Hindawi
Scientific Programming
Volume 2022, Article ID 3943419, 17 pages
https://doi.org/10.1155/2022/3943419

Research Article

Keywords-Driven Paper Recommendation Based on Mobile Edge Computing Environment Framework

Hanwen Liu,[1] Shuo Wang,[1] Huali Ren,[2] Shunmei Meng,[1] Jun Hou,[3,4] and Qianmu Li[1,5]

[1] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210096, China*
[2] *Institution of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou 510006, China*
[3] *School of Social Science, Nanjing Vocational University of Industry Technology, Nanjing 210046, China*
[4] *Intelligent Manufacturing Department, Wuyi University, Jiangmen 529020, China*
[5] *School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing 210096, China*

Correspondence should be addressed to Qianmu Li; qianmu@njust.edu.cn

Received 23 January 2022; Revised 1 March 2022; Accepted 24 March 2022; Published 8 April 2022

Academic Editor: Jiwei Huang

In the cloud computing era, a paper recommender system is usually deployed on the cloud server and return recommendation results to readers directly. However, considering the paper recommender system, processing tremendous paper citation data on the cloud cannot provide fine-grained personalized and real-time recommendations for each reader because these recommended papers from the cloud are far from readers and probably not correlated strongly with each other for helping each reader research further and deeper in the interested field. Recently, the edge-cloud collaboration-based recommender system has been used for releasing parts of the cloud computing task to the edge and provides the recommendation near the client. Based on the edge computing recommender system, a keywords-driven and weight-aware paper recommendation approach is presented, namely, $LP\text{-}PR_{k+}\ w$ (link prediction-paper recommendation), to enable intelligent, personalized, and efficient paper recommendation services in the mobile edge computing environment. Specifically, the whole paper recommendation process mainly covers two parts: optimizing the existing paper citation graph via introducing a weighted similarity (i.e., building a weighted paper correlation graph) and then recommending a set of correlated papers according to the weighted paper correlation graph and the users' query keywords. Experiments on a real-world paper correlation dataset, Hep-Th, show the capability of our proposal for improving the paper recommendation performance and its superiority against other related solutions.

1. Introduction

Generally, recommender system is based on the cloud-client structure that raises the risk of delays due to network bandwidth and latency, data leakage during remote transmission, and recommending uncorrelated results for users by processing tremendous multivariate heterogeneous data. In recent years, edge-cloud collaboration-based recommender systems are proposed to solve these problems [1, 2]. In the mobile edge computing environment, the recommender system can utilize the real-time information of users on the edge end to provide better recommendations.

Currently, the paper recommender system is one of the major tools and ways for readers to find their required papers. For example, popular academic paper search tools on the mobile edge end (e.g., Baidu Academic and Google Scholar) allow readers to look for their interesting papers from massive papers registered on the web, based on a set of desired keywords. In practice, one paper only covers readers' partial query keywords. Therefore, the paper search tools probably recommended a set of papers that contain all the query keywords to meet the readers' tastes for papers.

As shown in the right part in Figure 1, the fundamental paper recommendation process mainly consists of three phases in the cloud computing environment. First, a reader types a set of query keywords into the paper recommender system (e.g., $\{k_1, k_2, k_3, k_4, k_5\}$). The second phase is papers' discovery, in which vast candidate papers can be identified

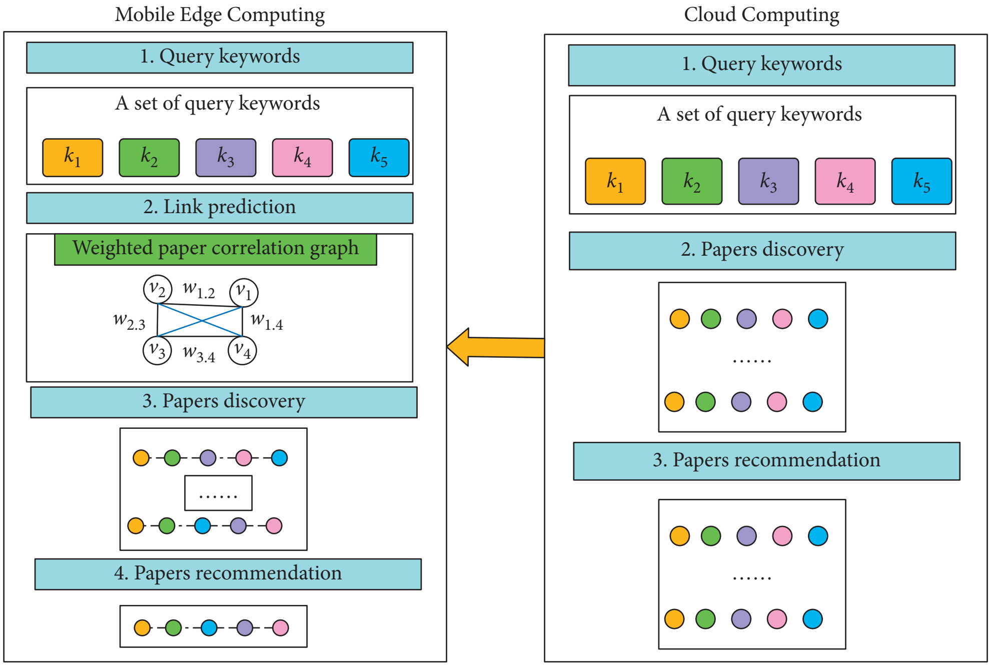


Figure 1: The recommendation process on the cloud computing and the mobile edge computing.

by using traditional keyword search methods [3]. The third phase is papers' recommendation, in which the recommender system will output all candidate papers to the reader.

However, finding a set of desirable papers from massive candidates on the cloud server for the reader is often a sophisticated job even for an experienced reader. The reasons are mainly threefold, which are clarified as follows:

(1) As the paper recommender system usually returns recommendation results to readers directly in the cloud computing, the paper recommender system cannot provide fine-grained personalized and real-time recommendation for readers.

(2) The returned paper list should meet the following conditions. First, these papers collectively cover all the query keywords from the reader. Second, the recommended papers should own certain correlations with each other so as to aid the reader in launching in-depth and continuous research on an identical idea.

(3) Traditional keywords-based paper recommendation approaches hardly analyze the possible correlations among different papers. Although paper citation graphs provide a good indicator of paper correlation relationships, they still face the serious problem of sparse data.

Currently, link prediction approaches are the best alternatives to address the sparse data problem of graphs (networks) [4], which aims to find the missing links and forecast future links based on known graphs (networks) information [5]. Specifically, link prediction approaches estimate the likelihood of building a new link among two unconnected nodes according to their graph structure and attribute information. Hence, applying link prediction to paper recommendations is of theoretical and practical significance.

To satisfy readers' requirements on paper search, we propose a novel keywords-driven and weight-aware paper recommendation approach deployed on the mobile edge end, that is, $LP-PR_{k+w}$ (link prediction-paper recommendation). As shown in the left part in Figure 1, $LP-PR_{k+w}$ integrates multiple operations: query keywords, link prediction, papers' discovery, and papers' recommendation. On the mobile edge end [6], through analyzing a reader's query keywords, $LP-PR_{k+w}$ can return a set of correlated papers (depicted by a connected subgraph) by mining the potential correlation patterns hidden in the weighted paper correlation graph. As output, the returned subgraph is interconnected by the target papers (containing query keywords) and bridging papers (containing no query keywords). Particularly, we consider interchangeability of one paper and its corresponding node in this paper, denoted as p or v.

Generally, we achieve the following main contributions:

(1) We propose a novel keywords-driven and weight-aware paper recommendation approach based on the mobile edge computing framework, i.e., $LP-PR_{k+w}$, which enables intelligent, personalized, and efficient paper recommendation services in the mobile edge computing environment.

(2) We optimize the existing paper citation graph model by introducing weighted similarity-based link prediction. Thus, we construct a weighted paper correlation graph.

(3) To evaluate the usefulness and feasibility of $LP - PR_{k+w}$, we conduct a group of experiments on a real-world paper citation dataset, Hep-Th.

The rest of the paper is organized as follows. Related researches are summarized in Section 2. Paper motivation is demonstrated in Section 3. In Section 4, we introduce the weighted similarity-based link prediction approach. Section 5 answers readers' keywords query via the proposed $LP - PR_{k+w}$ solution. Section 6 analyzes and evaluates $LP - PR_{k+w}$ through experimental comparisons. Finally, we summarize the paper and point out the future research directions in Section 7.

2. Related Work

2.1. Paper Recommendation. In general, Collaborative Filtering (CF) approach can calculate the similarity scores among different items; thus, paper recommender systems apply the CF approach and focus on ratings matrices. Furthermore, the CF approach can also calculate the ratings' matrixes that are created on a paper citation graph [7]. Therefore, the early works of paper recommendation mainly used the CF approach to recommend papers. With the application of the CF approach in the paper recommendation, researchers found that the CF approach is generally restricted to cold-start and data sparsity problems [8]. In addition, Content-Based Filtering (CBF) approach is similar to the CF approach. The CBF [9] approach mainly focuses on the content relevance [10, 11] among papers to recommend papers. However, these recommended papers hardly match readers' deep and continuous research around an identical focus. Furthermore, the CBF encounters the semantic ambiguity problem.

As the citation relationships between papers can reflect the correlations among papers' research content, a graph-based approach becomes research focus. For example, Meng et al. [12] regarded authors, papers, topics, and keywords as nodes and multiple relationships as edges; the approach executed a random walk on the constructed four-layer heterogeneous graph to recommend papers. Furthermore, [13] proposed a graph-based PageRank-like paper recommendation approach, which mainly executed a biased random walk on paper citation graphs to recommend papers. Although [12] and [13] take papers' citation relationships into account, these approaches do not tackle the data sparsity in the existing paper citation graphs.

2.2. Link Prediction. Link prediction [14] is an important approach of resolving links' sparsity problem as it can compute the likelihood of building new links between two unconnected nodes. At present, three types of link prediction have been identified, that is, similarity-based method, maximum likelihood approach, and probabilistic method. The similarity-based method was used in large-scale networks as it could compute the similarity score among any two nodes. The maximum likelihood approach predicted links by utilizing specific parameters, and the probabilistic method could employ the trained model to forecast links [15]. However, these two approaches did not apply to the large-scale networks. In the paper, our proposal mainly employs the similarity-based method considering that an existing paper citation graph is a large-scale paper relationships network [16].

Currently, more researchers considered utilizing the weight of links in link prediction approaches. For example, the work of [17] tried to identify strong ties (e.g., spouses or romantic partners) in social network, and these strong ties were considered as the different link weight among users. Furthermore, [18] investigated the use of link strength on link prediction approaches; their proposed weighting criterion was based absolutely on topological data (i.e., the frequency of interactions among nodes). In addition, link prediction model could be categorized into two types: an unsupervised link prediction model and a supervised link prediction model. For example, [18] and [19] made use of user's attributes information (i.e., gender, age, and sex) and interactive activities to estimate the social relationships strength on unsupervised link prediction model. Furthermore, [20] proposed a supervised link prediction model that used similarity metrics to calculate the similarity eigenvectors between a pair of nodes, and these eigenvectors were regarded as new databases for constructing the supervised model in predictive tasks [21].

In view of the above research content, a novel keywords-driven and weight-aware paper recommendation approach, that is, $LP - PR_{k+w}$, is proposed in the paper, to cope with the sparse data problem and recommend a set of correlated papers. Next, actual examples are presented in Section 3 to further demonstrate the research motivation of the paper.

3. Research Motivation

In this section, the examples of Figures 2 and 3 are recruited to directly demonstrate our research motivation. Figure 2 describes a scenario that a reader needs to perform five keywords research before he creates a new paper: (1) *keyword search* (i.e., k_1) is used by readers to search for interesting papers by typing query keywords; (2) *paper citation graph* (i.e., k_4) is for finding the correlation relationships among different papers (i.e., mining the potential correlation patterns); (3) *link prediction* (i.e., k_2) *is* to solve the data sparsity of paper citation graph; (4) *Steiner tree* (i.e., k_3) [22] is utilized to find a set of correlated papers; and (5) *dynamic programming* (i.e., k_5) [23] is applied to solve the Steiner tree problem. Thus, the reader obtains five corresponding query keywords, $Q = \{k_1, k_2, k_3, k_4, k_5\}$.

Figure 3 is a part of an undirected paper citation graph. It contains 14 nodes (papers), that is, $v_1, \ldots, v_{14}$, covering diverse keywords. The notation v_{13} $\{k_{11}, k_{13}\}$ $\{2021\}$ $\{a_1\}$ indicates that node v_{13} offers keywords k_{11} and k_{13}, an author a_1, and the published time 2021. $e(v_1, v_{10})$ indicates that nodes v_1 and v_{10} generate an undirected citation relationship.

According to the query keywords, the reader easily obtains a set of papers in Figure 3 (i.e., $R_p = \{v_1, v_2, v_3, v_4, v_5, v_6, v_{11}, v_{12}\}$). However, these returned papers fail to satisfy the reader deep and continuous

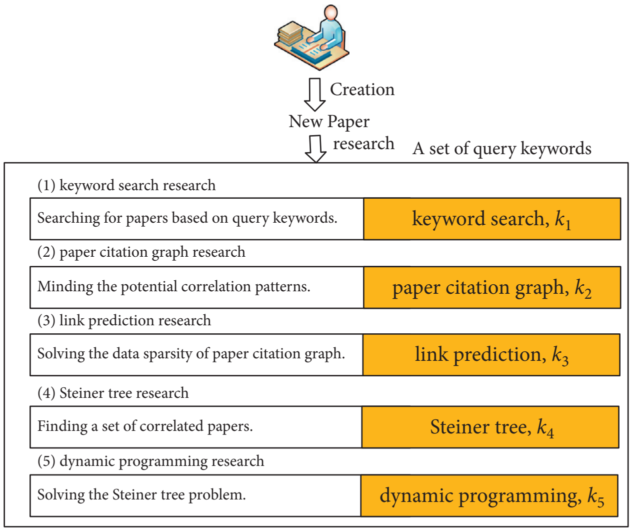

Creation
New Paper
research
A set of query keywords
(1) keyword search research
Searching for papers based on query keywords.
keyword search, k_1
(2) paper citation graph research
Minding the potential correlation patterns.
paper citation graph, k_2
(3) link prediction research
Solving the data sparsity of paper citation graph.
link prediction, k_3
(4) Steiner tree research
Finding a set of correlated papers.
Steiner tree, k_4
(5) dynamic programming research
Solving the Steiner tree problem.
dynamic programming, k_5

Figure 2: The paper research and creation tasks: an example.

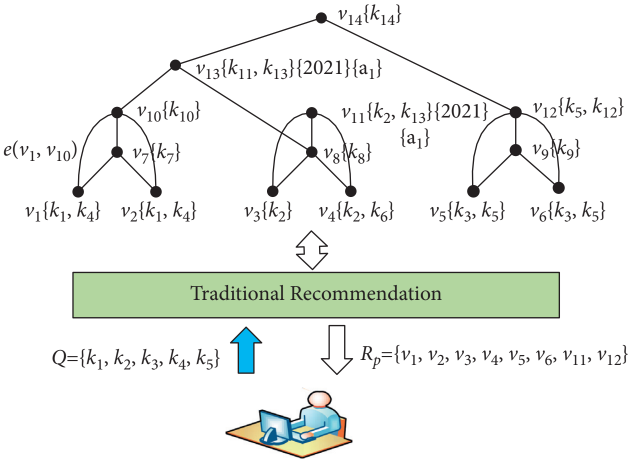


Figure 3: Tradition recommendation: an example.

academic research around an identical focus. In fact, a paper recommender system generally recommends a larger number of candidate papers to readers; furthermore, the correlation relationships among candidate papers are transparent to readers. Therefore, readers hardly select a set of correlated papers from these candidate papers. Luckily, a paper citation graph depicts citation relationships among diverse papers, so it provides a promising way to mine the potential correlation patterns. However, the paper citation graph faces to the sparse data problem; that is, it does not consider the potential correlations among different papers not connected in the graph. For example, nodes v_{11} and v_{13} have common research content, while they fail to build a correlation relationship in Figure 3.

According to the above examples, in the mobile edge computing environment, we firstly address the data sparsity of paper citation graph and then recommend a set of correlated papers, these contents are presented in detail in Section 4 and Section 5, respectively.

4. Link Prediction of *LP-PR*$_{k+w}$

4.1. Weighted Similarity-Based Link Prediction Approach. According to the analysis of the research motivation, we propose a weighted similarity-based link prediction schema that follows tasks' sequence [24]. In Figure 4, the link prediction process mainly comprises the following activities:

Activity 1. Preprocessing of Graph. For simplicity, a directed paper citation graph is treated as an undirected graph.

Activity 2. Nodes' Weighting. This activity mainly calculates the actual weight (i.e., w_{ij}) between all nodes of the undirected paper citation graph. Here, the weights of two connected nodes and two unconnected nodes are both computed by employing the following KTA weighting criterion.

Criterion. Keywords, Time, and Authors (KTA). Here, the number of common keywords of two papers increases and their published time are relatively close; thus, their weight (i.e., w_{ij}) will be larger. Furthermore, the theory of [25] states that two different papers containing common authors tend to refer to each other. Thus, the KTA weighting criterion is as follows:

$$\text{cosine}\left(K_{v_i}, K_{v_j}\right) = \frac{\left|K_{v_i} \cap K_{v_j}\right|}{\sqrt{\left|K_{v_i}\right|} * \sqrt{\left|K_{v_j}\right|}},$$

$$k(t) = \frac{1}{1 + e^{-\left|t_{v_i} - t_{v_j}\right|}},$$

$$\text{cosine}\left(A_{v_i}, A_{v_j}\right) = \frac{\left|A_{v_i} \cap A_{v_j}\right|}{\sqrt{\left|A_{v_i}\right|} * \sqrt{\left|A_{v_j}\right|}},$$

$$\begin{aligned} w_{ij} = w^{\text{KTA}}\left(v_i, v_j\right) = \beta^{\left(1-cosine\left(K_{v_i}, K_{vj}\right)\right)} \\ \times \lambda^{k(t)} \times \alpha^{\left(1-cosine\left(A_{v_i}, A_{vj}\right)\right)}, \end{aligned} \tag{1}$$

where β, λ, and α are arbitrary parameter values, i.e., β, λ, and $\alpha \in (0, 1)$. Higher (or lower) values of β, λ, and α intensify (or attenuate) the influence of keywords, published time, and authors in the weighting criterion, respectively. K_{v_i} (K_{v_j}) and A_{v_i} (A_{v_j}) denote a set of keywords and authors, respectively. $A_{v_i} \cap A_{v_j}$ and $K_{v_i} \cap K_{v_j}$ represent co-authors and common keywords, respectively. $\text{cosine}(A_{v_i}, A_{v_j})$ and $\text{cosine}(K_{v_i}, K_{v_j})$ denote that the method computes the similarity authors of and keywords between two nodes, respectively. t_{v_i} and t_{v_j} indicate the published time of two nodes, respectively.

Activity 3. Score Calculation and Ranking.

(1) Firstly, we get the actual weight of two connected nodes (e.g., v_i and v_z, v_j and v_z). Then, we calculate

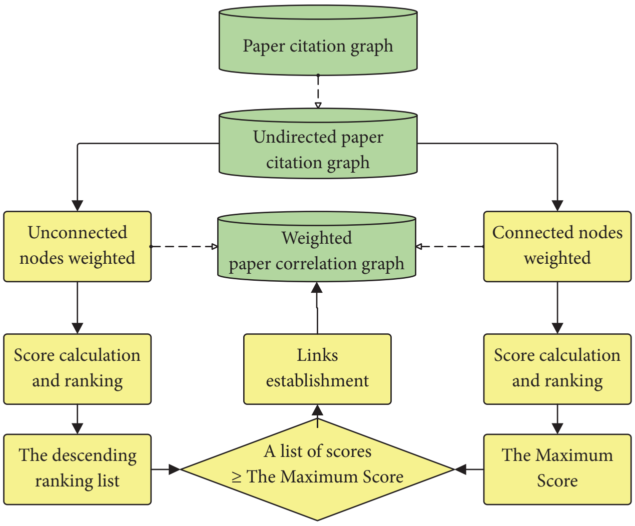

Paper citation graph
Undirected paper citation graph
Unconnected nodes weighted
Weighted paper correlation graph
Connected nodes weighted
Score calculation and ranking
Links establishment
Score calculation and ranking
The descending ranking list
A list of scores ≥ The Maximum Score
The Maximum Score

Figure 4: Process of the weighted similarity-based link prediction: overview.

the weight of two nonconnected nodes (e.g., v_i and v_j) by using the weighted similarity function (i.e., weighted common neighbor). Finally, we sort a descending list in order, and the maximum score [26] is saved in $w_{\max}\ (v_i,\ v_j)$.

Weighted Common Neighbor—WCN (v_i, v_j) [20]. It computes the average weight between two nonconnected nodes v_i and v_j; that is,

$$WCN(v_i, v_j) = \sum_{v_z \in \Gamma(v_i) \cap \Gamma(v_j)} \frac{w^{\mathrm{KTA}}(v_i, v_z) + w^{\mathrm{KTA}}(v_z, v_j)}{2},$$

$$WCN(v_i, v_j) = \frac{WCN(v_i, v_j)}{\left|\Gamma(v_i) \cap \Gamma(v_j)\right|},$$

$$w_{\max}(v_i, v_j) = \arg\max_{i,j=1,\ldots,N} w^{\mathrm{WCN}}(v_i, v_j), \qquad (2)$$

where (6) calculates the weight of two nonconnected nodes v_i and v_j by using the WCN. $|\Gamma(v_i) \cap \Gamma(v_j)|$ represents the number of common neighbor nodes. Equation (2) obtains a maximum weight (i.e., the maximum score) over the undirected paper citation graph.

(2) Here, we directly use the actual weight of all unconnected nodes to produce a descending-ranking list.

Activity 4. Links Establishment. This activity connects a pair of unconnected nodes. *LP* (link prediction) is defined as

$$LP = \{w_{ij} \ge w_{\max}(v_i, v_i)\}. \qquad (3)$$

4.2. Weighted Paper Correlation Graph. In the mobile edge computing environment, $LP - PR_{k+w}$ builds a weighted paper correlation graph by using the weighted similarity-based link prediction. The weighted paper correlation graph is defined as follows.

Definition 1 (Nodes). For one paper p, the weighted paper correlation graph has a mapped node v. And the node covers diverse keywords (i.e., $k_1 \ldots\ k_z$), where these keywords denote its main research content.

Definition 2 (Edges). For any pair of nodes $(v_i,\ v_j)$, the weighted paper correlation graph has a corresponding edge $e(v_i,\ v_j)$. And $e(v_i, v_j)$ denotes the fact that two papers have a correlation relationship.

Definition 3 (Weights). For any two connected nodes v_i and v_j, the weight of w_{ij}/w_{ij} (larger than 0) directly indicates the strength of correlation between nodes v_i and v_j. For example, considering two edges $e(v_1,\ v_2)$ and $e(v_2,\ v_3)$, if $w_{1,2} > w_{2,3}$, then node v_2 has better "similarity" (correlation) with node v_1 instead of node v_3.

Definition 4 (Weighted Paper Correlation Graph (W-PCG)). The W-PCG is expressed in $G_w(V_p, E_p, W_p)$, where V_p, E_p, and W_p represent a set of nodes, edges, and weights.

According to Definition 1, each node of G_w contains diverse keywords. To expediently answer the keywords query of Section 5, $LP - PR_{k+w}$ needs to preestablish an inverted index S_k. Concretely, given a query keyword k, we can faster search for all papers containing keyword k. For example, nodes v_1 and v_2 cover keywords k_1 (i.e., $S_k\{k_1\} = \{v_1,\ v_2\}$).

5. Paper Recommendation of $LP\text{-}PR_{k+}w$

5.1. Problem Formalization of Keyword-Driven Pattern Mining. On the mobile edge end [27], according to reader's query keywords, we will introduce our paper recommendation approach based on a W-PCG (i.e., G_w). Specifically, given a query Q containing $l\ (l \ge 2)$ query keywords (i.e., $Q = \{k_1,\ \cdots, k_l\}$), our proposal can find optimal answer trees on G_w, denoted as $T_w\ (Q)$, where $T_w\ (Q)$ is not only a connected tree containing all query keywords (i.e., Q) but also having highest correlation. To better clarify the paper, we summarize the symbols in Table 1.

Figure 5 is a part of G_w and contains the same nodes of Figure 3 (i.e., $v_1, \cdots, v_{14}$). And a new link $e(v_{11}, v_{13})$ (i.e., blue line) is added in Figure 5. According to the query keywords of Figure 2, i.e., $Q = \{k_1, k_2, k_3, k_4, k_5\}$, nodes v_1 and v_2 contain keywords k_1 and k_1, nodes v_3, v_4, and v_{11} contain keyword k_2, nodes v_5 and v_6 contain keywords k_3 and k_5, and node v_{12} contains keyword k_5. Thus, an answer tree $T_w\ (Q)$ connects one node from $\{v_1, v_2\}$, one node from $\{v_3, v_4, v_{11}\}$, one node from $\{v_5, v_6\}$, and one node from $\{v_5, v_6, v_{12}\}$; furthermore, $T_w\ (Q)$ may connect nodes that do not contain any query keywords, i.e., v_{10}, v_{13}, and v_{14}.

Table 1: Symbol definition.

Symbol	Definition
p/v	A paper
$k_1, \ldots, k_z$	The paper contains keywords
$e(v_i, v_j)$	A correlation relationship
Q/K	A set of query keywords
A_{v_i}/K_{v_i}	A set of authors/keywords
V_p	A set of nodes
E_p	A set of edges
W_p	A set of weights
$G_w(V_p, E_p, W_p)$	W-PCG
S_k	An inverted index
T_w (Q)	An optimal answer tree
Q^1/Q^2	Two queues
$T_{w\min}$ (v, K')	A minimum group weighted Steiner tree rooted at v
R_P	Recommendation result

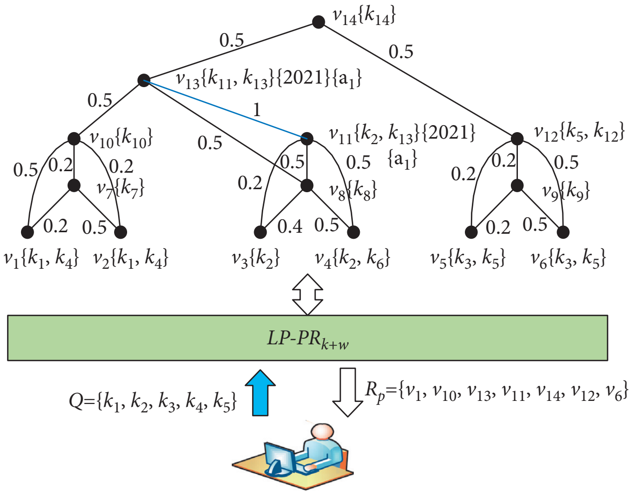


Figure 5: $LP - PR_{k+w}$: the same paper recommendation example.

Therefore, the reader will obtain a set of papers that is different from Figure 3 (i.e., $R_p = \{v_1, v_{10}, v_{13}, v_{11}, v_{14}, v_{12}, v_{16}\}$).

According to the above example, in the mobile edge computing environment, the reader initially pursues a weighted Steiner tree T_w (Q) [28], and the definition is as follows.

Definition 5 (Weighted Steiner Tree). Given a W-PCG (i.e., G_w) and a set of nodes $V_p' \subseteq V_p$, when T_w (Q) is a connected subgraph covering all nodes of V_p', T_w (Q) is a weighted Steiner tree.

Considering the inverted index of Section 4, we recognize multiple groups of nodes according to diverse query keywords from $Q = \{k_1, \ldots, k_l\}$, denoted as $V_{p1}, \ldots, V_{pl}$, where V_{pn} $(1 \le n \le l)$ is a set of nodes covering a query keyword k_n $(1 \le n \le l)$. Next, we need to find a group weighted Steiner tree that covers all query keywords. The definition is as follows.

Definition 6 (Group Weighted Steiner Tree). Given G_w and multiple groups of nodes $V_{p1}, \ldots, V_{pl} \subseteq V_p$, when a weighted Steiner tree w_{ij} (Q) selects nicely one node from each group V_{pn} $(1 \le n \le l)$, T_w (Q) is a group weighted Steiner tree.

There may be obtained multiple diverse group weighted Steiner trees in answer to reader's keywords query. In fact, our recommendation goal, in the mobile edge computing environment, is simply to recommend a set of "most correlative" papers covering all query keywords. Here, the weight T_w of G_w denotes the "similarity" (correlations) of nodes v_i and v_i. Hence, the recommendation goal is finding a group weighted Steiner tree with maximal weight. In practice, $LP - PR_{k+w}$ considers the weight-aware (i.e., correlation-aware) paper recommendation problem as an optimization problem, and the object function is of "the smaller weight the better" case. Here, we transform Figure 5 into Figure 6. In Figure 6, the weight w_{ij} of edge $e(v_i, v_j)$ is represented by $1/w_{ij}$. For example, when $w_{1,10} = 0.5$ holds in Figure 5, $w_{1,10} = 2$ holds in Figure 6. In the remainder of this paper, we only use a converted W-PCG (i.e., G_w) as illustration. Next, we must search for a group weighted Steiner tree with minimum weight. The definition is as follows.

Definition 7 (Minimum Group Weighted Steiner Tree). Given a set of alternative group weighted Steiner trees, that is, T_{w1} $(Q), \ldots, T_{wm}$ (Q), when w $(T_{wi}$ $(Q)) = min$ $(w$ $(T_{w1}$ $(Q)), \ldots, w$ $(T_{wm}$ $(Q)))$ $(1 \le i \le m)$, T_{wi} (Q) is a minimum group weighted Steiner tree.

5.2. Searching for Optimal Answer via Pattern Mining. According to readers' keywords query Q, $LP - PR_{k+w}$ must search a minimum group weighted Steiner tree T_w (Q) representing final solution for readers via mining on a W-PCG (i.e., G_w). However, the computation of the T_w (Q) is the NP-complete problem. Therefore, we will utilize a dynamic programming (DP) technique [28] to solve this problem.

Generally, the DP technique firstly divides the MGWST (minimum group weighted Steiner tree) problem into a set of easier subproblems. Next, each same subproblem is addressed only once and the DP technique stores the corresponding result. Finally, the DP technique can exactly provide optimal solutions to readers by combining previously saved results.

In DP model, T_w (v, K') $(K' \subseteq K = Q)$ is a state, which covers a set of query keywords K'. Furthermore, w $(T_w$ $(v, K'))$ denotes the weight of T_w (v, K'). The model state transitions equations are defined as follows.$w(\underset{\forall v \in V_p}{T_{w\min}}(v, K')) = \infty$ if $\underset{\forall v \in V_p}{T_{w\min}}(v, K')$ does not contain any query keywords:

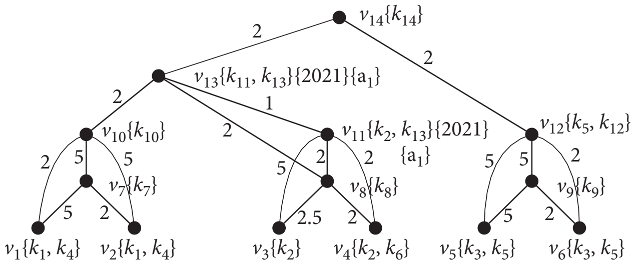


Figure 6: The W-PCG converted from Figure 5.

$$w\left(\underset{\forall v\in V_k\subseteq V_p}{T_{w\min}}(v,K')\right) = 0\ \text{if}\left|\underset{\forall v\in V_k\subseteq V_p}{T_{w\min}}(v,K')\right| = 1, \tag{4}$$

$$w(T_{w\min}(v,K')) = \min\left(w\left(T_{wg}(v,K')\right), w(T_{wm}(v,K'))\right), \tag{5}$$

$$w\left(T_{wg}(v,K')\right) = \min_{u\in N(v)}\left\{w(T_{w\min}(u,K'_u)) + w_{u,v}\right\}, \tag{6}$$

$$w\left(T_{wg}(v,K')\right) = \min_{\left(\substack{K'_u\cap K'\neq K'_u \,\&\,\& K'_u\cap K'=K' \\ \|\ K'_u\cap K'\neq K' \,\&\,\& K'_u\cap K'=K'_u}\right)}\{w(T_{w\min}(v,K')) + w(T_{w\min}(u,K'_u))\}, \tag{7}$$

$$w(T_{wm}(v,K')) = \min_{\left(\substack{K'_u\subseteq K\,\&\,\&K'\subseteq K\,\&\,\& \\ K'_u\cap K'\neq K'\,\&\,\&K'_u\cap K'\neq K'_u}\right)}\{w(T_{w\min}(v,K')) + w(T_{w\min}(u,K'_u))\}, \tag{8}$$

where $N(v)$ is a set of neighbor nodes of node v; u, v $\in\in G_w(V_p, E_p, W_p)$, $u \in\in N(v)$, $e(u,v) \in E_p$, and $\mathrm{w}_{\mathrm{u,v}} \in\in W_p$. Since tree $T_{w\min}$ (v, K') does not contain any query keywords, the weight of the tree is infinite in formula (3). In formula (10), a tree covers one query nodes and its weight is 0. Formula (5) denotes that tree $T_{w\min}$ (v, K') is acquired by doing the following operations: tree growth operation (i.e., formulas (7) and (8)) and tree merging operation (i.e., formulas (9) and (10)). In Figure 7(a), the tree growth operation creates new tree T_w (v, K') by adding new node u, and the pseudocode of the tree growth operation is specified in Algorithm 1. In Figure 7(b), the tree merging operation creates new tree T_w (v, K') by merging two trees; these two trees are both rooted at v. And the pseudocode of the tree merging operation is specified in Algorithm 2. When $K' = K$, $LP - PR_{k+w}$ searches for a minimum group weighted Steiner tree $T_{w\min}$ (v, K') in formula (11), and $T_{w\min}$ (v, K') contains at least one node.

$$\begin{aligned} w(T_{wm}(v,K')) &= \min_{\substack{K'_1\cap K'_2=\Phi \\ K'_1\cup K'_2=K'}}\{w(T_{w\min}(v,K'_1)) + w(T_{w\min}(v,K'_2))\}, \\ T_w(Q) &= T_{w\min}(v,K')\ \text{if}\ K' = K. \end{aligned} \tag{9}$$

According to formulas (4)–(11), T_w (Q) is returned by repeating the tree growth and the tree merging operations, and the pseudocode of the MGWST algorithm is specified in Algorithm 3. Furthermore, the algorithm maintains two queues: Q^1 and Q^2: the queue Q^1 records intermediate trees only containing partial query keywords, and the queue Q^2 records qualified trees covering all query keywords K.

Note that, in the worst-case scenario, the MGWST algorithm may return the entire W-PCG as its output. In addition, the algorithm does not take into account the role of

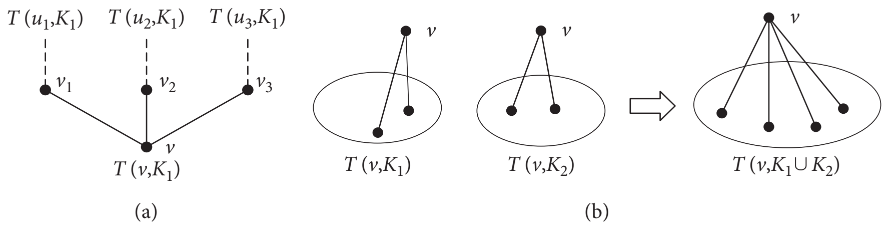


FIGURE 7: Tree operations. (a) Tree growth. (b) Tree merging.

```
Input: $K = \{k_1, k_2, \ldots, k_l^{\},}\ Q^1 = \Phi$
Output: $Q^1$
For each $u \in N(v)$ do
    If $w_{u,v} + w(T_{w\min}(u, K_u')) < w(T_{w\min}(v, K'))$
        $T_{w\min}(v, K') = e(u, v) + T_{w\min}(u, K_u')$
        enqueue $T_{w\min}(v, K')$ into Q1
        update Q1
    End If
    If $(K_u' \cap K' \neq K' \&\& K_u' \cap K' = K_u' \,||\, K_u' \cap K' \neq K_u' \&\& K_u' \cap K' = K')$
            $w(T_{w\min}(u, K_u')) + w(T_{w\min}(v, K')) < w(T_{w\min}(v, K'))$
    $w(T_{w\min}(v, K')) = w(T_{w\min}(u, K_u')) + w(T_{w\min}(v, K'))$
        enqueue $T_{w\min}(v, K')$ into Q1
        update Q1
    End If
    Return Q1
End For
```

ALGORITHM 1: Tree growth.

```
Input: $K = \{k_1, k_2, \ldots, k_l\}$, Q1 $= \Phi$
Output: Q1
For each $u \in N(v)$ do
    If $(K_u' \subseteq K \&\& K' \subseteq K \&\& K_u' \cap K' \neq K_u' \&\& K_u' \cap K' \neq K')$
        $T_{w\min}(v, K') = T_{w\min}(u, K_u') + T_{w\min}(v, K')$
        enqueue $T_{w\min}(v, K')$ into Q1
        update Q1
    End If
End For
$K_1' = K'$
    For each $T_{wmw\min in}(v, K_2')$ in Q1 s.t$(K_1' \cap K_2' = \Phi \,\&\& K_1' \bigcup K_2' = K')$
        If $w(T_{w\min}(v, K_1')) + w(T_{w\min}(v, K_2')) < w(T_{w\min}(v, K_1' \bigcup K_2'))$
            $T_{w\min}(v, K_1' \bigcup K_2') = T_{w\min}(v, K_1') \oplus T_{w\min}(v, K_2')$
            enqueue $T_{w\min}(v, K_1' \bigcup K_2')$ into Q1
            update Q1
    End If
    Return Q1
End for
```

ALGORITHM 2: Tree merging.

the synonymy and word inflections in the paper recommendation [29–35] process.

6. Experiments

To evaluate the usefulness and feasibility of $LP - PR_{k+w}$, the large-scale experiments are tested on Hep-Th dataset [36].

6.1. Experimental Environment

Dataset. We select part of the Hep-Th data set for the experiment (i.e., these papers are published between 1997 and 2003); the partial data contains 8721 papers and forms a paper citation graph. Here, each paper node stores paper published time and authors' information. Furthermore, the keywords' information of each paper is constructed by employing the RAKE (rapid automatic keyword extraction) technique.

Experiment Settings. To obtain an optimal W-PCG (i.e., G_w), we execute the link prediction process depicted in Figure 4. Furthermore, we also set the parameters values of the KTA weighting criterion: α, $ß \in \{0.3, 0.5, 0.7, 0.9\}$, $\lambda \in \{0.3, 0.9\}$.

In the keywords query experiments, three diverse sets of the keywords are set, i.e., set A, set B, and set C. In set A, the keywords of one paper are regarded as the query keywords, where readers can provide all query keywords for their research content. In set B, the query keywords are randomly selected from different papers (in excess of one paper) as the readers' research content covers diverse research topics. In set C, the query keywords are randomly selected from any two papers to further verify the feasibility of the MGWST algorithm. Furthermore, an author of each paper basically creates the number of keywords with up to 6, so there are up to six query keywords in each set. Each experiment is repeated 100 times and the average experiment results are adopted.

Evaluation Criteria. We compare the following evaluation criteria:

(1) Number of new edges: the larger, the better; that is, more new edges denote that our link prediction approach can better solve the sparse problems of the existing paper citation graph.

(2) Number of nodes: number of recommended papers (the smaller, the better, i.e., fewer papers denote that high correlation). Note that the return solution contains at least one paper.

(3) Success rate [37]: when the number of papers of a recommendation result is less than twice the number of the readers' query keywords, the recommendation result is successful (the larger, the better).

(4) Weight: the weight of a recommended result (the smaller, the better).

(5) Computation time: the time for generating an answer tree (the smaller, the better). Here, the computation time can be well described by logarithmic function (log_2).

(6) Precision [20] is defined as

$$\text{precision} = \frac{TP}{R_p}, \tag{10}$$

where TP denotes a set of papers containing query keywords and R_p denotes a recommendation result.

(7) Recall [38] is defined as

$$\text{recall} = \frac{1}{|p|} \sum_{p_a \in p} \frac{R_p \cap T_{p_a}}{T_{p_a}}, \tag{11}$$

where $|p|$ is the number of papers (e.g., $|p| = 2$) in set C. T_{p_a} is a set of papers cited by p_a.

(8) $F1$ score is defined as

$$F1\,\text{score} = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}}. \tag{12}$$

Here, we compare $LP - PR_{k+w}$ with several similar paper recommendation approaches:

Baseline 1. Link Prediction-Random (*LP-Random*) [28]. The approach is randomly finding a set of nodes from the G_w, and these nodes collectively cover readers' query keywords. Finally, this approach grows a minimum weighted spanning tree.

Baseline 2. Link Prediction-Greedy (LP-Greedy) [28]. Likewise, the approach selects a set of nodes contained all query keywords from the G_w. Next, this approach regards these nodes as the initial root nodes and continuously generates a tree until these nodes are interconnected.

Baseline 3. Link Prediction-Random Walk (LP-RW) [13]. First, using papers' keywords and the correlation relationships of G_w to build 2-layer graph, and then the approach runs on the 2-layer graph to recommend papers. Furthermore, each query only uses readers' entered keywords: $q = [0, qW]$.

Baseline 4. Link Prediction-Random Walk Restart (LP-RWR) [12]. Likewise, the approach also uses the 2-layer graph for paper recommendation. If the state vectors of LP-RWR steadily grows linearly in the experiment, then we consider the approach achieves linear convergence.

Experimental Tools. All experiments are implemented by python and executed with Intel® Core® CPU @ 3.0 GHz, 16 GB RAM, and Windows 10 @ 1809, 64-bit operating system.

6.2. Experimental Results

(1) *Profile 1: The Number of New Edges.*

In this profile, we compare the number of new edges to select a set of appropriate parameters values that are better at solving the sparsity of an existing paper citation graph in some extent. Tables 2–3 present that we mostly gain different experimental results as paper keywords, paper published time, and paper authors' information have a significant role in the link prediction process. According to the above experiments results, when $\alpha = 0.3$, $ß = 0.5$, and $\lambda = 0.9$, we get the best experimental results; that is, the number of new edges is 348. Therefore, we select the set of appropriate parameters values to build a W-PCG.

(2) *Profile 2: The Number of Nodes of Different Approaches.*

```
Input: K = $\{k_1, k_2, \ldots, k_l\}$, Q1 = $\Phi$, Q2 = $\Phi$
Output: Q1, Q2, $T_{w\min}(v, K)$
Let Q1 = $\Phi$, Q2 = $\Phi$
For each $v \in V_p$ do
    If v contains any nonempty keyword set $K' \subset K$
        $w(T_{w\min}(v, K')) \neq \infty$
        enqueue $T_{w\min}(v, K')$ into Q1
    End If
End For
While Q1 $\neq \Phi$ do
    dequeue Q1 to $T_{w\min}(v, K')$
    If $K'$ = K
        enqueue $T_{w\min}(v, K')$ into Q2
        Continue
    End If
    Else tree growth
    Else tree merging
    Return $T_{w\min}(v, K)$ = Q2.top ()
End While
```

Algorithm 3: **MGWST** (G_w, K).

Table 2: α, β, and $\lambda = 0.3$ are employed in the link prediction to obtain the number of new edges.

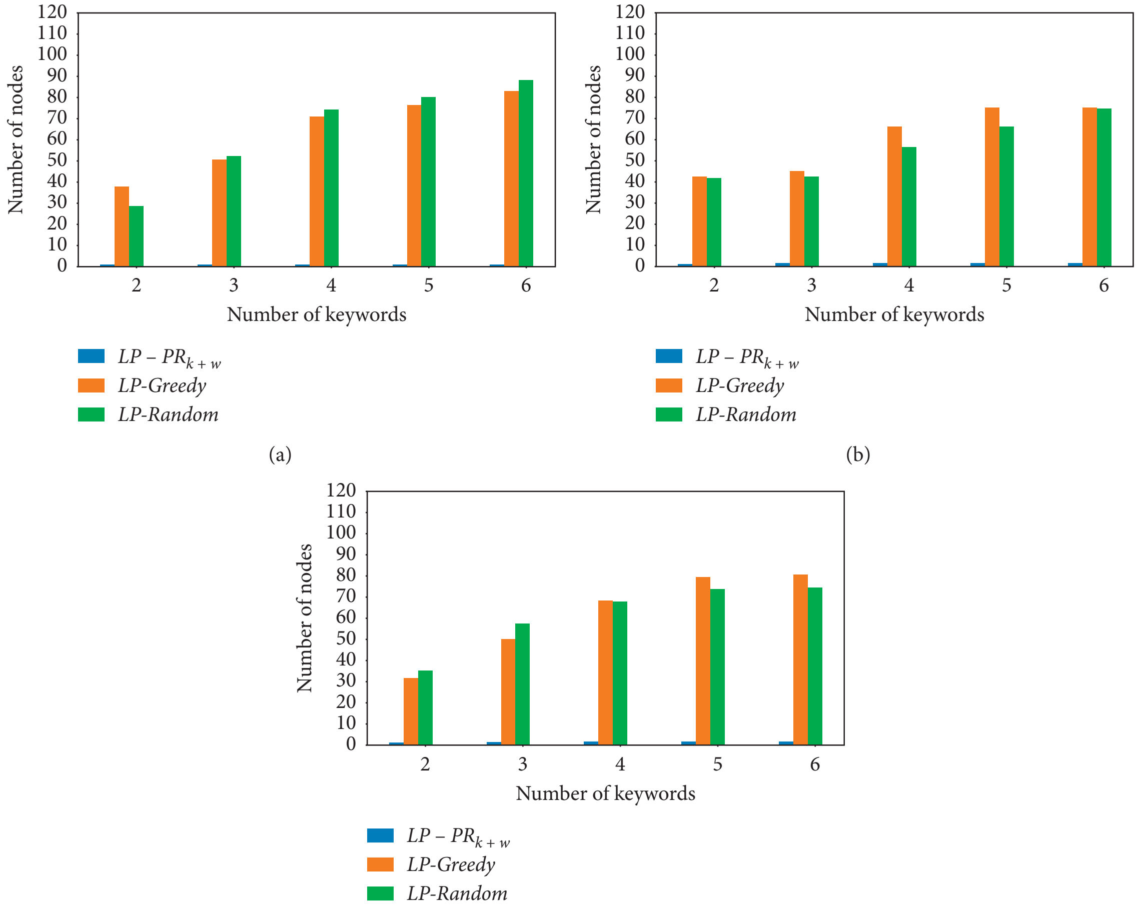
$\alpha\beta$	0.3	0.5	0.7	0.9
0.3	190	192	136	160
0.5	264	270	276	238
0.7	258	258	264	310
0.9	258	258	258	264

Table 3: α, β, and $\lambda = 0.9$ are employed in the link prediction to obtain the number of new edges.

$\alpha\beta$	0.3	0.5	0.7	0.9
0.3	208	152	160	168
0.5	348	234	216	176
0.7	342	342	314	226
0.9	288	288	288	294

In the experiment, we test the number of recommended nodes (papers) of different approaches. Here, the number of readers'query keywords ranges from 2 to 6. As shown in Figure 8, these experiment results show that the number of recommended including more papers can satisfy requirements of readers on more keywords query. When readers can accurately provide all query keywords, Figure 8(a) presents that our approach can accurately answer readers' keywords query; that is, the number of papers is 1. Furthermore, the experiment results of Figure 8(b) demonstrate that $LP - PR_{k+w}$ can find necessary bridging nodes between target nodes. In the case of same keywords query, Figure 8 shows that the number of recommended papers of $LP - PR_{k+w}$ is fewer than that of the other two approaches (i.e., *LP-Random* and *LP-Greedy*). In fact, a recommended result contains a handful of papers, which means these recommended papers have higher correlation. Thus, the experiment results of Figure 8 can show directly that our paper recommendation approach is superior to *LP-Random* and *LP-Greedy*.

(3) *Profile 3: The Success Rate of Different Approaches.* As shown in Figure 9, we compare the success rate of different approaches in the different sets. According to [39], if the number of query keywords equals 6, then the number of papers of a successful recommended result must range from 1 to 12. When readers can accurately provide all query keywords, Figure 9(a) presents that the success rate of our approach is 100%. For the other keywords query cases, the experiment results of Figures 9(b) and 9(c) again present that the success rate of $LP - PR_{k+w}$ is 100%. However, *LP-Random* and *LP-Greedy* are difficult to obtain successful paper recommendation results, especially with the number of query keywords equal to 6. In conclusion, the experiment



(a)

(b)

(c)

FIGURE 8: The number of recommended nodes of different approaches. (a) The number of recommended nodes in set A. (b) The number of recommended nodes in set B. (c) The number of recommended nodes in set C.

results of the set C can verify the feasibility of our paper recommendation algorithm (i.e., the MGWST algorithm). Furthermore, the experiment results of Figure 9 can show that $LP - PR_{k+w}$ can recommend more sets of papers with the higher correlations than other two approaches.

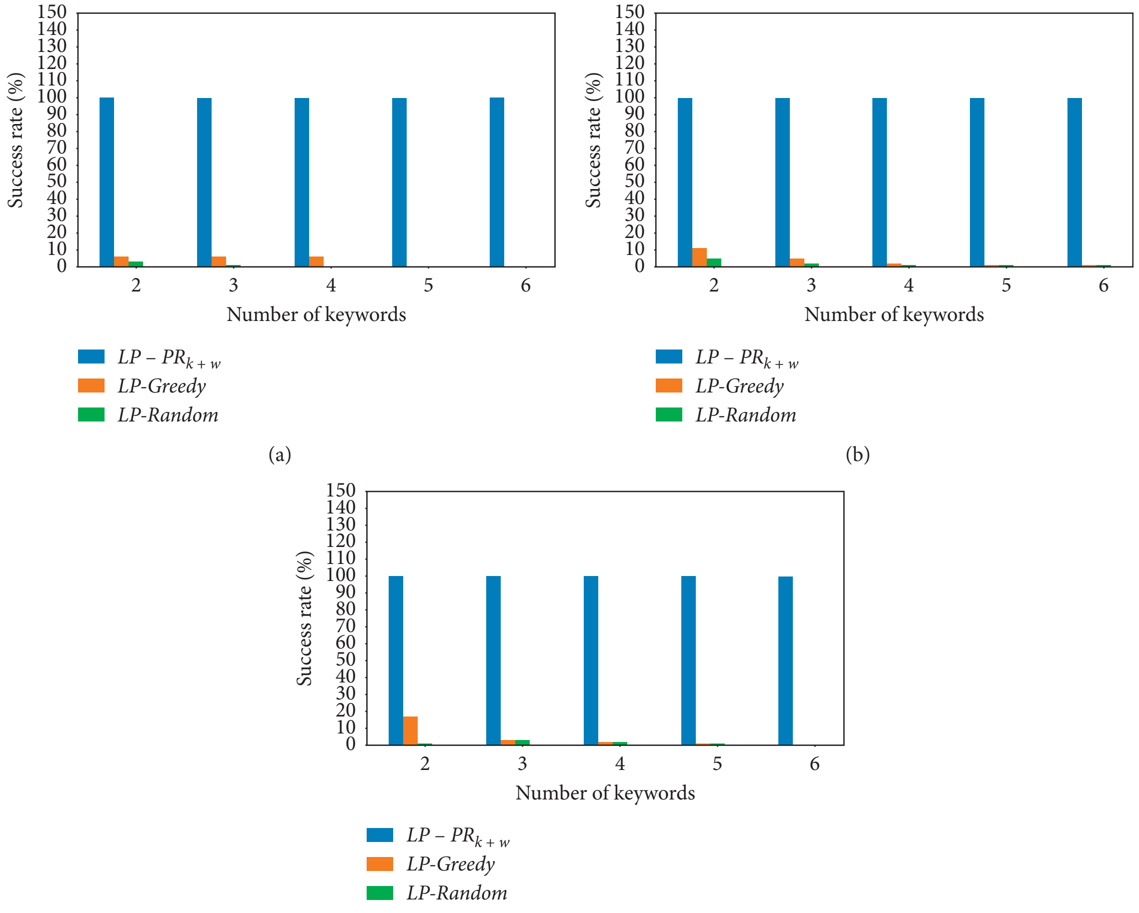
(4) *Profile 4: The Weight of Different Approaches.*

In the paper, $LP - PR_{k+w}$, *LP-Random*, and *LP-Greedy* are both returning a tree containing all query keywords to readers. Furthermore, the weight of a tree can reflect the correlation of a set of papers; the smaller the weight, the higher the correlation. Hence, we compare the weight of different approaches in this experiment. As shown in Figure 10, these experiment results show that the weights of different approaches increase with the number of query keywords increasing, as these approaches need to find more papers to meet readers' keywords query requirements. In the case of the same keywords query, the weight of our approach is less than the other two approaches (i.e., *LP-Random* and *LP-Greedy*) as $LP - PR_{k+w}$ adopts the weight optimization strategy. Furthermore, Figure 10(a) shows that the weight of $LP - PR_{k+w}$ equals 0, which further shows that our proposal can find one corresponding paper containing all query keywords. According to the above experiment results, $LP - PR_{k+w}$ can guarantee to return a set of papers with the minimum weight; that is, these papers have the higher correlation.

(5) *Profile 5: The Computation Time of Different Approaches.*

In the paper, a computation time is defined as the time of finding paper recommendation results. Thus, we compare the computation time of different approaches. As *LP-RW* and *LP-RWR* only do iterative operations and matrix operations in the process of paper recommendation, the computation time of these two approaches are both fixed values in the sets B and C.

Moreover, we only count the time when *LP-RW* achieves linear convergence, so the computation time of the approach is smaller than that of our approach. As shown in Figure 11, the computation time of $LP - PR_{k+w}$, *LP-Random*, and *LP-Greedy* generally increases as the number of query keywords increases; that is, these three approaches all take

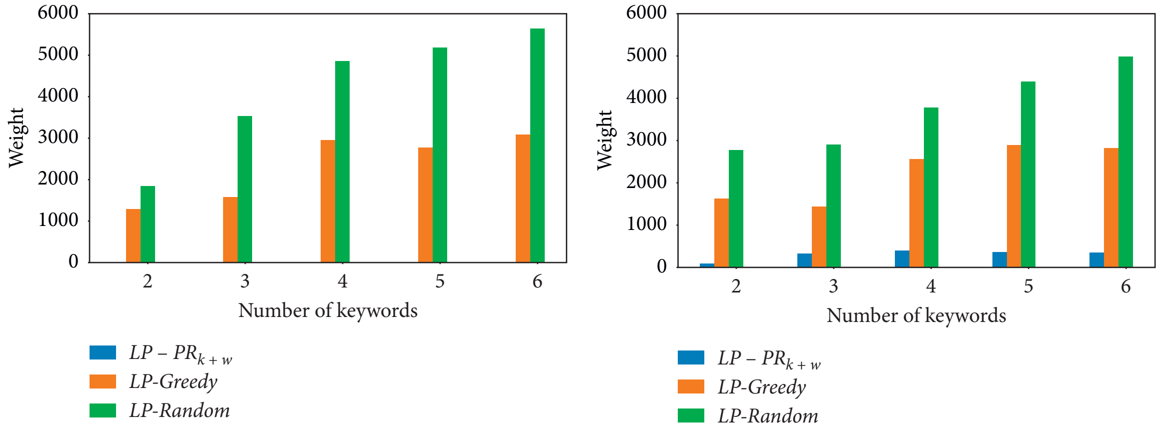


(a)

(b)

(c)

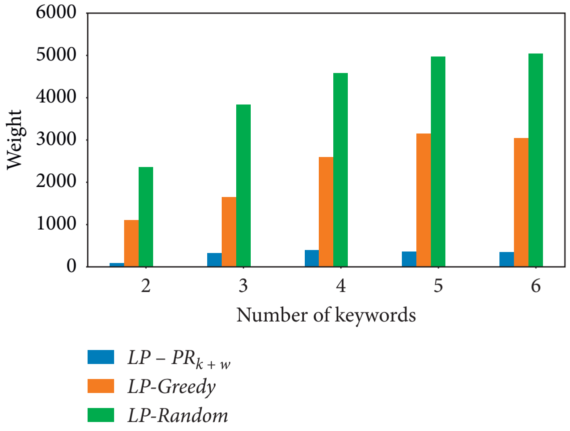
Figure 9: The success rate of different approaches. (a) The success rate in set A. (b) The success rate in set B. (c) The success rate in set C.



(a)

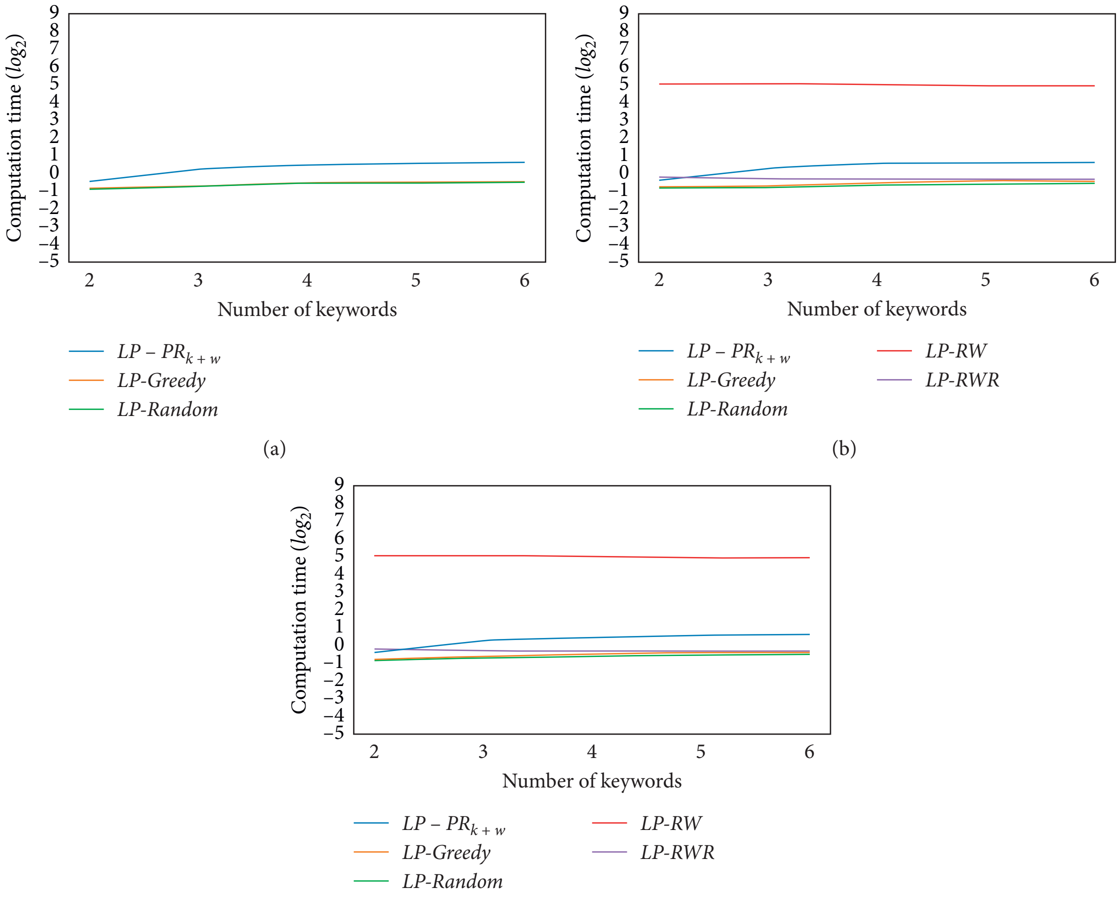
(b)

Figure 10: Continued.



(c)

Figure 10: The weight of different approaches. (a) The weight in set A. (b) The weight in set B. (c) The weight in set C.



(a)

(b)

(c)

Figure 11: The computation time of different approaches. (a) The computation time in set A. (b) The computation time in set B. (c) The computation time in set C.

more time to find an answer tree with the increasing number of query keywords. Furthermore, the computation time of these three approaches increases exponentially. As *LP-Random* and *LP-Greedy* both use extremely simple heuristic for selecting the bridging nodes from G_w, these two approaches find the answer tree faster than our approach. In fact, the recommended results of *LP-Random* and *LP-Greedy*

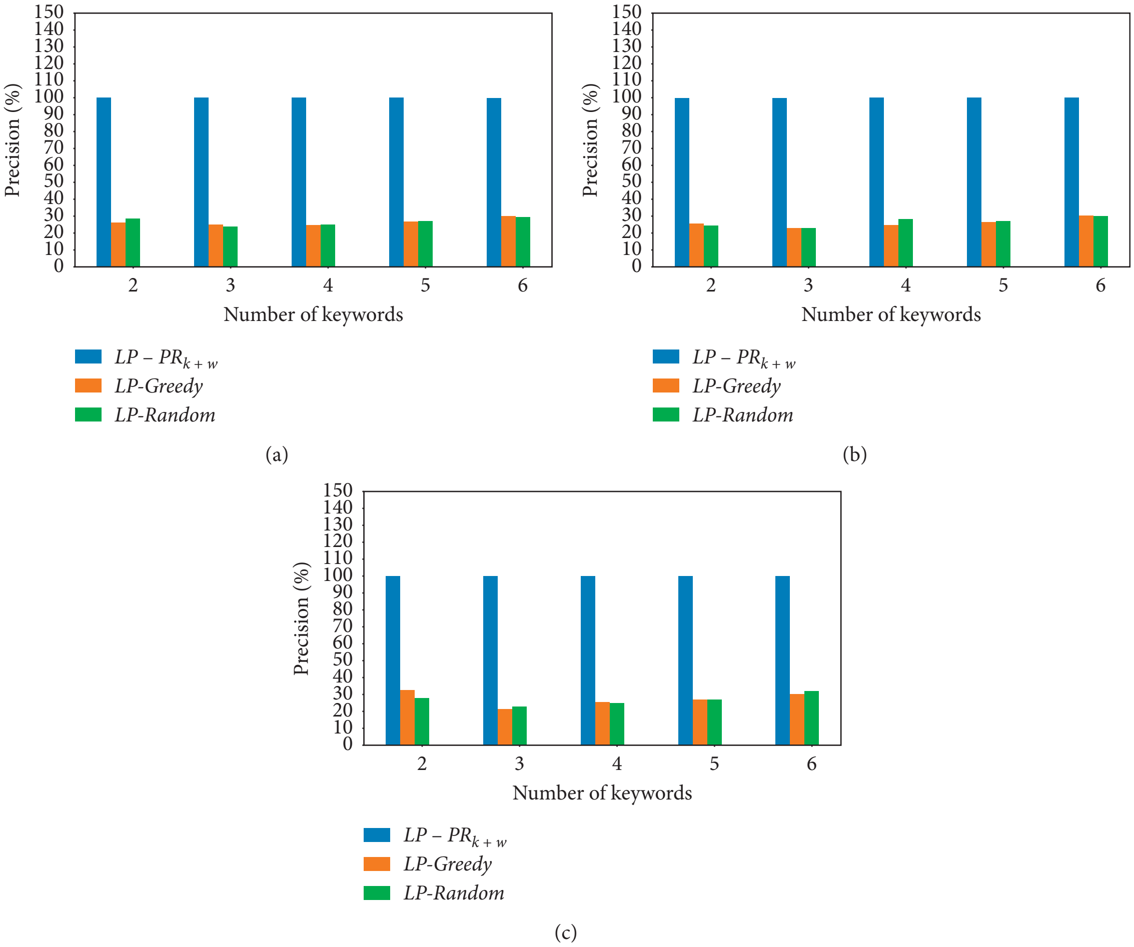


Figure 12: The precision of different approaches. (a) The precision in set A. (b) The precision in set B. (c) The precision in set C.

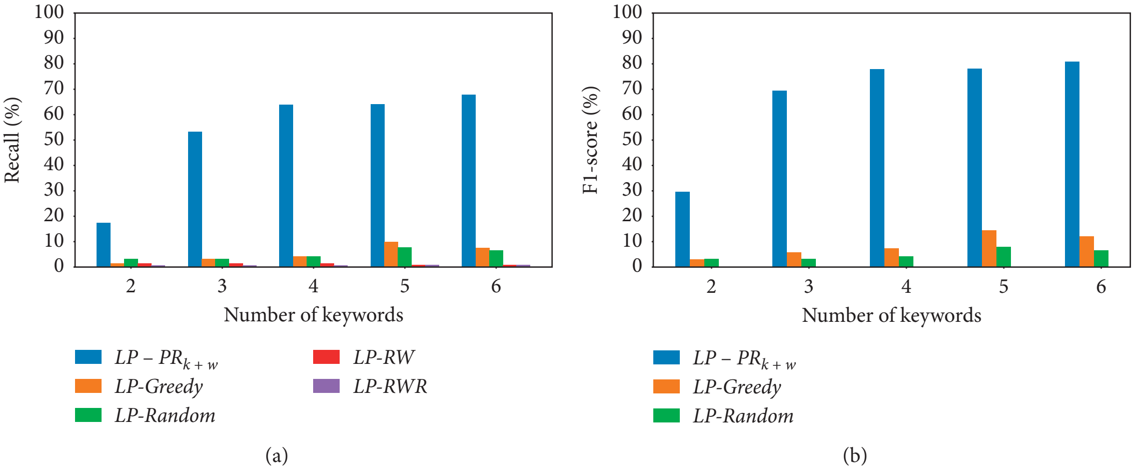


Figure 13: Continued.

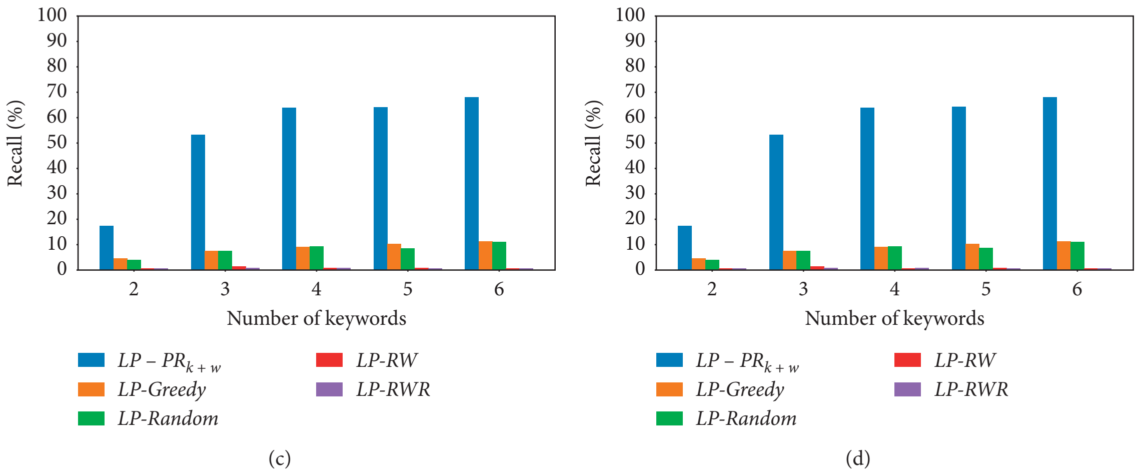


Figure 13: The recall and $F1$ score of different approaches. (a) The recall in set B. (b) The $F1$ score in set B. (c) The recall in set C. (d) The $F1$ score in set C.

are not ideal for readers as their recommended results contain some unnecessary bridging nodes (see Figure 8). In most really cases, the computation time of $LP-PR_{k+w}$ is allowable and receivable, which is the price to pay if readers save more time and energy on achieving their research goal.

(6) *Profile 6: The Precision of Different Approaches.*

For diverse paper recommendation approaches, we compare the precision of their recommended results. As shown in Figure 12, these experiment results show that the precision of diverse approaches is very different in the same keywords query cases. Whether readers accurately or randomly provide their query keywords, $LP-PR_{k+w}$ can accurately answer the readers' keywords query, and the precision equals 100%. However, the precision of *LP-Random* and *LP-Greedy* has both a range from 22% to 33% in the experiment. Therefore, the experiment results of Figure 12 further show that readers will easily realize their research aim by using our recommended results.

(7) *Profile 7: The Recall and F1 Score of Different Approaches.*

In this profile, we further compare the recall and $F1$ score of different approaches in sets B and C. Here, the numbers of recommended papers of *LP-RW* and *LP-RWR* are both equal to 20. As can be seen Figures 13(a) and 13(b), the recall and $F1$ score of $LP-PR_{k+w}$ range from 17% to 69% and from 29% to 81%, respectively; meanwhile, the other four approaches have smaller recall and $F1$ score than our approach. Likewise, Figures 13(c) and 13(d) also show that the recall and $F1$ score of other four approaches are smaller than $LP-PR_{k+w}$. Therefore, the experiment results of Figure 13 are not only demonstrating that our proposal can satisfy readers' keywords query requirements but also verifying further the feasibility of the MGWST algorithm (i.e., Algorithm 3).

7. Conclusions

In this paper, a novel keywords-driven and weight-aware paper recommendation approach based on the mobile edge computing framework, that is, $LP-PR_{k+w}$, is put forward to address the sparsity and compatibility issues existing in paper recommendations. The recommended papers are not only collectively containing readers' query keywords but also having the highest correlation degree. Therefore, the recommended papers can significantly promote readers' in-depth and continuous research around an identical focus. Furthermore, our proposal indirectly solves the problem of latency of recommendation results by adjusting the order of paper recommendation in the mobile edge computing environment. Finally, the experimental results show the feasibility of $LP-PR_{k+w}$, in terms of multiple evaluation metrics. In summary, our proposal can provide intelligent and personalized paper recommendation services in the mobile edge computing environment and further promote the quality-of-retrieval experience of readers.

Although our work shows desirable results, there are still several issues remaining unsolved. Firstly, our adopted link prediction approach in recommender system [40–46] is a bit naive and straightforward. Therefore, more refinements are needed in the future. Second, readers' paper search is often a multiobjective decision-making problem that involves a number of influencing factors [47–54] in the mobile edge computing environment. We will further improve our proposal by considering these factors.

Data Availability

The experiment dataset Hep-Th used to support the findings of this study has been deposited in "http://snap.stanford.edu/data/cit-HepTh.html."

Additional Points

This work is intended to improve and perfect the model in [40]. Thus, the work enables intelligent, personalized, and efficient paper recommendation services in the mobile edge computing environment.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Key Research Base of Philosophy and Social Sciences in Jiangsu Universities "Academic Center of Huang Yanpei Vocational Education Thought Research Association," Jiangsu University Philosophy and Social Science Research Project (2020SJA0675), Scientific Research Project of Nanjing Vocational University of Industry Technology (2020SKYJ03), Jiangsu Province Modern Education Technology Research Project (2020-R-84365), and National Vocational Education Teacher Enterprise Practice Base "Integration of Industry and Education" Special Project (Study on Evaluation Standard of Artificial Intelligence Vocational Skilled Level).

References

[1] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1964–1973, 2022.

[2] Y. Gong, Z. Jiang, Y. Feng et al., "EdgeRec," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2477–2484, Association for Computing Machinery, New York, NY, USA, October 2020.

[3] A. Xiong, D. Liu, H. Tian, Z. Liu, P. Yu, and M. Kadoch, "News keyword extraction algorithm based on semantic clustering and word graph model," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 886–893, 2021.

[4] T. Murata and S. Moriyasu, "Link Prediction of Social Networks Based on Weighted Proximity Measures," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, pp. 85–88, Fremont, CA, USA, November 2007.

[5] Z. Wang, J. Liang, R. Li, and Y. Qian, "An approach to cold-start link prediction: establishing connections between non-topological and topological information," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 11, pp. 2857–2870, 2016.

[6] J. Huang, Z. Tong, and Z. Feng, "Geographical POI recommendation for internet of things: a federated learning approach using matrix factorization," *International Journal of Communication Systems*, 2022.

[7] S. M. McNee, I. Albert, D. Cosley et al., "On the recommending of citations for research papers," in *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW '02)*, pp. 116–125, Association for Computing Machinery, NY, USA, November 2002.

[8] S. Meng, L. Qi, Q. Li, W. Lin, X. Xu, and S. Wan, "Privacy-preserving and sparsity-aware location-based prediction method for collaborative recommender systems," *Future Generation Computer Systems*, vol. 96, pp. 324–335, 2019.

[9] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, "Diversified service recommendation with high accuracy and efficiency," *Knowledge-Based Systems*, vol. 204, Article ID 106196, 2020.

[10] P. Nitu, J. Coelho, and P. Madiraju, "Improvising personalized travel recommendation system with recency effects," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 139–154, 2021.

[11] Y. Xu, L. Qi, W. Dou, and J. Yu, "Privacy-preserving and scalable service recommendation based on SimHash in A distributed cloud environment," *Complexity*, vol. 2017, p. 9, Article ID 3437854, 2017.

[12] F. Meng, D. Gao, W. Li, X. Sun, and Y. Hou, "A unified graph model for personalized query-oriented reference paper recommendation," in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM '13)*, pp. 1509–1512, Association for Computing Machinery, NY, USA, October 2013.

[13] M. Gori and A. Pucci, "Research paper recommender systems: a random-walk based approach," in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, pp. 778–781, Hong Kong, China, December 2006.

[14] H. Liu, H. Kou, C. Yan, L. Qi, and P. Link, "Paper citation network to construct paper correlation graph," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–12, 2019.

[15] R.-H. Li, J. X. Yu, and J. Liu, "Link prediction," *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, Association for Computing Machinery, in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*, pp. 1147–1156, October 2011.

[16] Y. Chen, W. Gu, and K. Li, "Dynamic task offloading for internet of things in mobile edge computing via deep reinforcement learning," *International Journal of Communication Systems*, 2022.

[17] L. Backstrom and J. Kleinberg, "Romantic partnerships and the dispersion of social ties," *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, Association for Computing Machinery, in *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*, pp. 831–841, February 2014.

[18] C. P. Muniz, R. Goldschmidt, and R. Choren, "Combining contextual, temporal and topological information for unsupervised link prediction in social networks," *Knowledge-Based Systems*, vol. 156, pp. 129–137, 2018.

[19] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks," in *Proceedings of the 19th International Conference on World Wide Web*, pp. 981–990, Association for Computing Machinery, NY, USA, April 2010.

[20] P. M. Chuan, L. H. Son, M. Ali, T. D. Khang, L. T. Huong, and N. Dey, "Link prediction in co-authorship networks based on hybrid content similarity metric," *Applied Intelligence*, vol. 48, no. 8, pp. 2470–2486, 2018.

[21] P. R. S. Soares and R. B. C. Prudêncio, "Proximity measures for link prediction based on temporal events," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6652–6660, 2013.

[22] B. Filipecki and M. Van Vyve, "Stronger path-based extended formulation for the Steiner tree problem," *Networks*, vol. 75, no. 1, pp. 3–17, 2020.

[23] N. Löhndorf and A. Shapiro, "Modeling time-dependent randomness in stochastic dual dynamic programming," *European Journal of Operational Research*, vol. 273, no. 2, pp. 650–661, 2019.

[24] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[25] M. Lu, Z. Qu, M. Wang, and Z. Qin, "Recommending authors and papers based on ACTTM community and bilayer citation network," *China Communications*, vol. 15, no. 7, pp. 111–130, 2018.

[26] H. Liu, H. Kou, X. Chi, and L. Qi, "Combining time, keywords and authors information to construct papers correlation graph," in *Proceedings of the The 27th International Conference on Software Engineering and Knowledge Engineering (SEKE'2019)*, pp. 11–19, USA, July 2019.

[27] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic task offloading for mobile edge computing with hybrid energy supply," *Tsinghua Science and Technology*, 2021.

[28] L. Qi, Q. He, F. Chen et al., "Finding all you need: web APIs recommendation in web of things through keywords search," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1063–1072, 2019.

[29] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: challenges, progress and prospects," *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

[30] S. Wang, L. Hu, Y. Wang et al., "Graph learning based recommender systems: a review," *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, Survey Track, , January 2021.

[31] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[32] X. Zhou, X. Yang, J. Ma, and K. I.-K. Wang, "Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT," *IEEE Internet of Things Journal*1 page, 2021.

[33] J. Hu, Y. Pan, T. Li, and Y. Yang, "TW-Co-MFC: two-level weighted collaborative fuzzy clustering based on maximum entropy for multi-view data," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 185–198, 2021.

[34] X. Zhou, Y. Li, and W. Liang, "CNN-RNN based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, 2021.

[35] Z. Xue and H. Wang, "Effective density-based clustering algorithms for incomplete data," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 183–194, 2021.

[36] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (KDD '05)*, pp. 177–187, Association for Computing Machinery, NY, USA, August 2005.

[37] H. Liu, H. Kou, C. Yan, and L. Qi, "Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph," *Complexity*, vol. 2020, Article ID 2085638, 2020.

[38] L. Guo, X. Cai, F. Hao, D. Mu, C. Fang, and L. Yang, "Exploiting fine-grained co-authorship for personalized citation recommendation," *IEEE Access*, vol. 5, pp. 12714–12725, 2017.

[39] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, and Q. Ni, "Data-driven web APIs recommendation for building web applications," *IEEE Transactions on Big Data*, IEEE Transactions on Big Data1 page, IEEE Transactions on Big Data, 2020.

[40] H. Liu, J. Hou, Q. Li, and J. Jiang, "Keywords-driven and weight-aware paper recommendation via paper correlation pattern mining," *Research Square*, 2021.

[41] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," *ACM Computing Surveys*, vol. 54, no. 7, pp. 1–38, 2022.

[42] J. Li, H. Peng, Y. Cao et al., "Higher-order attribute-enhancing heterogeneous graph neural networks," *IEEE Transactions on Knowledge and Data Engineering*1 page, 2021.

[43] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[44] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. I.-K. Wang, "Hierarchical adversarial attacks against graph neural network based IoT network intrusion detection system," *IEEE Internet of Things Journal*1 page, 2021.

[45] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[46] Z. Cai and Z. He, "Trading private range counting over big IoT data," *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, in *Proceedings of the The 39th IEEE International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, July 2019.

[47] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task scheduling for sensor hubs," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 242–247, 2020.

[48] W. Fan, P. Chen, D. Shi, X. Guo, and L. Kou, "Multi-agent modeling and simulation in the AI age," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 608–624, 2021.

[49] Z. Tong, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[50] X. Zhou, W. Liang, K. I.-K. Wang, and L. T. Yang, "Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 171–178, 2021.

[51] M. Wang, W. Han, and W. Chen, "MetaOJ: a massive distributed online judge system," *Tsinghua Science and Technology*, vol. 26, no. 4, pp. 548–557, 2021.

[52] H. Liu, S. Meng, J. Hou, S. Wang, Q. Li, and C. Huang, "Locality-sensitive hashing-based link prediction process on smart campus education or online social platform," *Journal of Circuits, Systems, and Computers*, Article ID 2250160, 2022.

[53] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, "Deep-learning-enhanced multitarget detection for end-edge-cloud surveillance in smart IoT," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12588–12596, 2021.

[54] Y. Bie and Y. Yang, "A multitask multiview neural network for end-to-end aspect-based sentiment analysis," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 195–207, 2021.