IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 24, 2020, accepted June 5, 2020, date of publication June 9, 2020, date of current version June 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001137

Graph Kernel Based Clustering Algorithm in MANETs

YING SONG[1], **HONGWEI LUO**[2], **SHANGCHAO PI**[3], **CHAO GUI**[1], **AND BAOLIN SUN**[1]

[1]School of Information and Communication Engineering, Hubei University of Economics, Wuhan 430205, China
[2]School of Information and Engineering, Wuhan International Trade University, Wuhan 430205, China
[3]Science College, Zhongyuan University of Technology, Zhengzhou 450007, China

Corresponding authors: Shangchao Pi (psc1000@126.com) and Baolin Sun (blsun@163.com)

This work was supported in part by the Natural Science Foundation of Hubei Province, China, under Grant 2018CFB661, and in part by the National Natural Science Foundation of China under Grant 61572012.

ABSTRACT The mobile ad hoc network (MANET) is a kind of dynamic, easy to construct and universal network, which has been widely concerned by a large number of researchers. Graph theory provides an effective theoretical tool for MANETs modeling and analysis. Clustering is one of the most effective methods to measure network performance with different attributes. This paper gives the basic concept of graph kernel and discusses the principle of optimizing graph kernel and multi-graph kernel. In this paper, we propose a Graph Kernel based Clustering Algorithm in MANETs (GKCA). The GKCA algorithm gives the basic concept of graph kernel, discusses the principle of optimizing graph kernel and multi-graph kernel, and proposes the basic principle based on d-hop graph kernel. GKCA algorithm uses shortest path (SP) to connect different cluster head nodes for packet transmission. The performance of GKCA algorithm, such as the control packets ratio, packets loss ratio, and average end-to-end delay are experimentally evaluated using network simulation (NS2) software. Experimental analysis shows that the proposed approach is efficient, and its performance advantage in dynamic mobile networks is promising.

INDEX TERMS MANET, graph kernel, clustering, performance analysis.

I. INTRODUCTION

A mobile ad hoc network (MANET) is an Internet made up of mobile nodes that are not supported by any base stations or infrastructure. Due to rapid development, it has been widely used in many fields such as military and civil [1]–[3]. In MANET, the nodes are mobile and the links are wireless. However, the network has many limitations, which include the transmission capacity of nodes, bandwidth, limited energy, unpredictable node connections, and many others. In order to maximize the data transmission capacity of MANET and prolong its lifetime, it is necessary to select the optimal routing and the node with the best performance for data processing and transmission [1]–[3].

The MANET structure can be divided into distributed network structure and cluster network structure. In distributed network structure, the roles of all nodes are equal, and the network topology can be adjusted freely according to the mobility of nodes. In clustering networks, the selection of cluster-head node is a key problem. Since the performance of cluster-head nodes is better than that of ordinary nodes, which requires stronger processing capacity and better energy. Thus, the selection of cluster-head nodes is an important factor in clustering network. Clustering can overcome routing protocol problems, improve data transmission quality, and help improve network scalability [4]. Clusters in MANET provide a stable way to connect mobile nodes and allocate resources efficiently, and provide a network layered environment through cluster to ensure the stability of MANET structure. The main feature of a MANET is that the network can be connected using a cluster-based hierarchy, while a larger network can be divided into smaller subgroups [4].

Graph theory provides an effective theoretical knowledge and technical tool for network modeling and analysis of Internet networks. Graph theory as an ancient branch of mathematics. Dating back to the famous Konigsberg's bridge problem, Euler proposed an elegant graph theory solution in 1736 [5]–[7]. Since then, graph theory has been advancing in a variety of successful applications, from traditional postman, traffic scheduling to modern Internet, mobile communication, cloud computing, artificial intelligence (AI) and many others [5]–[7].

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

In graph theory, graphs are usually abstracted from real world problems and can be generated in advance by random methods. For example, the cornerstone of a timing graph is to bridge the gap between point-based and period-based semantics, and between time graph traversal and static graph traversal [6]. The famous postman problem [7] can be modeled as an undirected weighted graph, whose vertices are cities, edges are roads, and edge weights are road lengths. Therefore, it can be used for analyzing the valuable attributes hidden in the graph or mining information patterns, such as small world events, global optimization, etc. However, for Internet analysis problems, neural network analysis problems, and artificial intelligence analysis problems, which cannot be solved by traditional chart analysis. Thus, learning from data is needed and graph mining learning methods needs to be designed.

This paper reviews the research progress of kernel theory and clustering protocol. The authors attempt to analyze and discuss the clustering performance by using the graph kernel theory, considering the properties between the cluster-head node and the surrounding node of MANET. The purpose of this paper is to provide a complete knowledge of graph kernel theory, as well as the design of clustering algorithm. Thus, the clustering environment of MANET can be correctly understood.

The major contributions of our work are as follows:

Firstly, we develop the basic concept of graph kernel, discuss the principle of optimizing graph kernel and multi-graph kernel, and put forward a basic principle based on d-hop graph kernel.

Secondly, we discuss the algorithm of clustering. Combining with the d-hop graph kernel, we propose a Graph Kernel based Clustering Algorithm in MANETs (GKCA). GKCA algorithm uses shortest path (SP) to connect different cluster-head nodes for packet transmission.

Finally, in the simulation environment, the GKCA algorithm is simulated and the simulation results are given. The key idea of the algorithm is to find the clustering schemes in the process of cluster-head selection criteria. The control packets ratio, packets loss ratio, and the average end-to-end delay are combined evaluation to stability and reliability of MANET.

This paper is organized as follows: Section II reviews the research work of MANET and graph kernel. Section III presents the theory and calculation methods of graph kernel, and the improved graph technique are also given. In Section IV, we provide multiple graph kernels, prediction model and GKCA algorithm. Simulation results are provided in section V. Finally, section VI concludes the paper.

II. RELATED WORK

In this section, we discuss relevant works with respect to our proposed GKCA algorithm. For the sake of convenience, the discussion is organized based on several aspects, including cluster algorithm, cluster-head selection, graph kernel theory, d-hop mechanism, multi-graph accounting, and others.

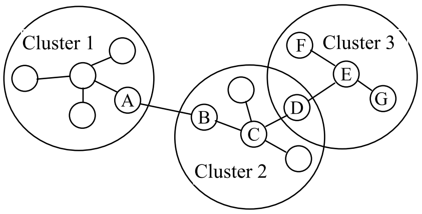


FIGURE 1. Cluster architecture.

A. THE CLUSTER ARCHITECTURE AND ALGORITHM

In MANET, the mobile nodes in the mobile cluster usually contain the following categories: cluster-head, border node, and cluster member.

The cluster-head nodes: a node is the cluster-head if it is a central node, or if it has a strong performance (or greater than a predicted threshold) link to the central node of another cluster.

The border node: a node is called a border node if its neighbor node belongs to another cluster.

Cluster member node: if a node is neither the cluster-head node nor a border node, it is a cluster member node.

Fig. 1 shows the cluster architecture. Mobile node A and node B are the border node connecting cluster 1 and cluster 2, mobile node C is the cluster-head node of cluster 2, mobile node D is the border node between cluster 2 and cluster 3, and mobile node E is the cluster-head node of cluster 3. The neighborhood set nodes of node E are node D, node F and node G.

In order to realize the organization of the nodes in the cluster and the generation of the cluster-head nodes, we adopt a progressive method, that is, each node is determined according to its neighbors and the signals it receives. The cluster-headed node declares that it is the cluster-headed node. All nodes adjacent to the cluster-head node check the signal strength and node performance received from it and declares that they are the cluster-head node. At the same time, the other nodes in the cluster examine the signal strength received from the adjacent nodes to determine the category of their nodes. If a node has a neighbor belonging to another cluster, it declares itself as a boundary node.

Aboutorab *et al.* [8] analyzed the problem of data grouping generation and recovery scenarios, which improved the performance of the network, but they did not consider the extraction and sharing of grouping. Aiming at the problem of free view video streaming in the network, Zhang *et al.* [9] proposed an algorithm to extract a part of anchor points from the server through a main channel for each user, which can improve the transmission of real-time free view video streaming in the network. Bayat *et al.* [10] proposed a peer-to-peer (P2P) video streaming framework in the overlay network, which solves the application problem of deploying real-time video streaming on the P2P overlay network. The framework supports decentralized decision making, fast crowding, and uses network coding algorithms to improve bandwidth utilization. Mobility Prediction-Based Clustering (MPBC) [11] algorithm estimated the relative speed of mobile nodes and proposed algorithms based on independent and random moving nodes. In MPBC algorithm, the velocity information of each moving node is first obtained, and then the cluster-head node information of the cluster is maintained, thus solving the relative movement problem of nodes.

In literature [12], a comparative study was conducted on the cluster scheme for locating mobile nodes and beacon sensor nodes, and a precise positioning algorithm based on distance and angle was proposed. This method can improve the energy efficiency of nodes in MANETs. Bentaleb *et al.* [13] proposed a QoS (Quality of Service) topology management and efficient k-hop scalability scheme for large-scale MANETs, which is suitable for urban environments. This scheme focuses on the construction and maintenance of MANET topology, including cluster formation stage, node joining stage and gateway selection stage. Prabha and Jeyanthi [14] proposed a new trust model, which uses behavioral trust, neighborhood trust, and historical trust to isolate malicious nodes in the routing process. Fuzzy rules can be used to determine the size of cluster, the optimal distance between cluster-head and member node, the optimal selection of cluster-head and the energy consumption of member node. The energy of each node is compared with the energy of the adjacent nodes with the level of movement. Through fuzzy modeling and energy modeling, efficient cluster-heads are selected. Aftab *et al.* [15] proposed a self-organizing clustering scheme based on regional group mobility in MANET to improve the stability and scalability of the overall network. The algorithm utilizes the biologically-inspired behavior of bird clusters to form and maintain MANET clusters. A dynamic cluster scale management mechanism is proposed to reduce network congestion and improve MANETs performance in cluster movement.

B. NEIGHBORHOOD AGGREGATION APPROACHES

As early as 2003, Kashima *et al.* [16] proposed that the comparison method of graphs is the basic theory of graph kernel, which has been widely used in the research of various graph theories and information theories since its birth. Since 2012, scholars have put forward several kernel theories specially designed for graphs with continuous attributes, and proved the feature representation technology of feature space of the graph kernel, etc., but the study of the graph kernel is still a challenging work. In the following, we will give an overview of some of the typical model pairs that work with the graphics kernel [17]. The working principle of neighborhood aggregation method is to assign an attribute to each node according to the local structure of neighboring nodes around the node, and so on [17]. For each node in the graph, the attributes of its neighbors will be aggregated into the cluster-head node to calculate a new attribute, which will eventually be extended to other neighborhood structures. Shervashidze *et al.* [18] proposed a heuristic algorithm based on 1-dimensional Weisfeiler-Lehman (1-WL), which is based on a class of highly influential neighborhood aggregation neural algorithms. The goal of graph cluster is to identify the connections between internal nodes, so as to establish a more compact cluster than external nodes. Ma *et al.* [19] introduced a new cluster mass fraction based on local motif rate, which can effectively respond to the density of clusters in high-order graphs, and proposed a motif-based local extended optimization algorithm (MLEO) to improve the clustering of local high-order graphs.

Gong and Ai [20] proposed a neighborhood adaptive graph convolutional network (NAGCN) based on efficient learning nodes. The NAGCN algorithm abstracts the neighborhood adaptive kernel from the diffusion process in order to learn and integrate the relevant neighborhood node information of each node more accurately. Wang *et al.* [21] used graph kernels to capture the local to global structural information of functional connectivity networks, and proposed a novel graph-kernel based structured feature selection (gk-SFS) method for brain disease classification based on functional connectivity networks.

C. NETWORK CONNECTIVITY

To predict the connectivity and correlation between two mobile nodes, the data interaction between the two mobile nodes is usually used for measurement [22]. The more interactive and similar the two mobile nodes are, the more likely the positive correlation between the two nodes is. The less interactive and similar two nodes are, the more likely they are to be negatively correlated. Given an undirected graph $G(V, E)$, V is the node set in G, E is the link set in G, suppose mobile node v_i and mobile node v_j are two nodes of graph G, the similarity of v_i and v_j is: $\mathrm{S}(v_i, v_j) = |N_1(v_i) \cap N_2(v_j)|$, where $N_1(v_i)$ is the neighbors of v_i in G, $N_2(v_j)$ is the neighbors of v_j in G, and $| \cdot |$ means the number of '$\cdot$'. Graph structure balance theory [23] considers four different ternary relationships between node v_i, node v_j and their common neighbor v_k. The structural balance theory is used to predict the links and cluster-heads. The related work first finds all the triad relations containing the target links, and then assigns the symbols to the target links to maintain the balance of the triad relations. Node information, link information and cluster-heads information reflect part information of MANETs. Nodes are widely connected in a MANETs. They are affected not only by themselves or the links that connect them, but also by other nodes and links that are not directly connected to them. Therefore, we consider structural information to predict link and cluster-heads of MANET.

In the G, minimum spanning tree (MST) is an acyclic connected subgraph with all vertices, and a tree with minimum weight is generated by search algorithm [24].

Given a set of vertices V, the Delaunay triangulation (DT) is defined as a circular hypersphere in which no vertices in V are located in any simplex circular hypersphere. The ε-N method gets the graph topology E by simply setting up the connection matrix $\mathbf{C}$, where ε is a pre-defined threshold. ε-N is a commonly used method for sparsifying both MANETs and social networks, where the connected matrix $\mathbf{C}$ is often a

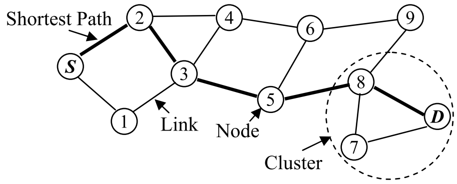


FIGURE 2. Shortest path graph.

Pearson's correlation matrix [24]. Kumar *et al.* [25] aimed at network connectivity problems such as dynamic network failure and network link disconnection caused by landslide prone areas and bad weather, and improved network connectivity according to geological attributes and demographic characteristics of nodes. A common way to predict whether two nodes are linked is to measure the interactivity and similarity between two nodes. If the two nodes are more interactive, the more positive relationship exists between the two nodes. On the contrary, there are different similarity relations, which are called negative correlation [26].

D. GRAPH KERNEL AND SHORTEST PATH (SP)

If there is a path between any pair of nodes in $V(G)$, graph G is called connected graph G, otherwise it is disconnected graph G. Paths, nodes, links and cluster are illustrated in Fig. 2. In graph theory, sparsity-inducing graph can provide very good robustness, high efficiency and interactivity, which provides an important promotion for the application of graph theory. Qiao *et al.* [27] designed a sparsely retained projection algorithm to reduce the dimension of L1 graphs by preserving them in a low-dimensional space. Low-rank graph learning algorithm is a joint lowest-rank representation method for finding the entire node set, which can better capture the global structure of data [28]. The theory and practice of graphs prove that the low-rank method is effective, especially in matrix compensation and robust subspace recovery [28].

In literature [29], the author used a similar local regularizer to learn low-rank graphs and further improved the theory. The most common way to compare two paths or subgraphs is to determine the best match between the nodes that make up the two objects, or to map the nodes of one subgraph to the structure of another. This method can also be applied to graph kernels, such as optimizing the allocation of graph kernels (Kriege *et al.* [30]). Lanneau *et al.* [31] proved the new polynomial calculation of the subfamily of perfect graphs, including claw-free perfect graphs and chord graphs, and based on the design of the kernel calculation method, gave two graph operations: clique-cutset decomposition and augmentation of flat edges.

The shortest path (SP) kernel is a typical application of graph kernel. The idea of a shortest path kernel is to compare the length and properties of the shortest path between all vertex pairs in two subgraph cores. A labeled graph $G(V, E, f)$ is a label function $f\colon V(G) \rightarrow N$ on the previous graph $G(V, E)$, where N is a positive integer, let $k(i, j)$ represent the SP distance between node i and node j in the same graph. The graph kernel is defined as,

$$k_{SP}(G, H) = \sum\sum k(i,j) \tag{1}$$

Using this algorithm, the time complexity of SP kernel is reduced to that of the existing the Weisfeiler-Lehman algorithm, which is in $O(n^3)$ [17]. Kriege *et al.* [32] have studied that under certain conditions, the algorithm of explicit calculation of graph kernel can process feature graph with higher efficiency. The algorithm is combined with several graph kernels (such as SP kernel) to improve the accuracy and efficiency of the dataset.

We supposed to have a random variable ζ_{SP} for each arc SP, and denote $\zeta_{SP} = \sum_{v \in SP} \zeta_v$ for each path SP. Given a source node $\boldsymbol{S}$ and a destination node $\boldsymbol{D}$, a SP problem typically seeks an $\boldsymbol{S}$–$\boldsymbol{D}$ path P minimizing a probability functional μ

$$\min_{P \in P_{SD}} \mu(\sum_{v \in P} \zeta_v) \tag{2}$$

under P path constraints of the form

$$\overline{\mu}(\sum_{v \in P} \zeta_v) \leq \beta \tag{3}$$

where $\overline{\mu}$ is a probability functional.

III. GRAPH KERNEL

MANET graph can be expressed as undirected graph $G = G(V, E)$, where V represents the set of wireless nodes in G, and E represents the set of wireless undirected edges in G. Wireless link $e = (i, j) \in E$ means that mobile node i can directly transmit packets to wireless node j, that is, node i is directly connected to node j. We assume that the wireless link is symmetric, that is, $(i, j) = (j, i) \in E$.

A. IMPROVED KERNEL

For an undirected graph G, several reduction rules for this problem can be given [17], [21], [32], [33].

Definition 1: Graph $G^f = (V, E, f)$ is a label graph, V is the node set, E is the edge set, and function $f\colon V(G) \rightarrow N$ to the graph $G(V, E)$, where N is a positive integer.

Definition 2: All the nodes directly connected by node j are called the neighbor set of node $j \in V$, represented by $N(j)$, i.e., $N(j) = \{i \in V | (j, i) \in E\}$. The size of its neighborhood is called the degree of the node, $\deg(i) = |N(i)|$.

Definition 3: A path in G can be represented as an ordered sequence of nodes, $P_{ij} = (i, \ldots, j)$. If $P_{ij} = (i, j)$, then node i and node j are directly connected.

Definition 4: In G, if there is a path between any pair of nodes in V, graph G is said to be connected, otherwise it is not connected.

Rule 1: In graph G, if the degree of node i is 0, node i can be removed from G.

Rule 2: For a node v in graph G, if the node v contains at least two neighbors of degree 1, denoted by $\{u_1, u_2, \ldots, u_i\}$ $(i \geq 2)$, then delete arbitrarily i-1 nodes from $\{u_1, u_2, \ldots, u_i\}$.

Rule 3: If a node i has two distinct neighbors x, y of degree 1, then delete node x or node y.

Rule 4 [17], [33]*:* If node i and j are two nodes such that $|N(i)\cap N(j)| \geq 2$ and if there exist two nodes $x, y \in N(i)\cap N(j)$ with deg(x) $=$ deg(y) $=$2, then node x can be deleted.

Definition 5: Assuming that undirected graph G has n nodes, if each node induced subgraph has an exact match of n-1 nodes, then G is called a factor-critical graph. For a matching M in G, if there happens to be a mismatched node in G, M is called a near-perfect match of G.

Theorem 1 [33]: Let G is an undirected graph reduced with respect to the rules 1, 2, 3 and 4, for which any induced matching contains at most k nodes. Then $|V| = O(k)$.

Proof: We assume that there is a maximum induced matching subgraph of size M at k, the maximum value of graph G. Thus, it can be proved that: either $|V| = O(k)$ is true, or M cannot be the maximum induced matching subgraph. According to the setting, we have: if M is the maximum induced matching subgraph of graph G, then for each node i, there is a node u, so that $d(i, j) \leq$2. Otherwise, we can add an edge to the match M to get a bigger induced match. Roughly speaking, each node in the graph G is at most two nodes $V(M)$ away, and each edge at M is at most four to at least one other edge M away. This leads the idea of regions "between" to the edge of matching each other. Thus, it can be obtained that if the graph is reduced according to the above data reduction rules, these regions are not too large. Furthermore, we have shown that it is impossible to have many nodes that are not contained in such a region.

Lemma 2: Given an undirected graph G and the maximum induced match M of graph G, there exists an algorithm to construct the maximum M region decomposition with $O(|M|)$ region.

Lemma 3: In an undirected graph G, each region contains $O(1)$ nodes in M regions simplified for decomposition.

Theorem 2: It is assumed that the undirected graph G is a reduced plane graph and M is a maximum induced match of G. Then there is an M region decomposition, so that the total number of nodes in all regions is $O(|M|)$.

Proof: Using lemma 2, the maximum M region decomposition on the maximum $O(|M|)$ region of the undirected graph G can be obtained. According to lemma 3, the number of nodes in each region is constant. Therefore, there are $O(|M|)$ nodes in the region.

B. SUBGRAPH GENERATION

In MANET, in order to obtain the information of wireless nodes, the similarity of wireless nodes can be obtained. In an undirected network graph, the information of a node can be represented by a set of subgraphs. The shortest path distance between nodes reflects the strength of the relationship between nodes: the shorter the distance, the stronger the relationship between nodes; the further the distance, the weaker the relationship between the nodes. Therefore, the shortest path distance between nodes can be used to construct the generation of subgraphs, and the nodes in different subgraphs have different strength of mutual relationship.

One drawback of the node and link label kernels is that they ignore the structure of the graph and the interaction between the labels, and they have almost no information for unlabeled graphs. The kernel can be calculated as a subgraph pattern. To avoid the problem of graph normalization, a graph invariant can be used, which in rare cases can map a non-isomorphic neighborhood subgraph to the same path. Then, the shortest path distance between these neighborhood graph pairs and their center nodes is characterized. We can first define the isomorphic subgraph and the d-hop subgraph, as shown below.

Definition 6: Two graphs are considered isomorphic if they have the same marker graph.

Definition 7: Given two graphs G_1 and G_2, a common subgraph G' is isomorphic to G_2 if it is isomorphic to G_1.

Algorithm 1 Common Subgraph Algorithm

Input: Graph G_1, G_2;

Output: Common subgraph G';

```
$G' = \phi$;
for all $v_i$ in $G_1$ do
  for all $v_j$ in $G_2$ do
    if $f(v_i) = f(v_j)$ then
      if match($v_i$, $v_j$) then
        add nodes to $G'$
      end if
    end if
  end for
end for
```

Algorithm 1 is mainly to extract the common subgraph. The algorithm first traverses the two graphs to find two similar nodes, and then creates a new node for the common subgraph structure to grow the rest of the common subgraph as a seed node. The algorithm recursively attempts to add new nodes to the graph.

Denfinition 8 [33]: (d-hop subgraph) Let $G = (V, E)$ represents a graph containing a set of vertices $V = \{v_1, v_2, \ldots, v_n\}$, also called nodes, and a set of undirected binary edges $E = \{(v_i, v_j)|\ (v_i, v_j) \in [-1, 1], v_i, v_j \in V\}$. $|V| = n$, $|E| = m$. Let SP(v_i, v_j) be the length of the shortest path between v_i and v_j. Node v_i's d-hop subgraph is defined as $N_d^G(v_i) = (V', E')$, in which d is the allowable length of the shortest hop between v_i and v_j, i.e., $\forall v_j \in V$ if SP(v_i, v_j) $\leq d$ then $v_j \in V'$, $d \in Z^+$; $\forall v_j, v_t \in V'$, if $(v_j, v_t) \in E$ then $(v_j, v_t) \in E'$.

In order to predict the characteristics of link (v_i, v_j), a set of subgraphs of node v_i and node v_j with hop 1 can be defined:

$$K_d^G(v_p) = \{N_1^G(v_p), N_2^G(v_p), \ldots, N_d^G(v_p)\} \tag{4}$$

where $v_p \in [v_i, v_j]$, subgraphs belonging to $K_d^G(v_p)$ represent the structural information of v_p according to different wireless connection strength.

C. GRAPH KERNEL CALCULATION

Graph kernel is a function of similarity between degree subgraph pairs, which allows artificial intelligence algorithms and optimization algorithms to operate directly on the graph kernel [34]. The method in this paper uses the graph kernel function proposed by Neumann *et al.* [35], because this function can calculate the number of links in time linearity and has good scalability in experiments.

First, each subgraph can be described by the k-order Krylov subspace, which is a set of vectors derived from its truncated power iterations. In this work, the k-order Krylov subspace can be used to mathematically represent the subgraph generated above. The k-order Krylov subspace is represented by mathematical notation that makes sense, and also produces some of the fastest linear algebraic algorithms for sparse matrices.

Secondly, the graph kernel of the Bhattacharyya kernel function quantum graph was used to calculate the similarity of k-dimensional Gaussian distributions representing k-order Krylov subspaces.

The similarity of two k-dimensional Gaussian distributions was calculated using the Bhattacharyya kernel [17], [25]. For two multidimensional Gaussian distributions $D_1(x)$ and $D_2(x)$, the similarity is:

$$K(D_1, D_2) = \int \sqrt{D_1(x)D_2(x)}dx \tag{5}$$

Since $D_1(x)$ and $D_2(x)$ follow Gaussian distributions, (5) can be transformed to (6).

$$K(D_1, D_2) = \frac{\left|\frac{Cov_1+Cov_2}{2}\right|}{\sqrt{|Cov_1|\,|Cov_2|}} \tag{6}$$

where Cov_1 is the covariance matrix of $D_1(x)$, Cov_2 is the covariance matrix of $D_2(x)$, and $|\cdot|$ is the determinant of matrix.

Since the multi-dimensional Gaussian distribution can be used to represent the subgraph pairs based on formula (6), the similarity of the subgraph G_1 and G_2 can be calculated as

$$S(G_1, G_2) = K(D_1, D_2) \tag{7}$$

where $D_1(x)$ and $D_2(x)$ are the multidimensional Gaussian distributions corresponding to graph G_1 and G_2, respectively. $K(D_1, D_2)$ is calculated by formula (6).

The similarity of node v_i and node v_j can be represented as

$$\begin{aligned} x(v_i, v_j) = [&S(N_1^G(v_i), N_1^G(v_j)), S(N_2^G(v_i), N_2^G(v_j)), \\ &\cdots, S(N_r^G(v_i), N_r^G(v_j))] \end{aligned} \tag{8}$$

IV. MULTIPLE KERNELS AND PREDICTION MODEL

A. MULTIPLE KERNELS RIDGE REGRESSION

In this section, the representation and model of ridge regression are considered, and a new model that performs clustering tasks and learning similarity relationship in kernel space is introduced. Kernel ridge regression (KRR) [36] is a nonlinear regression method, which uses the well-known graph technology to transfer time series $\{t_1, t_2, \ldots, t_n\}$ data schema is transformed nonlinearly into a high-dimensional feature space determined by a kernel function satisfying the Mercer's condition. Let the training data set contain n pairs, denote $(x_1, t_1), (x_2, t_2), \ldots, (x_n, t_n)$, where n is the number of inputs and a nonlinear mapping function $\varphi(x_i)$, while the original input space is transformed into a higher-dimensional feature space. The linear regression model is expressed as

$$y_i = \beta \bullet \varphi(x_i), \quad i = 1, 2, \ldots, n \tag{9}$$

where y_i is the i-th output, and β is the weight vector and is given by

$$\beta = [\beta_1, \beta_2, \ldots, \beta_n]^{\mathrm{T}} \tag{10}$$

The kernel ridge regression uses regularized least square method to obtain weight vector β by minimizing the objective function as follows:

$$\min L = \frac{1}{2}\|\beta\|^2 + \frac{1}{2}\cdot C \cdot \sum_{i=1}^{n} \zeta_i^2 \tag{11}$$

subject to

$$t_i - \beta \cdot \varphi(x_i) = \zeta_i, \quad i = 1, 2, \ldots, n \tag{12}$$

In Eq. (11), parameter $C(C > 0)$ is a regularization parameter and is the positive constant adjusted by the user, and is equivalent to the penalty coefficient of the squared error. The choice of C is $C = 2^\lambda$, $\lambda > 0$. Once the graph's parameter values are trained and the output weights are fixed, the graph assumes that the predictive time series data is ready.

Applying Lagrange multipliers to Eq. (12) the following expression is obtained:

$$L = \frac{1}{2}\|\beta\|^2 + \frac{1}{2}\cdot C \cdot \sum_{i=1}^{n} \zeta_i^2 + \sum_{i=1}^{n} \alpha_i(t_i - \beta\cdot\varphi(x_i) - \zeta_i) \tag{13}$$

By taking the derivative of L with respect to β, ζ, and α equating the resulting equations to zero, the output weight vector β is obtained as.

$$\beta = (\varphi^T\varphi + \tfrac{1}{C})^{-1}\varphi^T T \tag{14}$$

and the target vector $T = [t_1, t_2, \ldots, t_N]^T$.

Therefore, in the case of N node spaces, the obtained kernel matrix is

$$K = \begin{bmatrix} K(x_1, x_1) & K(x_1, x_2) & \ldots & K(x_1, x_N) \\ K(x_2, x_1) & K(x_2, x_2) & \ldots & K(x_2, x_N) \\ & \ldots & & \\ K(x_N, x_1) & K(x_N, x_2) & \ldots & K(x_N, x_N) \end{bmatrix} \tag{15}$$

B. CLUSTER HEAD ALGORITHM

The cluster-head algorithm is mainly used to form k-hop clusters. However, in order to select stable cluster-head nodes, a cluster-head selection algorithm based on graph kernel is proposed in MANET.

In MANET, each node has information about relative speed. The cluster-head algorithm can contain two parts: the

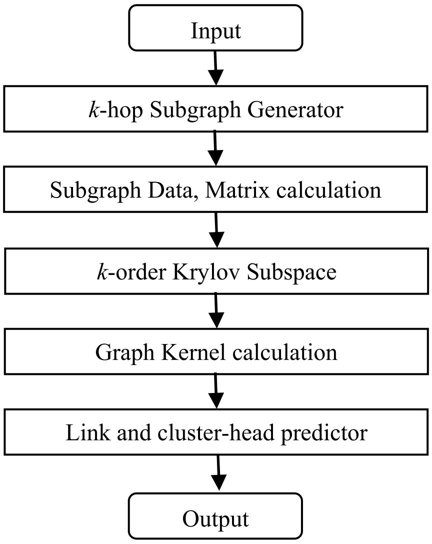

Input
k-hop Subgraph Generator
Subgraph Data, Matrix calculation
k-order Krylov Subspace
Graph Kernel calculation
Link and cluster-head predictor
Output

FIGURE 3. The architecture of our proposed algorithm.

cluster-head mechanism based on graph kernel selection and the cluster-head maintenance mechanism [37].

In the cluster-head algorithm, the maintenance part of the cluster uses the method based on graph kernel to solve the stability problem in the process of node movement. The maintenance part of the cluster-head is responsible for dealing with the problem of k hop and the node situation of the graph kernel, which guarantees the stability and connectivity of the cluster-head node.

The proposed algorithm compares the structural information of nodes to obtain their similarity and uses these similarities to predict links and cluster-heads. Our algorithm architecture is shown in Fig. 3. The input is the entire MANET. The output is the predicted link and cluster-header. The proposed method includes three stages: subgraph generation, kernel calculation and kernel classification.

C. GKCA ALGORITHM

It can be seen from literature that the singular KRR algorithm cannot produce accurate results in the prediction research. Therefore, multiple kernels ridge regression (MKRR) learning can be expressed as a combination of base nucleus and structural parameters of KRR [36]. This extension handles different heterogeneous data efficiently and performs better across a wider range of applications.

MKRR refers to the process of linearly combining the M specified kernels into a kernel K_{MKRR}:

$$K_{MKRR} = \sum_{m=1}^{M} \beta_m K^m \tag{16}$$

where, $\beta_i \geq 0$, (i $= 1, 2, \ldots, M$), $\beta_1 + \beta_2 + \ldots + \beta_M = 1$.

By definition, the kernel K_{MKRR} is symmetric and positive, and a feature space and a feature map are formed. Therefore, this kernel can be used for subsequent analysis, as it can provide a full sample summary. The combined kernel computes a kernel that minimizes the distortion between all input kernels. Algorithm 2 is the algorithm structure of GKCA.

Algorithm 2 Algorithm of GKCA

```
Input: Randomly generated network graph G;
       Predefined kernel matrices K;
       Hop count $h$;
       Parameters $\lambda$;
       The number of clusters $C$;
Output: Similarity matrix $K$;
        Calculate the cluster and cluster-head node;
initialize network graph $G$;
initialize L to identity matrix and parameters $\lambda$;
for $i = 1$ to $n$
    calculate $\beta, x_i, y_i$;
    calculate $\zeta_i, \alpha$;
    update $L$;
end for
for $i = 1$ to $n$
    calculate $K_d^G(v_p)$ in Eq. (4);
    calculate $K(D_1, D_2)$ in Eq. (6);
    calculate matrix $K$ in Eq. (15);
    calculate kernel $K_{MKRR}$ in Eq. (16);
    determine the cluster size and cluster-head node;
end for
```

D. TIME COMPLEXITY ANALYSIS

Here, the computational cost of GKCA is discussed. From Algorithm 2, the time complexity of GKCA is $O(n^3)$, where n denotes the number of instances. Besides the iterative updates, our model also needs $O(n^2 k)$ to construct the kernel, where $k = \sum_v k^v$ and k^v is the number of features of the v-th view. Thus the overall cost for GKCA is $O(n^3)$.

V. SIMULATION EXPERIMENT

A. SIMULATION MODEL

Several network scenarios datasets are used to evaluate the performance of the proposed approach. Network randomly generated datasets have been widely used in MANET network research, such as wireless link connection, wireless link bandwidth and mobile node transmission capacity.

In this section, we show the efficiency of our scheme through simulations conducted on NS2 (Network Simulator 2). The simulations range is 1000m×1000m in a 2-D free space with 100 mobile nodes. The radio transmission range is assumed to be 250 m. The source node and the destination node are randomly selected. The data sending speed of the source node is a constant bit rate (CBR), and each source node generates corresponding data packets according to the protocol for sending. In the simulation, the nodes move according to the random waypoint mobility model (RWP) with the minimum and maximum speeds setting to 0 and 20 m/s, respectively. Each simulation execution time is 600 seconds. Several simulation runs with different parameter values were carried out for each scenario execution,

TABLE 1. Simulation parameters.

Name	Parameter
Number of nodes	100
Map Size	1000m×1000 m
Transmission range	250 m
Average node degree	3-5
Simulation time	600 seconds
Node's mobility speed	0-20 m/s
Mobility model	Random waypoint mobility model
Communication model	Constant bit rate (CBR)
Connection Rate	1 Mbps
Node pause time	2 seconds
Examined routing protocol	GKCA, MPBC, NAGCN

and the average data was selected in these simulation runs. The free space propagation model is used in the simulation experiment. Table 1 lists some parameters in the simulation experiment.

B. PERFORMANCE METRICS

The performances of GKCA algorithm are compared with that of typical MPBC algorithm [11] and NAGCN algorithm [20] under the same movement model and communication model. MPBC algorithm is a clustering algorithm based on mobility prediction in MANET, which is more suitable for the rapid movement of nodes and the change of cluster-heads, and has some typical characteristics. NAGCN algorithm can construct a neighborhood adaptive kernel efficiently and collect more useful information about the neighborhood. NAGCN algorithm is a typical neighborhood cluster algorithm. The main performance parameters can be defined as follows:

The control packets ratio: the ratio of the number of control packets generated to recover the cluster-head to the data generated by the cluster-head.

The packets loss ratio: the ratio of the number of lost packets sent by the source node to the destination node to the total number of packets sent by the source node to the destination node.

Average end-to-end delay: the average value of the time that the received data packets take to reach the destination from their origin.

C. PERFORMANCE ANALYSIS

In order to evaluate the performance of GKCA algorithm based on graph kernel selection, we used NS-2 [38] simulation software recommended by IEEE 802.11 and with complete implementation extension mechanism to conduct simulation experiments. NS-2 is a discrete event simulator for network problems research. NS-2 provides a lot of simulation support for simulating Transmission Control Protocol, routing and cluster-head protocols on wired and wireless networks.

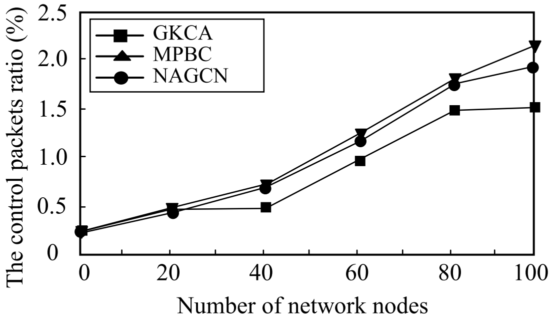


FIGURE 4. The control packets ratio vs. Network size.

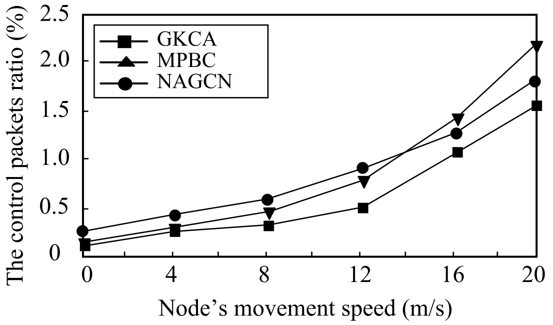


FIGURE 5. The control packets ratio vs. Node's movement speed.

Normalized network information, integrity, and accuracy are used to evaluate mobile node clusters. These index parameters are widely used in mobile node clustering with good positive correlation. The larger the number of mobile node clusters, the better the performance of the cluster-head.

Fig. 4 shows the performance comparison of GKCA algorithm with MPBC algorithm and NAGCN algorithm in control packets ratio as the number of MANET mobile nodes increases. When the number of network nodes increases, the number of cluster and cluster-head selection control packets also increases, so the control packets rate also increases. It can be seen from the experimental results in Fig. 4 that the control packets rate of GKCA algorithm is lower than that of MPBC algorithm and NAGCN algorithm, because GKCA algorithm uses the graph kernel method to select clusters and cluster-heads, resulting in relatively stable clusters and cluster-head nodes.

Fig. 5 compares the performance of GKCA algorithm in MANET with MPBC algorithm and NAGCN algorithm in control packet rate when the node movement speed increases. It can be seen in Fig. 5 that when the movement speed of mobile nodes increases, the link changes between mobile nodes are relatively large, and the changes of cluster construction and cluster head selection will increase, requiring more control groups to construct clusters and cluster head nodes. Fig. 5 also shows that GKCA algorithm selects clusters and cluster head nodes with stable performance, making the control packet rate of GKCA algorithm better than that of MPBC algorithm. This is mainly because GKCA algorithm uses the graph kernel mechanism to select clusters and keeps the stability of clusters.

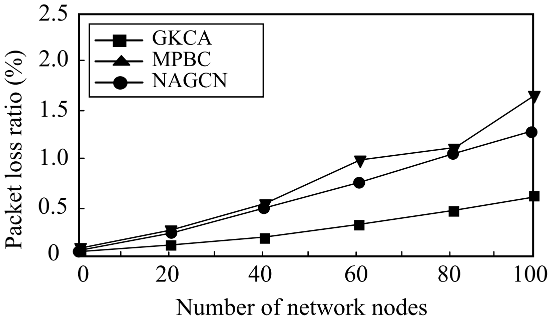


FIGURE 6. The packet loss ratio vs. Network size.

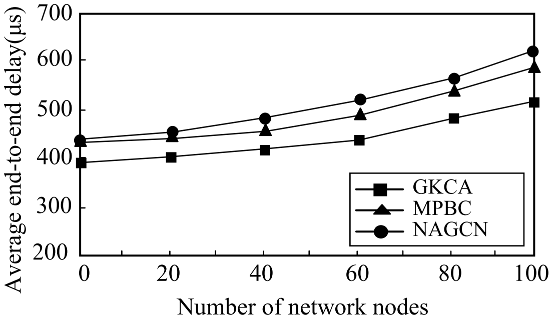


FIGURE 8. The average end-to-end delay vs. Network size.

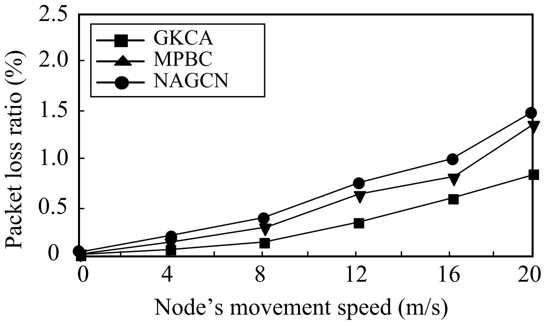


FIGURE 7. The packet loss ratio vs. Node's movement speed (m/s).

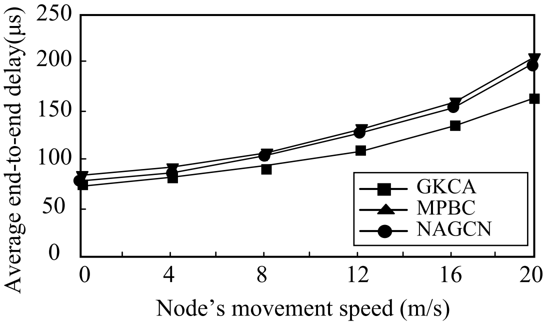


FIGURE 9. The average end-to-end delay vs. Node's movement speed (m/s).

Fig. 6 shows the relationship between packet loss rate and network size. When the network size is small, the packet loss rate of GKCA algorithm, MPBC algorithm and NAGCN algorithm is very small. When the number of network nodes increases gradually, the packet loss rate also increases gradually. When the network scale increases, the packet loss rate of GKCA algorithm does not increase significantly. This is mainly because GKCA algorithm uses the structure of graph kernel to select the cluster and cluster-head node with excellent performance, thus ensuring the better transmission of data packet.

Fig. 7 shows the relation between the packet loss rate and the node movement speed. When the node mobile speed is low, the packet loss rate of GKCA algorithm is very similar to MPBC algorithm and NAGCN algorithm. When the node movement speed is high, the cluster head node, the stability of the cluster and the connectivity between nodes are also poor, and the packet loss rate is also increasing, but the packet loss rate of GKCA algorithm is the lowest. GKCA algorithm can choose the cluster with better performance and the cluster head node. This is mainly because the GKCA algorithm uses the structure of the graph kernel to select the cluster and cluster head nodes with good performance, selects the cluster and cluster head nodes with better performance, and guarantees the transmission of data packets.

Fig. 8 shows the performance comparison of GKCA algorithm with MPBC algorithm and NAGCN algorithm when the size of network nodes increases. With the increasing of node number, GKCA algorithm is obviously better than that of MPBC algorithm and NAGCN algorithm. As shown in Fig. 8, the average end-to-end delay of GKCA is at most 10-20% larger than that of MPBC algorithm and NAGCN algorithm. It demonstrates that the GKCA algorithm is more stable with the variation of the network size.

Fig. 9 shows the comparison of the average end-to-end delay performance of network data packets when the movement speed of network mobile nodes changes from 0 to 20 m/s. As can be seen from Fig. 9, when the movement speeds of mobile nodes increases, the data packet transmission delay also increases slowly, but the performance of GKCA algorithm is obviously better than that of MPBC algorithm and NAGCN algorithm. The increase of motion speed leads to more frequent topological changes, which leads to an increase in the probability of chain break and a longer reconnection time of links. From Fig. 9, it can be seen that when the node's mobility speed increases, GKCA algorithm has lower average end-to-end delay in higher mobility environment.

VI. CONCLUSION

Due to the good scalability and adaptability of MANET network in the case of environmental change, it can be deployed as an emergency network when other networks fail in the case of disaster or combat. We propose a Graph Kernel based Clustering Algorithm in MANETs (GKCA). The key idea of the protocol is to find the clustering schemes in the process of cluster-head selection criteria. The control packets ratio, packets loss ratio, average end-to-end delay are combined to evaluate the cluster-head. The performance evaluation of our proposed methods is accomplished via modeling and simulation. The simulation results demonstrate that the proposed approach and parameters provide an efficient method of estimating and evaluating the cluster-head stability in dynamic mobile networks. Further work to improve the algorithm includes the support of nodes with limited mobility.

REFERENCES

[1] B. L. Sun, C. Gui, Y. Song, and H. Chen, "Network coding-based maximum lifetime algorithm for sliding window in WSNs," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 3, pp. 1298–1310, 2016, doi: 10.3837/tiis.2019.03.010.

[2] Q. Zhang, Y. Song, B. Sun, and Z. Dai, "Design of routing protocol for opportunistic network based on adaptive motion," *IEEE Access*, vol. 8, pp. 18228–18239, 2020, doi: 10.1109/ACCESS.2020.2968598.

[3] T. Rahman, I. Ullah, A. U. Rehman, and R. A. Naqvi, "Clustering schemes in MANETs: Performance evaluation, open challenges, and proposed solutions," *IEEE Access*, vol. 8, pp. 25135–25158, 2020, doi: 10.1109/ACCESS.2020.2970481.

[4] L. J. Gerstein, "Integral quadratic forms and graphs," *Linear Algebra Appl.*, vol. 585, pp. 60–70, Jan. 2020, doi: 10.1016/j.laa.2019.09.032.

[5] Y. Wang, H. Sun, Y. Zhao, W. Zhou, and S. Zhu, "A heterogeneous graph embedding framework for location-based social network analysis in smart cities," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2747–2755, Apr. 2020, doi: 10.1109/TII.2019.2953973.

[6] J. Byun, S. Woo, and D. Kim, "ChronoGraph: Enabling temporal graph traversals for efficient information diffusion analysis over time," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 424–437, Mar. 2020, doi: 10.1109/TKDE.2019.2891565.

[7] K. Wang, G. Wu, X. Hou, D. Wei, H. Liu, and J. Qiu, "From groupwise to individual brain functional networks parcellation and application," *Chin. Sci. Bull.*, vol. 61, no. 27, pp. 3022–3035, Sep. 2016, doi: 10.1360/N972016-00585.

[8] N. Aboutorab, P. Sadeghi, and S. Sorour, "Enabling a tradeoff between completion time and decoding delay in instantly decodable network coded systems," *IEEE Trans. Commun.*, vol. 62, no. 4, pp. 1296–1309, Apr. 2014, doi: 10.1109/TCOMM.2014.021614.130172.

[9] B. Zhang, Z. Liu, S.-H.-G. Chan, and G. Cheung, "Collaborative wireless freeview video streaming with network coding," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 521–536, Mar. 2016, doi: 10.1109/TMM.2016.2518485.

[10] N. Bayat and H. Lutfiyya, "Network coding for coping with flash crowd in P2P multi-channel live video streaming," in *Proc. 11th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Mar. 2015, pp. 24–27, doi: 10.1109/DRCN.2015.7149019.

[11] M. Ni, Z. Zhong, and D. Zhao, "MPBC: A mobility prediction-based clustering scheme for ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 9, pp. 4549–4559, Nov. 2011, doi: 10.1109/TVT.2011.2172473.

[12] I. Ullah, J. Chen, X. Su, C. Esposito, and C. Choi, "Localization and detection of targets in underwater wireless sensor using distance and angle based algorithms," *IEEE Access*, vol. 7, pp. 45693–45704, 2019, doi: 10.1109/ACCESS.2019.2909133.

[13] A. Bentaleb, S. Harous, and A. Boubetra, "A k-hop scalable management scheme for large scale mobile ad hoc networks," *Wireless Pers. Commun.*, vol. 96, no. 4, pp. 6239–6271, Oct. 2017, doi: 10.1007/s11277-017-4475-x.

[14] K. P. Rama Prabha and N. Jeyanthi, "A trust and fuzzy cluster based dynamic secure routing algorithm for mobile ad hoc networks," *Wireless Pers. Commun.*, vol. 98, no. 3, pp. 2959–2974, Feb. 2018, doi: 10.1007/s11277-017-5010-9.

[15] F. Aftab, Z. Zhang, and A. Ahmad, "Self-organization based clustering in MANETs using zone based group mobility," *IEEE Access*, vol. 5, pp. 27464–27476, 2017, doi: 10.1109/ACCESS.2017.2778019.

[16] H. Kashima, K. Tsuda, and A. Inokuchi, "Marginalized kernels between labeled graphs," in *Proc. 20th Int. Conf. Mach. Learn.*, Aug. 2003, pp. 321–328.

[17] N. M. Kriege, F. D. Johansson, C. Morris, "A survey on graph kernels," *Appl. Netw. Sci.*, vol. 5, pp. 1–42, Dec. 2020, doi: 10.1007/s41109-019-0195-3.

[18] N. Shervashidze, P. Schweitzer, E. J. V. Leeuwen, K. Mehlhorn, K. M. Borgwardt, "Weisfeiler-lehman graph kernels," *J. Mach. Learn. Res.*, vol. 12, no. 9, pp. 2539–2561, 2011.

[19] W. Ma, L. Cai, T. He, L. Chen, Z. Cao, and R. Li, "Local expansion and optimization for higher-order graph clustering," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8702–8713, Oct. 2019, doi: 10.1109/JIOT.2019.2923228.

[20] P. Gong and L. Ai, "Neighborhood adaptive graph convolutional network for node classification," *IEEE Access*, vol. 7, pp. 170578–170588, 2019, doi: 10.1109/ACCESS.2019.2955487.

[21] M. Wang, B. Jie, W. Bian, X. Ding, W. Zhou, Z. Wang, and M. Liu, "Graph-kernel based structured feature selection for brain disease classification using functional connectivity networks," *IEEE Access*, vol. 7, pp. 35001–35011, 2019, doi: 10.1109/ACCESS.2019.2903332.

[22] A. Nocaj, M. Ortmann, and U. Brandes, "Adaptive disentanglement based on local clustering in small-world network visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 6, pp. 1662–1671, Jun. 2016, doi: 10.1109/TVCG.2016.2534559.

[23] C. Lu, J. X. Yu, R.-H. Li, and H. Wei, "Exploring hierarchies in online social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2086–2100, Aug. 2016, doi: 10.1109/TKDE.2016.2546243.

[24] L. Qiao, H. Zhang, M. Kim, S. Teng, L. Zhang, and D. Shen, "Estimating functional brain networks by incorporating a modularity prior," *NeuroImage*, vol. 141, pp. 399–407, Nov. 2016, doi: 10.1016/j.neuroimage.2016.07.058.

[25] S. Kumar, S. Duttagupta, V. Rangan, M. V. Ramesh, "Reliable network connectivity in wireless sensor networks for remote monitoring of landslides," *Wireless Netw.*, vol. 26, no. 3, pp. 2137–2152, 2020, doi: 10.1007/s11276-019-02059-7.

[26] C. Si, L. Jiao, J. Wu, and J. Zhao, "A group evolving-based framework with perturbations for link prediction," *Phys. A, Stat. Mech. Appl.*, vol. 475, pp. 117–128, Jun. 2017, doi: 10.1016/j.physa.2017.01.087.

[27] L. Qiao, S. Chen, and X. Tan, "Sparsity preserving projections with applications to face recognition," *Pattern Recognit.*, vol. 43, no. 1, pp. 331–341, Jan. 2010, doi: 10.1016/j.patcog.2009.05.005.

[28] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013, doi: 10.1109/TPAMI.2012.88.

[29] K. Tang, Y. You, Z. Su, W. Jiang, and J. Zhang, "Structure-constrained low-rank representation," *J. Comput.-Aided Des. Comput. Graph.*, vol. 31, no. 4, pp. 589–595, 2019, doi: 10.3724/SP.J.1089.2019.17348.

[30] N. M. Kriege, P.-L. Giscard, and R. C. Wilson, "On valid optimal assignment kernels and applications to graph classification," in *Proc. 30th Conf. Neural Inf. Process. Syst. (NIPS)*, Barcelona, Spain, Dec. 2016, 1623-1631.

[31] A. Pass-Lanneau, A. Igarashi, and F. Meunier, "Perfect graphs with polynomially computable kernels," *Discrete Appl. Math.*, vol. 272, pp. 69–74, Jan. 2020, doi: 10.1016/j.dam.2018.09.027.

[32] N. M. Kriege, M. Neumann, C. Morris, K. Kersting, and P. Mutzel, "A unifying view of explicit and implicit feature maps of graph kernels," *Data Mining Knowl. Discovery*, vol. 33, no. 6, pp. 1505–1547, Nov. 2019, doi: 10.1007/s10618-019-00652-0.

[33] G. Tan, Q. Feng, B. Zhuo, N. Huang, and J. Wang, "New kernels for several problems on planar graphs," *Theor. Comput. Sci.*, vol. 806, pp. 587–594, Feb. 2020, doi: 10.1016/j.tcs.2019.09.024.

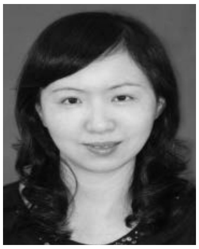
[34] W. Yuan, K. He, D. Guan, L. Zhou, and C. Li, "Graph kernel based link prediction for signed social networks," *Inf. Fusion*, vol. 46, pp. 1–10, Mar. 2019, doi: 10.1016/j.inffus.2018.04.004.

[35] M. Neumann, R. Garnett, C. Bauckhage, and K. Kersting, "Propagation kernels: Efficient graph kernels from propagated information," *Mach. Learn.*, vol. 102, no. 2, pp. 209–245, Feb. 2016, doi: 10.1007/s10994-015-5517-9.

[36] J. Naik, R. Bisoi, and P. K. Dash, "Prediction interval forecasting of wind speed and wind power using modes decomposition based low rank multi-kernel ridge regression," *Renew. Energy*, vol. 129, pp. 357–383, Dec. 2018, doi: 10.1016/j.renene.2018.05.031.

[37] S. Huang, Z. Kang, I. W. Tsang, and Z. Xu, "Auto-weighted multi-view clustering via kernelized graph learning," *Pattern Recognit.*, vol. 88, pp. 174–184, Apr. 2019, doi: 10.1016/j.patcog.2018.11.007.

[38] *The Network Simulator—NS-2*. [Online]. Available: http://www.isi.edu/nsnam/ns/



YING SONG received the master's degree in geographical information system and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2004 and 2011, respectively. She is currently an Associate Professor at the School of Information and Engineering, Hubei University of Economics, Wuhan. She has published over 30 research articles. Her research interests include wireless communication, mesh networks, and network protocol.

HONGWEI LUO received the M.S. degree from Wuhan University, China, in 2003. He is currently a Lecturer of computer science and technology at Wuhan International Trade University, China. His research interests include wireless communication and network optimization.

SHANGCHAO PI received the B.Sc. degree from the Mathematics Department, Central China Normal University, Wuhan, China, in 1983. He is currently a Professor at the Science College, Zhongyuan University of Technology, Zhengzhou, China. He has published over 30 research articles. His research interest includes differential equation and its application.

CHAO GUI received the master's degree in computer science and technology from Wuhan University, Wuhan, China, in 1989. He is currently a Professor at the School of Information and Engineering, Hubei University of Economics, Wuhan. He has published over 60 research articles. His research interests include wireless communication, performance analysis, and modeling.

BAOLIN SUN received the master's and Ph.D. degrees in computer science and technology from the Wuhan University of Technology, China, in 1999 and 2006, respectively. He is currently a Professor at the School of Information and Engineering, Hubei University of Economics, Wuhan, China. He was awarded the Province Special Prize by the Hubei Province Government, in 2007. He is a member of IAENG, one of the Editorial Board Guest Members of World Sci-Tech Research and Development, and also an International Standard Draft organizing members of ISO/IEC JTC1/SC6. His research interests include multipath routing, parallel and distributed computing, network optimization, and ad hoc networks. He has published over 150 journal and conference papers and has authored four books in the above areas.