Trusted Opportunistic Routing Based on Node Trust Model

BING SU, CHENHUI DU, AND JUAN HUAN
School of Information and Engineering, Changzhou University, Changzhou 213164, China
Corresponding author: Bing Su (subing@cczu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61803050.

ABSTRACT In vehicular ad hoc networks, opportunistic routing can effectively improve the reliability and throughput. However, opportunistic routing also has security issues. For example, malicious nodes can easily mix into node candidate sets, which can interfere with network performance. In this paper, a trust model based on node behavior is proposed for solving the problem of malicious nodes in the opportunistic routing and forwarding candidate set. The proposed trust model uses pruning and filtering mechanisms to remove malicious suggestions, and uses dynamic weight calculation methods to combine direct trust and indirect trust when calculating the comprehensive trust value, which can screen and filter low-trust nodes in the network. Then, combining the ETX (Expected Transmission Count) value and the node trust value, an opportunity routing algorithm based on trust model (BTOR) is proposed. Extensive simulation results represent that the algorithm can significantly improve the network performance and reduce the interference of malicious nodes to the network system.

INDEX TERMS Trust model, opportunistic routing, node trust, pruning filtering mechanism, anti-interference.

I. INTRODUCTION

At present, vehicular ad hoc network is mainly realized by sensor technology, Internet technology and wireless communication technology. As vehicular ad hoc network can reduce traffic accidents, improve traffic efficiency and safety, provide information and entertainment services and the dissemination of related information, it is considered to be one of the most important applications of mobile self-organizing network, so it is widely concerned by researchers. Vehicular ad hoc network has the characteristics of changeable topology, diverse environment and fast scene change, which leads to the problems and challenges different from the general network environment [1]. For example, the unpredictable mobility of nodes makes it not always possible to route packets on the best path [2]. Protocols that use several metrics have been shown to be the most adequate to VANETs due to their effectiveness in dealing with dynamic environment changes due to vehicle mobility [3]. OR (Opportunistic Routing, OR) is designed to take advantage of the broadcast characteristics of wireless channels to a certain extent to improve the transmission reliability and end-to-end throughput of vehicular ad hoc networks. In recent years, researchers have conducted a lot of research. In contrast to traditional methods for routing (such as AODV, DSR, and OLSR [4]), which selects only one node as the relay node, the OR methods benefit from the broadcast characteristics of the wireless medium, and a group of neighbor nodes is candidate to forward to the destination more reliably Packet. In the traditional routing protocol, if the only selectable next-hop node has never received a data packet, this situation will cause the loss of data packets and reduce network performance. Therefore, by selecting a group of nodes as potential next hops, these nodes can cooperate with each other to more reliably forward data packets to their destinations [5]. However, security issues also ensue. Malicious nodes can join the set of candidate forwarding nodes through collusion and identity forgery to interfere with the forwarding of data packets. Therefore, it becomes very important to solve the security problem of nodes in the opportunistic routing candidate set. At present, it is a feasible method to solve the routing security problem to evaluate the credibility of the candidate nodes in the route by establishing a trust model. The trust model effectively identifies and screens malicious nodes, enhances the security and stability of the network, and reduces network latency.
In existing methods, Li and Das [6] proposes a new trust model being applicable in opportunistic networks. In this algorithm, the concept of positive feedback message (PFM) is introduced in which once a node receives and forwards a packet to the next hop, the next hop node generates a PFM and sends it back to the previous hop in order to verify the behaviour of the current forwarder node. Salehi et al. [7] proposed a novel opportunistic routing protocol, which not only selects the next hop to send node based on the link transfer probability between nodes, but also based on the trust level calculated by the node to other nodes when communicating data packets. However, the above method only has a good effect on general malicious nodes, and it has no effect on the collusion behavior of nodes and cannot solve the collusion behavior of malicious nodes.

The most important contributions of this paper are as follows:

1) Using dynamic coefficients to balance direct trust and recommended trust, and a model based on node trust degree is proposed (Based on Trust model, BT model). Aiming at the problem of dishonest nodes and malicious recommendation in recommended nodes, a pruning filtering mechanism is used to filter nodes and recommended trust values. Comparing the established model with the SAT model [8] and the DCT model [9], the results show that with the increase of malicious nodes in the network, the trust mechanism established in this paper has better anti-interference ability.

2) Combine the node trust degree and ETX value, and compare with the four routing algorithms TBOR [10], RTOR [11], TAODV, and EXOR to conduct comparative experiments on throughput, end-to-end delay, and packet delivery rate. Simulation results show that combining the trust model and OR protocol proposed in this paper can significantly improve the communication performance in harsh environments.

The rest of the paper is organized as follows: Section 2 includes related work in the field of the OR protocol and a comprehensive description of the proposed trust model. Section 3 introduces the establishment of the trust model. Section 4 introduces the BTOR algorithm of integrated node trust and gives The algorithm flow is presented. Section 5 contains two comparative experiments, including the comparison between trust models and the comparison of different trusted routing protocols in an interference environment, finally, the full text is summarized.

II. RELATED WORK

A. TRUST MODEL

In vehicular ad hoc network, nodes are highly dependent on node-to-node communication, thereby allowing the spread of malicious communications generated by dishonest nodes. Due to the sensitivity of the information involved in VANET, a safe and reliable environment is essential [12]. However, because of its poor defense capabilities against various attacks (such as man-in-the-middle attacks, Sybil attacks, black hole attacks, or denial of service attacks), ensuring the security of VANET is very challenging [13], [14]. In recent years, the trust model has been introduced as an effective security technology to the trusted environment for message propagation between vehicles. Based on the previous work of researchers, trust models are roughly divided into three categories: data-oriented trust models (DTM), entity-oriented trust models (ETM) and hybrid Trust Models (HTM).

Data-oriented trust model (DTM) is defined as the trust of information received from various sources, which is established based on the evidence obtained from these sources [15], [16]. Bayes’ theorem, belief propagation, DempsterShafer theory and weighted voting are common techniques for establishing DTM [17], [18]. In DTM, trust in messages (data) is calculated based on opinions generated by adjacent vehicles or previous interactions between nodes. In recent years, researchers have proposed a variety of digital ground models. Gazdar et al. adopted a layer-based analysis method. Among them, the vehicle continuously evaluates the credibility of the received data based on its direct experience [19]. In this method, the trust of each participating vehicle is evaluated, and its main purpose is to identify high-trust vehicle nodes and dishonest vehicle nodes based on the exchanged data. Wu et al. proposed a centralized trust model framework for using the advantage evaluation data of adjacent infrastructure (road-side units) [20]. In this framework, the calculation of trust is based on two factors: 1) observation and 2) feedback. The vehicle detects an event and generates observations, as well as their confidence, depending on the distance from the event, its maximum message detection rate, and the number of embedded sensors that detect the event.

Unlike DTM, ETM evaluates the credibility of entities (vehicles). These vehicle management systems rely heavily on nearby vehicles. Khan et al. uses a cluster-based technology, in which the cluster head (Cluster Head, CH) is selected first, and is responsible for evaluating the trust degree in the network [21]. In this technical manual, CH implements a watchdog mechanism to provide reports on the existence of bad vehicles in vehicles in its vicinity. After detecting these vehicles, the CH notifies the trusted authority (TA) responsible for revoking these vehicles to maintain a trusted network. However, the main limitation of this method is the communication overhead generated by the exchange of messages with the CH, thereby reducing the efficiency of the entire network. Similarly, Yang uses a similar mining method for trust calculation [22], which calculates the similarity between the received messages based on the Euclidean distance and the reputation of the sending vehicle when receiving messages from the vehicle. Since trust is calculated locally using Euclidean distance, the TM cannot provide any global information about the similarity of messages.

HTM evaluates the credibility of vehicles and exchanges information. In other words, the trust of the data is calculated by using the trust of the vehicle. However, since many control messages must be evaluated within a limited time, HTM
will incur higher computational overhead. S. Ahmed et al. proposed a logic-based trust calculation for HTM method used to identify nodes injected with false information in the network [23]. In this trust model, when neighboring vehicles share messages, the model can identify the credibility of the event. Once the real event is identified, this information is used to determine whether the behavior of the sender node is legal or malicious, and to calculate trust through weighted voting and logistic trust functions. This trust model is effective in identifying dishonest vehicles that spread erroneous information. However, the main limitation of this trust model is its dependence on weighted voting. In the case of most dishonest vehicles, weighted voting may be biased. Another HTM was proposed by Shrestha et al. It calculates trust in vehicles in a fully distributed manner [24]. The trust calculation is divided into two steps, namely the evaluation of the trust of the vehicle, and the second step involves the trust calculation of the information. Trust is achieved by dividing honest and dishonest vehicles into two independent groups to identify the credibility of neighboring nodes. Next, an improved threshold random walk algorithm is used to evaluate the trustworthiness of received messages. The main disadvantage of this scheme is the assumption of an even distribution of dishonest nodes in the network.

The trust model proposed in this paper is a combination of direct and indirect trust [16]. Direct trust (DT) is based on a node’s own experience with another node, and indirect trust (IDT) is obtained from the recommendations of a given node. In order to objectively assess trust, this paper introduces a dynamic balance coefficient between direct trust and recommended trust, which changes with the direct interaction of the node in the network. IDT can be obtained from Beta Reputation or DST. However, if you trust to use a linear function, it may be more stable and less sensitive to bad behavior. Therefore, Y. Wang et al. [25] used the logistic function to update the trust of the service provider. The logistics function uses the historical and environmental attributes provided by the service. Use the expectation maximization algorithm to calculate the weight of the environmental attributes. Although the system generates fewer false alarms, it requires a lot of observations to obtain these weights, which makes the system slower. Trust is vulnerable to referral attacks, such as ballots. Both attacks can be handled by similar techniques. For example, the revised trust value is used to calculate the reputation of the node and the credibility of its recommendation [26], but it lacks a mechanism to identify false suggestions. Another node will pass the trust relationship to other nodes. Then, the receiving node calculates the similarity between its own opinions and the received suggestions.

In the trust model, correct trust measurement is the key to achieving objective and correct trust assessment. The DCT model is an application type and node permission level, a dynamic entity center trust model based on weights. The SAT model is a feedback-based dynamic trust calculation model that considers various factors that determine node trust, including satisfaction, Similarity, feedback credibility, recent trust, historical trust, etc. According to literature [27], there are some attributes related to trust establishment in the vehicular ad hoc networks, which serve as a trust metric in many existing trust models. The most commonly used metrics are the success rate of data forwarding, the frequency of interaction, and so on. The specific measurement factors will be described in detail in the third chapter.

B. OPPORTUNITY ROUTING

An important number of multihop communications strategies (ie, routing techniques) dedicated to urban VANETs have been proposed, which can be grouped into three categories [2]: (a) greedy-based routing, (b) delay tolerant-based routing, and (c) reactive-based routing. The opportunistic routing studied in this paper is one of the delay-tolerant routing. The first category is characterized by choosing the shortest routing path between a pair of communicating vehicles. Biswas and Morris [28] proposed the first and most famous OR protocol, namely EXOR. In EXOR, a node uses a metric called Expected Transmission Count (ETX) to select its candidate set. ETX calculates how many packets a sender node needs to send on a particular link until it is finally received by the corresponding receiver node. The candidate nodes in EXOR are selected from the neighbor group. These neighbors use ETX metrics to minimize the shortest path between the source node and the target node. Dubois-Ferriere et al. [29] introduced the LCOR protocol, which uses the best algorithm for candidate selection. According to LCOR, nodes use the expected arbitrary path transmission (EAX) metric proposed in [30]. Although EAX is designed to meet the requirements of the OR protocol, the calculation of EAX is computationally expensive, especially in large network scenarios. Rozner et al. introduced the OAPF algorithm [31]. In OAPF, first use the ETX index to generate a set of initial candidates, and then continue to force all potential nodes in the initial candidate set to select their own candidates first. Finally, the source node uses EAX metrics to complete the candidate selection. The node that provides the best EAX between the source and target will be added to the candidate set. Darehshoorzadeh et al. [32] proposed a heuristic-based candidate selection method based on the link quality between nodes, and compared this method with the EXOR protocol. Li et al. proposed CodePipe [33], which considers energy efficiency, reliability, and communication throughput in noisy networks.

In addition to considering only the link transfer probability between nodes to propose a metric and algorithm method for candidate selection, there is another type of algorithm in which the geographic location of nodes is considered. For example, Widmer et al. proposed the CBF algorithm [34]. In CBF, when a node decides to send a data packet to a specific destination, it not only locates itself, but also locates the geographic information of the target node into the data packet and propagates it. By receiving such packets, neighbor nodes will decide whether they should be rebroadcasted and when they should be propagated in them.
Each neighbor node can adjust its retransmission time according to the distance progress from itself to the destination node. In POR [35], a node selects its candidate set using the amount of progress through each neighboring node. The distance progress with the geographic distance between each neighbor and the destination can be calculated. In addition, when a candidate node receives a data packet and needs to determine its forwarding time, a timer-based coordination method will be applied between the candidate nodes. Compared with POR that only benefits from the geographic information of neighboring nodes for candidate selection, Darehshoorzadeh et al. proposed the DPOR protocol [36], which combines the location information of nodes with their link transmission probabilities. Introduce different candidate selection metrics. In fact, DPOR recommends selecting candidates from the neighbor list. These candidates not only provide acceptable distance progress, but also provide a reasonable link transfer probability.

The comparison routing protocol selected in this paper, in addition to the two classic routing protocols EXOR and TAODV, TBOR and RTOR are relatively new trusted opportunistic routing. Among them, TBOR routing algorithm is based on Bayesian network, introduces uncertainty factors, and calculates more realistic node credibility based on behavioral positive factors. RTOR routing algorithm not only considers the forwarding behavior of nodes, but also considers the quality of links between nodes. Three different next-hop selection indicators are introduced to enable nodes to use link quality, node’s geographic location, and their trust level to select their next-hop sending node more finely.

III. NODE TRUST DEGREE CALCULATION MODEL

In order to calculate the node trust value more accurately and objectively, the trust model designed in this paper consists of two parts: direct trust and indirect trust. Direct trust is obtained through direct interaction between nodes, and indirect trust is obtained through recommendation from neighboring nodes. In addition, filtering mechanisms are used in the process of indirect trust collection to eliminate false recommendations.

A. DIRECT TRUST

In the Internet of Vehicles environment, some of the main methods for malicious nodes to interfere with the network include: discarding data, forwarding false data, and intentionally delaying the forwarding time. Combined with the vehicle-to-vehicle communication environment, it is concluded that the determined behavior trust factors are mainly the following, where is the evaluation node and j is the node to be tested:

1) DATA FORWARDING INTEGRITY RATE (DI)
Forwarding data integrity rate is the most intuitive and direct way to describe the trust information of nodes. When the number of packet forwarding failures increases, the probability of the node being judged as a malicious node increases. For this kind of malicious behavior, the trust value of the node should be greatly reduced, so as to judge the malicious node from the intuitive change of the trust value:

\[
DI = \frac{IP_{ij}(t)}{DTP_{ij}(t)}
\]

IP is the number of data forwarded by the node, DTP is the number of packets sent by the node.

2) FORWARDING DATA FRESHNESS (DF)
In order to identify malicious nodes and improve the effectiveness of network data collection, it is necessary to analyze the temporal correlation of data content sent by nodes. If the evaluation subject i detects that the neighbor node repeatedly sends packets with the same content, it can be considered that it is performing replay attack:

\[
DF = \frac{NRP_{ij}(t)}{RP_{ij}(t) + NRP_{ij}(t)}
\]

RP is the number of packets with duplicate content, while NRP is the number of fresh packets.

3) FORWARD DATA consistency(DS)
In order to prevent malicious nodes from forging packets, it is necessary to analyze the spatial consistency of data sent by neighboring nodes. In the local network, the data between neighbor nodes is generally highly correlated. Subject i monitors the content of data packets sent by object j, and compares it with the data collected by itself. If the difference between the two is within a certain range, it can be considered that the evaluation subject and object have a consensus on the monitoring object:

\[
DS = \frac{CP_{ij}(t)}{CP_{ij}(t) + NCP_{ij}(t)}
\]

CP is the number of consistent packets and NCP is the number of inconsistent packets.

4) INTERACTION FREQUENCY (IF)
In social relations, acquaintances often have a high degree of trust. In the Internet of vehicles, only when two nodes enter each other’s transmission range (i.e. neighbors) in the process of moving, it is possible to establish a route and realize message forwarding. For nodes with more interaction history, they should be given a higher degree of trust. Therefore, the interaction frequency between nodes is an important standard to measure the level of trust:

\[
IF = \frac{NI_{ij}(t)}{ANI_{ij}(t)}
\]

NI is the number of interactions in t time period, and an is the average number of interactions in communication environment in this time period.
5) DELAY RATE OF DATA FORWARDING (FD)

Networks are often used in the field of traffic safety, and require high timeliness of information. Therefore, the timeliness of node processing information can be included in the category of trust. For a node with a smaller delay in forwarding packets, it can be considered that the trust is higher, otherwise the trust is lower:

\[
FD = \frac{TD_{ij}(t)}{ATD_{ij}(t)}
\]  

(5)

TD is the delay time of data forwarding in t time period, ATD is the average delay time in this time period. FD

In the evaluation process, the evaluation subject i calculates the evaluation value of each trust factor of the evaluation object j according to the record of direct communication with the evaluation object j, and then obtains the direct trust value of the evaluation object j. According to the listed trust factors, the direct trust value (DTV) of evaluation object j is:

\[
DTV_{ij} = W1 \times DI + W2 \times DF + W3 \times DS + W4 \times IF + W5 \times FD
\]  

(6)

Among them: w1, w2, w3, w4, w5 are weighting coefficients, and w1+w2+w3+w4+w5=1. In general, it is considered that these five behaviors have the same weight when evaluating the trust value of a node. In different environments, different weight distribution schemes can be allowed.

B. INDIRECT TRUST

The evaluation node i evaluates the evaluation trust value of the node j to be tested. Through the direct communication behavior record with node j, not only the direct trust value of each node can be calculated, but also the indirect trust value can be calculated by other nodes (k1, k2, . . . , kn) in the public communication range between node i and node j. However, in recommendation trust, there is no guarantee that the recommender is not malicious. And malicious nodes will produce collusive behaviors and gather together to conduct malicious behaviors. Therefore, before calculating indirect trust, we must first filter out malicious and false suggestions.

Inspired by the hierarchical pruning filtering mechanism, this paper makes such a filtering mechanism, which embodies the process of filtering useful information in accepting recommendation trust. This mechanism is a hierarchical structure. First, it eliminates the \( DTV_{kn} \) with low trust, and then selects recommendation information by considering various factors. For the recommender with high trust, if its recommendation deviates from the direct trust, it is also acceptable within a certain deviation range, for the recommender with low trust, the corresponding acceptable deviation range will be smaller.

Among these m indirect trust values, there may be false indirect trust values recommended by malicious nodes. Therefore, comprehensive analysis of indirect trust is needed to filter out abnormal results.

The process of trust value filtering is as follows:

1) prune according to the trust level of the subject to the recommender. In the case of \{k1, k2, . . . , kn\} classification, firstly, we prune the lower trust set on the left and directly exclude the recommendation of the lower trust set.

For how to define which trust is lower. The root mean square value (RMS) is used here, that is:

\[
DTV_{knj}(rms) = \sqrt{\frac{\sum_{t=1}^{n} DTV^2_{kj}}{n}}
\]  

(7)

If \( DTV_{knj} DTV_{knj}(rms) \) then the trust degree given by the node to the evaluation object j will be considered as untrusted and deleted.

2) prune according to the recommended trust level. Each recommender k has different direct trust value to object j. According to these trust values, different ranges are divided. Those within the range are accepted and those outside the range are excluded.

The following definitions are made here:

**Definition 1 (Deviation):** \( DTV_{ink} \) is the direct trust degree of subject i to recommender k, and \( DTV_{knj} \) is the recommendation trust degree of recommender k to object j and k needs to recommend this trust degree to subject i, so the deviation is:

\[
d = |DTV_{ij} - DTV_{knj}|
\]  

(8)

Obviously, the more biased the recommendation, the less acceptable it is.

**Definition 2 (Degree of Understanding):** The best understanding degree of the recommended entity \( k_n \) to object j is:

\[
u = n \sum e^{-\lambda(t-t_0)}
\]  

(9)

where n represents the number of interactions required to fully understand the object in a period of time, n represents the actual number of successful interactions, \( \lambda \) is the coefficient, and to represents the starting point of time. The degree of understanding increases with the number of successful interactions, decreases with time, and \( u \in [0, 1] \).

**Definition 3 (Recommended Conflict):** The average difference between the recommendation entity \( k_n \) and other recommenders \( k_m \) is:

\[
c = \frac{\sum |DTV_{kn} - DTV_{km}|}{n}
\]  

(10)

The larger C is, the more k recommendation deviates from the majority of recommenders, the less acceptable it is.

It can be seen that the higher the similarity between the subject and the recommender, the larger the tolerance range for the deviation of recommender, and vice versa. Namely:

\[
DTV_{knj} \in [DTV_{ij} - \varepsilon, DTV_{ij} + \varepsilon]
\]  

(11)
where $\varepsilon = u - d - c$. If $DTV_{kaj}$ is in this range, it will be accepted by subject $i$, otherwise it will be excluded. The recommendation trust degree assumes that after hierarchical pruning filtering, there are $m$ recommendation nodes remaining, the recommendation trust degree is:

$$ITV_{ij} = \frac{1}{m} \sum_{a=1}^{m} DTV_{ika} \times DTV_{kaj} \quad (12)$$

### C. COMPREHENSIVE TRUST VALUE

The direct trust value $DTV_{ij}$ of the evaluation subject $i$ to the evaluation object $j$, and the indirect trust value $ITV_{ij}^{1}, ITV_{ij}^{2}, \ldots, ITV_{ij}^{m}$ obtained from the neighboring nodes $k1, k2, \ldots, km$, get the whole assessment process.

Thus, the final trust value of subject $i$ to object $j$ is obtained:

$$FTV_{ij} = \frac{1}{1 + \beta(j)} \times DTV_{ij} + \frac{\beta(j)}{1 + \beta(j)} \times ITV_{ij} \quad (13)$$

When calculating the final trust value, we need to combine the direct trust value with the indirect trust value, so how much weight each of them takes is a question. Here, the final trust calculation method is an adaptive method. The determination of classification weight is completely calculated automatically by the system according to the established mathematical model, and it does not need to be determined artificially. Here, the calculation formula of the function $\beta(j)$ defined in this model is as follows:

$$\beta(j) = \frac{1}{2} \left[ \phi(L_j) + \phi(n_{total}) \right] \quad (14)$$

where $\phi(x) = 1 - \frac{1}{x + \pi}$. $L_j$ is the number of respondents, and $n_{total}$ is the number of all entities that interact with $j$ monitored by the system. The adjustment function $\phi(x)$ is an arbitrary constant greater than $\theta$, and is used to control the speed at which $\phi(x)$ tends to 1. The larger the value of $\theta$, the faster the speed at which $\phi(x)$ approaches 1.

It can be seen from the formula that the entity activity $\beta(j)$ is determined by two variables, $L_j$ and $n_{total}$. The more other entities are traded with the entity, the greater the value of $\beta(j)$. At the same time, the more the number of feedbacks, the greater the value of $\beta(j)$. The number of variables $L_j$ and $n_{total}$ really reflects the activity of the node in the network environment.

### IV. BTOR ROUTING ALGORITHM

Using the broadcast characteristics of wireless media, opportunistic routing can select multiple nodes as forwarding relay nodes, so that there will be more forwarding and acceptance opportunities, as shown in Figure 1: Suppose the forwarding success rate from the source node $i$ to each intermediate node is 50%, and the forwarding success rate from each intermediate node to the destination node $j$ is 100%. If the traditional routing method is used, the source node will select only one node from the four intermediate nodes as the next hop sending node. At this time, the success rate of forwarding from the source node to the destination node is only 50%. If you use the opportunistic routing forwarding method, establish a forwarding candidate set, and use 4 relay nodes as candidate forwarding nodes at the same time. As long as one of the packets received from the source node can be forwarded to the destination node, the forwarding rate can be increased to $(1 - (1 - 0.5)^4) \times 100\% \approx 93.75\%$, the forwarding rate has increased from 50% to 93.75%, thus significantly improving the end-to-end throughput and success rate of forwarding.

However, since malicious nodes may enter the candidate set to interfere with forwarding, a trust model is introduced to filter and screen malicious nodes. The basic idea of the BTOR routing algorithm is: first calculate the expected transmission times $ETX$ value of the neighbor node within the communication range of the sending node. If the $ETX$ value from the neighbor node to the destination node is less than the $ETX$ value from the source node to the destination node, then the node enters the initialization candidate set. Then use the node trust model proposed in the previous section to calculate the trustworthiness of each node in the initialization candidate set, filter the candidate set nodes with lower trust according to the set trust threshold, and then use the $ETX$ value to the candidate set The nodes prioritize, finally, the data packets are forwarded according to the coordinated forwarding mechanism.

Based on the node trust calculation method proposed in the previous section, the node trust value is increased on the basis of the classic $ETX$ value. The original $ETX$ value is weighted to a comprehensive trust degree $T$ to obtain the weighted expected transmission number $TETX$:

$$TETX = ETX \times \frac{1}{T} \quad (15)$$

According to the calculated $TETX$ value, the packet forwarding strategy is executed. The flow chart of the entire algorithm, as shown in Figure 2: In this algorithm, identify malicious nodes through the trust model, exclude malicious nodes from the candidate set, help each level of forwarding nodes determine a reliable forwarding set, and re-sort the
candidate node trust values in the candidate set, the main process is 4 steps, as shown in Figure 3:

1) The initial candidate forwarding set selection is performed for n nodes within the communication range of the starting node i, and ETX (i,j) and ETX (kn,j) are calculated. If \( \text{ETX}(kn,j) < \text{ETX}(i,j) \), the node is added to the initialization candidate set, otherwise, the node is not added to the initialization candidate set.

2) According to the node trust calculation method proposed in the previous section, calculate the trust value of all nodes in the initial candidate set. After obtaining the trust values of all candidate nodes, the set trust threshold \( \alpha \) is used to filter the potential malicious nodes in the initial candidate set, and add the nodes that meet the conditions to the candidate forwarding set.

3) Weighting the original ETX value to synthesize the node trust degree \( T_n \), the new expected transmission number \( \text{TETX} \) after weighting the feedback credibility is obtained.

4) The filtered candidate nodes are sorted in ascending order according to the \( \text{TETX} \) value to obtain the final candidate set.

The specific algorithm is shown in Algorithm 1:

V. EXPERIMENT SIMULATION AND RESULT ANALYSIS
This part is mainly divided into two parts, namely the comparison experiment between trust models and the comparison experiment between routing algorithms. First, the performance of the BT model under different confrontation strategies was evaluated, and compared with the other two models, its effectiveness was demonstrated. The second is the comparison between routing performance. The following three metrics are used when comparing network performance: end-to-end delay, throughput, and packet delivery rate. The end-to-end delay is the time required for data packets to be transmitted from the source to the receiver, throughput is the total amount of data packets received and sent by network nodes in a unit time, the packet delivery rate
Algorithm 1 BT Model

| Input: ETX(i,u) from sending node to destination node and ETX(j,u) from all neighbor nodes of the sending node to the destination node.
|  
| 1 Begin CandSet(i,u) ← ∅,
| 2 If s == d then,
| 3 Return,
| 4 End if,
| 5 IniCandSet(i,u) ← ∅,
| 6 For all j ∈ NeighborSet(i) do,
| 7 If ETX(j,u) < TX(i,u) then,
| 8 IniCandSet(i,u) ← CandSet(i,u) ∪ j,
| 9 End if,
| 10 End for,
| 11 For all j ∈ IniCandSet(i,u) do,
| 12 Calculating DTV and ITV by formula (1)-(12),
| 13 Calculating FTV by formula (13)-(14),
| 14 If (0 ≤ FTV ≤ α) then,
| 15 Node j cant enter the CandSet(i,u),
| 16 Else if (α < FTV < 1) then,
| 17 CandSet(i,u) ← CandSet(i,u) ∪ j,
| 18 End if,
| 19 End if,
| 20 For all j ∈ IniCandSet(i,u) do,
| 21 Calculating TETX by formula (15),
| 22 End for,
| 23 Sort CandSet(i,u) by TETX from j to u,
| 24 Return formula,
| 25 End,
| Output: CandSet(i,u) of the sending node’s next hop candidate forwarding set.

is the ratio between the total data packets received by the receiving node and the total data packets generated by the sending node.

A. SIMULATION EXPERIMENT ENVIRONMENT AND PARAMETER SETTING

In order to verify the validity of the trust model and the BTOR routing algorithm proposed in this paper, MATLAB was used for simulation experiments. The simulation environment was set as follows: 100 point nodes were distributed in the area of 1000m × 1000m. In addition, in order to evaluate the effectiveness of the trust model, the proposed model should be evaluated when there are malicious nodes in the network, so the number of malicious nodes is set between 0 and 40. Other specific parameters are shown in Table 1:

B. COMPARATIVE EXPERIMENT BETWEEN TRUST MODELS

Before conducting trust model comparison experiments, it is necessary to discuss the behavior of malicious nodes and the weight distribution scheme.

1) BEHAVIOR MODE OF MODES AND DEFINITION OF MALICIOUS NODES

The behavior patterns of trusted nodes are easy to simulate because they provide good service and honest feedback. However, it is a challenge to realistically simulate the malicious behavior of nodes. Malicious nodes deceive and provide erroneous feedback to other agents during data transmission, that is, they have a low evaluation of good agents, and a high evaluation of malicious agents, and perform a large number of false transactions to improve their own ratings while denigrating others. Excellent nodes to help each other definitely.

Taking data forwarding integrity, data freshness, and consistency as examples, the initial values are all selected to test the trust value of trusted nodes and malicious nodes:

1) The malicious node forwards the data packet, but will randomly discard some data during the forwarding process.
2) The malicious node repeatedly sends the same data packet in several interactions, which is intended to consume network resources.
3) The malicious node modifies the received data packet before sending it out, which affects the authenticity of the data.

The trust value generated by the three malicious behaviors is shown in Figure 4:

It can be found through experiments that the trust value generated by each trust factor of trusted nodes fluctuates between 0.85 and 0.95, while the trust value generated by malicious nodes fluctuates between 0.15 and 0.25.

2) RELATED PARAMETERS

In this experiment, three parameters are used to simulate the behavior of malicious nodes, namely: per (percentage of malicious nodes),res (percentage of malicious agents giving erroneous feedback in this interaction) and collusion (percentage of malicious agents forming a conspiracy group). Malicious agents form a conspiracy group to improve their rating by executing a large number of false transactions, while discrediting other excellent agents. To help each other definitely.

is the ratio between the total data packets received by the receiving node and the total data packets generated by the sending node.

TABLE 1. Simulation parameters.

| parameter             | value         |
|-----------------------|---------------|
| Network topology area | 1000m × 1000m  |
| Number of nodes       | 100           |
| Node communication radius | 250m         |
| MAC Protocol          | IEEE 802.11p  |
| Network Protocol      | IEEE 1609.4   |
| Number of malicious nodes | 0-40      |
| Number of interactions| 50            |
| Send data packet rate | 50/packet/s   |
| Packet size           | 10-512 byte   |
| Low confidence threshold| 0.5          |
| High confidence threshold| 0.8          |
| Simulation time       | 600s          |
In this way, the evaluator (roadside infrastructure) becomes more experienced with the nodes in his network environment, it is more inclined to rely on his own judgment, and the direct trust value is slightly higher than the indirect trust value. However, in the DCT model, the value of the direct trust value is directly used as its own weight. This means that when the direct trust value is higher, the weight is higher, and when the direct trust value is lower, the weight of the indirect trust value is higher. The advantage of this is that on the one hand, direct trust has been given enough attention, and on the other hand, when nodes have doubts or direct trust is not high enough, recommendations are used to estimate entity trust.

Similar to most models, this model calculates the weights of direct trust value and indirect trust value according to the activity of nodes in the environment. Through formula (14), it can be seen that when the value of \( \theta \) is too large, the size of \( X \) will be meaningless, so only the case of small value of \( \theta \) is discussed here. When there are many interaction times and interaction nodes for trusted nodes, \( \phi(L_j) \) and \( \phi(n_{total}) \) approach to 1, then \( \beta(j) \) also approach to 1, so the weights of direct trust and indirect trust tend to be equal, each of which is 0.5. For malicious nodes, it has a low number of interactions with other nodes in the network. In this case, formula (14) can calculate that the weight of direct trust value is slightly greater than that of indirect trust value, which is the opposite of the calculation concept of DCT model weight.

In this section, the same model will be used to compare different weight distribution experiments. According to the previous analysis, four different \( \theta \) values will be used for experiments. Option 1 uses most weight calculation methods, and \( \theta \) values are 0.5 and 0.6 respectively. Second, the DCT model weight calculation method is adopted, and \( \theta \) values are 0.8 and 0.2, respectively. The experimental results are shown in Figures 6:

It can be seen from Figures 6 that the distribution of weight values affects the fluctuation of the measured trust value. Compared with Option 2, Option 1 has a smaller fluctuation range, which indicates that the weight of the direct trust value and indirect trust value is greater. The closer the value, the smaller the fluctuation range of the overall trust value, because the greater the weight of a single trust value, the greater the fluctuation of the overall trust value. Such a weight allocation scheme will be explained in detail in the model comparison experiment in the next section.

4) TRUST MODEL COMPARISON EXPERIMENT
Use Matlab to establish simulation environment and experiment. First, the same vehicle node is compared with the SAT model and the DCT model under the same conditions. These three models use different trust factors, so it is difficult to compare directly when comparing, but they all use a combination of direct trust and indirect trust. Therefore, in the comparison test, the different trust factors are ignored, and the node’s trust value is directly assigned. Through previous experiments, it can be determined that the trust value of the

---

**FIGURE 4.** Comparison of behavior trust values between trusted nodes and malicious nodes.

**FIGURE 5.** The effect of res on the trust value under different collusions.

---

3) ALLOCATION OF WEIGHTS
When combining the direct trust value and the indirect trust value, the weight distribution of the two has been the topic of many trust models. Most trust models believe that the greater the number and number of times a node interacts with other nodes in a network environment, the higher its active value.
FIGURE 6. The influence of different weight allocation schemes on trusted nodes (a), malicious nodes (b).

FIGURE 7. Comparison of three models for measuring trust value of trusted nodes (a), malicious nodes (b).

roadside infrastructure to trusted nodes after the interaction fluctuates between 0.85 and 0.95, and the trust value generated by malicious nodes fluctuates between 0.15 and 0.25. Interaction between nodes will also generate trust values. Trusted nodes will send real trust values to RSU, while malicious nodes will send false trust values. The false setting here is opposite to the real trust value.

There are 100 recommended nodes in the network environment, where pes = 20%, res = 40%, and collusion = 100%. After a total of 50 interactions, the trust value fluctuation graphs of the three models for trusted nodes and malicious nodes are shown in Figures 7. It can be seen from Figures 7 that the trust values measured by DCT model fluctuate greatly compared with the other two models, while the trust values of the other two models are relatively stable. The initial trust value of SAT model is low because the author added the historical trust value to the design of the model. So there is a process similar to understanding when the model calculates the trust value of nodes, so the trust value measured by sat model will slowly rise, and then gradually tend to be stable. However, when the number of interactions is small, because the understanding process of SAT is slow, it may cause the base station to misjudge the vehicle node.

The BT model designed in this paper can accurately calculate the trust value of this point when the number of interactions is small, and with the increase of the number of interactions, it tends to be stable, and the decay rate of indirect trust value is higher and higher as time goes on. This model has already added such attenuation function to the trust model of this model.

The next experiment is to test the anti-interference of the model. The main content is that the number of malicious recommendation nodes in the network increases gradually, so as to compare the trust values calculated by the three models. In the setting of this model, the malicious node has camouflage, that is, the malicious node will show reliability in the interaction with RSU, but it will send false trust value of the node to be tested to RSU. In the experiment, the number of malicious nodes increases gradually. In the network
environment set by this model, PES increases gradually, but it will not exceed 40%. Next, res = 40%, aggregation = 100%. The experimental results are shown in Figure 8:

![Figure 8: The impact of the increase of malicious recommendation nodes on the model.](image)

It can be seen from Figure 8 that with the increase of the number of malicious recommendation nodes in the network, the trust value calculated by trusted nodes, sat model and DCT model receives malicious recommendation, so the trust value will decrease, however, the BT model designed in this paper adopts pruning filtering mechanism, which can eliminate the malicious recommendation of dishonest nodes, so that the trust value calculated by the base station to the tested nodes starts Finally, it is more accurate and stable.

In the next experiment, we will consider that the collusion rate of malicious nodes will increase gradually, that is, malicious nodes will form a group to interfere with the network together. In the environment set in this experiment, pes = 40%, res = 40%, and collusion will increase from 0 to 100%. The experimental results are shown in Figure 9:

![Figure 9: The impact of increasing collusion rate of malicious nodes on the model.](image)

It can be seen from Figure 9 that with the increase of the collusion rate of malicious nodes, the other two models can not actively resist the collective interference of malicious nodes to the network. Because there are many malicious nodes with the same behavior, the malicious evaluation of trusted nodes can not be made even if other trusted nodes based on the correct suggestions of the base station for the node. The filtering mechanism adopted in this model can resist the clustering behavior of malicious nodes.

5) DYNAMIC CHANGE OF NODE IDENTITY

The purpose of this experiment is to show how the BT model can deal with the dynamic changes of node behavior. In the last experiment, we have shown the effectiveness of BT model to filter the false feedback submitted by malicious nodes, so in this experiment, we focus on the alternate behavior of malicious nodes. In this experiment, we simulate a malicious agent to establish its reputation first, then consume the established reputation, and finally try to re-establish its reputation mode, that is, the agent swings between establishing and consuming reputation.

In this experiment, there are 100 nodes, including PES = 20%, res = 40%, and aggregation = 100%. A total of 100 interactions are performed, which are divided into three consecutive timeslots. Figure 10 shows the trust value calculated by the node when the behavior mode is changed:

![Figure 10: Effectiveness against node changes.](image)

It can be seen from figure 10 that BT model can quickly reflect the sudden decline of node reputation, which is shown by the sharp decline of the curve. Once the trust value is reduced to zero, a large number of continuous good services are needed to make its trust value rise again, that is to say, its cooperative nature must be proved. From the figure, we can see that although the malicious agent has good behavior in the later interaction, its trust value rises very slowly, even in this case, it will not rise to the original trust value. Therefore, the cost of rebuilding reputation is actually higher than the cost of extracting reputation. The model successfully
integrates the principle of “fast decline and slow rise of trust value.

C. BTOR ROUTING ALGORITHM COMPARISON EXPERIMENT

1) CANDIDATE SET NODE
The main function of the trust model is to prevent malicious nodes from entering the candidate set. Figure 11 shows the change of the average expected transmission times with the number of candidate set nodes:

As shown in Figure 11, in the case of the same number of nodes, the expected transmission times of BTOR, TBOR, and RTOR are significantly lower than EXOR, because these three routing algorithms all use different algorithm trust models. In the EXOR algorithm, malicious nodes join the candidate set, resulting in a high expected transmission times.

The number of malicious nodes in the candidate set directly affects the performance of the routing algorithm. With the increase in the number of malicious nodes in the network environment, malicious nodes will more easily enter each level of forwarding candidate sets through deception such as collusion. Figure 12 shows the changes in the number of malicious nodes in different candidate routing algorithm candidate sets as the number of malicious nodes in the network increases:

Figure 12 verifies the influence of the number of malicious nodes on the trusted forwarding set. When the number of malicious nodes increases, the nodes in the forwarding candidate set of BTOR, TBOR, and RTOR gradually decrease, proving that these three improved routing algorithms isolate malicious nodes from the candidate set, while the EXOR routing forwarding candidate set has malicious nodes and cannot detected. At this time, although the number of nodes in the forwarding candidate set in BTOR, TBOR, and RTOR is small, most of them are trusted forwarding nodes.

2) END-TO-END AVERAGE DELAY
Figure 13 shows the comparison of the average end-to-end delay of the five routing algorithms in the case of increasing malicious nodes. With the increasing number of malicious nodes, the end-to-end average delay of the five routing algorithms is constantly increasing. This is because more malicious nodes in the network delay the transmission and discard the packets sent by the source node. The upper-layer transmission node has to re-establish the connection and retransmit the data packet to the destination, so the delay increases. Therefore, those algorithms that can detect and eliminate malicious nodes can quickly deliver data packets to the destination, resulting in lower end-to-end latency:

As can be seen from Figure 13, the three routing algorithms BTOR, TBOR, and RTOR increase the calculation of the node’s credibility, resulting in additional algorithm consumption. Although the TAODV routing algorithm also discriminates against malicious attacks, the calculation of TAODV is not as complicated as the calculation of the above three routing algorithms, so its additional consumption is less.
than the above three routing algorithms. The EXOR routing algorithm does not calculate credibility, so it cannot respond to the interference of malicious nodes. As the number of malicious nodes increases, its communication process will be seriously affected.

3) THROUGHPUT
Figure 14 shows the comparison of the throughput of the five routing algorithms with the increase of malicious nodes. Because malicious behaviors such as malicious packet loss and selective forwarding of malicious nodes continue to rise, resulting in the loss of a large number of data packets, the throughput is decreasing:

As can be seen from Figure 14, with the increase of malicious nodes, the EXOR routing algorithm uses a trust mechanism to prevent malicious nodes from entering the candidate set. Due to the interference of malicious nodes, its throughput declines the fastest, and the BTOR routing algorithm proposed in this paper has a significant advantage in detecting malicious nodes compared to TBOR and RTOR routing algorithms. The filtering mechanism can eliminate more malicious candidates in the forwarding node set. So choose a more reliable node to reduce the packet loss rate, it can reduce the impact of malicious nodes on network throughput. The TAODV routing algorithm has preventive measures against malicious attacks, but the effect is not obvious.

4) PACKET DELIVERY RATE
The packet delivery rate is an important indicator of network performance. The higher the delivery rate, the more secure the network and the better the performance. However, the behavior of malicious nodes in the network, such as packet loss and incomplete transmission, will seriously affect the delivery rate. With the increase of malicious nodes, the change in packet delivery rate of the five routing algorithms is shown in Figure 15:

As can be seen from Figure 15, as the number of malicious nodes increases, the packet delivery rate of the EXOR routing algorithm drops significantly. When the number of malicious nodes increases to 40, the packet delivery rate of the EXOR routing algorithm decreases to a low value. However, the packet delivery rate of the BTOR routing algorithm has not decreased significantly, because it can identify malicious nodes and isolate them from the candidate set, thereby reducing malicious behaviors such as packet loss by nodes in the network.

VI. CONCLUSION
This paper analyzes the attack behavior of malicious nodes in the vehicle ad hoc network environment. Combining with the current research on trust mechanism, a trust model based on node behavior is proposed. Experiments show that the model can still accurately and quickly measure the trust value of the node under test when there are many malicious nodes in the network. By combining the trust model and opportunistic routing, a trusted opportunistic routing forwarding model based on node trust is proposed. Through simulation experiments, compared with the remaining four protocols, the method proposed in this paper can better resist the malicious behavior of malicious nodes entering the candidate set and interfering with the network. In future research, we will consider introducing a watchdog mechanism, using the priority of candidates in the candidate set to accurately monitor the behavior of each candidate node in the candidate set.

REFERENCES
[1] L. Yanbing, S. Xiuli, and X. Yonggang, “Authentication mechanism and trust model of IoV,” J. Beijing Univ. Posts Telecommun., no. 3, pp. 5–22, 2017.
[2] N. Chaib, O. S. Oubbati, M. L. Bensaad, A. Lakas, P. Lorenz, and A. Jamalipour, “BRT: Bus-based routing technique in urban vehicular networks,” IEEE Trans. Intell. Transp. Syst., early access, Sep. 12, 2019, doi: 10.1109/TITS.2019.2938871.
[3] C. Tripp-Barba, A. Zaldivar-Colado, L. Uquiza-Aguir, and J. A. Aguilar-Calderón, “Survey on routing protocols for vehicular ad hoc networks based on multimetrics,” Electronics, vol. 8, no. 10, p. 1177, Oct. 2019.
[4] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, “Routing protocols in ad hoc networks: A survey,” Comput. Netw., vol. 55, no. 13, pp. 3032–3080, 2011.
[5] C. J. Hsu and H. I. W. K. Liu & Seah, “Opportunistic routing—A review and the challenges ahead,” Comput. Netw., vol. 55, no. 15, pp. 3592–3603, 2011.

[6] N. Li and S. K. Das, “A trust-based framework for data forwarding in opportunistic networks,” Ad Hoc Netw., vol. 11, no. 4, pp. 1497–1509, Jun. 2013.

[7] M. Salehi and A. Boukerche, “Trust-aware opportunistic routing protocol for wireless networks,” in Proc. 10th ACM Symp. QoS Secur. Wireless Mobile Netw. (Q2SWinet), New York, NY, USA, 2014, pp. 79–86.

[8] A. Das and M. M. Islam, “SecureTrust: A dynamic trust computation model for secured communication in multiagent systems,” IEEE Trans. Dependable Secure Comput., vol. 9, no. 2, pp. 261–274, Mar. 2012.

[9] X. Yao, X. Zhang, H. Ning, and P. Li, “Using trust model to ensure reliable data acquisition in VANETS,” Ad Hoc Netw., vol. 55, pp. 107–118, Feb. 2017.

[10] D. Keyi and S. Fanjun, “Opportunistic routing algorithm based on node trust degree in WSNs,” Softw. Guide, vol. 18, no. 6, pp. 70–74, 2019.

[11] M. Salehi, A. Boukerche, A. Darehshoorzadeh, and A. Mammi, “Towards a novel trust-based opportunistic routing protocol for wireless networks,” Wireless Netw., vol. 22, no. 3, pp. 927–943, Apr. 2016.

[12] F. Ahmad and A. Adnane, “A novel context-based risk assessment approach in vehicular networks,” in Proc. 30th Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA), Mar. 2016, pp. 466–474.

[13] Q. Ali, N. Ahmad, A. Malik, G. Ali, and W. Rehman, “Issues, challenges, and research opportunities in intelligent transport system for security and privacy,” Appl. Sci., vol. 8, no. 10, p. 1964, Oct. 2018.

[14] S. Ahmed, M. U. Rehman, A. Ishfaq, S. Khan, A. Ali, and S. Begum, “VANSec: Attack-resistant VANET security algorithm in terms of trust computation error and normalised routing overhead,” J. Sensors, vol. 2018, Jul. 2018, Art. no. 6576841, doi: 10.1155/2018/6576841.

[15] S. Ruj, M. A. Cavenagni, Z. Huang, A. Nayak, and I. Stoimenov, “On data-centric misbehavior detection in VANEETS,” in Proc. IEEE Veh. Technol. Conf. (VTC Fall), Sep. 2011, pp. 1–5.

[16] F. Ahmad, A. Adnane, F. Kurugolu, and R. Hussain, “A comparative analysis of trust models for safety applications in IoT-enabled vehicular networks,” in Proc. Wireless Days (WD), Apr. 2019, pp. 1–8.

[17] Y. A. Kim and M. A. Ahmad, “Trust, distrust and lack of confidence of users in online social media-sharing communities,” Knowl.-Based Syst., vol. 37, pp. 438–450, Jan. 2013.

[18] M. A. Azad, S. Bag, S. Parkinson, and F. Hao, “TrustVote: Privacy-preserving node ranking in vehicular networks,” IEEE Internet Things J., vol. 6, no. 4, pp. 5878–5891, Aug. 2019.

[19] T. Gardar, A. Belgith, and H. Abutair, “An enhanced distributed trust computing protocol for VANETS,” IEEE Access, vol. 6, pp. 380–392, 2017.

[20] A. Wu, J. Ma, and S. Zhang, “RATE: A RSU-aided scheme for data-centric trust establishment in VANEETS,” in Proc. 7th Int. Conf. Wireless Commun., Netw. Mobile Comput., Sep. 2011, pp. 1–6.

[21] U. Khan, S. Agrawal, and S. Silakari, “Detection of malicious nodes (DMN) in vehicular ad-hoc networks,” in Proc. Int. Conf. Inf. Commun. Technol. (ICICT), Dec. 2014, pp. 965–972.

[22] N. Yang, “A similarity based trust and reputation management framework for VANETS,” Int. J. Future Gener. Commun. Netw., vol. 6, no. 2, pp. 25–34, 2013.

[23] S. Ahmed and K. Tepe, “Using logistic trust for event learning and misbehaviour detection,” in Proc. IEEE 84th Veh. Technol. Conf. (VTC-Fall), Sep. 2016, pp. 1–5.

[24] R. Shrestha and S. Y. Nam, “Trustworthy event-information dissemination in vehicular ad hoc networks,” Mobile Inf. Syst., vol. 2017, Nov. 2017, Art. no. 9050787, doi: 10.1155/2017/9050787.

[25] Y. Wang, Y.-C. Lu, I.-R. Chen, J.-H. Cho, A. Swami, and C.-T. Lu, “Logit-Trust: A logit regression-based trust model for mobile ad hoc networks,” Dept. Comput. Sci., Army Res. Lab., Virginia Tech, Blacksburg, VA, USA, Tech. Rep., 2015.

[26] H. El Sayed, S. Zeadally, and D. Pathal, “Design and evaluation of a novel hierarchical trust assessment approach for vehicular networks,” Veh. Commun., vol. 24, Aug. 2020, Art. no. 100227.

[27] Q. Li, A. Malip, K. M. Martin, S.-L. Ng, and J. Zhang, “A reputation-based announcement scheme for VANETS,” IEEE Trans. Veh. Technol., vol. 61, no. 9, pp. 4095–4108, Nov. 2012.

[28] S. Biswas and R. Morris, “Opportunistic routing in multi-hop wireless networks,” ACM SIGCOMM Comput. Commun. Rev., vol. 34, no. 1, pp. 69–74, 2004.

[29] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, “Valuable detours: Least-cost anypath routing,” IEEE/ACM Trans. Netw., vol. 19, no. 2, pp. 333–346, Apr. 2011.

[30] Z. Zhong, J. Wang, S. Nelahuditi, and G. H. Lu, “On selection of candidates for opportunistic anypath forwarding,” ACM SIGMOBILE Mobile Comput. Commun. Rev., vol. 10, no. 4, pp. 1–2, 2006.

[31] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu, “SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks,” IEEE Trans. Mobile Comput., vol. 8, no. 12, pp. 1622–1635, Dec. 2009.

[32] A. Darehshoorzadeh and A. Boukerche, “An efficient heuristic candidate selection algorithm for opportunistic routing in wireless multihop networks,” in Proc. IEEE Symp. Comput. Commun. (ISCC), Jun. 2014, pp. 1–6.

[33] P. Li, S. Guo, S. Yu, and A. V. Vasilakos, “Reliable multicast with pipelined network coding using opportunistic feeding and routing,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 12, pp. 3262–3273, Dec. 2014.

[34] H. Füller, J. Widmer, M. Käsmann, M. Maurer, and H. Hartenstein, “Contention-based forwarding for mobile ad hoc networks,” Ad Hoc Netw., vol. 1, no. 4, pp. 351–369, Nov. 2003.

[35] S. Yang, F. Zhong, C. K. Yeo, B. S. Lee, and J. Boleng, “Position based opportunistic routing for robust data delivery in MANETS,” in Proc. IEEE Global Telecommun. Conf. (GLOBECOM), Nov. 2009, pp. 1–6.

[36] A. Darehshoorzadeh and L. Cerda-Alabern, “Distance progress based opportunistic routing for wireless mesh networks,” in Proc. 8th Int. Wirel. Less Commun. Mobile Comput. Mobile Inf. Comput. (IWCMC), Aug. 2012, pp. 179–184.

**BING SU** received the B.S. and Ph.D. degrees from the Nanjing University of Aeronautics and Astronautics (NUAA), China. He is currently an Associate Professor with the Department of Computer Science, School of Information and Mathematics, Changzhou University. His current research interests include network security, wireless sensor networks, the Internet of Things, routing protocols, and cloudy computing.

**CHENHUI DU** was born in September 1994. He received the degree in computer information engineering from the School of Information and Mathematics, Changzhou University, in 2018, where he is currently pursuing the master’s degree with the Computer Science Department. His current research interests include network security, wireless sensor networks, and trusted computing.

**JUAN HUAN** received the Ph.D. degree in agricultural electrification and automation from Jiangsu University, in 2019. She is currently an Associate Professor with the School of Information Science and Engineering, Changzhou University. Her research interest is information intelligence processing.