Received November 13, 2021, accepted December 19, 2021, date of publication January 31, 2022,
date of current version March 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3146579

Community Clustering Routing Algorithm Based on Information Entropy in Mobile Opportunity Network

QINGHUA LI[1], LIMIN ZHANG[1], FENG ZENG[2], (Member, IEEE), YONG PAN[3], AND JUNJIE YANG[1]

[1]School of Information Engineering, Lingnan Normal University, Zhanjiang 524048, China
[2]School of Computer Science and Engineering, Central South University, Changsha 410083, China
[3]School of Tourism Data, Guilin Tourism University, Guilin 541006, China

Corresponding authors: Limin Zhang (zhanglm@lingnan.edu.cn) and Feng Zeng (fengzeng@csu.edu.cn)

This work was supported in part by the National Social Science Fund of China under Grant 18BMZ057, in part by the Nature Science Foundation of Hunan Province of China under Grant 2020JJ4756, and in part by the Project of Educational Commission of Guangdong Province of China under Grant 2021ZDZX4055.

ABSTRACT Aiming at the difficulty of traditional routing clustering algorithms to deal with the different characteristics between communities and the inefficient nodes after community clustering, this paper proposes a Community clustering Routing protocol based on information Entropy in mobile opportunity Networks(CREN). The proposed protocol uses the K-Modes algorithm with unsupervised learning, combined with the pre-selected initial clustering center node to divide the network nodes into the initial clustering community. Then, the communities with similar characteristics are clustered and merged according to the change of information entropy. At the end, a number of different types of communities are formed in the network, and the nodes in the community have a high degree of similarity, which improves the efficiency of message forwarding. At the same time, in order to eliminate the inefficient nodes in the community, based on the information entropy and the social attributes of the nodes, this paper proposes a mechanism for dynamically updating the community to ensure the efficiency of the nodes in the community. The simulation results show that the transmission success rate of this algorithm is better than other classic routing algorithms, meanwhile, it also has lower transmission delay and routing overhead.

INDEX TERMS Opportunistic networks, routing protocol, node community, information entropy, unsupervised learning.

I. INTRODUCTION

With the rapid development of network technology and mobile communication applications [1], [2], various smart portable devices have emerged and are widely used, such as tablet computers, smart watches, smart bracelets, and so on. Through these devices, users can communicate anywhere and anytime to share information. However, due to the mobility of these devices, in some extreme environments, the network performance will be degraded or even unable to communicate due to frequent network breaks. These problems make traditional wireless network-related technologies no longer applicable [3]. Mobile opportunistic network, the current research hotspot, can effectively deal with these issues [4]–[8]. Mobile opportunistic network is a network model suitable for wireless communication, compared with the traditional wireless sensor network, the biggest feature of this network is that the information transmission between nodes needs to find ‘‘opportunities’’ for communication, as there may not exist persistent network connection between these two nodes, each node stores the data which need to forward and then sends it to the meeting node. Moreover, in mobile opportunistic network, due to frequent movement between nodes, the network structure is constantly changing, so that the nodes in the network can organize themselves to form a local self-organizing network [9], [10], such as community structure, group structure and so on, then this networks will add some certain fixed facilities to fulfill communication [11].

In mobile opportunistic networks, generally, network device is carried by people, so that each node has certain

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

social attributes. So some research works [12]–[14] use historical data carried by network nodes to predict the possibility of node encounters, and then design community routing algorithm for message forwarding. A number of research works [15], [16] divide the nodes in network into communities according to their time characteristics, and then proposes a community-based routing message forwarding mechanism. However, although there are more community-based network routing protocols, how to divide communities effectively is still a valuable problem. Although most of the community division methods take into account the interest of nodes, the similarity of nodes, and social relations in the real scene. However, for the divided communities, the existed algorithms do not consider re-merging or re-clustering these communities with the same properties. At the same time, because not every node in the community can forward messages effectively, those nodes with low transmission efficiency and high transmission delay should be removed from the community in time, so as to reduce the routing overhead and improve the success rate of message transmission.

Based on the problems of the above work, we formulate the community cluster algorithm, and explore a dynamic community update strategy and then finally propose a community clustering routing algorithm based on information entropy. The main works of this paper are summarized as follows.

(1) We first analyze the related community routing algorithms and then use the idea of unsupervised learning to define how communities are divided.

(2) After the community is established, we use information entropy [18] to measure the amount of node information, and further merge and cluster the initially divided communities to ensure the integrity and unity of the communities in the entire network.

(3) We propose a dynamic community update strategy based on information entropy, this strategy comprehensively consider the various social attributes of nodes, and then assign different weights to each node's social attributes, so as to ensure that all nodes in the community are efficient nodes for message forwarding.

The remainder of the paper is organized as follows. In Section 2, we describe and analyze the related community routing algorithm. Section 3 presents the mode of community clustering routing based on information entropy in detail. Simulations are presented and discussed in section 4. Finally, we conclude the paper in section 5.

II. RELATED WORK

In mobile opportunistic networks, how to effectively forward the data carried by mobile smart devices has become a hot issue in this field. At present, Relevant researchers in this field have proposed various routing algorithms for data storage and forward. Among them, the Epidemic protocol [12] is based on the flooding algorithm strategy for message transmission, the main idea of this protocol is that when two nodes meet, the source node carrying the message sends the message to the meeting node. After a series of intermediate nodes forwarding, the message will finally reach the destination node. Because each node has a high probability to obtain a copy of the message, then the destination node also has a high probability to meet the node which carrying the copy of the message and causing the message to be forwarded successfully. Theoretically the epidemic protocol has a higher message transmission success rate but also has a larger routing overhead. The Direct Transmission Protocol [20] requires the source node to store the message to be forwarded and can only forward the message to the destination node. This process will continue until it encounters the destination node or the end of the message life cycle. Therefore, the transmission overhead of the Direct Transmission protocol is low, the data transmission delay is high, and the success rate is low. In [21], the authors proposed a multicopy routing algorithm based on a single branch diffusion strategy to dynamically control the number of replicas of a packet. The algorithm comprehensively takes into account the energy consumption of the nodes, the probability of message arriving at the destination node, the time to reach the destination node, and the real-time status of the packets. However, the algorithm has a high latency, and the performance of the routing algorithm is relatively low in some areas where network services are not available. In [22], the authors explained the influence of node cache on routing algorithm and suggested that if the node lacks connection and cannot forward the message, it will buffer the message according to the queue strategy, and then transmit the message after the connection is established again. In the case of congestion, the node discards the message according to the discarding strategy. The scheduling and discarding strategy together constitute the management of the node's buffer, which can affect routing performance.

In addition to the traditional meeting and forwarding message model, there are also some routing protocols based on probability, among them the PROPHET algorithm [23] is a more classical one. In this algorithm, the authors proposed a transitive probability prediction method, which used historical data of node encounters and transmission records to calculate the forwarding probability of messages between nodes, and to judge the probability of the message being delivered to the destination node. Based on the research of probabilistic routing protocols using the historical information of encounters and the transferability, in [24] the authors used the encounter interval to estimate the probability of encounter, and then selected the retransmission node according to the probability of encounter. Aiming to imbalanced data problem, an algorithm of 3D Augmented Convolutional Network to extract time series information was proposed in [25]. A novel routing algorithm based on mobility prediction was proposed in [26], which calculated the probability of message transmission to a node in a certain area based on the semi-Markov model. However, in actual situations, the performance of this algorithm will be affected by the movement of the device, as it is difficult to update the mobile data immediately. In [27] the authors proposed an efficient routing protocol based on machine learning. This protocol uses the concepts of decision trees and neural networks to predict the success rate of packet delivery.

Since smart devices are often carried by people, the movement trajectory of the node represents the behavior trajectory of the person, so the movement of the node has certain social attributes. Some research works have designed efficient routing algorithms based on the sociality of nodes. For example, Bubble Rap routing protocol [28] is one of the classical algorithms that apply node social relationships to design forwarding strategies. The authors use the centrality and community of nodes to calculate the global ranking of nodes in the entire network and the local ranking of nodes in the community. The message will only be forwarded to the higher-ranked node to increase the success rate of message transmission. However, as this algorithm uses centrality as the only indicator for message transmission in the routing process, several nodes with higher centrality may be overloaded, while other remaining nodes are too idle. Considering the correlation between cluster set and specific time, a message forwarding routing algorithm based on cluster set named DRAFT was proposed in [29]. In [30], a clustering routing protocol based on the social attributes of nodes is proposed. This protocol defines the dominant and recessive social relationship characteristics of nodes, and assigns the weights of the dominant and recessive features according to the historical data information of the nodes. Finally, combined with the proposed feature information, the node can select the remaining nodes that have a close social relationship with itself to form a cluster, so as to forward the message to his objective node. Although the protocol takes into account the explicit and invisible social characteristics of the node, it increases the probability of message forwarding in the network, but at the same time it also increases the route overhead and brings a larger load to the network environment. In paper [31], the authors introduced dynamic social features to collect the contact behavior information of nodes, at the same time they also considered the multiple social relationships of nodes. Then, based on the multicast comparison (that is, when the message meets multiple community structures that conform to the forwarding, select a node that is more similar to the target community for forwarding), the community structure is used to select the best relay node to improve the routing transmission efficiency. In [28] the authors proposed a routing algorithm based on social relations, this algorithm uses improved median centrality to evaluate the heterogeneity of nodes, and then introduces a forwarding judgment factor in the community to speed up message forwarding. Finally, this algorithm selects forwarding nodes according to a simple community identification algorithm with a decay mechanism. This scheme can better balance the two routing evaluation metrics of node transmission success rate and average transmission delay, but the algorithm proposed by this scheme has higher network routing overhead. A distributed algorithm based on social graph is proposed to detect overlapping communities and solve the problem of community evolution by using the frequency and duration of encounter between nodes to generate social graph. Based on this, a community-based routing scheme is designed in [31].

Different from the existing research, we proposed a method to select the initial cluster center node by comprehensively considering a variety of attributes between nodes. Then, according to the selected initial cluster center node, we used K-Modes algorithm to divide the initial cluster community. Since the divided communities may have similar types of communities, we re-cluster and merge the preliminarily divided communities according to the change of information entropy to ensure that the communities divided in the network have a high degree of unity. At the same time, we propose a dynamic community updating method based on information entropy to reduce the invalid nodes in the community and ensure the efficient transmission of messages in the community.

III. SYSTEM MODEL DESIGN

With the rapid development of mobile network technology and the widespread use of mobile devices and software, the information acquisition and forwarding between different people can be more convenient. As shown in Figure 1, sometimes the exchange of information between users does not require face-to-face, the inter-communication by using software on mobile devices for text, voice and video is very convenient. This feature is more convenient for long-distance information communication. Therefore, in order to improve the QoS of wireless communication, it is necessary to study network routing with high transmission rate and low latency.

FIGURE 1. The social relationships that mobile devices have when communicating as network nodes.

In mobile opportunity network, the distribution of nodes has the characteristics of clustering in a certain period of time, which is similar to human living in a form of community. Therefore, the characteristics of community can be used as a reference indicator in the mobile opportunity network, and during this period, nodes in the network will be in a state of long-term cooperation, which is easier to succeed in forwarding messages than isolated nodes. Therefore, the performance of community-based routing algorithms in opportunistic social networks is better than traditional routing algorithms. This chapter is mainly used to define how communities are divided and how to update their strategies and explain the process of message transmission across the network.

A. CLUSTER-BASED COMMUNITY PARTITION METHOD

Cluster-based community division method proposed in this paper can be divided into three stages. In the first stage, we mainly pre-process the data. As there are many kinds of information contained in the nodes, and each has its own unit of calculation. In order to facilitate the calculation and improve the accuracy of the results, we should pre-process the data firstly. The second stage is to select the initial clustering center in the entire network, which can determine the number of initial communities to a certain extent. Then the K-Modes algorithm is used to cluster nodes and divide communities. The third stage is to determine the final number of clustered communities, that is, to further cluster and merge the communities identified in the second stage using information entropy, which to determine the number of eventually divided communities and to ensure that the categories of nodes within the community are to the maximum extent similar. The process of dividing communities described above is shown in Figure 2. By performing a clustering process, nodes in the entire network can be divided into the following communities.

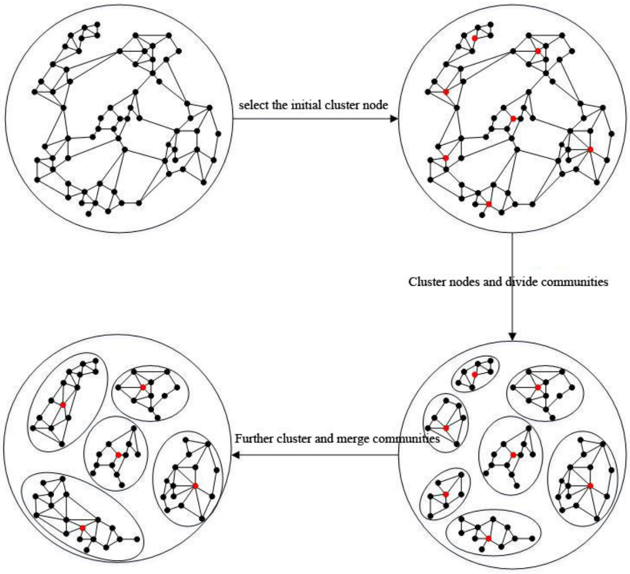
select the initial cluster node
Cluster nodes and divide communities
Further cluster and merge communities

FIGURE 2. The process of dividing the community by network nodes.

(1) Node data preprocessing

In the mobile opportunity network, each node represents a user or a mobile device, which contains a variety of attribute information, such as the number of times the nodes meet *Nm*, the time required to deliver the message *Tm*, the number of times the message was successfully delivered *Nd*, and so on. Normally, when there is a message need to forward, the node will forward the message to an encounter node that meets the forwarding criteria. However, if the node carrying the message encounters more than one node that meets the forwarding criteria at the current time t, the message transmission will experience less latency only if the node carrying the message forwards the message to the node that meets the destination node with a higher probability. Therefore, how to select the meeting nodes that satisfy the criteria is very important. This selection process can be judged by the priority of the node, which is an important factor affecting the success rate of message forwarding. The equation is expressed as:

$$\mathrm{N}PR_i^t = \frac{SeN_i^t}{T_{ij}^{end} - T_{ij}^{start}} \tag{1}$$

where $\mathrm{N}PR_i^t$ represents the priority value of node i at the current time t. As mentioned in the previous section, SeN_i^t represents the number of times that node i has successfully forwarded the message at the current time t, $\mathrm{T}_{\mathrm{ij}}^{\mathrm{start}}$ and $\mathrm{T}_{\mathrm{ij}}^{\mathrm{end}}$ respectively represent the sending time of node i and the receiving time of node j, and the range represented by node j is determined by node i that carries the message. The meaning of this formula is that the priority of a node consists of two parts, that is, the time it takes a node to forward a message and the number of times it successfully forwards a message. Only the less time it takes a node to forward a message, and the more successful a node forwards a message, the higher the priority of the current node.

In order to understand the attribute characteristics of nodes more intuitively and conveniently, and make the data better processed in the following context. Based on the related characteristics of the nodes mentioned above, we can build an evaluation matrix for these characteristics as follow:

$$Q = \begin{bmatrix} Nm_1 & Tm_1 & Nd_1 & NPR_1^t \\ Nm_2 & Tm_2 & Nd_2 & NPR_2^t \\ \ldots & \ldots & \ldots & \ldots \\ Nm_n & Tm_n & Nd_n & NPR_n^t \end{bmatrix} \tag{2}$$

In the above matrix Q, the node set is defined as

$\mathrm{V} = \{V_i | i = 1, 2, 3 \ldots\}$, the standard set of node attributes is defined as NF=$\{Nm, Tm, Nd, NPR^{\mathrm{t}}\}$, Due to the different units of calculation of the information, the meaning of the information is also different. If it is used directly, the result error may be large. Therefore, before the calculation, in order to eliminate the dimensional effect between the data variables and make each variable have the same expressive force, the data is standardized. The formula is as follows

$$R_{ij} = \frac{q_{ij} - \bar{q}_j}{S_j}, \quad i \in V, j \in NF \tag{3}$$

where q_{ij} represents the j-th attribute value of the current node i, $\bar{q}_j$ represents the average value of the j-th attribute of all nodes in the network, and S_j represents the standard deviation of the jth attribute of all nodes in the network. R_{ij} represents the new data obtained after processing these index data. Among then $\bar{q}_j$ and S_j can be express as:

$$\bar{q}_j = \frac{1}{n}\sum\nolimits_{i=1}^{n} q_{ij} \tag{4}$$

$$S_j = \sqrt{\frac{1}{n-1}\sum\nolimits_{i=1}^{n}(q_{ij}-\bar{q}_j)^2} \tag{5}$$

(2) The selection of cluster centers and the division of cluster communities

1) DETERMINATION OF THE NUMBER OF NODE CLUSTERING COMMUNITIES

In real environment, the trajectory of human life presents a certain social phenomenon, such as the community where people live, shopping malls, cinemas, hospitals and so on. Compared with other activity areas, the communication between people in these areas is more frequent, and it is more conducive to the forwarding of messages. This feature can be reflected in the activity of nodes in mobile opportunity network. The relatively active nodes in the network may forward more messages in a certain period of time, and also have a higher probability of forwarding messages to the destination node. While the less active nodes will have a higher delay of transmitting messages than the active nodes, and the success rate of transmitting messages is also lower. We know that the clustering process is actually the process of other nodes grouping around the nodes with high activity. If the nodes with low activity are used for clustering nodes, then the message forwarding may cause the failure, delay and loss of message forwarding.

The activity of a node can be expressed as the number of times a node forwards messages. When calculating the activity of a node, an initial value is assigned to it. Then, every time node i successfully forwards a message, the activity value of node i is $\mathrm{Se}N_i$ plus one. In addition, the activity value of a node depends not only on the number of times the node successfully forwards messages, but also on the number of times the node receives messages $\mathrm{Re}N_i$. In general, the activity value of a node can be expressed as the number of times a node successfully forwards a message plus the number of times it receives a message. The equation can be expressed as:

$$A_i^t = SeN_i^t + ReN_i^t \tag{6}$$

In (6) where SeN_i^t represents the number of messages forwarded by node i at time t, and its initial default value is set to 1, ReN_i^t represents the number of messages received by node i at time t. A_i^t represents the activity value of node i at time t. In order to compare the activity value of the node at the current time in the network more conveniently, we normalized the equation as follows:

$$(A_i^t)^* = \frac{A_i^t - \min\left\{A_j^t\right\}}{\max\left\{A_j^t\right\} - \min\left\{A_j^t\right\}}, \quad j = 1, 2, 3 \cdots n \tag{7}$$

In (7), *max* represents the maximum activity value of the node at the current moment, and *min* represents the minimum activity value of the node at the current moment. The size of the node activity value can preliminarily determine the number of node clusters. If the node activity is greater than the average activity value $\mathrm{A}p_t$ of all nodes in the network at the current moment, the number of clusters of nodes in the current network N_k is increased by one. The formula for the average activity value of the nodes in the network at the current moment is:

$$\mathrm{A}p_t = \frac{1}{n}\sum\nolimits_{i=1}^{n}(A_i^t)^* \tag{8}$$

From (8) we can get the number of clustering nodes. Firstly, we use the K-Modes algorithm to cluster the network nodes and divide the communities directly, then we perform hierarchical clustering, instead of starting from a single node to perform hierarchical clustering. The purpose of hierarchical clustering is to merge communities with similar properties.

2) SELECTION OF INITIAL CLUSTER CENTER

It can be seen from 1) that the node set is defined as $V = \{v_i|\mathrm{i} \in \mathrm{V}\}\mathrm{i} \in \mathrm{V}\}$, and the attribute feature set form of the node is defined as $NF = \{nf_1|\mathrm{l} \in NF\}$. Let $x_i, x_j \in V$, $\mathrm{nf} \in NF$, $\mathrm{U} = \bigcup_{nf \in NF} U_{nf}$, among then U represents the union of each attribute range, and U_{nf} represents the range of attribute nf. So $V * NF \to$ úU is an information function, which assigns an information value to each attribute of each node, namely $\forall nf \in NF, \mathrm{x} \in V$ and $\mathrm{f}(\mathrm{x}, \mathrm{nf}) \in U_{nf}$,then the distance between the nodes x_i, x_j is:

$$Dis_{NF}\left(x_i, x_j\right) = \sum\nolimits_{l \in NF, l=1}^{m} f(x_i, nf_l) \cdot f(x_j, nf_l) \tag{9}$$

where:

$$f\left(x_i, nf_l\right) \cdot f\left(x_j, nf_l\right) = \begin{cases} 0, & f\left(x_i, nf_l\right) = f\left(x_j, nf_l\right) \\ 1, & f\left(x_i, nf_l\right) \neq f\left(x_j, nf_l\right) \end{cases} \tag{10}$$

When selecting the cluster center node, if we only consider the distance between the selected node and the initial cluster center node, it may cause the remote isolated node to be selected as the cluster center node, and the node that is really suitable for the cluster center will not be selected. Therefore, in addition to considering the distance of the node, we also need to consider the surrounding condition of the node, that is, the surrounding density of the selected node, which can be expressed as the following expression:

$$\mathrm{ND}\left(x_i\right) = \frac{\sum_{l=1}^{m}\frac{\{x_i, x_j|f(x_i, nf_l) = f\left(x_j, nf_l\right)\}}{|V|}}{|N|} \tag{11}$$

It can be seen from the above equation that the greater value $ND\left(x_i\right)$ of the density of node i, the more data nodes are distributed around node i, and the higher the probability that node i will be selected as the initial cluster center node. Therefore, the node with the largest density can be selected as the first initial cluster center node x_1', and the density expression is as follows:

$$ND\left(x_1'\right) = \max(ND\left(\mathrm{x_i}\right), \quad i = 1, 2, 3 \cdots n\} \tag{12}$$

After the first cluster center node is determined, it can be put into the set $CN = CN \cup \{x'_1$. When selecting the remaining cluster centers x'_j, in addition to the density factor of the node, it is also necessary to consider the distance between the node and the selected cluster center node. Therefore, the traditional maximum and minimum distance algorithm is further extended, and the equation can be expressed as:

$$ND(\mathrm{x}'_{\mathrm{j}}) = \max_{i=1}^{n}\{\min\{\mathrm{Dis}_{\mathrm{NF}}(\mathrm{x_i}, \mathrm{x_h}) \cdot ND(\mathrm{x_i})\}\}, \ \mathrm{x_h} \in CN \tag{13}$$

Each time a cluster center node is selected, it should be put into the CN set, and compare whether the size of the set CN is less than or equal to the number of clusters N_k, if it is less, the steps of selecting the remaining cluster centers should be continue to perform, otherwise, the algorithm will directly output the final the cluster center node set CN.

3) DIVISION OF INITIAL CLUSTERING COMMUNITIES

After the initial clustering center node selection is completed, we can use the K-Modes algorithm to divide the entire network into clustering communities. The detailed steps are as follows:

Step 1: Select the initial cluster center node set CN, and assuming the set $CN=\{\mathrm{cn_1}, \mathrm{cn_2}, \mathrm{cn_3} \ldots \mathrm{cn_k}\}$.

Step 2: For the set of nodes in the network $V=\{\mathrm{v_i}|\mathrm{i} = 1, 2, 3 \ldots \mathrm{n}\}$, we compare the difference between its and the k initial clustering center nodes, and then add the current node i to the community which the initial clustering center node with the smallest difference. The formula is expressed as:

$$NDF_i = \min_{1\le j<<k}\{\sqrt{\sum\nolimits_{l\in NF}(v_{il} - cn_{jl})^2}\} \tag{14}$$

In (14), v_{il} represents the l-th eigenvalue of node i, cn_{jl} represents the l-th eigenvalue of the initial cluster center node j, and NDF_i represents the minimum difference value between the current node i and the initial cluster center node j.

Step 3: Repeat Step 2 to add eligible nodes to the cluster community where the corresponding initial cluster center is located, until all nodes in the network are traversed.

Step 4: Output the final initial clustered community set $C=\{\mathrm{c_1}, \mathrm{c_2}, \mathrm{c_3} \ldots \mathrm{c_k}\}$.

(3) Cluster and merge communities and then determine the final community size

It can be seen from the above 2) that the K-Modes algorithm finally selects a certain initial cluster center node and the divided cluster community set. Each cluster community set contains at least one or more nodes. However, the K-Modes algorithm is clustering algorithm based on selected cluster center nodes. In the same area, this algorithm classifies nodes with the same attribute according to the selected cluster center, but nodes in different areas may be clustered according to the same attribute index, to divide the scale of clustered communities with certain attributes. Therefore, we can merge clustered communities with the same attributes. If the two communities that choose to merge are more similar in structure, the disorder caused by the merger will be smaller. If the differences between the two merged communities are larger, the greater disorder they will bring. Since information entropy can better reflect the distribution characteristics of data, we can use the amount of change in entropy after the merger of the two communities to measure the dissimilarity between the two communities.

Information entropy is a quantitative index to measure the information content of a system, and it is a description of the uncertainty of things happening. The size of information is related to the probability of the event. The smaller the probability event, the greater the amount of information, and the larger the probability event, the smaller the amount of information. In the process of community clustering, the nodes that are divided into the same community have a closer relationship. Compared with the nodes between communities, the probability of transmitting information is greater, and therefore, the amount of information is also smaller. On the contrary, the greater the amount of information. Information entropy can be expressed as:

$$\mathrm{E}(\mathrm{n}f_i) = \sum\nolimits_{i=1}^{m} p(x_i)\log_2(x_i) \quad (i = 1, 2, 3\ldots m) \tag{15}$$

In (15), $\mathrm{E}(\mathrm{n}f_i)$ represents the information entropy value of each feature $\mathrm{n}f_i$ of the node, and $p(x_i)$ represents the probability function of the feature $\mathrm{n}f_i$. Set $p(x_i) = 0$, then $p(x_i)\log_2(x\_i) = 0$, that is to say, the information entropy value of feature i is 0. It can be seen from the formula that the function $\mathrm{E}(\mathrm{n}f_i)$ satisfies monotonicity, non-negativity, and accumulation. When using entropy to analyze node clustering characteristics, these properties must also be satisfied. At the same time, because information entropy requires multiple logarithmic calculations when calculating, it often leads to data overflow and the calculation is abnormally time-consuming, so we reconstruct the equation (15) according to the above principles as follows:

$$\mathrm{E}(\mathrm{n}f_i)=\sum\nolimits_{i=1}^{m} p(x_i)(1-p(x_i))^2 \quad (i = 1, 2, 3\ldots m) \tag{16}$$

It can be seen from this expression that the feature $\mathrm{n}f_i's$ probability function is $p(x_i)$ and its value range is $p(x_i) = [0, 1]$. When the probability $p(x_i)$ obtained by the feature $\mathrm{n}f_i$ of the node is smaller, the value $(1 - p(x_i))^2$ is larger, and the entropy value $\mathrm{E}(\mathrm{n}f_i)$ of the feature $\mathrm{n}f_i$ obtained finally is also larger, so the more information that can be generated. On the contrary, the larger the value $p(x_i)$, the smaller the value $(1 - p(x_i))^2$, the smaller the entropy value $\mathrm{E}(\mathrm{n}f_i)$, so the amount of information is smaller. Therefore, the equation (16) satisfies the entropy-related properties defined by Shannon, the father of information theory, that is, monotonicity, non-negative and additive.

For those community collection set $C = \{\mathrm{c_1}, \mathrm{c_2}, \mathrm{c_3} \ldots \mathrm{c_k}\}$ divided by the K-Modes algorithm, $C_\mathrm{i}, C_\mathrm{j} \in C$, $\mathrm{i} \neq \mathrm{j}$, and $C_i \neq \emptyset, C_j \neq \emptyset, C_i \cap C_j \neq \emptyset$. Assuming that the number of nodes in the C_i subset is h, based on the accumulation of information entropy, then $E(C_i) = \sum_{x=0}^{h} E_x(\mathrm{nf})(x = 1, 2, 3 \ldots h)$, where E_x is the information entropy of the xth node, and the amount of information entropy change between C_i and C_j can be expressed as:

$$\mathrm{IE}\left(C_i, C_j\right) = \left(|C_i| + \left|C_j\right|\right) E\left(C_i \cup C_j\right) - (|C_i| E\left(C_i\right) + \left|C_j\right| E\left(C_j\right) \quad (17)$$

In equation (17), $|C_i|$ *and* $\left|C_j\right|$ respectively represent the number of node elements in the community set *i* and *j*, $E\left(C_i\right)$ and $E\left(C_j\right)$ respectively represent the information entropy value of the community set *i* and *j*. $E\left(C_i \cup C_j\right)$ represent the information entropy value of the community set *i* and *j* after the merging. *IE* $\left(C_i, C_j\right)$ represents the change of information entropy of the community set *i* and *j* after merging compared with that before merging.

The change of information entropy defined by the above expression can reflect the difference in the clustering structure between the two communities. If the amount of change is zero, it means that no impurities are brought into the clustering result during the merging process. Similarly, a small amount of change indicates that the clustering structure has not changed significantly, and a large amount of change indicates that the clustering results have changed significantly. As the merger between the two communities is between the selected community and the new generated community. In order to further express the change rule of community structure among different communities in the process of clustering, we expressed the relationship of entropy change as follows:

$$I\,(\mathrm{k}-1) = \frac{\min\{\mathrm{IE}\left(C_i, C_j\right)|i = 1, 2 \ldots k-1; j = i+1 \ldots k\}}{|C_i| + \left|C_j\right|} \quad (18)$$

Among them, I (k − l) represents the minimum value of the change in entropy in the network from the current *k* clustered communities to *k−1* clustered communities. Related research shows that when $I(k)$ is approximately equal to $I(k-1)$, it can be considered that the change of entropy in the process of community merging is approximately equal. In other words, as long as the structure of the cluster community has not undergone major changes, it can be merged into one community to increase the effective transmission probability of nodes.

The flowchart of clustering and dividing communities is shown in Figure 3. The community division method adopted in this paper uses the idea of unsupervised clustering, the number of communities that can be divided is obtained through pre-processing before clustering, so as to finally form a community structure with similar characteristics.

B. DYNAMIC UPDATING STRATEGY OF COMMUNITY NODES

The network community is composed of multiple closely connected nodes, but not every node in the community has the ability to become a relay node. Nodes in the community that do not meet the transmission conditions can be considered as inefficient nodes. Deleting these inefficient nodes can reduce transmission overhead and improve community transmission efficiency. For each community in the network, we analyzed various social attributes of the node and measured the impact of various social attributes on information transmission. By weighting and combining these attributes, we can get a comprehensive indicator that can be used to evaluate the number of nodes in the community and reduce the number of invalid nodes.

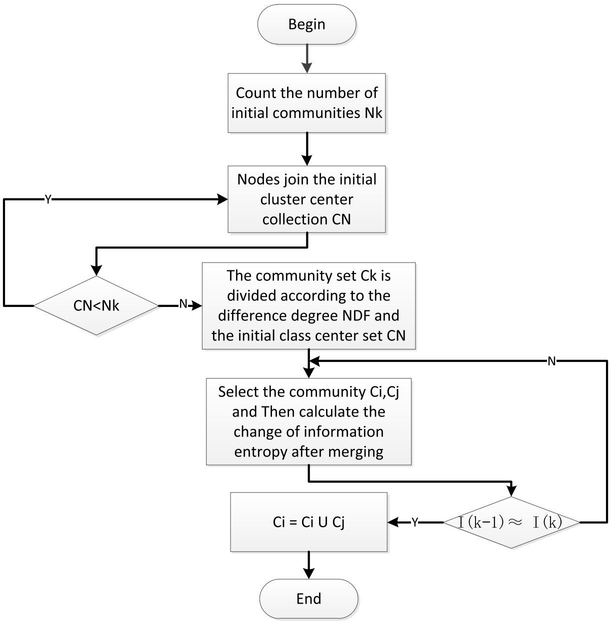
Begin
Count the number of initial communities Nk
Nodes join the initial cluster center collection CN
Y
CN<Nk
N
The community set Ck is divided according to the difference degree NDF and the initial class center set CN
N
Select the community Ci,Cj and Then calculate the change of information entropy after merging
Ci = Ci U Cj
Y
I (k−1) ≈ I (k)
End

FIGURE 3. Flowchart of clustering division of community.

It can be seen from A.(1) that we have performed preprocessing operations on the data of the nodes, and according to equation (2), we have normalized the data matrix Q. Then we calculate the probability P_{ij} of each attribute value of the current node i in the entire network on the processed matrix, the formula is as follows:

$$P_{ij} = \frac{q_{ij}}{\sum_{i=1}^{n} q_{ij}}, \quad j\epsilon NF \quad (19)$$

According to the reconstructed information entropy expression (16), the information entropy value of the attribute can be calculated as shown below:

$$E\left(\mathrm{nf}_j\right) = \sum_{i=1}^{m} p_{ij}(1-p_{ij})^2 \quad (j = 1, 2, 3 \ldots m) \quad (20)$$

In mobile opportunistic networks, the importance of each attribute is different. It is usually necessary to make a quantitative description of the importance, that is, to determine the weight of each attribute. Therefore, we will normalize the weight of each index and then define the weight of the j-th attribute of the current node i as:

$$\mathrm{W}\left(\mathrm{nf}_{ij}\right) = \frac{1 - E(\mathrm{nf}_{ij})}{\sum_{l=1}^{m}(1 - E(\mathrm{nf}_{il}))} \quad (21)$$

Then we can estimate the evaluation value of the current node i, and the formula is as follows:

$$\mathrm{A}V_i = \sum_{j=1}^{m} \mathrm{W}\left(\mathrm{n}f_{ij}\right) \times \mathrm{n}f_{ij} \tag{22}$$

According to equation (22), we can judge whether the current node j can meet the needs of the relay node according to the evaluation value, which helps to delete inefficient nodes from the current community, and can better control the size of the community to ensure all nodes in the community have high intimacy. Through this scheme of reducing nodes in the community, we can filter nodes and delete some of them that do not meet the transmission requirements in the community. After reducing the number of nodes that do not meet the transmission conditions, the relationship between nodes in the community is closer, and all nodes have higher transmission capabilities.

Figure 4 below this paragraphs shows the process of clustering nodes in the network and merging and dividing communities. Some inefficient nodes are deleted to improve the performance of the algorithm. Compared with the original single-node algorithm, it not only improves the transmission success rate and transmission speed, but also reduces the transmission delay and routing cost.

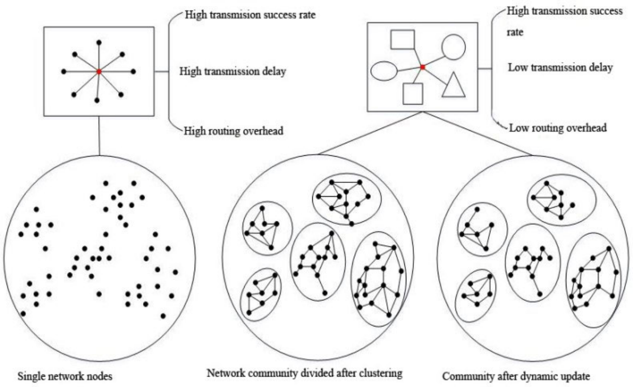
High transmision success rate
High transmission delay
High routing overhead
High transmission success rate
Low transmission delay
Low routing overhead
Single network nodes
Network community divided after clustering
Community after dynamic update

FIGURE 4. **Comparison between the traditional routing algorithm and the proposed algorithm.**

C. THE COMMUNITY-BASED MESSAGE FORWARDING STRATEGY

In summary, as mentioned in above A and B, we can get several close communities from the network, and the nodes in these communities will be more favorable for message forwarding. The CREN algorithm proposed in this paper can effectively improve the success rate of data transmission and reduce the routing overhead. When two nodes meet, message forwarding will occur only when the meeting node is in the same community as the target node or is the target node itself. The specific forwarding process is as follows:

When node V_i carrying a message meets node V_j, node V_i will first determine whether node V_j is the destination node. If it is the destination node, node V_i will forward the message directly to node V_j and deletes the message in its own cache.

Algorithm 1 Community Clustering Algorithm Based on Information Entropy

```
Begin
 For each Node
   Judge whether Node $V_i$ can join set CN
   If (Node $V_i$ is suitable for joining community $C_j$)
     Join Node i to community $C_j$
   End if
End for
For each community
   Calculate $E(C_i)$, $E(C_j)$, $E(C_i \cup C_j)$ of community $C_i$, $C_j$
     Calculate the combined entropy difference $IE(C_i, C_j)$
  If ($I(k) \approx I(k-l)$)
       Merge community $C_i$, $C_j$
     End if
  End for
  Forward meesage from source node S
   If (destination node D and S is in the same community)
     Message forwarded to node in the community S allocate
  Else
     If (the message node $V_i$ in the community D allocate
       Message forwarded to node $V_i$
   End if
  End if
  End
```

If V_j is not the destination node, node V_i will forward the message according to the following two situations:

(1) Send messages within the community

If node V_i and the target node are in the same community, in addition, node $\mathrm{V_j}$ and the destination node are also in the same community, then node V_i will forward a message copy to node V_j. Otherwise, the message will not be forwarded.

(2) Messages forwarding between different communities

If the target node and node V_i are not in the same community, node V_i will ask node V_j to query whether they are in the same community. Then node V_j will determine whether they are in the same community according to its own community information. If they are in the same community, node V_i will directly forward the message to node V_j. If it is not in the same community, the message will not be forwarded.

The purpose of this method is to prevent useless messages from flooding the whole network and to avoid the increase of network load and the collapse of the network. Before node V_i forwards the message, it will save the message in its own buffer for a period of time, The length of time depending on the size of node V_i's buffer. During this period of time, if node V_i encounters a qualified forwarding target, it will forward the message. Otherwise, when the message times out, node V_i will delete the message from the buffer. Therefore, the size of the node's cache is also crucial for message forwarding. In order to have a deeper understanding of the message forwarding process, the detailed algorithm flow is as follows:

IV. SIMULATIONS

The experiment of this paper is completed on the platform of opportunistic network simulator ONE [32], through this platform the performance of CREN algorithm proposed in this paper is analyzed. Compared with EPIC [12], BUBBLE [27] and DRAFT [28] algorithm, the performance of CREN algorithm is demonstrated. Among them, EPIC algorithm is a classic routing algorithm, which is based on flooding strategy for message transmission. BUBBLE algorithm is a routing algorithm based on dividing communities and passing messages by comparing the centrality between communities. DRAFT algorithm is also an algorithm based on clustering, which mainly considers the correlation between clustering set and specific time, so as to forward messages based on clustering set. After considering the data information required by the algorithm, four real datasets of Infocom5, Infocom6, Cambridge and Intel are selected for simulation experiment. The specific information of each dataset is shown as the table below:

TABLE 1. Simulation parameters of FOU experimental datasets in one.

Dataset	Infocom5	Infocom6	Cambridge	Intel
Time(day)	**3.5**	**4**	**11.5**	**4**
Number of Nodes	**41**	**98**	**52**	**9**
Number of node collisions	**22459**	**170601**	**10873**	**1364**

In the simulations, we use iMote as experiment nodes, each node is mobile with speed in the range 0.5-1.5m/s and each node has 5M cache. In order to verify CREN algorithm, we uses three performance evaluation indexes, transmission success rate, routing overhead and average end-to-end delay. **T**ransmission success **r**atio *Tr* refers to the ratio of the total number of data packets successfully arriving at the destination node and the total number of data packets to be transmitted sent by the source node in a given time.

$$Tr = \frac{success\_number}{created\_number}$$

This metric describes the ability of routing algorithm to correctly forward data packets to the destination node. Routing overhead refers to the total number of data packets forwarded by a node in a certain period of time. It is usually evaluated by the overhead rate, that is, the ratio of the total number of data packets generated by the source node to the total number of data packets forwarded by all nodes. The formula is expressed as:

$$Or = \frac{forward\_number - delivered\_number}{delivered\_number}$$

Transmission delay refers to the time required for data packets to arrive at the target node from the source node, which is usually evaluated by the average transmission delay. Compared with routing algorithm, transmission delay is small, which has strong transmission ability and high transmission efficiency. At the same time, it means that data packets will occupy less resources in transmission process.

A. TRANSMISSION SUCCESS RATIO

The delivery ratio is defined as the ratio of the total number of messages successfully reached their destination and the amount of total messages. The higher delivery ratio is, the greater the probability that the message will reach the destination node successfully.

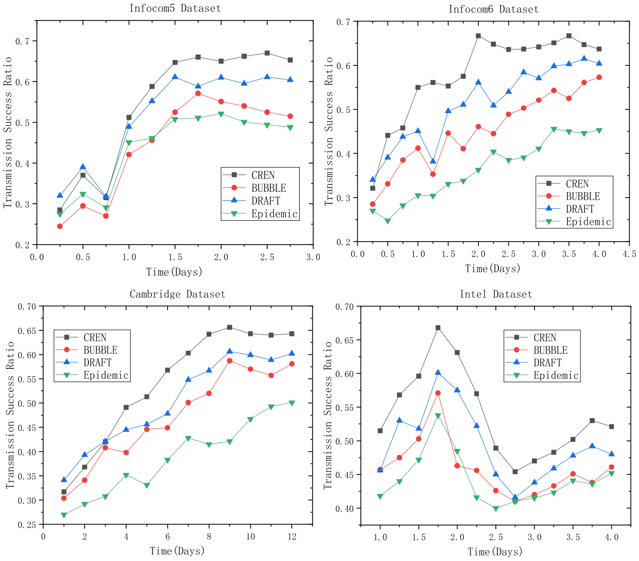

FIGURE 5. Comparison for transmission success ratio of various routing algorithm.

Figure 5 shows the impact of time changes on the packet transmission success rate of CREN, BUBBLE, DRAFT, and Epidemic algorithms. When the simulation time is less than one day, the advantage of the algorithm CREN is not obvious, because the process of forming a community in the network takes a certain amount of time, so in a relatively short period of time, the nodes in the community are not enough for effective data forwarding. As the simulation time increases, we can find that the transmission success rate of the CREN algorithm is always the highest among these algorithms. In the CREN algorithm, the nodes in the network are divided into several communities, and all nodes in the community may often communicate with each other. At the same time, the CREN algorithm proposes a dynamic update community strategy based on node attributes to calculate the entropy value, which can minimize the number of unqualified nodes in the community. The BUBBLE algorithm is a community-based routing algorithm, but it considers the time relationship of the nodes without considering the factors of social attributes, while the CREN algorithm takes into account a variety of related attributes of the nodes, so it has better performance. For the Epidemic algorithm, it does not take into account the characteristics of the node, but forwards the message according to the probability, which has a large randomness, and is not suitable for scenarios that require high stability. For the DRAFT algorithm, the priority of the node is not considered, so it is possible that the node that is more conducive to message forwarding does not receive the message, which causes the failure of message transmission.

B. TRANSMISSION DELAY

The average end-to-end transmission delay of each algorithm is shown in Figure 6.Compared with other algorithms, the CREN algorithm has the lowest average end-to-end delay. As the CREN algorithm proposes a community dynamic update strategy by analyzing the relevant attributes of nodes, it can reduce inefficient nodes in the community, thereby reducing the average end-to-end delay. Compared with the Epidemic algorithm, when encountering the node, the message is directly forwarded, which may cause a sharp increase in routing and forwarding delay. The BUBBLE algorithm combines the characteristics of community and centrality, and can delete the number of copies in time when forwarding messages, so the transmission delay is lower than traditional algorithms. For the DRAFT algorithm, in the process of forming a cluster set to forward messages, it takes into account the attributes of the node's time range, so it is a traditional routing algorithm with low latency. Among the above algorithms, the average end-to-end transmission delay of the CREN algorithm is the smallest.

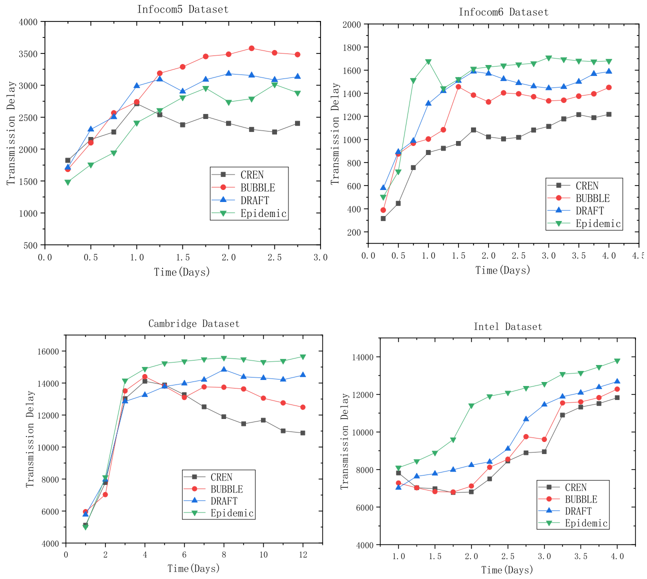

FIGURE 6. Comparison for transmission delay of various routing algorithm.

C. ROUTING OVERHEAD

Figure 7 shows the comparison of routing costs between the above algorithms. Since the CREN algorithm adopts the idea of dynamic community and fully considers the relevant attributes of the node, the average cost of CREN is always kept to a minimum. In the CREN algorithm, instead of blindly forwarding messages to any node, nodes selectively forward messages to other nodes to distinguish them through the community. As a result, the number of copies of messages in the network is low, and the cost of routing is greatly reduced. Compared with the CREN algorithm, the Epidemic algorithm, because of its arbitrary message forwarding mechanism, requires a lot of time and resources for redundant message copies, resulting in very high routing overhead. Although the DRAFT algorithm takes into account the characteristics of the clustering set, it may cause excessive waiting time for message forwarding, which will lead to a part of the routing overhead. Although the BUBBLE algorithm takes into account the centrality of the nodes, it can reduce the resource consumption of some unavailable nodes, so routing overhead can be optimized. In short, compared to other algorithms, the routing overhead of the CREN algorithm is the lowest.

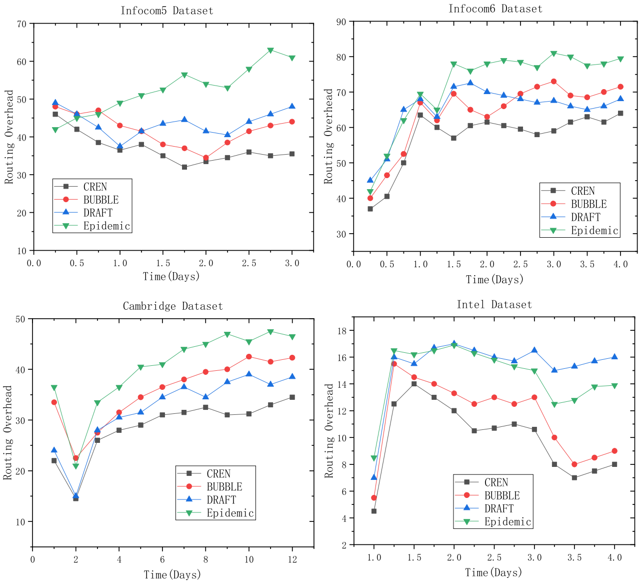

FIGURE 7. Comparison for routing overhead of various routing algorithm.

Based on the experimental results in Fig. 5, Fig. 6 and Fig. 7, when the simulation time is less than one day, the proposed algorithm CREN has no obvious advantages over other algorithms. However, when the network simulation time is extended to at least 3 days, the CREN algorithm can increase the transmission success rate by 34.6%, reduce the transmission delay by 31%, and reduce the routing overhead by 26%.CREN has the characteristics of high transmission success rate, low transmission delay and low routing overhead, so it is a good routing protocol with high performance.

V. CONCLUSION

This paper proposes a community clustering routing algorithm based on information entropy and the social relationship between nodes in the mobile opportunistic network.

Compared with other routing algorithm, our algorithm perform good performance based on the experimental results. The algorithm first proposed a method for selecting initial clustering nodes based on the density and distance of nodes and their attribute relationships. Then, the K-Modes algorithm combined with the selected initial clustering nodes was used to divide the entire network into communities. In order to ensure that the divided communities have a high degree of unity, a scheme is proposed to merge and cluster the communities in the network using the change of information entropy. At the same time, in order to ensure the efficiency of information transmission by nodes in the community, a community dynamic update strategy based on information entropy is proposed. This method analyzes various attributes of nodes, comprehensively evaluates nodes, and deletes inefficient nodes. Combining the above methods, the network node forwards the message in combination with the community state, so that the message can reach the destination node in an efficient and stable transmission mode. Because the algorithm in this article requires the message to be forwarded in the form of a copy, its has higher requirements for the cache of the network nodes, so in the future research work, we will further improve the performance of the algorithm in terms of caching and security.

REFERENCES

[1] M.-A. Storey, A. Zagalsky, F. F. Filho, L. Singer, and D. M. German, "How social and communication channels shape and challenge a participatory culture in software development," *IEEE Trans. Softw. Eng.*, vol. 43, no. 2, pp. 185–204, Feb. 2017.

[2] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: Social-based forwarding in delay-tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1576–1589, Nov. 2011.

[3] O. Richard, O. Fadahunsi, M. Maheswaran, and M. F. Zhani, "Opportunistic edge computing: Concepts, opportunities and research challenges," *Future Gener. Comput. Syst.*, vol. 89, pp. 633–645, Dec. 2018.

[4] Y. Liu, T. Wang, S. Zhang, X. Liu, and X. Liu, "Artificial intelligence aware and security-enhanced traceback technique in mobile edge computing," *Comput. Commun.*, vol. 161, pp. 375–386, Sep. 2020.

[5] M. Song, Z. Wang, Z. Zhang, Y. Song, Q. Wang, J. Ren, and H. Qi, "Analyzing user-level privacy attack against federated learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2430–2444, Oct. 2020.

[6] Y. Li, G. Su, D. O. Wu, D. Jin, L. Su, and L. Zeng, "The impact of node selfishness on multicasting in delay tolerant networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 5, pp. 2224–2238, Jun. 2011.

[7] Q. Li, F. Zeng, Q. Wu, and J. Yang, "An incentive mechanism based on Bertrand game for opportunistic edge computing," *IEEE Access*, vol. 8, pp. 229173–229183, 2020.

[8] M. Yu, A. Liu, N. N. Xiong, and T. Wang, "An intelligent game based offloading scheme for maximizing benefits of IoT-edge-cloud ecosystems," *IEEE Internet Things J.*, early access, Nov. 23, 2020, doi: 10.1109/JIOT.2020.3039828.

[9] Y. Ouyang, A. Liu, N. Xiong, and T. Wang, "An effective early message ahead join adaptive data aggregation scheme for sustainable IoT," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 201–219, Jan. 2021.

[10] X. Yu, Z. Liu, and B. Yan, "Routing algorithm based on social relationship in mobile opportunistic network," *Comput. Eng.*, vol. 43, no. 12, pp. 98–102, 2017.

[11] M. Shen, A. Liu, G. Huang, N. N. Xiong, and H. Lu, "ATTDC: An active and traceable trust data collection scheme for industrial security in smart cities," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6437–6453, Apr. 2021.

[12] P. Mundur, M. Seligman, and G. Lee, "Epidemic routing with immunity in delay tolerant networks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2008, pp. 1–7.

[13] E. Tasnádi and G. Berend, "Supervised prediction of social network links using implicit sources of information," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1117–1122.

[14] T.-K. Huang, C.-K. Lee, and L.-J. Chen, "PRoPHET+: An adaptive PRoPHET-based routing protocol for opportunistic network," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 10–18.

[15] F. Zeng, N. Zhao, and W. Li, "Effective social relationship measurement and cluster based routing in mobile opportunistic networks," *Sensors*, vol. 17, no. 5, pp. 1109–1122, 2017.

[16] F. Xia, J. Wang, X. Kong, D. Zhang, and Z. Wang, "Ranking station importance with human mobility patterns using subway network datasets," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2840–2852, Jul. 2020.

[17] C. Huang and D. Wang, "Unsupervised interesting places discovery in location-based social sensing," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, Washington, DC, USA, May 2016, pp. 67–74, doi: 10.1109/DCOSS.2016.12.

[18] P. Yuan, H. Ma, and H. Fu, "Hotspot-entropy based data forwarding in opportunistic social networks," *Pervas. Mobile Comput.*, vol. 16, pp. 136–154, Jan. 2015, doi: 10.1016/j.pmcj.2014.06.003.

[19] F. Zeng, Q. Chen, L. Meng, and J. Wu, "Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3247–3257, Jun. 2021.

[20] H. Zhou, X. Chen, S. He, J. Chen, and J. Wu, "DRAIM: A novel delay-constraint and reverse auction-based incentive mechanism for WiFi offloading," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 4, pp. 711–722, Apr. 2020.

[21] Y. Chen, S. He, F. Hou, Z. Shi, and J. Chen, "An efficient incentive mechanism for device-to-device multicast communication in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 7922–7935, Dec. 2018.

[22] Z. Zhang, S. He, J. Chen, and J. Zhang, "REAP: An efficient incentive mechanism for reconciling aggregation accuracy and individual privacy in crowdsensing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 12, pp. 2995–3007, Dec. 2018, doi: 10.1109/TIFS.2018.2834232.

[23] K. Chen, H. Shen, and L. Yan, "Multicent: A multifunctional incentive scheme adaptive to diverse performance objectives for DTN routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1643–1653, Jun. 2015.

[24] L. Li, Y. Qin, X. Zhong, and H. Chen, "An incentive aware routing for selfish opportunistic networks: A game theoretic approach," in *Proc. 8th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2016, pp. 1–5.

[25] S. Pei, T. Shen, C. Gu, Z. Ning, X. Ye, and N. Xiong, "3DACN: 3D augmented convolutional network for time series data," *Inf. Sci.*, vol. 513, no. 2020, pp. 17–29, Nov. 2020.

[26] L. Zhang, Z. Cai, J. Lu, and X. Wang, "Mobility-aware routing in delay tolerant networks," *Pers. Ubiquitous Comput.*, vol. 19, no. 7, pp. 1111–1123, 2015.

[27] D. K. Sharma, S. K. Dhurandher, I. Woungang, A. Mohananey, and J. J. Rodrigues, "A machine learning-based protocol for efficient routing in opportunistic networks," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2207–2213, Sep. 2018.

[28] X. Chen, C. Shang, B. Wong, W. Li, and S. Oh, "Efficient multicast algorithms in opportunistic mobile social networks using community and social features," *Comput. Netw.*, vol. 111, no. 1, pp. 71–81, 2016.

[29] M. Orlinski and N. Filer, "The rise and fall of spatio-temporal clusters in mobile ad hoc networks," *Ad Hoc Netw.*, vol. 11, no. 5, pp. 1641–1654, Jul. 2013.

[30] F. Zeng, Y. Chen, L. Yao, and J. Wu, "A novel reputation incentive mechanism and game theory analysis for service caching in software-defined vehicle edge computing," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 2, pp. 467–481, Mar. 2021.

[31] K. Wei, D. Zeng, S. Guo, and K. Xu, "On social delay-tolerant networking: Aggregation, tie detection, and routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 6, pp. 1563–1573, Jun. 2014, doi: 10.1109/TPDS.2013.264.

[32] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proc. 2nd Int. ICST Conf. Simul. Tools Techn.*, 2009, pp. 1–10.

QINGHUA LI received the Ph.D. degree from the School of Information Science and Engineering, Central South University, Changsha, China, in 2010. He has been working as an Associate Professor with Lingnan Normal University, Zhanjiang, China, since 2018. His research interests include wireless networks, network quality of service, edge computing, and big data.

LIMIN ZHANG received the M.Eng. degree from Hunan Normal University, Changsha, China, in 2008. He has been working as a Lecturer in computer science with Lingnan Normal University, Zhanjiang, China, since 2009. His research interests include intelligence algorithms, computer networks, and data mining with their applications.

FENG ZENG (Member, IEEE) received the B.Eng. and M.Eng. degrees in computer science from Hunan University, in 2000 and 2005, respectively, and the Ph.D. degree in computer sciences from Central South University, in 2010. He is currently an Associate Professor with the School of Computer Science and Engineering, Central South University. His current research interests include wireless networks, crowdsourcing, and opportunistic routing.

YONG PAN received the Ph.D. degree from the School of Geosciences and Info-Physics, Central South University, Changsha, China, in 2016. He has been working as an Associate Professor with Guilin Tourism University, Guilin, China, since 2021. His research interests include edge computing and big data.

JUNJIE YANG received the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2006. He has been working as a Professor with Lingnan Normal University, Zhanjiang, China, since 2009. His research interests include computer networks, optimization, and data mining with their applications.