increased using RSR-IDS in VANETs.

Real-time management of urban traffic for densely populated smart cities is crucial. RSR-IDS is a multi-hop protocol in urban environments that consumes short times to detect attacks, process data, and calculate USs. On the other hand, the excessive speed of vehicles on highways, especially outside of cities (roads between villages), may hamper the opportunity for intrusion detection and scoring. The low number of vehicles on the highways also contradicts the multi-hop nature of RSR-IDS. Therefore, designing RSR-IDS for rural or highway areas to cover the mentioned challenges is suggested for the future. Implementing RSR-IDS in a real-time industrial scenario involving self-driving vehicles can show this protocol's security and efficiency. RSR-IDS can also be deployed efficiently in real-time scenarios of intelligent transportation systems like emergency vehicles, critical response management, and logistic operations. Emergency response vehicles require rapid and secure message routing quickly. Establishing communication priority for emergency vehicles in the RSR-IDS algorithm can ensure that real-time information is received free from security threats. RSR-IDS can also establish a dedicated communication channel between emergency vehicles, RSUs, and fog servers to provide real-time updates. Deploying with a logistics firm collaboration is one of the RSR-IDS applications in real scenarios. Installing fog servers near traffic signals also facilitates real-time data processing and communication. Lightweight encryption/decryption methods are also suggested in some secure and sensitive situations where maintaining the privacy of messages is more critical than delay.

Conclusion

This paper proposed a reliable score-based routing protocol called RSR-IDS in vehicular networks. We deployed a centralized fog-assisted detection system to propose a score-based unreliability-calculation methodology for vehicles. The suggested fog-enabled intrusion detection strategy solved the data traffic problem compared to an infrastructure-less strategy. RSR-IDS selected the appropriate route based on the scores and hop counts. The chosen route was reliable and short for establishing inter-vehicle communications. RSR-IDS increased the routing reliability and security in vehicular networks. Simulation results on OMNeT++ showed a 14.1% increase in the accuracy of RSR-IDS compared to the baseline works focused on the support vector machine-based cooperative detection systems.

The proposed model increased the F1-score by 11.4% compared to the investigated baseline method. RSR-IDS showed a better false negative rate than XGBoost-based detection systems. RSR-IDS also reduces end-to-end delay by 11% compared to the cache agent-based protocol using distance and direction, while decreasing packet loss ratio by 10.9% and packet delivery ratio by 2.2%.

In future work, we suggest an attack-aware routing protocol with a multi-class classifier to detect attack types and assign specific weights for each one, especially black-hole and Sybil attacks, in the scoring mechanism. Moreover, we propose a low-computational intrusion detection strategy as a scoring mechanism to deal with attacks in the routing. We also suggest designing a lightweight IDS to support scalability in high-demand and dense transportation systems like city centers. Designing RSR-IDS for rural or highway areas is suggested for the future.

Data availability

The basic datasets during the current study are available in the UNSW research repository, <https://research.unsw.edu.au/projects/unsw-nb15-dataset>. The generated datasets used during the current study are also available from the corresponding author on reasonable request.

Received: 5 December 2024; Accepted: 19 June 2025

Published online: 16 July 2025

References

1. Al-Jarrah, O. Y. et al. Intrusion detection systems for intra-vehicle networks: A review. *IEEE Access* **7**, 21266–21289 (2019).
2. Zarei, M. Traffic-centric mesoscopic analysis of connectivity in VANETs. *Comput. J.* **63**(2), 203–219 (2020).
3. Al-Mayouf, Y. R. B. et al. Survey on VANET technologies and simulation models. *ARN J. Eng. Appl. Sci.* **11**(15), 9414–9427 (2016).
4. Safwat, M. et al. Survey and taxonomy of information-centric vehicular networking security attacks. *Ad Hoc Netw.* **124**, 102696 (2022).
5. Farooqi, A. M. et al. A fog computing model for VANET to reduce latency and delay using 5G network in smart city transportation. *Appl. Sci.* **12**(4), 2083 (2022).
6. Lokman, S.-F., Othman, A. T. & Abu-Bakar, M.-H. Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review. *EURASIP J. Wirel. Commun. Netw.* **2019**(1), 1–17 (2019).
7. Banafshehvaragh, S. T. & Rahmani, A. M. Intrusion, anomaly, and attack detection in smart vehicles. *Microprocess. Microsyst.* **96**, 104726 (2023).
8. Afzal, Z. & Kumar, M. Security of vehicular Ad-hoc networks (VANET): A survey. In *Journal of Physics: Conference Series*. IOP Publishing (2020).
9. Wu, W. et al. A survey of intrusion detection for in-vehicle networks. *IEEE Trans. Intell. Transp. Syst.* **21**(3), 919–933 (2019).
10. Dibaei, M. et al. Attacks and defences on intelligent connected vehicles: a survey. *Digital Commun. Netw.* **6**(4), 399–421 (2020).
11. Li, Y. et al. Methods and challenges of cryptography-based privacy-protection algorithms for vehicular networks. *Electronics* **13**(12), 2372 (2024).
12. Sharma, S. & Kaul, A. A survey on intrusion detection systems and honeypot based proactive security mechanisms in VANETs and VANET cloud. *Veh. Commun.* **12**, 138–164 (2018).
13. Sakiz, F. & Sen, S. A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV. *Ad Hoc Netw.* **61**, 33–50 (2017).
14. Yaqoob, S. et al. Deep learning based anomaly detection for fog-assisted IoVs network. *IEEE Access* **11**, 19024–19038 (2023).
15. Huang, C., Lu, R. & Choo, K.-K.R. Vehicular fog computing: Architecture, use case, and security and forensic challenges. *IEEE Commun. Mag.* **55**(11), 105–111 (2017).
16. Al-Mayouf, Y. R. B., Mahdi, O. A., Uliyan, D. M. An intersection-based segment aware algorithm for geographic routing in VANETs. In *2018 9th International Conference on Information and Communication Systems (ICICS)*. IEEE (2018).
17. Hemmati, A. & Zarei, M. UFC3: UAV-aided fog computing based congestion control strategy for emergency message dissemination in 5G internet of vehicles. *Autom. Innov.* **7**(3), 456–472 (2024).
18. Divya, N. S., Bobba, V., & Vatambeti, R. An adaptive cluster based vehicular routing protocol for secure communication. *Wirel. Person. Commun.* 1–20 (2021).
19. Kumar, A. et al. Black hole attack detection in vehicular ad-hoc network using secure AODV routing algorithm. *Microprocess. Microsyst.* **80**, 103352 (2021).
20. Magaia, N., et al., Group'n route: An edge learning-based clustering and efficient routing scheme leveraging social strength for the internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* (2022).
21. Sharma, P., Pandey, S. & Jain, S. Implementation of efficient security algorithm and performance improvement through ODMRP protocol in VANET environment. *Wirel. Pers. Commun.* **123**(3), 2555–2579 (2022).
22. Al-Mayouf, Y.R.B., et al. Efficient routing algorithm for VANETs based on distance factor. In *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEEES)*. IEEE (2016).
23. Bangui, H., Ge, M., & Buhnova, B. A hybrid machine learning model for intrusion detection in VANET. *Computing* 1–29 (2021).
24. Yang, L., Moubayed, A., & Shami, A. MTH-IDS: A multi-tiered hybrid intrusion detection system for internet of vehicles. *IEEE Internet Things J.* (2021).
25. Partovi, Z., Zarei, M. & Rahmani, A. M. Data-centric approaches in the internet of vehicles: A systematic review on techniques, open issues, and future directions. *Int. J. Commun. Syst.* **36**(3), e5383 (2023).
26. Mahmood, J. et al. Security in vehicular ad hoc networks: Challenges and countermeasures. *Secur. Commun. Netw.* **2021**, 1–20 (2021).

27. Alsarhan, A., et al., Machine Learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks. *J. Ambient Intell. Human. Comput.* 1–10 (2021).
28. Karthiga, B., et al., Intelligent intrusion detection system for VANET using machine learning and deep learning approaches. *Wirel. Commun. Mobile Comput.* **2022** (2022).
29. Amaouche, S. et al. FSCB-IDS: Feature selection and minority class balancing for attacks detection in VANETS. *Appl. Sci.* **13**(13), 7488 (2023).
30. Arya, M. et al. Intruder detection in VANET data streams using federated learning for smart city environments. *Electronics* **12**(4), 894 (2023).
31. Ajjaj, S. et al. A new multivariate approach for real time detection of routing security attacks in VANETs. *Information* **13**(6), 282 (2022).
32. Al-Otaibi, B., Al-Nabhan, N. & Tian, Y. Privacy-preserving vehicular rogue node detection scheme for fog computing. *Sensors* **19**(4), 965 (2019).
33. Paranjothi, A. & Atiquzzaman, M. Enhancing security in vanets with efficient sybil attack detection using fog computing. arXiv preprint [arXiv:2108.10319](https://arxiv.org/abs/2108.10319), 2021.
34. Jo, J.-M. Effectiveness of normalization pre-processing of big data to the machine learning performance. *J. Korea Inst. Electron. Commun. Sci.* **14**(3), 547–552 (2019).
35. Moustafa, N. & Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf. Secur. J. A Global Perspect.* **25**(1–3), 18–31 (2016).
36. Korkmaz, M. A study over the general formula of regression sum of squares in multiple linear regression. *Numer. Methods Partial Differ. Equ.* **37**(1), 406–421 (2021).
37. Dhankhar, S. & Agrawal, S. VANETs: A survey on routing protocols and issues. *Int. J. Innovat. Res. Sci. Eng. Technol.* **3**(6), 13427–13435 (2014).
38. Saini, T. K. & Sharma, S. C. Recent advancements, review analysis, and extensions of the AODV with the illustration of the applied concept. *Ad Hoc Netw.* **103**, 102148 (2020).
39. El Madani, S., Motahhir, S., & El Ghzizal, A. Internet of vehicles: Concept, process, security aspects and solutions. *Multimedia Tools Appl.* 1–25 (2022).
40. Abdollah, S. & Zarei, M. A traffic-centric fuzzy approach for solving the starvation problem of cooperative awareness messages in vehicular ad hoc networks. *Int. J. Commun. Syst.* **34**(18), e4999 (2021).
41. Hemmati, A., Zarei, M., & Rahmani, A.M. Big data challenges and opportunities in Internet of Vehicles: A systematic review. *Int. J. Pervasive Comput. Commun.* (2024).
42. Varga, A. and R. Hornig. An overview of the OMNeT++ simulation environment. In *1st International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems*. 2010.
43. Sommer, C., German, R. & Dressler, F. Bidirectionally coupled network and road traffic simulation for improved IVC analysis. *IEEE Trans. Mob. Comput.* **10**(1), 3–15 (2010).
44. Haklay, M. & Weber, P. Openstreetmap: User-generated street maps. *IEEE Pervasive Comput.* **7**(4), 12–18 (2008).
45. Muna, A.-H., Moustafa, N. & Sitnikova, E. Identification of malicious activities in industrial internet of things based on deep learning models. *J. Inf. Secur. Appl.* **41**, 1–11 (2018).
46. Gao, Y. et al. A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network. *IEEE Access* **7**, 154560–154571 (2019).
47. Song, H. M., Woo, J. & Kim, H. K. In-vehicle network intrusion detection using deep convolutional neural network. *Veh. Commun.* **21**, 100198 (2020).
48. Moridi, E. & Barati, H. RMRPTS: A reliable multi-level routing protocol with tabu search in VANET. *Telecommun. Syst.* **65**(1), 127–137 (2017).
49. Fatemidokht, H. & Kuchaki Rafsanjani, M. F-Ant: An effective routing protocol for ant colony optimization based on fuzzy logic in vehicular ad hoc networks. *Neural Comput. Appl.* **29**, 1127–1137 (2018).
50. Zhang, T. & Zhu, Q. Distributed privacy-preserving collaborative intrusion detection systems for VANETs. *IEEE Trans. Signal Inf. Process. Over Netw.* **4**(1), 148–161 (2018).
51. Ghaleb, A. et al. Misbehavior-aware on-demand collaborative intrusion detection system using distributed ensemble learning for VANET. *Electronics* **9**(9), 1411 (2020).
52. Gurumoorthi, E. & Ayyasamy, A. An intelligent fuzzy based location aided routing in vehicular ad hoc networks. *Int. J. Innov. Technol. Explor. Eng.* **8**, 1946–1955 (2019).
53. Gurumoorthi, E. & Ayyasamy, A. Performance analysis of Geocast based location aided routing using Cache agent in VANET. *Int. J. Inf. Technol.* **14**(1), 125–134 (2022).
54. Gurumoorthi, E. & Ayyasamy, A. Cache agent based location aided routing protocol using direction for performance enhancement in VANET. *Wirel. Pers. Commun.* **109**(2), 1195–1216 (2019).
55. Gurumoorthi, E. & Ayyasamy, A. Cache agent based location aided routing using distance and direction for performance enhancement in VANET. *Telecommun. Syst.* **73**(3), 419–432 (2020).

Author contributions

Conceptualization, Methodology, Software, Writing-Original draft preparation: S.T.B.; Investigation, Supervision, Validation, Reviewing and Editing: M.Z.; Supervision: A.M.R. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025