IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 31, 2019, accepted January 14, 2020, date of publication January 17, 2020, date of current version January 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967407

A Novel Link Prediction Method for Opportunistic Networks Based on Random Walk and a Deep Belief Network

ZILIANG LIAO[1], LINLAN LIU[1], AND YUBIN CHEN[2]
[1]School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China
[2]School of Software, Nanchang Hangkong University, Nanchang 330063, China

Corresponding author: Linlan Liu (liulinlan@nchu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61762065 and Grant 61962037, in part by the Natural Science Foundation of Jiang Province under Grant 20181BAB202015 and Grant 20171ACB20018, in part by the Innovation Foundation for Postgraduate Student of Jiangxi Province under Grant YC2018-S371, and in part by the Natural Science Foundation of Jiang Province under Grant 20171BBH80022.

ABSTRACT Link prediction is to estimate the possibility of future links among nodes by utilizing known information such as network topology and node attributes. According to the characteristics of opportunistic networks (topological time-variation, node mobility and intermittent connections), this paper proposes a novel link prediction approach (IRWR-DBN) for opportunistic networks that is based on random walk and a deep belief network. First, we reconstruct the Markov probability transition matrix and define a similarity index– improved random walk with restart (IRWR). Second, we divide the opportunistic network into network snapshots. Then, the similarity matrix of each snapshot is calculated by using the IRWR index to construct a sample set. Finally, a predictive model is constructed based on a deep belief network which extracts the time-domain characteristics in the process of dynamic evolution of the opportunistic network. The experimental results on the ITC and MIT Reality datasets show that compared with methods, such as the similarity-based index (CN, AA, Katz, RA, RWR), convolutional neural network, and recurrent neural network, the proposed method is more accurate and stable.

INDEX TERMS Opportunistic network, link prediction, random walk with restart, deep belief network, similarity index.

I. INTRODUCTION

Opportunistic networks [1] are mobile ad hoc networks that do not require an end-to-end link between the source node and the target node and establish communication through the movement of network nodes. Data transmission of the opportunistic network is realized by the story-carry-forward routing mechanism. The process of data transmission of opportunistic networks is shown in figure 1.

Source node S wants to send a message to target node D. At time t_1, the message is generated by source node S, and because source node S is not in the communication range of target node D, the source node S sends the message to node 3. At time $t_2(t_2 > t_1)$, node 3 transfers the message to node 8. At time $t_3(t_3 > t_2)$, node 8 arrives in the communication range of target node D, and the message is successfully sent to target node D.

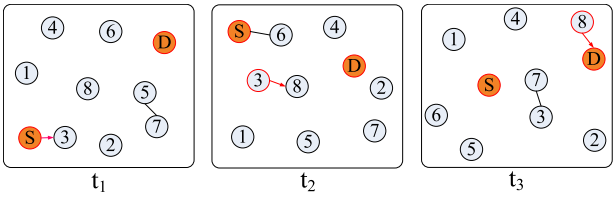


FIGURE 1. The process of data transmission of opportunistic networks (the solid line indicates the connection between nodes, and the arrow indicates the direction of data transmission).

These are the characteristics of an opportunistic network: topology time-varying, node mobility and intermittent connectivity. An opportunistic network better satisfies the requirements of mobile ad hoc networks in the real world. To analyse and understand the evolution of the network structure and propose a better message transmission mechanism for opportunistic networks, researchers have recently focused more attention on the problem of link prediction in opportunistic networks [2], [3]. Link prediction is to estimate the

The associate editor coordinating the review of this manuscript and approving it for publication was Xiao-Sheng Si.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

possibility of future links among nodes by utilizing known information, such as network topology and node attributes. Link prediction can help researchers further analyse and understand the evolution of the network topology, build the node mobile model, and provide theoretical support for routing algorithms [4].

In opportunistic networks, topology temporal information is a key factor in calculating the similarity between nodes. However, most existing link prediction methods are proposed for static networks or social networks where the network topology does not change or changes slowly over time. These methods do not sufficiently apply temporal information. Therefore, it is necessary to study effective methods for predicting links in opportunistic networks. In this paper, we present a similarity index–improved random walk with restart (IRWR). Furthermore, we propose a link prediction model based on IRWR and deep belief network (IRWR-DBN), which can extract the time-domain characteristics in the process of dynamic evolution of the opportunistic networks.

The main contributions of this paper are as follows.

(1) We reconstruct the Markov probability transition matrix according to the information of the second-order neighbours of the node and propose a similarity index IRWR. Then, a sample set is constructed based on the IRWR similarity index.

(2) We propose a link prediction model, IRWR-DBN, which extracts the time-domain characteristics of sample set, and achieves a significantly high prediction performance.

The rest of the paper is organized as follows. Related work is discussed in Section 2. The construction of the similarity index IRWR is described in Section 3. Sample construction is described in Section 4. The prediction model IRWR-DBN is addressed in Section 5. The experiment and analysis are presented in Section 6. Section 7 provides the conclusion.

II. RELATED WORK

Link prediction methods of opportunistic networks fall into the following categories: predictions based on similarity indexes, matrix decomposition, and machine learning.

A. SIMILARITY INDEX-BASED PREDICTION METHODS

Similarity index-based prediction methods mean that the more similar the structural information, attribute information and behaviour of nodes in the network, the greater the possibility of a connection between them. For example, if two people have the same friends or prefer to go to the same place, they have a higher probability of meeting or communicating with each other.

Liu *et al.* [5] defined an influence set of nodes and a common influences set between nodes. The similarity value between nodes is calculated according to the common influences set. Guo *et al.* [6] proposed a common neighbour tightness index based on node degree and clustering coefficient, they proposed that the closer the relationship between the common neighbours of the nodes, the higher the possibility of connections between nodes. Rahman *et al.* [7] considered user activities and common neighbours, and defined the local and global link prediction algorithm to evaluate the similarity between nodes. Yang *et al.* [8] introduced the concept of gravitational field and proposed the LP-GFCN algorithm based on indirect and direct gravitational. Shang *et al.* [9] discussed the role of link direction in link prediction problems. They proposed that the bi-directional link contains more network information, and found that a pair of nodes with bi-directional links has greater probability to connect to the common neighbours with bi-directional links. Hu *et al.* [10] considered the effectiveness of quad motifs in calculating the similarity between nodes in the directed network, and defined closed quad, relative open quad, no-crossing closed quad and quad motifs. Li *et al.* [11] defined a community relationship strength index (CRS) to estimate the closeness between communities, and proposed a link prediction framework based on node similarity and community information. Then, in the framework, they combined CRS with traditional similarity indexes to measure the connection likelihood. Wang *et al.* [12] proposed an adjustable parameter based on community information and applied it to nine similarity indexes. Then, they verified that the parameter can improve the accuracy of link prediction in ten real datasets. Additionally, a parallelization algorithm was proposed to apply the above nine indexes to the link prediction of large-scale complex networks.

The abovementioned prediction methods mainly have better performance in networks with slow or no change in topologies, such as static networks or social networks. Because these studies do not consider the relationship between topology and time information, they have a poor predictive effect in networks with frequent topology changes over time.

B. MATRIX FACTORIZATION-BASED PREDICTION METHODS

Matrix factorization-based link prediction methods use the low-rank matrix obtained by matrix factorization to solve the problem of link prediction. Among them, the matrix consists of an adjacency matrix or a matrix constructed by extracting other network information. At present, matrix factorization mainly falls into the following categories: singular value decomposition, non-negative matrix factorization and probability matrix factorization.

Li *et al.* [13] proposed a link prediction method for dynamic attribute networks. In the proposed method, the feature matrix was obtained from the first network snapshot, and its low-rank matrix is calculated. Then, the low-rank matrix was continuously updated according to the subsequent network snapshot. After the update, the residual error was used to estimate the possibility of the existing links between future nodes. Wang *et al.* [14] noted that there is a lack of effective fusion of topological information and non-topological information in social-information networks in the existing link prediction algorithms. Therefore, they defined a user topic similarity index based on user topic information,

and a topic similarity matrix is constructed based on the user topic similarity index. Then, the information of the following/followed network and the topic similarity matrix were fused into the framework of probabilistic matrix factorization, based on which the representation of the network nodes are obtained. Finally, the linking probability between network nodes was calculated based on the obtained latent-feature representation. Ahmed *et al.* [15] fused the temporal and structural properties of dynamic networks into non-negative matrix factorization, and presented a novel iterative rules of non-negative matrix factorization. Then, they obtained the similarity value matrix of the network according to the result of matrix factorization, and link prediction was performed. Mutinda *et al.* [16] applied non-negative matrix factorization to extract the latent features of the time-series graphs, and the time-series prediction method Holt-Winters was used to learn and extract the time-domain information of the above features to solve the link prediction problem.

The abovementioned prediction methods mainly perform low-rank factorization on the adjacency matrix and extract time-domain information of the dynamic network by iterative factorization. However, in large-scale dynamic networks, iterative matrix factorization can lead to great time complexity.

C. MACHINE LEARNING-BASED PREDICTION METHODS

As is known, machine learning algorithms have a powerful ability to extract features. The machine learning-based prediction method extracts the features of the data in the network by using a machine learning algorithm from a certain perspective and realizes link prediction.

Yang *et al.* [17] proposed that one kind of object can be considered as the features of another kind of object in bi-typed heterogeneous networks. According to this method, the features representation of nodes were obtained, and they were clustered. Then, they determined whether there was a connection between nodes based on the clustering result and the decision tree model. Li *et al.* [18] introduced utility analysis into the link prediction method by considering that individual preference is the main reason for forming a link, and they also focused on the meeting process that is a latent variable during the process of forming links. Accordingly, the link prediction problem was formulated as a machine learning process with latent variables. Therefore, an expectation-maximization (EM) algorithm was adopted to address the estimation problem. Lei *et al.* [19] applied graph convolutional network (GCN), long short-term memory (LSTM) and generative adversarial network (GAN) to extract the nonlinear characteristics of link changes in weighted dynamic networks. They utilized GCN to extract local features of each snapshot, then employed LSTM to characterize the evolving features of the dynamic networks, and utilized GAN to improve the accuracy of the model. Shao *et al.* [20] proposed an adaptive link prediction method based on density peak clustering in order to solve the problem that single link prediction index cannot be applied to all networks. They utilized different prediction indexes (CN, JC, RA and AA) as attributes of the link, and used clustering analysis to transform the link prediction into classification. Li *et al.* [21] introduced the structural subgraph and proposed the structure subgraph feature. Then they applied it to the linear regression model and neural machine to predict the link. Cai *et al.* [22] divided the network into a series of time-series snapshots based on the time-varying characteristics of the opportunistic network and defined a set of vectors based on the attributes of the edge, such as the endpoint ID and the start time of the edge. They also constructed a predictive model (RNN-LP) based on a recurrent neural network to extract the features of the edge attribute vector over time to predict the link. Shu *et al.* [23] transformed the multi-node link prediction problem into pattern classification. They divided the network to obtain a series of network snapshots, and calculated the state map of each snapshot. Then, they constructed a prediction model based on a convolutional neural network (CNN) to extract the pattern variation features and realize the multi-node link prediction. Chen *et al.* [24] divided the dynamic network into a series of time-series snapshots and used an encoder to encode and characterize each network snapshot, and the long short-term memory (LSTM) model was constructed to extract the features in the encoded snapshots. Then, the features extracted by LSTM were amplified to the original dimensions of the network by a decoder to predict the topology of the network in the future. Sett *et al.* [25] analysed the characteristics of dynamic heterogeneous networks and proposed a feature set called time-aware multi-relational link prediction (TMLP). The TMLP includes CN, JC, AA, resource allocation (RA), and PA. Then, a predictive model based on random deep forest was constructed and trained to predict the link of a specific type. Winter *et al.* [26] analysed the impact of time attributes on link prediction in a dynamic network and proposed two methods to extend the node2vec model from a static network to a dynamic network.

In the abovementioned studies of the machine learning-based link prediction methods, the researchers considered the time characteristics of the network information in the dynamic network or the opportunistic network. However, they only used the information in a single network snapshot to construct data and then used machine learning algorithms to extract time characteristics of the data. They did not consider the correlation between network information in two or more adjacent snapshots. Therefore, according to the time-varying characteristics of the opportunistic network, this paper combines the impact of historical state of the link on its connection state at the next moment. We reconstruct a Markov probability transition matrix considers based on the information of the second-order neighbours of the node and define the IRWR index. Then, a similarity matrix for each network snapshot is constructed. We also construct a sample set based on the similarity matrix and the length of input data. We utilize the advantages of the DBN model in automatically extracting features to extract the time-domain characteristics in the process of dynamic evolution of the opportunistic network to realize future link prediction better.

III. THE CONSTRUCTION OF SIMILARITY INDEX

In real-world networks, nodes are primarily carried or manipulated by humans, and the intimacy between people is different according to factors such as common friends or distance. Therefore, considering the information of the second-order neighbours of the node, this paper reconstructs a Markov probability transition matrix, improves the random walk with the restart index in [27], and proposes a novel similarity index IRWR.

The RWR algorithm is a direct application of the PageRank algorithm. It can well grasp the global structural information of the network and has a better performance in large-scale and sparse networks [27]. The main idea of the RWR algorithm is as follows: it considers a random walker starting from node x, who will iteratively move to a random neighbour with probability c and return to node x with probability $1 - c$. P is the transition probability matrix of the network, and the element P_{xy} of the matrix P is defined in (1)

$$P_{xy} = \frac{a_{xy}}{k_x} \tag{1}$$

where P_{xy} indicates the probability that the random walker will move from node x to node y in the next step, with $a_{xy} = 1$ if node x and node y are connected, and $a_{xy} = 0$ otherwise. k_x denotes the degree of node x.

It can be known from equation (1) that the probability of a random walker moving to its neighbour nodes is equal in the RWR algorithm. However, this is not the case in the real world where the connections between nodes with the same hobbies or common neighbours are stronger, and we prefer to connect with people or things we are familiar with. Therefore, the probability of a random walker moving to its neighbour nodes is different, depending on the factors such as the structure and attributes of the node. We combine the influence of the second-order neighbours on the intimacy between nodes, reconstruct the Markov transition probability matrix of the network and mark it as P^{recon}, and the element P^{recon}_{xy} of P^{recon} is redefined in (2).

$$P^{recon}_{xy} = \begin{cases} \dfrac{k_y}{\tau_x + k_x}, & \textit{if } \text{node x and y are connected} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where τ_x is the number of nodes that node x can arrive at through two hops, and k_x denotes the degree of node x.

Let a random walker start from node x, and the reconstructed P^{recon} is the Markov transition probability matrix of the network. Denote by $q_{xy}(t)$ the probability of this random walker locating at node y after t steps, and we have

$$q_x(t) = c(P^{recon})^T q_x(t-1) + (1-c)e_x \tag{3}$$

where $(P^{recon})^T$ is the transposed matrix of the transition probability matrix P^{recon}, and e_x is the initial vector in which the x-th bit is 1 and the other bits are 0. The solution is straightforward, as shown in (4):

$$q_x = (1-c)(I - c(P^{recon})^T)^{-1} e_x \tag{4}$$

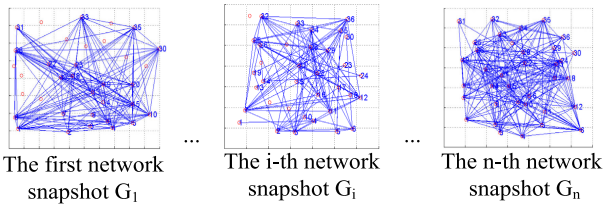


FIGURE 2. The evolution of the network topology of the opportunistic network.

The IRWR similarity index is thus defined in (5)

$$S^{irwr}_{xy} = q_{xy} + q_{yx} \tag{5}$$

where q_{xy} is the probability that the random walker starting from node x is located at node y in the steady state, and q_{yx} is the probability that the random walker starting from node y is located at node x in the steady state.

IV. SAMPLE CONSTRUCTION

This paper selects iMote Traces Cambridge (ITC) and MIT as the dataset in the experiment. However, the data in the ITC and MIT dataset is a continuous communication record between nodes, and it is not suitable as a direct input to the model. Therefore, we first need to process the initial data and construct the sample as input to the model. The sample construction has two steps. In the first step, we divide the opportunistic network into several continuous network snapshots according to the length of time slice and use the IRWR index to construct the similarity matrix of each network snapshot. In the second step, the sample set is constructed based on the similarity matrix and the length of the input data, and then used as training data for the prediction model.

A. CONSTRUCTION OF SIMILARITY MATRIX

This paper divides the opportunistic network $G = (V, E)$ according to the length of time slice Δt, where V and E denote the set of nodes and the set of edges in the network. Then, a network snapshot set $G = \{G_1, G_2, \ldots, G_{n-1}, G_n\}$ is obtained, where $G_t = (V_t, E_t)$ is the network topology map of the i-th network snapshot, V_t and E_t denote the set of nodes and the set of edges in the i-th network snapshot, and n denotes the number of network snapshots. The evolution of the network topology of the opportunistic network is shown in figure 2.

The process of constructing the similarity matrix is shown in figure 3. First, the opportunistic network is divided into several continuous network snapshots. Then, we use the IRWR index to calculate the similarity value between nodes in each network snapshot and construct the similarity matrix of each network snapshot. $Q = \{Q_1, Q_2, \ldots, Q_{n-1}, Q_n\}$ denotes an ordered set of similarity matrices, where Q_i represents the similarity matrix of the i-th network snapshot, and the element Q^i_{xy} of the matrix Q_i is defined in (6).

$$Q^i_{xy} = (S^{irwr}_{xy})^i \tag{6}$$

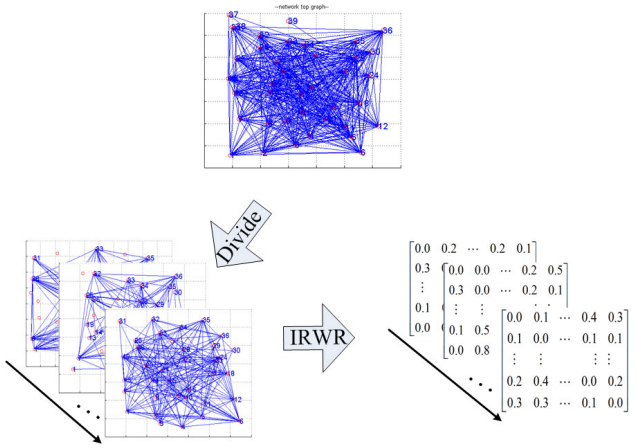


FIGURE 3. The process of constructing the similarity matrix.

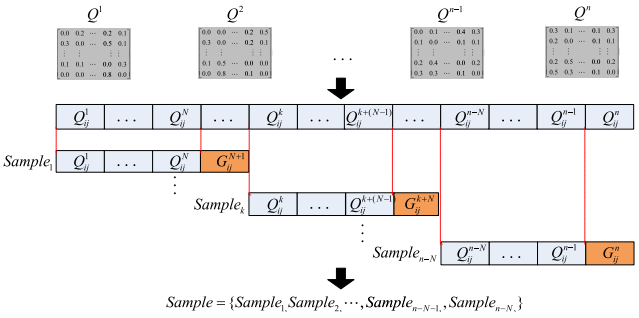


FIGURE 4. Sample set construction.

where $(S_{xy}^{irwr})^i$ represents the IRWR similarity value between nodes x and y in the i-th network snapshot.

B. CONSTRUCTION OF THE SAMPLE SET

Based on the above set Q and G, as shown in figure 4, the construction of the sample set can be performed into the following steps:

(1) We need to select the target node pair (V_i, V_j) used for prediction and determine the number of continuous snapshots used to predict the connection state of the node pair (V_i, V_j) in the next network snapshot (assumed to be N);

(2) We sequentially extract the IRWR similarity values of the node pair (V_i, V_j) in each similarity matrix and obtain a sequence of IRWR similarity values of the node pair (V_i, V_j);

(3) We set N as the length of the sliding window, and sequentially take the consecutive N similarity values in the sequence of IRWR similarity values as the data of the sample, and the connection state of the node pair (V_i, V_j) in the $(N + 1)$-th network snapshot is taken as the label of the sample.

According to the above steps, a sample set is constructed based on the N and node pair (V_i, V_j), and a total of $n-N-1$ samples can be constructed. The sample set is represented as $Sample = \{Sample_1, Sample_2, \cdots, Sample_k, \cdots, Sample_{n-N-1}\}$, where $Sample_k = \{Q_{ij}^k, Q_{ij}^{k+1}, \ldots, Q_{ij}^{k+(N-2)}, Q_{ij}^{k+(N-1)}, G_{ij}^{k+N}\}$ denotes the k-th sample. The pre-N dimension of the samples corresponds to the IRWR similarity values Q_{ij} of the node pairs (V_i, V_j) in the previous N snapshots. The $(N + 1)$-th dimension of the samples is the label of the sample G_{ij}^{k+N}, and G_{ij}^{k+N} denotes the connection state (0 or 1) of the node pair (V_i, V_j) in the $(k+N)$-th network snapshot.

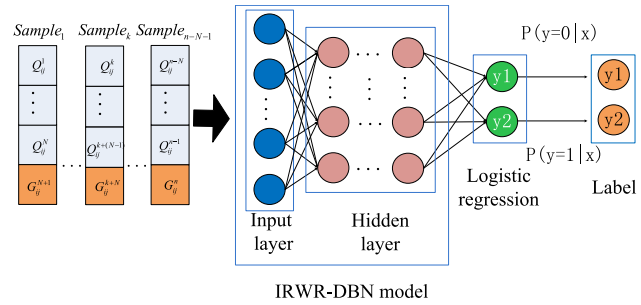


FIGURE 5. The structure of the IRWR-DBN model.

V. THE CONSTRUCTION OF DEEP BELIEF NETWORKS PREDICTION MODEL

On the basis of the sample set, this paper considers the result of link prediction to be yes or no (1 or 0); thus, we convert the link prediction problem into a classification problem, and we propose a link prediction model IRWR-DBN to solve this classification problem. The sample set *Sample* is used as the input of the IRWR-DBN model, and the connection state of node pairs is the output of the IRWR-DBN model. Benefiting from the DBN in automatically features extracting to extract the time-domain characteristics of data in the sample set, hence it predicts the connection state of node pairs in the next snapshot.

The main purpose of constructing the IRWR-DBN model is to capture the correlation between the information in multiple snapshots and the current connection state and then extract the characteristics of the link changes in the opportunistic network.

The IRWR-DBN model is constructed as follows: network structure, hyper-parameter setting and model training.

A. MODEL STRUCTURE

The IRWR-DBN aims to extract the relationship between the node pairs' historical information and its connection state in the time dimension. DBN effectively avoids the problem of local optimal solution and gradient explosion caused by initialization through pre-training. Therefore, this paper selects DBN to extract the time-domain characteristics of the data in the sample set. The structure of the IRWR-DBN model is shown in figure 5. The first layer is the input layer for receiving the sample set *Sample*, and the middle layers are the hidden layers, which are composed of multiple restricted Boltzmann machines (RBM) for extracting the high-order features of the sample set *Sample*. The last layer is a classifier (logistic regression). The output of the hidden layer is used as the input of the classifier to obtain the prediction results.

B. HYPER-PARAMETER SETTING

The accuracy of the model is closely related to the hyper-parameters of the model. To improve the ability to extract features of the IRWR-DBN model, it is necessary to set a good set of hyper-parameters for the model. The construction of the IRWR-DBN model in this paper involves the following hyper-parameters: the length of the input data, the number of nodes in the hidden layer and the number of hidden layers.

1) THE LENGTH OF THE INPUT DATA

The length of the input data, that is, the number N of network snapshots to be determined above, is a key part of extracting time-domain information. It defines the number of consecutive snapshots used to predict the connection state of a node pair in the next snapshot. Different lengths of the input data have different temporal characteristics. If the length is too long, it will lead to valid information in the network being covered, and the number of samples will be reduced. If the length of the input data is too short, it will lose the continuity of the data and affect the intrinsic association of the data. An appropriate length of the input data not only enables the model to make full use of the historical information of the network but also enables the model to have a better performance. However, there is no good theory for determining the length of the input data in existing research. The length of the input data is determined based on comparison experiments with multiple empirical values in most studies. In this paper, we determine the length of the input data of the IRWR-DBN model using the comparison experiment in experiment 2.

2) THE NUMBER OF NODES IN THE HIDDEN LAYER

The number of nodes in the hidden layer is a key parameter of the model. It is related to the connection between the hidden layer and the visible layer and whether the hidden layer can efficiently and accurately extract features in the sample. Hidden layers with different numbers of nodes have different abilities to extract features. If the number of nodes is very small, the error of reconstructing the input data will increase. If the number is very large, the complexity of the model will increase, and the model will easily over-fit. At present, the number of nodes in the DBN hidden layer is usually determined by an empirical formula, and the common empirical formulas are shown in (7), (8) and (9).

$$S = \sqrt{mn} + n \tag{7}$$

$$S = \sqrt{mn} + \frac{a}{2} \tag{8}$$

$$S \leq \sqrt{n(m-1)} + 1 \tag{9}$$

where S is the number of nodes in the hidden layer, m is the input length of the model, n is the output length of the model, and a is an integer within [1], [10]. Considering the size of the sample set and reducing the time complexity of model training, this paper selects equation (8) to determine the number of nodes in the hidden layer.

3) THE NUMBER OF HIDDEN LAYERS

The hidden layer in the proposed model is composed of multiple RBMs, and the output of the lower layer RBM is used as the input of the upper layer RBM. The bottom RBM extracts the low-order features of the input data, and the upper RBM extracts the more abstract and high-order features in the data. The number of hidden layers is defined as the number of RBMs stacked in the hidden layer. The effect of the model training is closely related to the number of hidden layers. If the number is too large, the complexity of model training will increase, and the model will easily over-fit. If the number is too small, the model will under-fit, which results in lower accuracy of the model. According to the characteristics of layer-by-layer training of the DBN, this paper uses the reconstruction error R$error$ between the original data and the data obtained by reconstructing the original data to determine the number of hidden layers [28]. The R$error$ is defined as shown in (10).

$$\mathrm{R}error = \sum_{i=1}^{n}\sum_{j=1}^{m}(p_{ij} - d_{ij})/nmp_x \tag{10}$$

where n is the number of samples, m is the dimension, p_{ij} is the reconstructed value, d_{ij} is the real value, and p_x is the number or range of values. The number of hidden layers is calculated, as shown in (11) [28].

$$L = \begin{cases} N_{RBM} + 1, & RError > \varepsilon \\ N_{RBM}, & RError < \varepsilon \end{cases} \tag{11}$$

where L is the number of hidden layers, and ε is the expected threshold of reconstruction error. We take $\varepsilon = 0.05$ [28]. The algorithm for determining the number of hidden layers is shown in algorithm 1.

C. MODEL TRAINING

Model training has the following two steps: bottom-up pre-training and top-down fine-tuning.

1) PRE-TRAINING

The purpose of pre-training is to obtain the initial parameters of the model that fit the sample data through an unsupervised greedy layer-by-layer method. Compared with the DBN network, other neural networks need to set initial weights for parameters and offsets in training. The initial weights directly affect the effect of model training. However, the DBN network determines the parameters of the model in a better range through pre-training and finally obtains the optimal parameters through fine-tuning; thus, it effectively avoids the problems of local optimal solution and gradient explosion caused by initialization.

In the pre-training, we first fully train the bottom RBM, and fix the weight and offset of the bottom RBM. Then, we use the output of the bottom RBM as the input of the upper RBM for training and repeat this step until all RBMs are fully trained. The structure of the RBM is shown in figure 6.

Algorithm 1 Algorithm for Determining the Number of Hidden Layers in a DBN

Inputs: Training dataset *trainData* (*trainData* ∈ *Sample*), threshold of reconstruction error ε, the number of nodes in the hidden layer *dimHid*.
Outputs: Pre-trained *DBN*.

```
Initialize n = 1
  If n == 1                                  // the first RBM
    Rerror = TrainerRBM(trainData, len
     (trainData[0]), dimHid)
                                  // len(trainData[0]) is the
                                     data length
    If Rerror < ε                // determine if the conditions
                                     are met
      Return DBN.append(RBM)
                                  // add the trained RBM to the
                                     DBN and return
    Else
      inpX = RBM.rbm_output(trainData)
                                  // obtain the input data of the
                                     next layer of RBM
    End If
  End If
While (Rerror > ε)                           // other layer
    Rerror = TrainerRBM(inpX, dimHid, dimHid)
                                  // Train an RBM and return
                                     the reconstruction error
    inpX = RBM.rbm_output(inpX)
    DBN.append(RBM)
End While
Return DBN
```

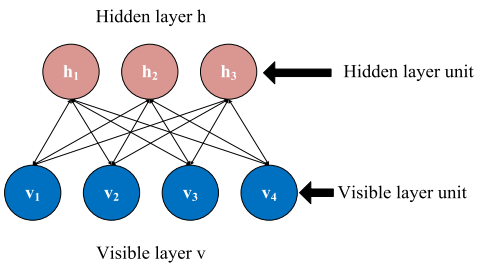


FIGURE 6. The structure of RBM.

As shown in figure 6, the visible layer and the hidden layer are composed of multiple neurons, and the neurons in the layer are not connected; the neurons between the layers are fully connected. The visible layer is used as an input layer of data, and the hidden layer extracts the features of the input data in the visible layer by calculation. Taking the first layer RBM in figure 5 as an example, based on the given input sample $Sample_k$, the state of the hidden layer is calculated, as shown in (12).

$$p(h_j = 1|v) = sigmoid(\sum_i W_{ij} Sample_i^k + b_j) \quad (12)$$

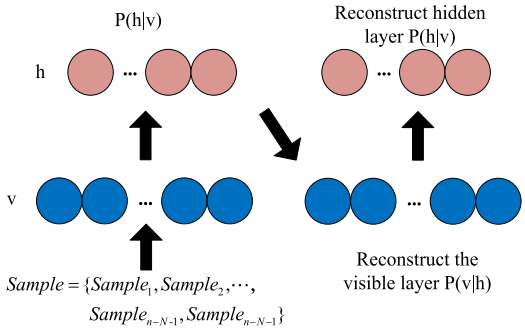


FIGURE 7. The main idea of the CD algorithm.

According to the state of the hidden layer, the state of the visible layer is calculated as shown in (12).

$$p(v_i = 1|h) = sigmoid(\sum_j W_{ij} h_j + a_i) \quad (13)$$

where W_{ij} denotes the weight between the i-th neuron of the input layer and the j-th neuron of the hidden layer, $Sample_i^k$ is the value of the i-th dimension in $Sample_k$, a_i is the bias of the neurons in the hidden layer, and b_j is the bias of the neurons in the visible layer.

This paper uses the classic contrast divergence (CD) [29] algorithm to pre-train the model. Taking the first layer RBM in figure 5 as an example, the main idea of the CD algorithm is shown in figure 7.

First, we set the activation state of neurons in the visible layer according to the *Sample* and calculate the state of all neurons in the hidden layers according to formula (12). Second, after the state of each neuron in the hidden layer is determined, the states of all neurons in the visible layer are calculated according to formula (13), and the reconstructed state of the visible layer is obtained. Finally, we adjust the parameters of the RBM by using the *Rerror* between the state of the visible layer and the reconstructed state of the visible layer.

2) FINE-TUNING

The above pre-training is an unsupervised learning process. Pre-training only ensures that the mapping within each RBM layer is optimal, and the mapping of the entire DBN network is not guaranteed to be optimal. Therefore, as shown in figure 5, this paper adds a logistic regression classifier layer at the end of the model and uses the labelled data to train the entire network after pre-training. We select the backpropagation algorithm [29] to fine-tune the entire network.

VI. EXPERIMENT AND ANALYSIS

For the sake of providing results under actual conditions, we select iMote Traces Cambridge (ITC) and MIT as the experimental dataset. The area under the receiver operating characteristic curve (AUC) and precision are adopted as evaluation indexes. Moreover, we set up three sets of experiments on the ITC and MIT dataset to verify the existence of

TABLE 1. The information of ITC and MIT dataset.

Data Sets	Device	Mobile Nodes	Duration (days)	Network Type	Sampling Interval(s)
ITC	iMote	50	12	Bluetooth	10
MIT	Mobile phone	97	246	Bluetooth	300

information of second-order neighbours of nodes, determine the optimal parameters of the model and verify the validity and stability of the proposed IRWR-DBN model.

A. THE EXPERIMENTAL DESIGN

1) EXPERIMENTAL DATASETS

In this paper, we select the opportunistic network datasets ITC and MIT as experimental datasets, which have different numbers of nodes and sparsity of network connections. ITC [30] is derived from a visual experiment of the student trace of the Cambridge University campus. The MIT [31] dataset is a record of mobile phone communication between students on the MIT campus. They are all wireless ad hoc communication datasets archived by Dartmouth College. The information of the ITC and MIT dataset is shown in table 1.

2) EXPERIMENTAL CONFIGURATION

The experiment is mainly divided into three parts. The first part is experiment 1, which determines if there exist second-order neighbours of nodes in the ITC and MIT datasets; the second part is experiment 2, which determines the optimal IRWR-DBN by comparing its performance under different realistic parameters on the ITC and MIT dataset; The third part is experiment 3, and the effectiveness and rationality of the proposed model are verified by comparing the optimal IRWR-DBN with the traditional similarity indexes and other neural network models. In the experiments, we select the node pairs (21, 34) with the most connections in the ITC dataset and node pairs (27, 62) with the most connections in the MIT dataset as target node pair; 70% of training and 30% of testing are used in the sample set.

We also need to explain that in experiment 2, according to the research on the slice time on the ITC dataset in [22] and [32] (180s, 240s, 300s and 320s), we extended the length of the slice time on the ITC dataset to [100s, 360s], with a step size of 20s. According to the research on the slice time in [23] and [32] (300s and 600s) and the duration of the MIT dataset, we extend the length of the slice time on the MIT dataset to [100s, 1000s], with a step size of 100s. On this basis, in experiment 2, we set the length of the input data to [50,250] in both datasets, with a step size of 20.

B. EXPERIMENTAL RESULTS AND ANALYSIS

1) EXPERIMENT 1: THE ANALYSIS OF THE CONNECTION IN ITC AND MIT DATASET

The above proposed IRWR index utilizes the information of the second-order neighbour of the nodes in the network.

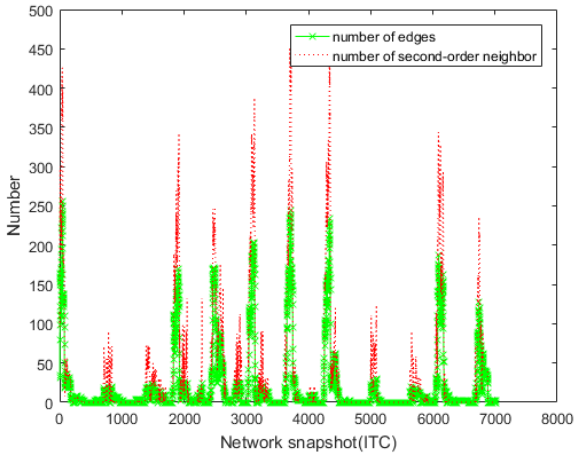


FIGURE 8. The connection of each network snapshot with $\Delta t = 140s$ in the ITC dataset.

To verify the existence of information of the second-order neighbour of nodes in the ITC and MIT dataset, the experiments in this section calculate the number of second-order neighbours of the nodes in the ITC and MIT dataset. In the ITC dataset, taking the length of slice time $\Delta t = 140s$ as an example, the ITC dataset is divided into 7,022 network snapshots, and the number of existing edges in each snapshot and the number of the two-hop nodes for each node in each snapshot are calculated, the results are shown in figure 8. In the MIT dataset, taking the length of slice time $\Delta t = 100s$ as an example, the MIT dataset is divided into 25,536 network snapshots, and the number of existing edges in each snapshot and the number of the two-hop nodes for each node in each snapshot are calculated, the results are shown in figure 9.

As shown in figure 8 and 9, whether in ITC or MIT dataset, in most snapshots, the number of the two-hop nodes for each node is more than the number of existing edges, where the dotted line denotes the sum of the number of the two-hop nodes for each node in each snapshot and the star line denotes the number of existing edges in each snapshot. The results show that there are information of second-order neighbours of nodes in the ITC and MIT datasets and it is feasible to combine the information of second-order neighbours of nodes in the ITC and MIT dataset for link prediction. Additionally, it can be seen in figure 8 that both the number of existing edges and the sum of the number of two-hop nodes for each node present a certain periodic law, because the node in the ITC dataset can be considered as a person carrying a Bluetooth communication device, and most humans have the characteristics of working during the day and sleeping at night. This characteristic results in more connections formed during the day than at night. The 12 peaks (or troughs) in figure 8 correspond exactly to the duration of the ITC dataset (12 days), and it can be seen in figure 9 that the number of two-hop nodes and existing edges of each node in the MIT dataset is less than that in the ITC dataset. This is because the number of nodes in the MIT dataset is larger than that of ITC, and the connections in the MIT are sparse than that of ITC.

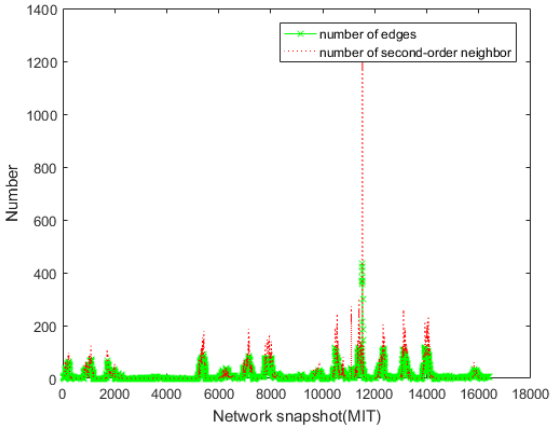


FIGURE 9. The connection of each network snapshot with $\Delta t = 100s$ in the MIT dataset.

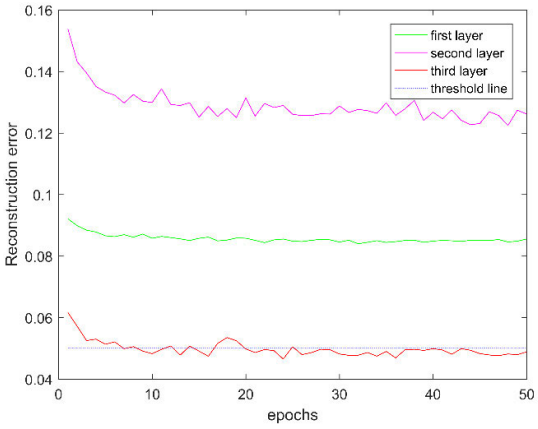


FIGURE 10. The reconstruction error of RBM in each layer of the model with $\Delta t = 140s$ and $N = 210$ on the ITC dataset.

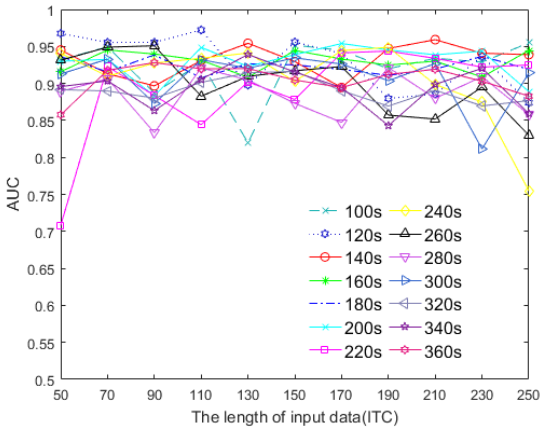


FIGURE 11. The AUC of models under different slice times and input data length on the ITC dataset.

2) EXPERIMENT 2: DETERMINATION OF OPTIMAL PARAMETERS OF THE MODEL

The experiments in this section are mainly to determine the optimal parameters of the model on the ITC and MIT dataset. Following are the parameters we need to determine: the number of hidden layers, the length of a time slice and the length of the input data. Then, the final precision and AUC of the model on the ITC and MIT dataset can be obtained under the optimal parameters.

a: THE NUMBER OF HIDDEN LAYERS

The number of hidden layers has a great influence on the effect of model training. If the number is too large, the complexity of model training will increase, and the model easily over-fits. In contrast, the model will under-fit, which results in a lower accuracy of the model. Based on the above 14 (10) different lengths of time slices and 11 different lengths of input data on the ITC (MIT) dataset, we obtain the number of hidden layers of the model under different parameters according to algorithm 1. Taking the model with $\Delta t = 140s$ and $N = 210$ on the ITC dataset as an example, the reconstruction error of each layer of the RBM is shown in figure 10.

As figure 10 shows, the reconstruction error of the first layer is smaller than that of the second layer because the number of nodes in the visible layer of the first RBM is large and the input data are sparse. The reconstruction error of the third layer is smaller than the second layer because the number of nodes in the hidden layers is the same. When the RBM is added, the learning ability of the model is enhanced, and the reconstruction error is reduced. Additionally, the reconstruction error after convergence in the third layer is less than the threshold. Therefore, the optimal number of hidden layers of this model is 3. The process of determining the number of hidden layers of the model under other parameters (on the ITC and MIT) is the same.

b: THE LENGTH OF SLICE TIME AND THE LENGTH OF INPUT DATA

The length of slice time also has a certain influence on the performance of the model. If the length of slice time is too large, it will cause the valid information in the network to be covered. In contrast, valid information will be truncated and useless information will be produced. An appropriate length for input data can better reflect the relationship between historical data and the current state of the node pair. If the length of the input data is too large, it will also cause the valid information in the network to be covered, and the number of samples will decrease. In contrast, the continuity of the data will be lost, and the intrinsic association of the data will be affected. According to the two sets of data that have been set above on the ITC and MIT dataset (the length of slice Δt time and input data N), AUC and precision of the model under different parameters and dataset obtained by calculating the mean of the results of 30 experiments are shown in figures 11, 12, 13 and 14, respectively.

As figure 11 and figure 12 show, on the ITC dataset, the proposed model can well extract the intrinsic features of link changes in different lengths of slice time and input data. When the length of slice time is 140s, the AUC and precision of the model are the best under different lengths of input data, which are more stable than other lengths of slice time. Additionally, the AUC and precision of the model are the best when the input data length is 210. Therefore, on the

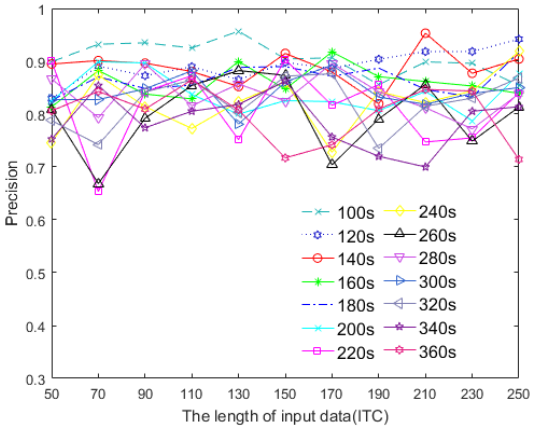


FIGURE 12. The precision of models under different slice times and input data length on the ITC dataset.

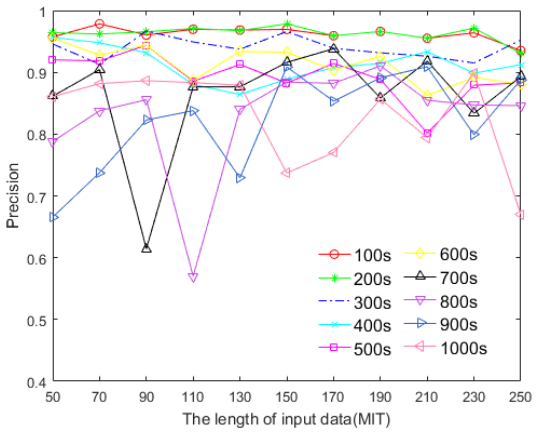


FIGURE 14. The precision of models under different slice times and input data length on the MIT dataset.

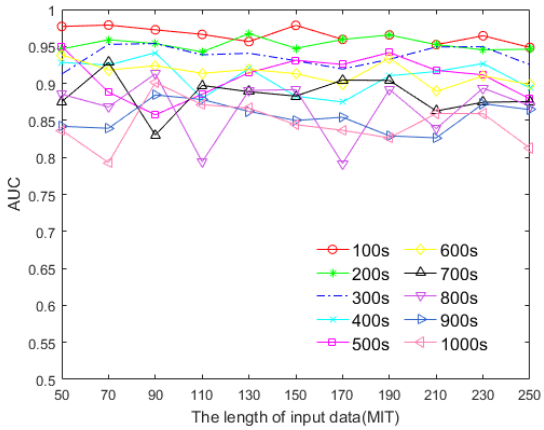


FIGURE 13. The AUC of models under different slice times and input data length on the MIT dataset.

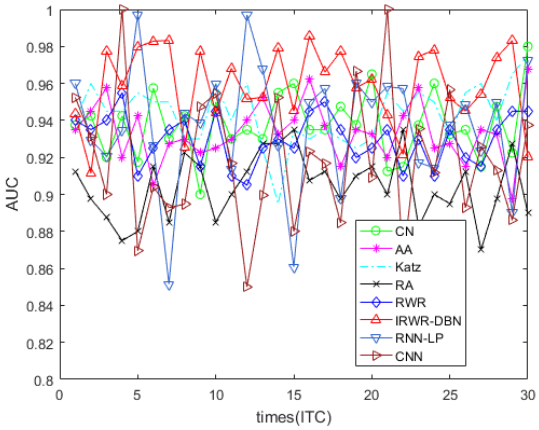


FIGURE 15. The AUC of different prediction methods on the ITC dataset.

ITC dataset, the optimal length of slice time is 140s, and the optimal length of input data is 210.

As figure 13 and figure 14 show, on the MIT dataset, the proposed model can also well extract the intrinsic features of link changes in different lengths of slice time and input data. When the length of slice time is 100s, the AUC and precision of the model are the best under different lengths of input data, which are more stable than other lengths of slice time. Additionally, the AUC and precision of the model are the best when the input data length is 70. Therefore, on the MIT dataset, the optimal length of slice time is 100s, and the optimal length of input data is 70.

3) EXPERIMENT 3: COMPARISON OF DIFFERENT PREDICTION METHODS

The optimal model (on the ITC and MIT dataset) is determined based on the optimal number of hidden layers, the optimal slice time and the optimal input data length, which are determined by the previous experiments. In this section, the effectiveness and rationality of the IRWR-DBN model (under the optimal parameters) are verified by comparison experiments based on the methods of CN, AA, RA, RWR [27], Katz, RNN-LP [22] and CNN [23]. We consider that these methods are used in different scenarios. In order to ensure the reasonability of the experiment, whether on the ITC dataset or the MIT dataset, all methods are implemented in the sample set under the optimal parameters determined above. The results of the experiments are shown in figures 15, 16, 17 and 18.

According to the above comparison experiments, on the ITC dataset, compared with other traditional similarity indexes, the Katz model has a better performance than other traditional similarity indexes on the AUC and precision, among which the performance of RA is the worst. Then, the performance of RNN-LP is better than Katz, but the proposed model IRWR-DBN has a better performance than RNN-LP and CNN in stability and accuracy. On the MIT dataset, the RWR model have a better performance than other traditional similarity indexes on the AUC and precision. Then, the performance of RNN-LP and CNN is better than RWR, and the proposed model IRWR-DBN also has a better performance than RNN-LP and CNN in stability and 97accuracy.

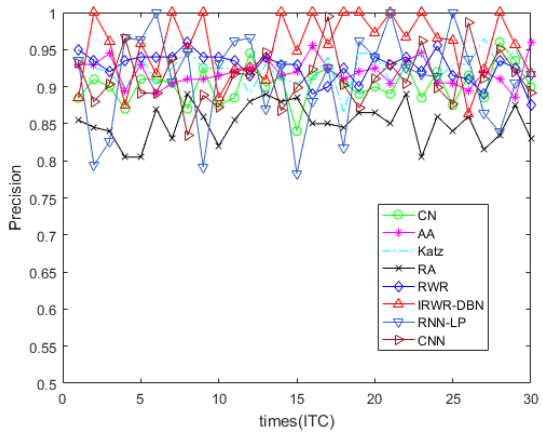


FIGURE 16. The precision of different prediction methods on the ITC dataset.

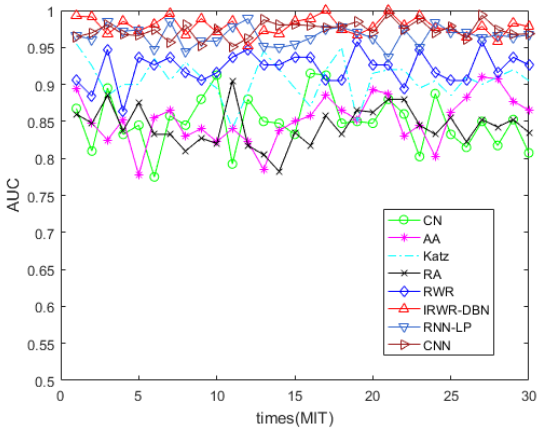


FIGURE 17. The AUC of different prediction methods on the MIT dataset.

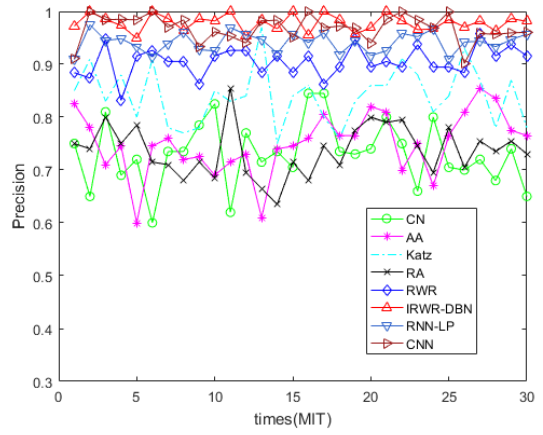


FIGURE 18. The precision of different prediction methods on the MIT dataset.

As figure 15, 16, 17 and 18 show, the proposed model has better performance on both datasets (ITC and MIT), which indicates that the proposed model has strong robustness. Additionally, the performance of CN, RA and AA on the MIT dataset is worse than on the ITC dataset because the number of nodes in the MIT is more than the number of nodes in the ITC dataset and the MIT dataset is sparse than ITC dataset.

TABLE 2. The mean values of AUC and precision for different prediction methods on the ITC dataset.

methods/indicators	mean of AUC	mean of precision
CN	0.9375	0.9028
AA	0.9335	0.9193
Katz	0.9433	0.9238
RA	0.9043	0.8515
RWR	0.9275	0.9266
RNN-LP	0.9370	0.9104
CNN	0.9189	0.9125
IRWR-DBN	**0.9591**	**0.9540**

TABLE 3. The mean valuesof AUC and precision for different prediction methods on the MIT dataset.

methods/indicators	mean of AUC	mean of precision
CN	0.8500	0.7315
AA	0.8523	0.7497
Katz	0.9063	0.841
RA	0.8435	0.7363
RWR	0.9225	0.9053
RNN-LP	0.9655	0.9415
CNN	0.9737	0.9405
IRWR-DBN	**0.9789**	**0.9784**

The mean values of precision and AUC (on the ITC and MIT dataset) of all methods are shown in table 2 and 3 (retaining 4 decimal places). As seen in table 2 and3, the proposed model is superior to other methods in AUC and precision on the ITC and MIT dataset, and it shows that our proposed model can better predict links in networks with different numbers of nodes and different sparsity. These experimental results show that IRWR-DBN has the best stability and accuracy, which indicates that our proposed method can effectively predict the future link of the opportunistic network.

A comprehensive analysis of the above experimental results shows that the proposed model has significantly high prediction performance. Hence, our model is effective and reasonable in the link prediction of opportunistic networks.

VII. CONCLUSION

This paper proposes a link prediction method IRWR-DBN for opportunistic networks. Considering the information of the second-order neighbours of the node, we reconstruct the Markov probability transition matrix and define the similarity index IRWR. We construct the similarity matrix based on IRWR index, so that a sample set is constructed. Then, we utilize the DBN model to extract the time-domain characteristics in the process of evolution of the opportunistic network. The experimental results on the ITC and MIT reality datasets show that the IRWR-DBN model achieves better accuracy

compared with other methods, such as the similarity-based index (CN, AA, Katz, RA, RWR), RNN-LP, and CNN.

REFERENCES

[1] Y. P. Xiong, L. M. Sun, J. W. Niu, and Y. Liu, "Opportunistic networks," *J. Softw.*, vol. 20, no. 1, pp. 124–137, 2009.

[2] S. Sivanantham, G. Balakrishnan, M. Jaidev, and S. Mahesh, "Efficient and opportunistic routing in MANET using link lifetime prediction," in *Proc. 3rd Int. Conf. Trends Electron. Informat. (ICOEI)*, Tirunelveli, India, Apr. 2019, pp. 590–594

[3] M. Naresh, A. Raje, and K. Varsha, "Link prediction algorithm for efficient routing in VANETs," in *Proc. 3rd Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Erode, India, Mar. 2019, pp. 1156–1161.

[4] F. Li, Y. L. Si, Z. Chen, N. Lu, and L. M. Shen, "Trust-based security routing decision method for opportunistic networks," *J. Softw.*, vol. 29, no. 9, pp. 2829–2843, 2018.

[5] Z. Liu, Y. Li, and H. Liu, "Link prediction in evolving networks base on information propagation," *IEEE Access*, vol. 7, pp. 140451–140459, 2019.

[6] J. Guo, L. Shi, and L. Liu, "Node degree and neighbourhood tightness based link prediction in social networks," in *Proc. 9th Int. Conf. Inf. Sci. Technol. (ICIST)*, Hulunbuir, China, Aug. 2019, pp. 135–140.

[7] M. S. Rahman, L. R. Dey, S. Haider, M. A. Uddin, and M. Islam, "Link prediction by correlation on social network," in *Proc. 20th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dhaka, Bangladesh, Dec. 2017, pp. 1–6.

[8] Y. Yang, Z. Ye, H. Zhao, and L. Meng, "Link prediction based on gravitational field of complex network," in *Proc. IEEE Int. Conf. Dependable, Autonomic Secure Comput., Int. Conf. Pervasive Intell. Comput., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, Fukuoka, Japan, Aug. 2019, pp. 499–504.

[9] K.-K. Shang, M. Small, and W.-S. Yan, "Link direction for link prediction," *Phys. A, Stat. Mech. Appl.*, vol. 469, pp. 767–776, Mar. 2017.

[10] X. Hu, S. Liu, S. Chang, and H. Li, "A quad motifs index for directed link prediction," *IEEE Access*, vol. 7, pp. 159527–159534, 2019.

[11] L. Li, S. Fang, S. Bai, S. Xu, J. Cheng, and X. Chen, "Effective link prediction based on community relationship strength," *IEEE Access*, vol. 7, pp. 43233–43248, 2019.

[12] J. Wang, Y. Ma, M. Liu, and W. Shen, "Link prediction based on community information and its parallelization," *IEEE Access*, vol. 7, pp. 62633–62645, 2019.

[13] J. Li, K. Cheng, L. Wu, and H. Liu, "Streaming link prediction on dynamic attributed networks," in *Proc. 11th ACM Int. Conf. Web Search Data Mining (WSDM)*, New York, NY, USA, 2018, pp. 369–377.

[14] Z. Q. Wang, J. Y. Liang, and R. Lu, "Probability matrix factorization for link prediction based on information fusion," *J. Comput. Res. Develop.*, vol. 56, no. 2, pp. 306–318, 2019.

[15] N. M. Ahmed, L. Chen, Y. Wang, B. Li, Y. Li, and W. Liu, "DeepEye: Link prediction in dynamic networks based on non-negative matrix factorization," *Big Data Mining Anal.*, vol. 1, no. 1, pp. 19–33, Mar. 2018.

[16] F. W. Mutinda, A. Nakashima, K. Takeuchi, Y. Sasaki, and M. Onizuka, "Time series link prediction using NMF," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Kyoto, Japan, 2019, pp. 1–8.

[17] N. Y. Yang, T. Peng, and L. Liu, "Link prediction method based on clustering and decision tree," *J. Comput. Res. Develop.*, vol. 54, no. 8, pp. 1795–1803, 2017.

[18] Y. Li, P. Luo, Z.-P. Fan, K. Chen, and J. Liu, "A utility-based link prediction method in social networks," *Eur. J. Oper. Res.*, vol. 260, no. 2, pp. 693–705, Jul. 2017.

[19] K. Lei, M. Qin, B. Bai, G. Zhang, and M. Yang, "GCN-GAN: A non-linear temporal link prediction model for weighted dynamic networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Paris, France, Apr. 2019, pp. 388–396.

[20] H. Shao, L. Wang, and J. Deng, "A link prediction algorithm by unsupervised machine learning," in *Proc. Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE)*, Haikou, China, Jul. 2019, pp. 622–625.

[21] X. Li, W. Liang, X. Zhang, X. Liu, and W. Wu, "A universal method based on structure subgraph feature for link prediction over dynamic networks," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Dallas, TX, USA, Jul. 2019, pp. 1210–1220.

[22] X. Cai, J. Shu, and M. Al-Kali, "Link prediction approach for opportunistic networks based on recurrent neural network," *IEEE Access*, vol. 7, pp. 2017–2025, 2019.

[23] J. Shu, X. P. Zhang, L. L. Liu, and Z. Y. Yang, "Multi-nodes link prediction method based on deep convolution neural networks," *Acta Electronica Sinica*, vol. 46, no. 12, pp. 2970–2977, 2018.

[24] J. Chen, J. Zhang, X. Xu, C. Fu, D. Zhang, Q. Zhang, and Q. Xuan, "E-LSTM-D: A deep learning framework for dynamic network link prediction," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[25] N. Sett, S. Basu, S. Nandi, and S. R. Singh, "Temporal link prediction in multi-relational network," *World Wide Web*, vol. 21, no. 2, pp. 395–419, Mar. 2018.

[26] S. De Winter, T. Decuypere, S. Mitrovic, B. Baesens, and J. De Weerdt, "Combining temporal aspects of dynamic networks with Node2Vec for a more efficient dynamic link prediction," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Barcelona, Spain, Aug. 2018, pp. 1234–1241.

[27] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Proc. 6th Int. Conf. Data Mining (ICDM)*, Hong Kong, Dec. 2006, pp. 613–622.

[28] G. Y. Pan, W. Chai, and J. F. Qiao, "Calculation for depth of deep belief network," *Control Decis.*, vol. 30, no. 2, pp. 256–260, Feb. 2015.

[29] K. Shi, Y. Lu, G. L. Liu, X. Bi, and H. Wang, "A deep belief networks training strategy based on multi-hidden layer Gibbs sampling," *Acta Automatica Sinica*, vol. 45, no. 5, pp. 975–984, 2019.

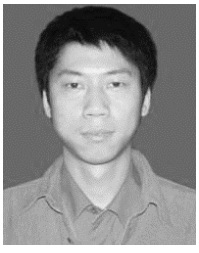
[30] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. (2009). *The Dataset Cambridge/Haggle(V.2009-05-29)*. [Online]. Available: http://crawdad.org/cambridge/haggle/20090529

[31] N. Eagle and A. Pentland. *CRAWDAD Dataset Mit/Reality (v. 2005-07-01)*. Accessed: 2005. [Online]. Available: https://crawdad.org/mit/reality/20050701

[32] M.-H. Zayani, V. Gauthier, I. Slama, and D. Zeghlache, "Tracking topology dynamicity for link prediction in intermittently connected wireless networks," in *Proc. 8th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Limassol, Cyprus, Aug. 2012, pp. 469–474.

ZILIANG LIAO was born in Fuzhou, Jiangxi, China, in 1996. He is currently pursuing the master's degree with Nanchang Hangkong University. His research interest includes opportunistic networks.

LINLAN LIU was born in 1968. She received the B.Sc. degree in computer application from the National University of Defense Technology. She is currently a Professor with Nanchang Hangkong University. Her research interests include wireless sensor networks, software engineering, and distributed systems. She is a member of CCF.



YUBIN CHEN was born in 1977. He received the M.Sc. degree in computer software and theory from Nanchang University. He is currently a Lecturer with the School of Software, Nanchang Hangkong University, China. His research interests include embedded systems and wireless sensor networks.