IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 17, 2020, accepted March 28, 2020, date of publication March 31, 2020, date of current version April 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984593

Resilient Routing Mechanism for Wireless Sensor Networks With Deep Learning Link Reliability Prediction

RU HUANG[1], LEI MA[1], GUANGTAO ZHAI[2], (Senior Member, IEEE), JIANHUA HE[3], (Senior Member, IEEE), XIAOLI CHU[4], (Senior Member, IEEE), AND HUAICHENG YAN[1], (Member, IEEE)

[1]School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
[2]Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China
[3]School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K.
[4]Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield S1 3JD, U.K.

Corresponding author: Ru Huang (huangrabbit@ecust.edu.cn)

This work was supported in part by the Shanghai Science and Technology Development Foundation under Grant 17511108604, in part by the National Natural Science Foundation of China under Grant 61501187, Grant 61673178, and Grant 61922063, in part by the Shanghai Natural Science Foundation under Grant 17ZR1444700, and in part by the Shanghai Shuguang Project under Grant 16SG28.

ABSTRACT Wireless sensor networks play an important role in Internet of Things systems and services but are prone and vulnerable to poor communication channel quality and network attacks. In this paper we are motivated to propose resilient routing algorithms for wireless sensor networks. The main idea is to exploit the link reliability along with other traditional routing metrics for routing algorithm design. We proposed firstly a novel deep-learning based link prediction model, which jointly exploits Weisfeiler-Lehman kernel and Dual Convolutional Neural Network (WL-DCNN) for lightweight subgraph extraction and labelling. It is leveraged to enhance self-learning ability of mining topological features with strong generality. Experimental results demonstrate that WL-DCNN outperforms all the studied 9 baseline schemes over 6 open complex networks datasets. The performance of AUC (Area Under the receiver operating characteristic Curve) is improved by 16% on average. Furthermore, we apply the WL-DCNN model in the design of resilient routing for wireless sensor networks, which can adaptively capture topological features to determine the reliability of target links, especially under the situations of routing table suffering from attack with varying degrees of damage to local link community. It is observed that, compared with other classical routing baselines, the proposed routing algorithm with link reliability prediction module can effectively improve the resilience of sensor networks while reserving high-energy-efficiency.

INDEX TERMS Deep learning, link prediction, routing, wireless sensor networks, reliability.

I. INTRODUCTION

Wireless sensor networks (WSNs) are formed by sensor nodes equipped with wireless communication devices. They are autonomous and are distributed in space, and they play a critical role in the Internet of Things (IoT) systems, enabling real-time monitor and surveillance of physical environment, providing huge data for analytics, and building intelligent models to control the IoT systems. However, the wireless sensor nodes and the networks are usually characterized by very limited storage, energy, and computing resources [1]. Therefore, reducing energy cost, enhancing invulnerability and prolonging networks life are the core goals of designing effective routing mechanism for sensor networks. Classical flat routing algorithms designed for mobile ad-hoc networks include Ad hoc On-Demand Distance Vector Routing (AODV) [2], Destination Sequence Distance Vector Routing (DSDV) and Dynamic Source Routing (DSR) [3], but they perform relatively poor under wireless sensor networks where communications links could be unavailable due to sleeping mechanisms. Routing schemes such as Flooding [4] and Threshold-sensitive Energy Efficient sensor

The associate editor coordinating the review of this manuscript and approving it for publication was Chih-Min Yu.

Network (TEEN) [5] have been proposed for wireless sensor networks, which are mainly designed to address the challenges of network scalability and energy efficiency.

With increasing popularity of huge data and business values of IoT applications and services, there are wide concerns on the malicious attacks to the wireless sensor networks. Due to the limited computation and communication capabilities of the wireless sensor nodes, the sensor nodes and the whole sensor network are vulnerable to the attacks, which can bring huge risks and damages to the IoT systems and services. For example, malicious nodes can pretend to be normal sensor nodes, dropping important messages that should be forwarded to the sink, or sending malicious messages. Due to the potential attacks, the resilience of wireless sensor networks and services can be significantly affected.

Wide research works including intrusion detection systems have been reported to tackle the network attacks, in which the computational load of the operation is transferred to the base station. However, such approach can be problematic with increasing network scale and dynamic communication links. Resilient routing mechanisms were proposed to tackle the problem of routing construction after attacks. Fan *et al.* [6] developed a local strategy to handle the problem of temporary rerouting in response to the faults of random topologies. In [7], autonomous underwater vehicles were adopted to transfer the packets towards the base station for the resilient routing needed by underwater wireless sensor networks. In the scenario of software defined networks, [8] designed a dynamic attack-resilient routing approach via optimizing multi-path routing to defend against the attack, and [9] proposed a fuzzy genetic framework which can provide multiple routes between two nodes when random faults happen in the network.

In this paper we propose a link reliability embedded algorithm for resilient routing in wireless sensor networks to improve the resilience and survivability of the networks. Here link reliability could be an integrated indicator of dynamic wireless communication channel quality, sleeping status of wireless sensor nodes and safety of the links due to potential external network attacks. In addition to the widely used routing metrics such as hops, link capacity and delay, link reliability can be used to solve the challenges posed by the dynamic network topology and network attacks. Link reliability in our paper is defined as the score of the links, which measures the possibility of the existence of a link calculated by the corresponding model and the network survivability, which is computed as function of transmission delay, state of residual energy or node failures state. Some papers have studied energy-efficient and reliable routing for various sensor networks. K. Haseeb *et al.* [10] presented a framework which utilize the end-to-end secure and multi-hop routing paths based on blockchain architecture to ensure the safe and efficient data delivery in sensor networks. Al-Qarni*et al.* [11] used a modified LEACH protocol specially designed for MANET to handle the problem of transmitting multimedia big data that can easily cause node failures because of the limited node energy in wireless networks. Neighbour discovery is also an important part in reliable routing construction. Nur *et al.* [12] developed a collaborative neighbour discovery (COND) mechanism for directional wireless sensor networks with high performance. Ahmed *et al*. [13] designed a QoS aware Routing Protocol originated from AODV which integrated a trust threshold and QoS parameters (transmission rate, link capacity and loss ratio) to construct reliable routing for the purpose of reducing the impact of the nodes with misbehaviour. Energy-efficient and QoS based routing mechanisms can be seen as resilient routing methods which can help improve the lifecycle of the wireless networks and reduce the energy consumption, thus enhance reliable data transmission in the networks [14]–[19].

To provide practical and effective link reliability performance for reliable and resilient routing, we propose to exploit link reliability prediction to cope with the dynamic change in networks.

According to the above methods, link reliability prediction plays an important role for the routing algorithm design. However, it is worth noting that although link prediction has been widely used for many applications such as recommendation system [20], knowledge graph completion [21], chemical molecular structure analysis [22], [23], and complex networks topology analysis [24]. There is very little work on the use of link prediction for wireless sensor networks in resilient routing construction. Link prediction can be utilized to extract missing information, identify false interactive connections, and evaluate the evolution mechanism of complex networks, which is a very useful tool for graph-structure based data mining. The structures of existing networks being studied with link prediction consist of weighted, heterogeneous, static, and dynamic networks [25], [26]. At present, there are four main methods for different link prediction tasks, which are based on local topology information of network, random walk, machine learning, and likelihood analysis. Due to the heterogeneity and dynamics of wireless sensor networks, these existing link-prediction algorithms are effective for our link-reliability-based routing [27].

In this paper we exploit the deep-learning-based link reliability prediction, which is a hot research topic used for data mining in complex networks. Graph Convolution Network (GCN) was proposed to creatively builds Laplacian spectrum convolution structure of complex networks [28], which is mainly used for semi-supervised node classification and link prediction in various networks. However, the depth of the model is limited because excessive layers will cause the graph Laplacian spectrum to be over smoothen [29]. Due to the irregular spatial graph structure and large network scale, traditional machine learning algorithms perform poor on graph-structure data [30]. Therefore, graph-kernel based solution has been adopted to extract and map subgraphs to Euclidean space to reduce data dimension [31]. Another technique called Graph Embedding has been adopted to keep the topological properties by learning graph representation. However, such dimensionality-reduction strategy

cannot guarantee the integrity of the original information [32]. Duvenaud *et al.* [23], extended the application of convolutional neural network to graph structure data, which is constructed to extract node features and perform the task of learning molecular fingerprints. Zhang *et al.* [33] proposed a Palette-Hash Weisfeiler-Lehman [34] graph-kernel based neural machine method (WLNM) to build a link prediction model by training a full connection neural network via using the target link sets in original graph. In this paper we will apply the WLNM to the link reliability prediction task, which has not been investigated in the literature. And based on the strong deep-learning-model based link reliability prediction approach, we design, implement and evaluate the link-reliability based routing algorithm for wireless sensor networks.

The main contributions of this paper are summarized as follows:

- Motivated by the problems mentioned above, we propose a deep-learning based link prediction model WL-DCNN, which combines Weisfeiler-Lehman subgraph labeling with 2-branch convolutional neural networks for the link reliability prediction of undirected graphs. In addition, extra consideration of the imbalance of training samples was taken to improve the way of forming the training set and modify the loss function. Extensive experiments on 6 public datasets show that our method can mine the topology features of complex networks efficiently with prediction results improved by up to 40% over baseline methods.
- Based on WL-DCNN, we propose a resilient routing algorithm (RRM-WLDCNN), which can effectively and accurately predict the reliability of vulnerable wireless links in sensor networks in order to ensure the integrality of transmitted data and shorten the length of transmission path selectively. It also contributes to reduce communication delay and energy consumption of nodes, thus enhance routing robustness and prolonging network lifetime.

The remainder of the paper is organized as follows. Section II illustrates the detailed WL-DCNN link prediction model and the routing mechanism based on the proposed deep learning model. Extensive experimental results and analysis are shown in Section III. Future challenges for our work and conclusion are presented in Section IV.

II. METHODOLOGY

In this paper we consider a general wireless sensor network with a few super sensor nodes, which have sufficient computing power and energy for running deep learning models and taking network control roles. These super sensors nodes will be responsible for link reliability collection for the normal sensor nodes, prediction of link reliability and distribution of the predicted link reliability to the normal sensor nodes. The normal sensor nodes will then utilize the received link reliability in computing routes to the sinks in a distributed way. It is noted that although the collecting and distributing the link reliability messages will generate control overhead, such overhead will not have a much adverse impact on network performance as the network topology is not going to change fast. And the overhead can be easily controlled by changing the frequency of updating the link reliability.

In this section, we will present the overall design methodology. We will firstly present the system framework of WL-DCNN. After that the approaches of extracting subgraphs and constructing samples are introduced. Then the deep learning model for link reliability prediction is designed. Finally, we show how to design a resilient routing algorithm for wireless sensor networks using both traditional path length and link reliability routing metrics.

A. SYSTEM FRAMEWORK OF WL-DCNN

The baseline algorithms for link prediction, which mainly depend on neighbor information or random walk, can be regarded as heuristic schemes. Most of them make use of the multi-level neighbor information from vertices. However, they cannot be consistently extended to general complex networks [22], [27]. In order to make algorithms of link prediction more general, supervised deep learning model is introduced into WL-DCNN for link prediction. Link prediction task can be treated as a supervised binary classification on regression problem. Deep learning model has the preponderance in adaptively mining topology characteristics of complex network. However, the link characteristics of complex networks shows that the number of existing edges is far less than that of non-existing ones. Hence, in the training process, current models should be biased towards negative samples due to the problem of imbalanced sample space. Our proposed model improves the solution to the above problems from two aspects, one is to expand the sample space, and the other is to modify the loss function. The framework of proposed WL-DCNN is shown in Fig. 1. The nodes marked in yellow establish the target links. The upper branch in framework represents the sampling and training process for positive samples, and the lower branch refers to the same operations to the negative samples.

B. GRAPHS AND NOTATIONS

Given a graph $G(V, E)$, V represents the set of all nodes $\{v_1, v_2, \ldots, v_n\}$ of the graph and E denotes a collection of all links in the network. S_{xy} is used to measure the possibility of the connection between each node pairs in the link prediction algorithm. E^T indicates the training set divided by the algorithm and E^P is the validation set. U stands for the set of all edges between any node pairs, called universe set. Define $\Gamma(x)$ as the set of all neighbors of node x, and $k(x) = |\Gamma(x)|$ denotes the degree of node x. In addition, the adjacency matrix $\boldsymbol{A}$ records the full connection details of all nodes. If $\boldsymbol{A}_{ij} = 1$, there exists a link between node i and node j, otherwise $\boldsymbol{A}_{ij} = 0$. The link reliability prediction problem is defined as follows: given an edge $e(u, v) \in E$, find a map function $f : e(u, v) \rightarrow s_{uv} \in \mathcal{R}$ to measure the possibility of the existence of $e(u, v)$.

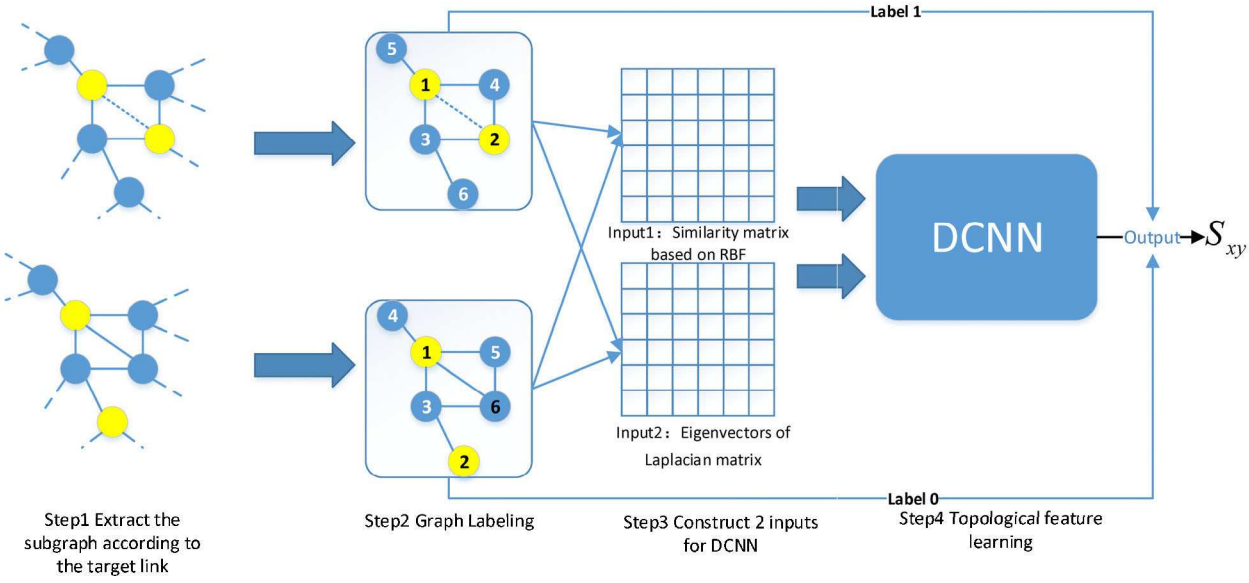


FIGURE 1. WL-DCNN framework. Based on Weisfeiler-Lehman [33] subgraph labeling on the original graph, WL-DCNN creatively constructs main and auxiliary input of DCNN according to the sampled target link, mines the spatial topological characteristics of link community, and finally outputs S_{xy} for each link, where S_{xy} is the possibility of existence of the target link set.

C. WEISFEILER-LEHMAN SUBGRAPH LABELING

Preliminary (Weisfeiler-Lehman Neural Machine (WLNM)): Given a graph $G(V, E)$, WLNM [33] first extracts the subgraph in the graph with the given target link (x, y) as the center to form a subgraph $G(V_K)$, and the number of vertices to be extracted is limited to integer K. Specifically, the k-order (k starts from 1) neighbors of target link are added into $G(V_K)$ until the number of subgraph nodes reaches K. If all the neighbors of the target link are taken whereas the number of subgraph vertices is still less than K, WLNM needs augment $K - |V_K|$ dummy nodes which are isolated from any other vertex. Otherwise, WLNM will simply discarding extra nodes if the number of neighbors of the target link is greater than K. Then we can unify the number of nodes in subgraph generated by each link so as to make use of machine learning model. After finishing the extraction process, $G(V_K)$ needs to be encoded, which is called labeling. Labeling is a map function $f : V^K \rightarrow C^K$, where $V^K = \{v_1, v_2, v_3 \ldots, v_k\}$ is the set of graph vertices, and C^K consists of a set of K unique integers starting from 1. Before running the subgraph labeling algorithm, it is necessary to allocate a precoding for the subgraph. Algorithm 1 illustrates the core process of extracting subgraphs and calculating precoding for each vertex. After allocating the precoding for each node, WLNM uses a hash function as shown in (1), namely the Palette Weisfeiler-Lehman graph kernel, to label each node after sorting the hash values of each node in an ascending order. Finally, the adjacency matrix is constructed and flattened into a vector, which will be fed into a fully-connected neural network (performing the classification task for link prediction).

$$h(x) = c(x) + \frac{\sum_{z \in \Gamma(x)} log(P(c(z))}{\left\lceil \sum_{z' \in V_K} log(P(c(z')) \right\rceil} \quad (1)$$

Algorithm 1 Subgraph Extraction and Precoding

Input: $G(V, E)$, target link (x, y), integer K

Output: subgraph $G(V_K)$ with precoding K vertices and extracting the order list $T(V_K)$

1. **while** $|G(V_K)| < K$ **do**
2. Add nodes in $\Gamma(x)$ and $\Gamma(y)$ into $G(V_K)$, record the order as $T(V_K)$ when adding the vertex
3. **end while**
4. Calculate the shortest distance to the target link of all vertices, using
5. $d(v) = \sqrt{d(v, x)d(v, y)}$ $\forall v \in V_{K/\{x,y\}}$, noting that $d(x) = d(y) = 1$
6. Update vertices labels in $G(V_K)$ using the node index of sorting $d(v)$ in ascending order, respectively

In the above formula, $\lceil \cdot \rceil$ gets the minimum integer greater than the input, $c(\cdot)$ is the current label of vertex, P denotes a set composed by all prime numbers sorted in ascending order, and $P(n)$ takes the n^{th} prime number from the prime set P. We apply (1) to the subgraph $G(V_K)$, and calculate the hash value $h(x)$ for each node, then the final label of each vertex is determined by the indexes after sorting all $h(x)$ in an ascending order.

We refer to the improved Palette Weisfeiler-Lehman with low time complexity to label the subgraph, which can reduce memory consumption and achieve faster convergence. The subgraph extraction process is inherited in our method. In literature [33], nodes with the same topology will get the identical hash value. Accordingly, different nodes get the same encoding label. In this paper, the extraction order list $T(V_K)$ of subgraph nodes is introduced to eliminate this situation

after Algorithm 1 achieves convergence, i.e., in the final iteration of calculating $h(x)$. The modified subgraph labeling function is shown as (2).

$$h^{i+1}(x) = T(x)\left\{c^i(x) + \frac{\sum_{z\in\Gamma(x)} log(P(c^i(z))}{\left\lceil\sum_{z'\in V_K} log(P(c^i(z'))\right\rceil}\cdot\right\} \quad (2)$$

In (2), i denotes the epoch at which the Palette-WL stops. Because of the order-preserving characteristic of Palette WL labeling algorithm [33], $\mathrm{h}^{\mathrm{i}}(\mathrm{x})$ and $\mathrm{h}^{\mathrm{i}}(y)$ can naturally keep order preserving. Now we need to prove that $\mathrm{h}^{\mathrm{i+1}}(\mathrm{x})$ and $h^{i+1}(y)$ are also order preserving. Because $T(x)$ is constructed by the extraction order which strictly follows the rule that closer node neighbors will be sampled earlier. In fact, $\mathrm{h}^{\mathrm{i}}(\mathrm{x})$ also acts as a metric to evaluate the relative distance of node x to the target central node pair. $\mathrm{h}^{\mathrm{i}}(\mathrm{x}) < \mathrm{h}^{\mathrm{i}}(y)$ means node x is closer to the link, then $\mathrm{T}(x)$ must be smaller than $\mathrm{T}(y)$ because of the extraction order. Hence, we can get $\mathrm{h}^{\mathrm{i+1}}(\mathrm{x}) = \mathrm{T}(x)\cdot\mathrm{h}^{\mathrm{i}}(\mathrm{x}) < \mathrm{T}(y)\cdot h^{\mathrm{i}}(y) = \mathrm{h}^{\mathrm{i+1}}(y)$.

In addition, the main improvements of our proposed WL-DCNN model compared with the WLNM method are summarized as follows:

1. The classifier adopted by WLNM is a fully-connected neural network, thus limiting the subgraph scale. Specifically, the number of trainable parameters will increase exponentially. Hence, in Section II-D, we illustrate our proposed DCNN model, whose parameters are less than the fully-connected neural networks even if our model has two branches.
2. WLNM flattens the subgraph adjacency matrix into a one-dimensional vector by concatenating each row of the matrix, which could lead to the problem of information loss. Besides, adjacency matrix is too sparse, which may result that the neural network stop training too early. Hence, we proposed 2 methods to construct the similarity matrix and Laplacian spectrum (Section II-D) respectively as the inputs to feed more information that can be learned by the DCNN deep learning model.
3. Because of the sparsity of complex network links, traditional classifier will be biased towards overfitting the negative samples. This problem is not considered by WLNM as it simply adopts down-sampling of negative samples. We modify the loss function to solve this problem.

Considering the drawbacks of WLNM, we proposed our DCNN deep learning model in the next section.

D. DCNN DEEP LEARNING MODEL

Spatial topology is an important structure feature in complex networks. The efficient solution of link reliability prediction problem is to use deep learning model to learn the unique connection patterns of links in different complex networks. When provided limit information (training set),predicting the possibility of links between the node pairs that do not have a reliable link at the present is of vital importance. The general fully-connected neural network is not sensitive to data with spatial displacement, which is mainly due to its limited ability of extracting local spatial features [35]. If the number of subgraph nodes is large, fully-connected neural networks will also bring about the problem of dimension explosion, resulting in poor robustness. Hence, the fully-connected neural network is not suitable to perform topology learning tasks in complex networks.

CNN has excellent ability of extracting spatial and abstract features in the field of computer vision, characterized by sharing weights. Inspired by the multimodal analysis in the field of MRI medical image processing [36], we designed a Dual Convolutional Neural Network for link reliability prediction task, which can extract and fuse the features of two different inputs of the same sample. Compared with single input source, we found DCNN can learn more comprehensive pattern of the target link. More specifically, the proposed DCNN model consists of independent 2-branch convolution layers, pooling layers, feature fusion layer, and logistic regression layer, then we can get the numerical existence possibility of the target link in the output layer.

The framework of DCNN is shown in Fig. 2, in which $K*K*1$ denotes $length * width * channels$ of an input sample, and K is also the number of nodes in the subgraph $G(V_K)$. As for convolution kernels, $3*3*n$ is defined as the $length * width * the\ number\ of\ kernels$. The upper branch accepts the similarity matrix of sampled link as the input, and the spectrum eigenvector matrix of the same subgraph generated by the target link will be inputted to the lower branch.

The element $\mathbf{0}$ in the similarity matrix has true physical meaning, implying that there is no link between node pairs or the link reliability is pretty low. Therefore, the boundary padding method of the convolution layers in the upper branch is set as None Zero Padding. As for the pooling layers and convolution layers in the lower branch, Zero Padding strategy is adopted. In the feature fusion layer, two dense layers will be concatenated.

DCNN model takes sigmoid function as the activation function in the output layer, shown as (3). We adopt the binary cross entropy function as the loss function, shown as (4), where $y^{(i)}$ is the label of the i^{th} sample. If there is a link between two nodes, then $y^{(i)} = 1$, otherwise $y^{(i)} = 0$. And $\hat{y}^{(i)}$ is the predicted value of the model for the i^{th} sample, forming the Logistic Regression in the output layer.

$$sigmoid\ (z) = \frac{1}{1+e^{-z}} \quad (3)$$

$$loss = -\sum_{i=1}^{N} y^{(i)} log\hat{y}^{(i)} + (1-y^{(i)}) log(1-\hat{y}^{(i)}) \quad (4)$$

Batch training is adopted during the training process, and batch size is determined for different datasets, which is set as $0.1\left|E^T\right|$.

Particularly, due to the unique characteristics of complex network links, the number of existing links is far less than that

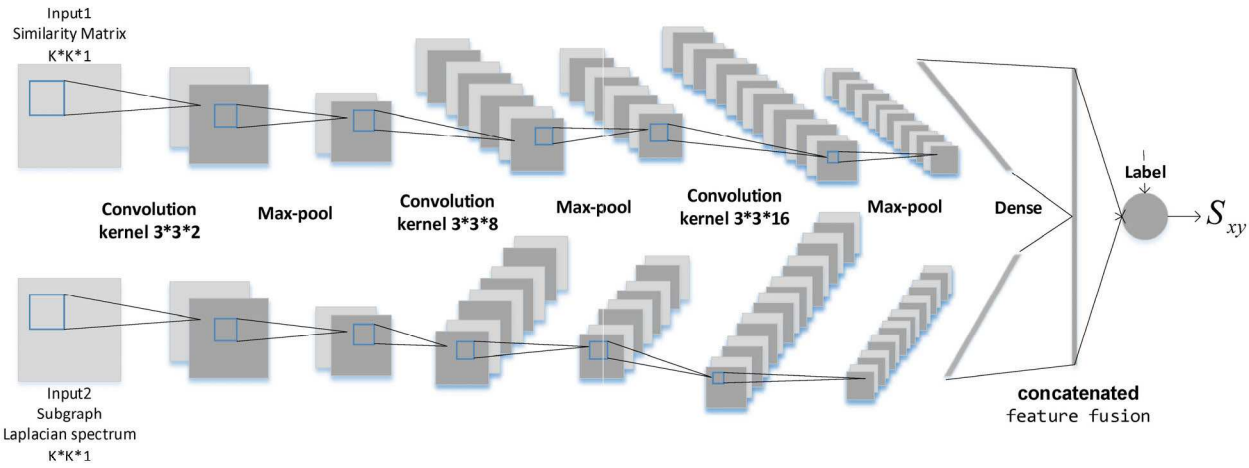


FIGURE 2. Link feature mining based on DCNN deep learning model. DCNN model has 2 branches, and each branch accepts an input from the same sample and merges its feature maps before the Logistics Regression output layer in order to learn a more comprehensive pattern of the sampled link.

of non-existing ones, resulting that the number of negative samples is higher than that of positive samples. Therefore, when training the DCNN model, it is necessary to modify the loss function and adjust sample space. When facing the problem of uneven samples, it can be solved from two aspects. One is to change the sample space, the other is to modify the classifier [24], such as down-sampling, up-sampling, or modifying the loss function. We can also change the threshold of the classifier in the problem of binary classification. In this paper, down-sampling of negative samples and modifying loss function are combined to solve the problem of uneven samples in the link reliability prediction task.

A complex network $G(V, E)$, composed of N nodes, has $C_N^2 = N(N-1)/2$ possible links in total, but the number of links that exist is lower than C_N^2. The training time of deep learning model can be reduced by down-sampling of negative samples, inevitably resulting in the situation that the pattern model learned is incomprehensive. Furthermore, adding virtual positive samples is infeasible. Hence, we reduce the number of negative samples to three times the number of positive samples in the training sets by random down-sampling. Then, the loss function is modified to increase the loss and penalty for positive samples, which should hold a small proportion in training sets. Equation (5) denotes the modified loss function, where $\omega^{(i)}$ is the penalty parameter, determined by the composition of sample space, i.e., the proportion of positive and negative samples. By combining the down-sampling with the modified loss function, our model can attach more importance to the positive samples, and the sample space can be balanced accordingly.

$$loss = -\sum_{i=1}^{N} \omega^{(i)} \left[y^{(i)} log\hat{y}^{(i)} + (1 - y^{(i)}) log(1 - \hat{y}^{(i)}) \right] \quad (5)$$

In order to avoid overfitting, we add a regularization item to the loss function, which is composed of the F-norm of convolution kernel weights W_k and L2-norm of weights W_l for fully-connected layer, shown in (6).

$$L_{reg} = \|W_l\|_2 + \sum_{k \in kernels} \|W_k\|_F^2 \quad (6)$$

The final loss function is defined as (7), where β is the importance of regularization item determined in the training process.

$$L = \beta L_{reg} + loss \quad (7)$$

To minimize the loss function (7), Root Mean Square Prop (RMSProp) optimizer is used to train the proposed DCNN model, thus making gradient fluctuation of the loss function smaller and achieving faster convergence [38]. The proposed DCNN model requires two different inputs from a sample at the same time. Section II-C gives the process of subgraph extraction and labeling, which can generate a subgraph $G(V_K)$ centered on the target link. However, if the adjacency matrix $\boldsymbol{A_{ij}}$ of subgraph $G(V_K)$ is directly used as the input, the learning process of DCNN will be terminated soon caused by the sparse property of $\boldsymbol{A_{ij}}$. Moreover, the adjacency matrix $\boldsymbol{A_{ij}}$ contains only the first-order neighbor information between nodes, which provides limited information. As a result, DCNN may not learn the comprehensive pattern of the subgraph generated by the target link. In addition, direct connection information of the target link should not be provided to DCNN. Therefore, we propose a reasonable way based on radial basis function to construct the main input $\boldsymbol{S_{ij}}$ of DCNN, namely the similarity matrix of subgraph. Algorithm 2 illustrates the detailed steps.

The similarity matrix constructed by Algorithm 2 does not contain the direct connection information of the target link, and the input is normalized by radial basis function, which can map low-dimensional data that is linearly non-separable into high-dimensional space [39] which is

Algorithm 2 Constructing the Main Input of DCNN

Input: $G(V_K)$

Output: $\boldsymbol{S}_{ij}$

1. **Let** $d(1,2) = d(2,1) = M$, where $M \gg K$.
2. Calculate the shortest path $d(i,j), \forall i,j \in V_{K/\{1,2\}}$ between node i and j. If there exists no path between i and j, **Let** $d(i,j) = inf$.
3. $\boldsymbol{S}_{ij} = exp\{-\frac{d(i,j)^2}{2\sigma^2}\}$, σ denotes the radical width.

linearly separable. This operation can improve the classification and representation ability of classifiers.

Compared with indexes such as the degree distribution, clustering coefficient and the betweenness centrality [40], Laplacian spectrum of complex networks is a more comprehensive measurement related to the network topological structure [41]. In this paper, the matrix $\boldsymbol{L}'_{ij}$ composed by Laplacian eigenvectors of subgraphs is used as the auxiliary input for DCNN model. The approach of constructing spectral eigenvector matrix is given by Algorithm 3.

Algorithm 3 Constructing the auxiliary input of DCNN

Input: $G(V_K)$

Output: Laplacian spectrum matrix $\boldsymbol{L}'_{ij}$.

1. Construct adjacency matrix $\boldsymbol{A}_{ij}$ of $G(V_K)$, and Let $\boldsymbol{A}_{12} = \boldsymbol{A}_{21} = 0$ to remove the connection information of the target link.
2. $\boldsymbol{L}_{ij} = \boldsymbol{D}^{+}(\boldsymbol{D} - \boldsymbol{A})$, diagonal matrix $\boldsymbol{D}_{ii} = \sum_{j=1}^{K} \boldsymbol{A}_{ij}$. $\boldsymbol{D}^{+}$ is the pseudo-inverse of $\boldsymbol{D}$.
3. Perform eigen decomposing on $\boldsymbol{L}_{ij}$. Sort its eigenvalues in ascending order $\lambda_1, \lambda_2, \lambda_3 \ldots \lambda_k$.
4. $\boldsymbol{L}'_{ij} = [\eta_1, \eta_2, \eta_3 \ldots \eta_k]$, η_i is the eigenvector corresponding to the sorted eigenvalue λ_i.

In the process of subgraph extraction, if the number of neighbors in target link set is lower than K, we need replenish dummy nodes to unify the subgraph size. Accordingly, it will cause the degree matrix $\boldsymbol{D}$ of the subgraph to become a singular matrix. Therefore, in Algorithm 3, pseudo-inverse of matrix $\boldsymbol{D}$ is used to solve the problem of calculating the inversion for a singular matrix.

E. RESILIENT ROUTING MECHANISM

WSNs are the powerful and economical tool to establish communication and perform the task of data transmission in the unsupervised area which is beyond the reach of human. The nodes in the networks could be exposed to various kinds of threats, such as deliberate attack and node failures. In this paper, the problem of resilient routing is defined as follows: when given a network which has been attacked, a resilient routing mechanism needs to be designed to eliminate the impact of the attack and resume as much as possible.

Most of current routing mechanisms contain the process of broadcasting to establish routing entries. Once the network is attacked in the process of broadcasting, the path it forms may no longer be the shortest one. Our motivation is to design such a routing method that can reduce the impact of the incomplete link information that is resulted from the attack in the period of the broadcasting as much as possible.

Hence, we propose a resilient routing mechanism based on the combination of current shortest path and link reliability metric in the case of incomplete and/or unsafe link information caused by malicious attack in sensor network. First, the cost function of the data flowing between arbitrary node pairs $\langle i,j \rangle$ is defined as follows:

$$Cost = \frac{|D\langle i,j\rangle|}{\prod_{k\in I} p\langle i,k\rangle p\langle k,j\rangle} \tag{8}$$

where $|D\langle i,j\rangle|$ is the distance of the currently shortest path between node i and node j in sensor network, whose link information may have been partially lost, $\prod_{k\in I} p\langle i,k\rangle p\langle k,j\rangle$ represents the product of the reliability of all single hop in the selected shortest path I. If the link reliability prediction model is not used, and only the currently known incomplete link information is used, we can get $\prod_{k\in I} p\langle i,k\rangle p\langle k,j\rangle = 1$. Then the cost function value is equal to the length of the selected path, and the above cost function degenerates to the shortest path route. The following theorem proves the feasibility of RRM-WLDCNN, which can efficiently abate the cost function via using link prediction model.

Theorem: There exists a possible path, which is shorter than the currently shortest path with certainty, and we can get a smaller cost, by using the defined cost function (8)

Proof: Write $|D\langle i,j\rangle| := Z$, $\prod_{k\in I} p\langle i,k\rangle p\langle k,j\rangle := W$, and adopt the natural logarithm of (8) without changing the monotonicity of the original function.

THEN we can rewrite the cost function as

$$Cost = ln\, Z - ln\, W \quad s.t.\, Z \in N^{+},\ W \leq 1 \tag{9}$$

Equation (8) is differentiable on its defined intervals, and its partial derivative is continuous, then there exists its directional derivative.

Define direction vector $\vec{l} = (z - z_0, w - 1)$ at arbitrary point$(z_0, 1)$, where z_0 denotes the hops of the current shortest path and 1 is the product of the link reliability in each hop between nodes i and j.

Then, we can get the directional derivative of (9) at arbitrary point $(z_0, 1)$ as follows:

$$\frac{\partial Cost}{\partial \vec{l}} = grad\ Cost \cdot \frac{\vec{l}}{\left\|\vec{l}\right\|_2} \tag{10}$$

$$\frac{\partial Cost}{\partial \vec{l}} = \frac{Z - z_0 W}{ZW\left\|\vec{l}\right\|_2} \tag{11}$$

$$s.t \begin{cases} 0 < W \leq 1 \\ Z < z_0 \end{cases}$$

In (10), *grad* indicates the gradient of the cost function (9). Because the minimum value of $Z - z_0 W$ is less than zero and its maximum value is greater than zero, we can derive that there exists an interval to make the directional derivative

$\partial Cost/\partial \vec{l} < 0$, which means we can get a new path whose cost is smaller than the current shortest path. Hence, we can prove that there exist possible paths between i and j, which has a smaller cost than the shortest path with certainty.

The above mathematical derivation for the directional derivative of cost function proves that the transmission cost can be reduced by using link-prediction-based routing mechanism under the premise of link information loss. Meanwhile, it is proved that the link prediction model can be used in route selection, considering that its cost function value less than that of current known shortest path. It means the mechanism can alleviate the influence that the attack brings and is energy-efficient, thus prolonging the lifecycle of the network. This is the core idea of our proposed mechanism. Our method can be also called semi-static routing, because the ordinary nodes in the network just need to use the routing entries and the reliability matrix P distributed by the super node to perform the transmission task. The super node will update the routing entries and the reliability matrix in fixed intervals under the assumption that the network topology will not change too fast. Hence, the energy consumption and transmission delay caused by incomplete link information can be greatly reduced, thus ultimately enhancing the survivability of sensor networks.

Algorithm 4 RRM-WLDCNN Routing Mechanism

Input: Source i, destination j, reliability matrix P, threshold γ, waiting time τ

Output: Updated routing entry of nodes i, j

1. Query the current routing table of the starting point i, get the shortest path $Path\langle i, j\rangle$ to the destination j, and prepare to start data transmission.
2. **for** each node $k \in Path\langle i, j\rangle$ of each hop in the transmission process **do**
3. Calculate the current cost C of transmitting to destination directly.
4. **if**[1] $P_{kj} > \gamma$ and $C > length(Path\langle i, j\rangle)$
5. The information is transmitted to both the destination j and the next hop in the initial shortest path.
6. **else**
7. The transmission will continue to the next hop of the initial shortest path
8. **end if**[1]
9. **if**[2] the receiving confirmation is returned from j within τ
10. Update the routing entry of nodes i and j according to the path of data transmission.
11. **end if**[2]
12. **end for**

Algorithm 4 depicts the detailed RRM-WLDCNN routing mechanism, and we set a reliability threshold γ to reduce the huge invalid transmission. Before running the Algorithm 4, the super node is needed for preprocessing, which consists of building the current shortest path routing table of each node,training the WL-DCNN deep learning model by using the complete link-information dataset, and distributing the calculated reliability matrix P which contains the link reliability of all links to each normal node in the network. The entire routing discovery and establishing phases is summarized as follows:

1. The super node in the network first uses AODV routing method to generate the shortest paths for all the nodes to initialize the routing entries and exploits the network topology feature to train the WL-DCNN model, calculating a link reliability matrix P composed of each links.
2. After establishing the routing table and the matrix P, the routing table and matrix P will be distributed to each node in the network.
3. Ordinary nodes in the network will use Algorithm 4 to finish the data transmission task.
4. The super node will update the whole routing entries and distribute it to other ordinary nodes at fixed intervals

III. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we present the experiment results on the link prediction model and the implementation for the routing algorithm with deep-learning-driven model validated by public complex network datasets. Based on the experiments' analysis results, we demonstrate the utility and performance of the proposed RRM-WLDCNN routing algorithm.

TABLE 1. Statistical data of complex network datasets.

Dataset	Nodes$\lvert V\rvert$	Edges$\lvert E\rvert$	$\langle k\rangle$	Radius$\langle r\rangle$
Yeast	2361	7182	6.084	6
Power	4941	6594	2.669	23
C.elegans	297	2148	14.465	3
Routers	6474	13895	4.293	5
NS	1589	2742	3.451	8
PB	1490	16718	22.44	4

A. DATASETS

We selected 6 public complex network datasets to test the performance of the proposed link prediction model, including protein interaction network (Yeast), American power grid (Power), neural networks of Caenorhabditis elegans (C.elegans), Internet routers (Routers), net of scientists (NS), and American political blogs (PB). The core topology statistics of these complex networks are given in Table 1. The selected six complex networks involve the field of biology, physical topology network, and interpersonal relationship, which are highly representative and valuable datasets in the field of link prediction, and the size of these datasets range from small to medium. $\langle k\rangle$ in Table 1 is defined as the average degree of networks. For each dataset in the experiment, 90% links which are randomly selected and all nodes in the original network $G(V, E)$ are utilized to construct positive

training samples by using the proposed Algorithm 2 and 3. Then, we randomly select the links that do not exist in each network to construct negative samples using the same methods, under the condition that the number of negative samples is three times of the positive samples. Hence, we can get a complete training set E^T. The remaining 10% of the edges in the network will be used to form the positive samples in the validation set. There are n negative samples in validation set, where n is determined by the number of independent comparisons when calculating AUC. It should be noted that there is no overlap among the negative samples between training set and validation set. Consequently, the construction of validation set E^P is completed.

B. BASELINE METHODS AND PARAMETERS SETTING

There exist many baseline methods for link prediction. In this paper, 9 classic baseline algorithms covering 5 main research directions of link prediction in complex network are selected for performance comparison with our proposed model. They are Common Neighbors (CN), Jaccard, Adamic-Adar (AA), Preferential Attachment (PA), Local Naive Bayes method-Common Neighbor (LBCN), Katz, Leicht-Holme-Newman-II (LHN-II), Superposed Random Walk (SRW), and Weisfeiler-Lehman Neural Machine (WLNM). In WLNM model, the number of subgraph nodes is unified to $K = 10$, and the hidden layers consist of three fully connected neural network with the activation functions being all ReLU (Rectified Linear Unit). Besides, the number of neurons in each hidden layer is set to [16], [32], [32] in WLNM model with Softmax layer being the output.

TABLE 2. Parameters setting of benchmark methods.

Algorithms	Parameters
Katz	Attenuation factor $\beta = 0.01$
LHNII	$\varphi = 0.9$
SWR	Particle return probability ρ=0.85，walking steps 4

Table 2 shows the specific parameter settings of baseline algorithms using path similarity and random walk.

In the WL-DCNN model, the number of extraction subgraph nodes is $K = 2\langle k \rangle$, where $\langle k \rangle$ denotes the average degree of tested network. The value of K ensures that at least the 1st order topological information of complex network links can be learned via the model. Radial basis function width is set as $\sigma = 2$. The maximal training iterations is 1000, and the early stopping condition is that the loss value keeps the same in consecutive 30 times. Table 3 gives the detailed hyper parameter settings of the DCNN.

C. EVALUATION METRICS

AUC (Area Under the receiver operating characteristic Curve) [42] can be used to evaluate the overall performance of

TABLE 3. Hyper parameters setting.

Layer Name	Parameters	Stride	Activation
[1] Convolution Layer 1	Kernel Size：3×3×2	1	ReLU
[2] Max Pooling Layer 1	Kernel Size：2×2	2	/
Convolution Layer 2	Kernel Size：3×3×8	1	ReLU
Max-Pooling Layer 2	Kernel Size：2×2	2	/
Convolution Layer 3	Kernel Size：3×3×16	1	ReLU
Max-Pooling Layer 3	Kernel Size：2×2	2	/
Dense(Input1)	Output dimension：128	/	ReLU
Dense(Input2)	Output dimension：128	/	ReLU
Concatenated Layer	Dimension：256	/	/
Output Layer	I/O Dimension：256/1	/	Sigmoid

[1,2] All convolution/pooling layers in both of the two branches of DCNN model have the same setting.

link prediction algorithm, which is shown as (12). We repeatedly and independently calculate the S_{xy} of each link for n times to compare the corresponding scores. Links should be randomly and respectively chosen from E^P and U/E. As a result, there are n' times that S_{xy} of links in E^P is greater than that of links in U/E, and n'' times for the opposite circumstance. Based on the random guess in link prediction execution, AUC will converge to 0.5. When AUC is less than 0.5, it shows that the algorithm is less effective than random guess.

$$AUC = \frac{n' + 0.5n''}{n} \tag{12}$$

$$Precision = \frac{m}{L} \tag{13}$$

$$RS = \frac{1}{|E^P|} \sum_{i \in E^P} RS_i = \frac{1}{|E^P|} \sum_{i \in E^P} \frac{r_i}{|H|} \tag{14}$$

In addition, Precision [43] and Ranking Score [44] can be used to evaluate the implementation accuracy in link prediction, which is defined as (13), i.e., the ratio of m to L, where m indicates the number of edges in E^P whose S_{xy} are in the top-L.

Ranking Score (RS) focuses on the ranking of predicted edges, shown as (14), where $H = U - E^T$, r_i is the ranking of the target link, and $|H|$ denotes the number of nodes in E^T

D. PERFORMANCE ANALYSIS ON LINK RELIABILITY PREDICTION MODELS

In this section, we compare the WL-DCNN link reliability prediction model with nine baseline methods. AUC and

TABLE 4. The AUC results of WL-DCNN Model and nine benchmark link prediction methods.

Algorithms	AUC									
	WLDCNN	CN	Jaccard	AA	PA	LBCN	Katz	LHNII	SRW	WLNM
Yeast	**0.986**	0.723	0.726	0.722	0.881	0.730	0.911	0.787	0.872	0.956
Power	**0.971**	0.618	0.620	0.619	0.580	0.617	0.936	0.930	0.746	0.848
Cleans	**0.908**	0.853	0.804	0.867	0.746	0.861	0.874	0.616	0.906	0.859
Routers	**0.988**	0.755	0.732	0.756	0.940	0.753	0.871	0.550	0.943	0.944
NS	**0.994**	0.943	0.943	0.921	0.745	0.951	0.991	0.930	0.910	0.984
PB	**0.945**	0.919	0.910	0.926	0.907	0.937	0.933	0.552	0.786	0.933

precision are selected as evaluation metrics in the experiments. The AUC results of baseline algorithms are derived from the average of 100 independent experiments, so as to minimize the accidental errors during the experiment. During the calculation of AUC, the number of independent comparisons n is set to 10000. The AUC experimental results of WL-DCNN are also taken from the average output from 100 independent experiments, whereas the deep learning model being trained once. In each independent experiment on WL-DCNN, the negative samples in the verification set are generated by randomly resampling from the set of U/E and inputting them into the model to obtain their S_{xy}, which is used to calculate the final AUC.

The input part of WL-DCNN is completed by MATLAB r2019a, and the deep learning model of DCNN is built by KERAS with the following experimental environment: AMD Ryzen R5 @ 3.5GHz × 4, 12G RAM, gtx1060 GPU, 6G VRAM.

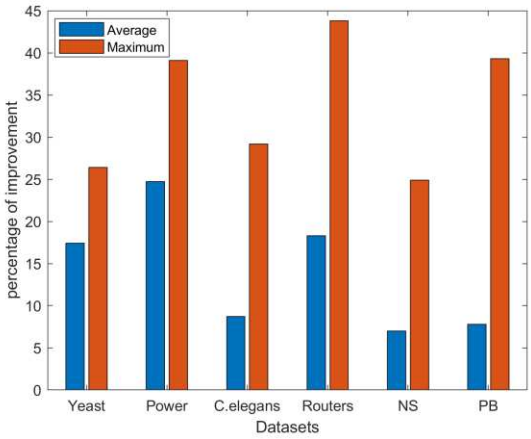


FIGURE 3. AUC improvement visualization of WL-DCNN model based on nine benchmark algorithms.

1) COMPARISON OF THE LINK PREDICTION ALGORITHMS

The AUC results of all the algorithms are summarized in Table 4, and the corresponding best results of each algorithm in the data set have been marked in bold. It can be seen from Table 4 that the proposed link prediction model, i.e., WL-DCNN is superior to the nine baseline algorithms in each dataset with AUC result improved up to 40%, especially on Power and Routers, where algorithms perform poor due to its complicated connection characteristics The improvement effect of the proposed link prediction model on six complex network data sets is shown in Fig. 3 calculated by Table 4.

Table 4 also verifies that the traditional link prediction algorithms, which are subjected to the unique topology characteristics of different networks, are not applicable to all kinds of complex network data sets. For example, for the NS dataset, each benchmark algorithm performs quite well in AUC results, which is under such the premise that the neighborhood characteristic of interpersonal networks is significantly marked and the neighborhood order is moderately small. As for the Power dataset, the performance of traditional benchmark algorithms based on neighbor information is poor, but the Katz, LHNII and other algorithms based on path information perform well. One of the main reasons is that the Power depends on the local sophisticated dataset has made it overwhelmingly vulnerable to link connection characteristics, whereas the path-based algorithms such as Katz can take the high-order neighbor into account. However, on the C.elegans dataset, the performance of nine benchmark algorithms is quite poor due to the complicated and large-scale connection features of Caenorhabditis elegans neurons, whereas the AUC value of the proposed WL-DCNN model still achieves more than 90%. WL-DCNN model utilize at least the first-order neighbor information of the target links in general. In addition, the features of links in complex network can be extracted by the proposed deep learning model with the link connection pattern being efficiently learned. The average AUC value of WL-DCNN model on six complex network datasets exceeds 90%, and it can be concluded that the WL-DCNN model can be taken as a more universal model in solving the link prediction problem, and can be more widely applied to various kinds of complex network datasets.

Precision can also be used as a criterion to evaluate the effectiveness and accuracy of the link prediction algorithms. Fig. 4 shows the comparison in precision results between the proposed model and eight benchmark algorithms.

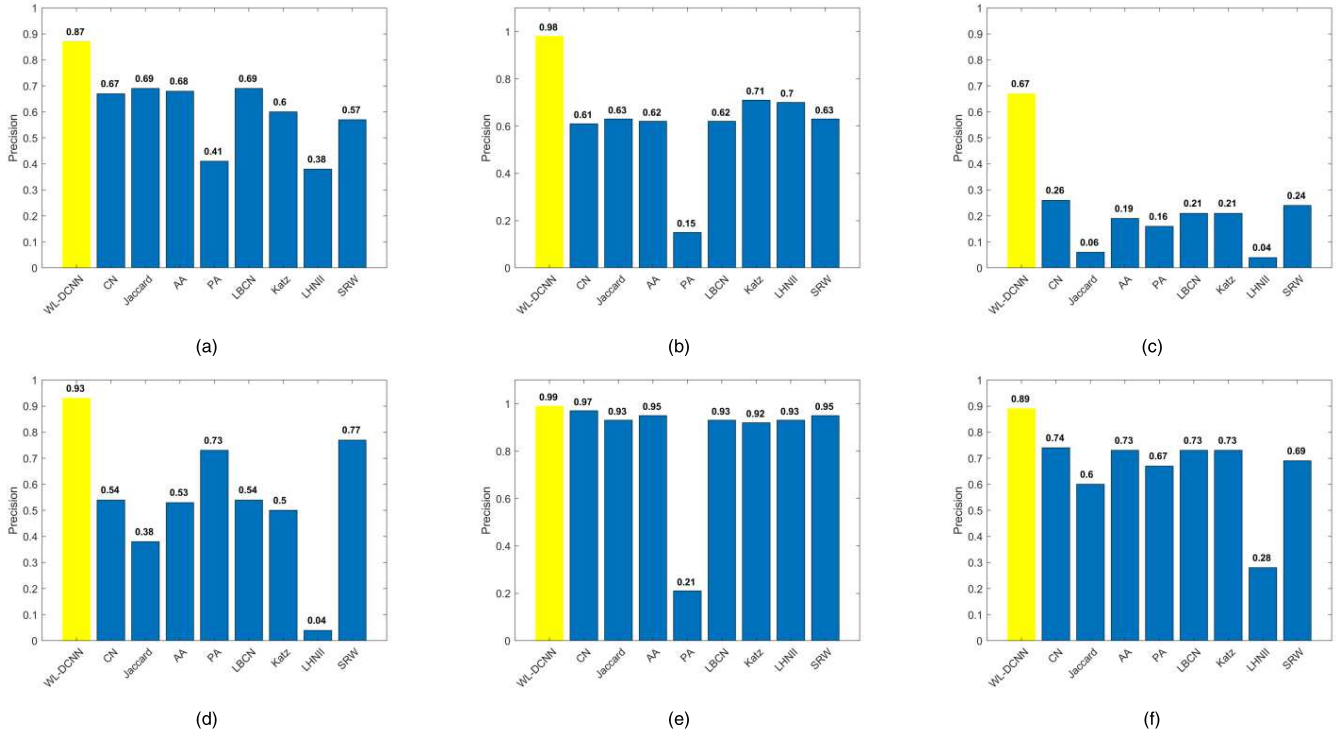


FIGURE 4. Experimental results of Precision compared with 8 benchmark algorithms. (a) Dataset: Yeast, $L = 665$; (b) Dataset: Power, $L = 660$; (c) Dataset: C.elegans, $L = 215$; (d) Dataset: Routers, $L = 1258$; (e) Dataset: NS, $L = 275$; (f) Dataset: PB, $L = 1672$.

To calculate precision, L needs to be specified. In the experiment, L is determined by the number of positive samples in validation set N_P. For each dataset, the algorithm with the best result is marked in yellow. It can be seen from Fig. 4 that the prediction results of WL-DCNN model outperforms the heuristic benchmark algorithms among the six tested complex network datasets, especially on C.elegans dataset, which can be ascribed to the excellent link feature extraction ability of our model. In the experiment on the NS dataset, except for PA algorithm, all the algorithms have achieved excellent prediction results, with precision value over 90%, which is due to the key elements including neighbor property, small network diameter, and uncomplicated connection characteristics of NS network. Consequently, all algorithms can basically use the low-order neighbor information to solve the link prediction problem. Fig. 4(b), (e), (d) also illustrate that the PA link prediction algorithm is exclusively suitable for the limited and special datasets such as Routers whose connection follows man-made routing guidelines, namely, more fixed link pattern.

The experimental results on 6 real world datasets in Table 4 and Fig. 4 demonstrate that WL-DCNN can adapt to multi-type complex network datasets, and the prediction results is better than heuristic link prediction methods. The reason is that the proposed model can make full use of the topological characteristics of complex networks, via extracting their link connection features and spontaneously learning the topological patterns of complex networks.

2) IMPACT OF K

For the subgraph $G(V_K)$, K is the number of subgraph nodes, and determines the neighbors' order extracted from the target link, which directly affects the topological features learned from the target link. Therefore, the self-contrast analysis of WL-DCNN is carried out under different K to determine the influence of K on the proposed model.

TABLE 5. Setting of parameter K.

Dataset	$\langle k \rangle$	K
Yeast	5.3	{3,5,7,9,11,13,15,17,19}
Power	2.4	{5,10,15,20,25,30,35,40,45}
C.elegans	13.5	{7,12,17,22,27,32,37,42,47}
Routers	7	{6,8,10,12,14,16,18,20,22}
NS	3.5	{7,12,17,22,27,32,37,42,47}
PB	20.4	{5,11,21,31,41,51,61,71,77}
Yeast	5.3	{3,5,7,9,11,13,15,17,19}

For the six complex network datasets in the experiment, we set 9 different K centered on $K = \lceil 2\langle k \rangle \rceil$, where $\langle k \rangle$ is the average degree of tested network, and $\lceil \cdot \rceil$ is the ceiling operation, calculating the smallest integer that is greater than the input. When $\langle k \rangle$ keeps in a very small range, K is set to start from $K = 2\langle k \rangle$. The specific settings of K is shown in Table 5 where the underlined numbers denote the central or the starting point of K.

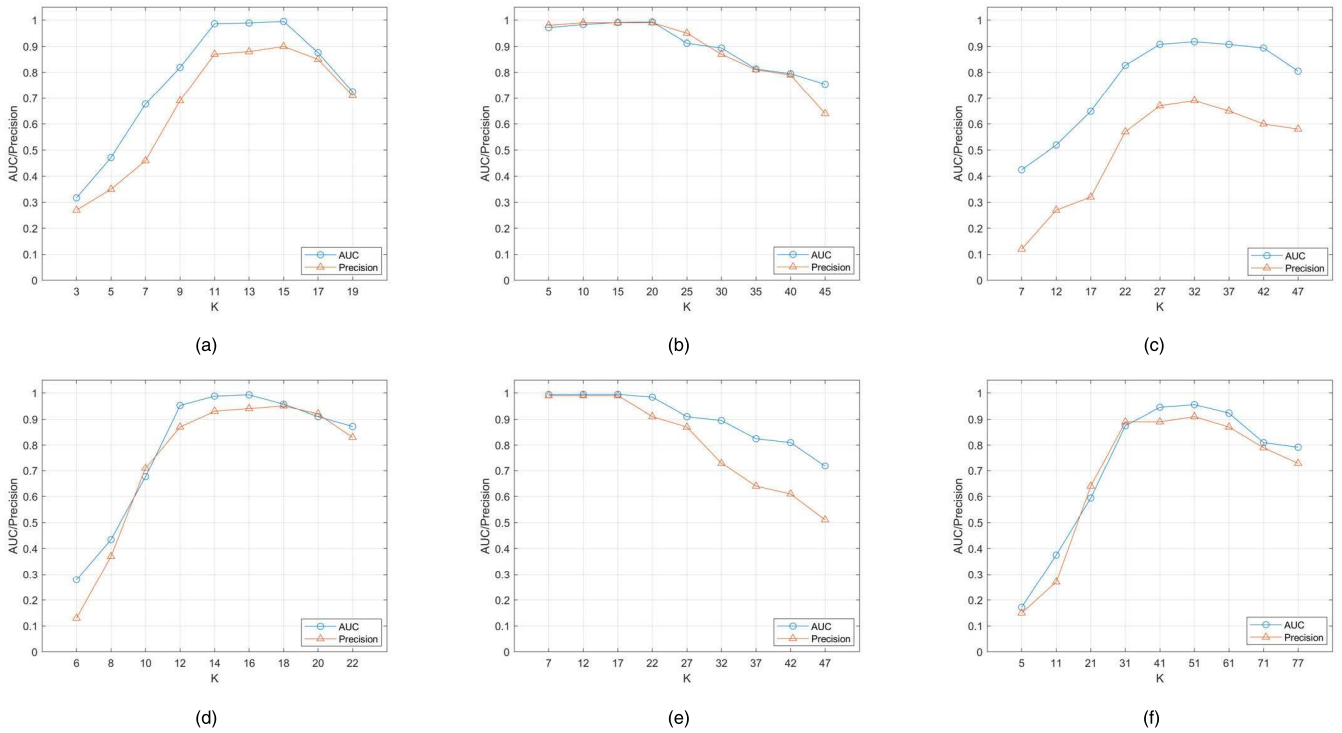


FIGURE 5. Comparison of AUC and Precision of the proposed model under different K; (a) Dataset: Yeast; (b) Dataset: Power; (c) Dataset: C.elegans; (d) Dataset: Routers; (e) Dataset: NS; (f) Dataset: PB.

The experimental results of AUC value and Precision of the proposed model under different K are shown in Fig. 5.

It can be seen from Fig. 5 that under the condition of taking different K, there are similar trends between AUC and Precision on all tested datasets. In other words, with the increasing of K, the link prediction ability of the model is correspondingly enhanced. The above experimental results show that the trade-off for K is $K = 2\langle k \rangle$. Obviously, the increasing of K can make the deep learning model extract more and more comprehensive link features of complex network, so the prediction effect will be enhanced. However, when K is too large, the prediction ability of the model starts becoming worse on all datasets. The reason is that in the process of subgraph extraction, not all graphs are fully-connected graphs. If the number of neighbor nodes of the target link keeps small, or the complex network itself is internally disconnected, our method should supplement virtual and isolated nodes to complete subgraph construction so as to unify the input size for deep learning model. A larger K will promote the number of false-isolated nodes in samples, which leads to redundant and misleading information fed into the subsequent deep learning model. Thus, prediction effect decreases accordingly. Moderate increment of K can make WL-DCNN achieve a better link prediction result, whereas increasing the input dimension in deep learning model will lead to training time being prolonged correspondingly. Furthermore, AUC and Precision result of our model tend to be consistently matched, which demonstrates the overall effect of our model with relatively high accuracy and generality.

TABLE 6. Simulation parameters and environment.

Parameters	Setting
Number of Nodes	100
Simulation Area	100m×100m
ETX	5nJ/bit
ERX	5nJ/bit
Initial Energy of Node	0.5J
Threshold γ, Waiting time τ	0.8, 30ms
Maximum Simulation Time Steps	800
Data generation rate	128 bit/ step
Number of Super Node	1

E. EVALUATION OF RRM-WLDCNN

The routing algorithm is implemented in MATLAB R2019a. The detailed simulation parameter settings and experimental environment of wireless sensor network are shown in Table 6.

In the 100m × 100m simulation area, 100 normal sensor network nodes are randomly deployed. Edge set E is generated according to the geographic location information. 10% of E is used as the WL-DCNN test set E^P. The remaining edges and all nodes are used to generate the training set E^T with incomplete link information, which can be regarded as a sensor network that has been attacked.

The final statistics of the simulated sensor network are listed as follows: $|E|$ is 364, positive samples $|E^T|$ is 328,

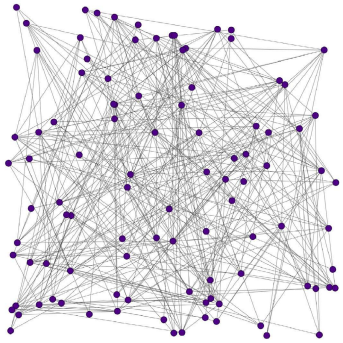


FIGURE 6. Visualization of the sensor network topology in the experiments.

$\left|E^P\right|$ is 36, the network diameter is and the average degree of $|E|$ is 4. Visualization of the simulated sensor network is shown as Fig. 6.

We use WL-DCNN model to extract, label and train the subgraph of the divided network $\left|E^T\right|$. The parameter settings of the deep learning network model, i.e., DCNN, are the same as shown in Section III-A, except that we need to set $K = 32$. Finally, the input samples constructed by the unknown edge set $U - E^T$ are fed into DCNN, and the numerical reliability of links between each node pairs is obtained by forward propagation, where U is the universal edge set of links in sensor network. It is not necessary to use link prediction model to calculate link reliability if the link already exists in the training set. The corresponding reliability is 1 for positive samples and 0 for negative ones. Accordingly, a complete link reliability matrix P for arbitrary node pairs is constructed.

In order to minimize random errors, the experimental results composed of the 4 indicators (the transmission delay, the system residual energy state, the average routing length, and the node failure curve) are all recorded as the average value of 100 independent and repeated experiments under the same sensor network topology, noting that DCNN only needs to be trained once. In the experiment, we choose Failure-Resilient Routing (FRR) [7], Secure and QoS Aware Routing (SQR) [13], Flooding [3], and AODV [4] as the baseline methods, then we simulate the scenario of network attacks by disturbing the broadcasting process of nodes during routing construction process. Specially, in order to make comparisons on the performance assessment between our model and the tested baseline methods, some modification for the experiment setting are needed. For SQR, which is also originated from AODV, the algorithm may provide QoS routing and alleviate the impact brought by the misbehaving nodes. Hence, we will use the sum of reliability score calculated by the DCNN model for each node and ETT as a substitute for the original metric Composite Routing Metric (CRM) presented by A. Ahmed *et al*. The trust value of SQR is removed because there are no malicious nodes in our experiment. As for the experiment setting of FRR mechanism, there is a natural transformation of application scenarios between IoT networks and sensor networks, and we set T=5 for FRR mechanism.

Fig. 7(a-d) illustrates the comparison on the transmission delay, the energy consumption, the average path length of transmission, and the state of system node failures between RRM-WLDCNN (adopting link reliability prediction model) and the tested routing baseline algorithms, i.e., FRR, SQR, Flooding, and AODV. Fig. 7 reflects the influence of the attack on the network, which is deployed to the network at the 400^{th} time epoch in each experiment.

In Fig.7 (a), compared with FRR, SQR and AODV, our proposed method shows the minimum transmission delay except for Flooding before the arrival of the attack, which can be ascribed to the semi-static nature of our method with the super node calculating routing entries and for the whole network and providing reliability matrix P for each node at fixed intervals. Once the attack started, the proposed RRM-WLDCNN experienced the smallest change in the transmission delay because our model exploits the link prediction model introduced into RRM-WLDCNN to construct optimal transmission routes based on the cost function, thus mitigating the impact of the incomplete link state caused by the attack. Since AODV is a passive route, broadcast is required to establish the new route for data transmission. If the link information is incomplete, the resulting path of AODV will be lengthened, which will lead to a significant increase in network transmission delay and node energy consumption. In fact, it even gets worse later due to the failures of nodes demonstrated in Fig. 7 (a). The transmission delay results of SQR shows that it has a higher delay than other methods whether or not the attack is deployed, and a plausible explanation is that SQR will construct the routing dynamically via calculating the metric of each path which requires time resource. Besides, it illustrates that SQR is vulnerable to this sort of deliberate attack. FRR perform moderately because of its mechanism of failure reaction which can update alternate route efficiently. Flooding method gets the minimum delay before the attack at the cost of huge energy consumption, and the robustness of Flooding is quite excellent in the earlier stage of our experiment depicted in Fig. 7(a). However, the transmission delay dramatically increased over time because of large amounts of node failures and the influence of the attack (simulated by special link pruning operations).

Fig.7(b) and Fig.7(d) jointly demonstrate the high energy efficiency of the proposed RRM-WLDCNN, which outperforms the other algorithms. It explains that our link-reliability-prediction based model can found a shortcut to the destination in the transmission task, efficiently reduce node energy consumption, and alleviate the influence caused by incomplete link information. The proportion of dead nodes in the network increases exponentially and the system residual energy decreases dramatically when adopting Flooding routing method. Besides, the attack also leads to increased node failures for AODV, SQR, and FRR, because the attack is aimed deliberately at the broadcast process. There is a similar trend towards the increase of energy consumption of the

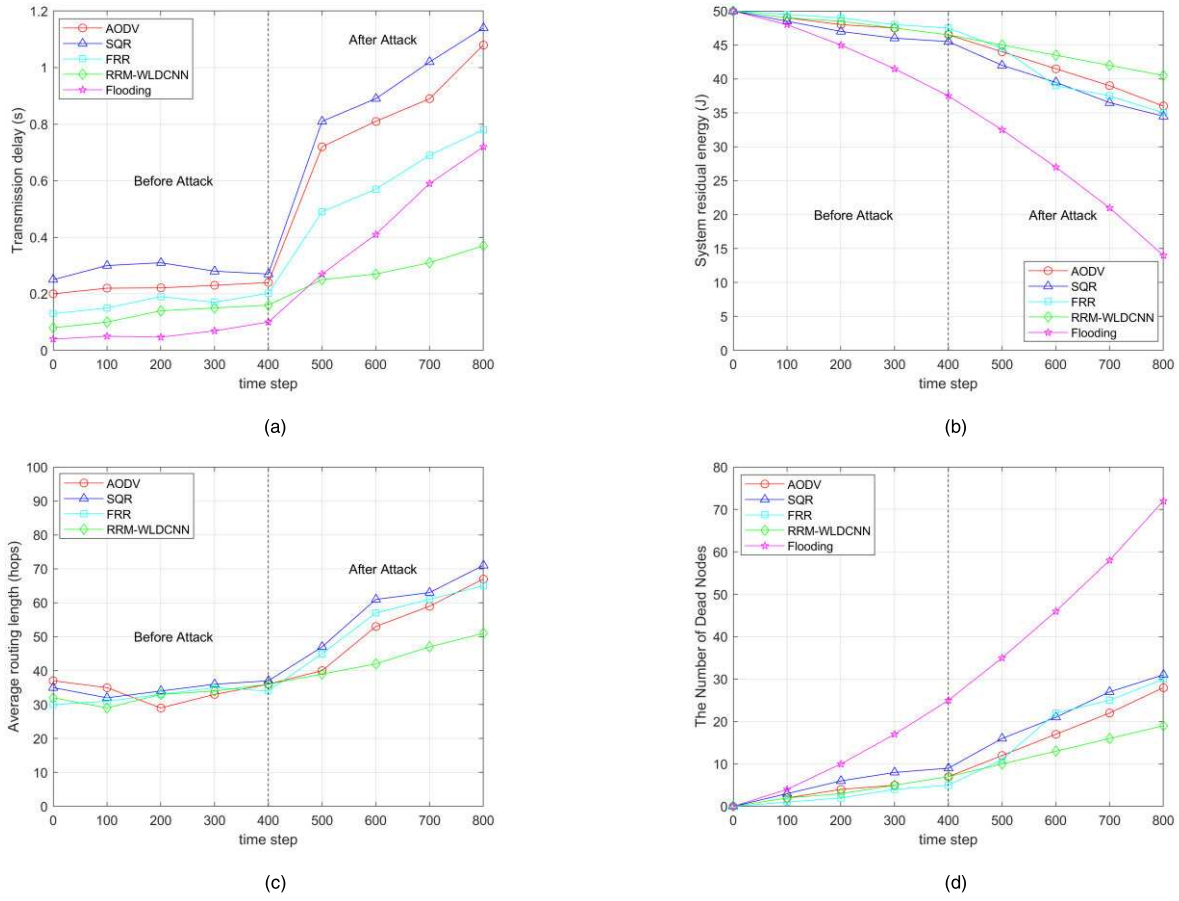


FIGURE 7. Comparisons between the baseline routing mechanisms and our proposed RRM-WLDCNN method. Black dotted line indicates the time when we attack the simulated network; (a) Average System Transmission Delay of the 5 methods, (b) System residual energy of the simulation network. (c) Average routing length of the 4 methods. (d) Node failures curve over time of the 5 routing methods.

5 routing methods which is resulted from the combination of the node failures and the attacks. However, our model shows its excellent resilience to the attacks with slight fluctuation in energy consumption among the tested algorithms.

Fig.7(c) shows the average routing length of the 4 mechanisms. The routing length result of Flooding is not recorded and shown because the routing length will be always equal to the current number of edges in the network when Flooding is adopted as the routing strategy. Before the attack starts, there is little difference on the routing length among the 4 routing mechanisms, whereas a significant differentiation can be found after the attack. The average path length increases for all the tested algorithms due to the node failures and the malicious attack. However, the attack has the least impact on the simulated network when RRM-WLDCNN is used as the routing method that can make full use of the network topology information and alleviate the impact of the attack. Furthermore, this attack operation intends to lead the network to constructing longer routes for nodes, which is caused by the incomplete link information. Hence, FRR, AODV, SQR perform relatively poor under this circumstance.

Through the experiment, it is validated that RRM-WLDCNN is effective and feasible to integrate link prediction into routing mechanism under the situation when the network is under attack with links being damaged, thus enhancing the survivability of sensor networks.

IV. CONCLUSION

In this paper, we investigated the problem of resilient routing under dynamic network topology and network attacks. We proposed a novel design which exploit link reliability to improve network and routing resilience. We transformed the link prediction problem into a trainable regression problem with evolving and high-dimensional topology information being processed simultaneously, which can be effectively handled by the proposed WL-DCNN model. We developed an efficient and effective approach of subgraph extraction and graph labelling to train a novel CNN with two branches. Specifically, WL-DCNN extracts and labels the subgraph generated by target links, via introducing the similarity matrix and the Laplacian spectrum to construct the main input and the auxiliary one for the deep learning model.

Excellent results are obtained over six real complex network datasets. Compared with 9 link prediction baseline algorithms, the link prediction performance of the proposed model is improved by 16% in average, and exceeds 40% over the Routers dataset. The results demonstrated that the model is feasible and effective in the link reliability prediction of undirected-graphs based sensor networks. Meanwhile, we made the feasibility proof of integrating link prediction into resilience mechanism, which can achieve lower transmission cost in the case of incomplete sensor network link information. Experiment shows that the RRM-WLDCNN rooting algorithm achieves good performance with the life cycle of the sensor network extended significantly, thus enhancing the network's survivability and resilience.

In the future we are interested in applying the proposed RRM-WLDCNN to improve the resilience of sensor network with more flexibility, adjusting and extending the model to more general complex network graphs, such as weighted, directed, dynamic, and heterogeneous graphs. We will continue to optimize the performance of the RRM-WLDCNN routing algorithm with anti-interference compatibility and real-time processing ability for green routing in sensor networks.

All data sets used in this paper, example python codes of our proposed WL-DCNN model and the sensor network topology information are available at [45].

ACKNOWLEDGMENT

The authors gratefully acknowledge Prof. Y. H. Hu of Wisconsin-Madison University for useful discussions and consultations, and Prof. J. Zhang of Sheffield University who performed the laboratory and data analysis for their proposed algorithm model. They would also like to thank the anonymous reviewers for their valuable comments and suggestions that should help improve the quality of this manuscript.

REFERENCES

[1] D. P. Agrawal, "Applications of sensor networks," in *Embedded Sensor Systems*. Singapore: Springer, 2017, pp. 35–63.

[2] N. C. V. N. Pereira and R. M. de Moraes, "Comparative analysis of AODV route recovery mechanisms in wireless ad hoc networks," *IEEE Latin Amer. Trans.*, vol. 8, no. 4, pp. 385–393, Aug. 2010.

[3] N. S. M. Usop, A. Abdullah, and A. F. A. Abidin, "Performance evaluation of AODV, DSDV & DSR routing protocol in grid environment," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 9, no. 7, pp. 261–268, 2009.

[4] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, pp. 1023–1032, Oct. 2001.

[5] C. Wang and S. Wang, "Research on uneven clustering APTEEN in CWSN based on ant colony algorithm," *IEEE Access*, vol. 7, pp. 163654–163664, 2019.

[6] K. Fan, J. Lu, D. Sun, Y. Jin, R. Shen, and B. Sheng, "Failure resilient routing via IoT networks," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Exeter, U.K., Jun. 2017, pp. 845–850.

[7] S. Kumari, P. K. Mishra, and V. Anand, "Fault resilient routing based on moth flame optimization scheme for underwater wireless sensor networks," *Wireless Netw.*, vol. 26, no. 2, pp. 1417–1431, Feb. 2020.

[8] P. Murali Mohan, M. Gurusamy, and T. J. Lim, "Dynamic attack-resilient routing in software defined networks," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 3, pp. 1146–1160, Sep. 2018.

[9] L. Veryard, H. Hagras, A. Starkey, and G. Owusu, "A fuzzy genetic system for resilient routing in uncertain & dynamic telecommunication networks," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, New Orleans, LA, USA, Jun. 2019, pp. 1–6.

[10] K. Haseeb, N. Islam, A. Almogren, and I. Ud Din, "Intrusion prevention framework for secure routing in WSN-based mobile Internet of Things," *IEEE Access*, vol. 7, pp. 185496–185505, 2019.

[11] B. H. Al-Qarni, A. Almogren, and M. M. Hassan, "An efficient networking protocol for Internet of Things to handle multimedia big data," *Multimedia Tools Appl.*, vol. 78, no. 21, pp. 30039–30056, Nov. 2019.

[12] F. N. Nur, S. Sharmin, M. A. Habib, M. A. Razzaque, and M. S. Islam, "COllaborative neighbor discovery in directional wireless sensor networks," in *Proc. IEEE Region Conf. (TENCON)*, Nov. 2016, pp. 1097–1100.

[13] A. Ahmed, P. Kumar, A. R. Bhangwar, and M. I. Channa, "A secure and QoS aware routing protocol for wireless sensor network," in *Proc. 11th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, Barcelona, Spain, Dec. 2016, pp. 313–317.

[14] A. Razaque, M. Abdulgader, C. Joshi, F. Amsaad, and M. Chauhan, "P-LEACH: Energy efficient routing protocol for Wireless Sensor Networks," in *Proc. IEEE Long Island Syst., Appl. Technol. Conf. (LISAT)*, Farmingdale, NY, USA, Apr. 2016, pp. 1–5.

[15] J. Yan, M. Zhou, and Z. Ding, "Recent advances in energy-efficient routing protocols for wireless sensor networks: A review," *IEEE Access*, vol. 4, pp. 5673–5686, 2016.

[16] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H.-J. Kim, "Energy efficient routing algorithm with mobile sink support for wireless sensor networks," *Sensors*, vol. 19, no. 7, p. 1494, 2019.

[17] M. Selvi, R. Logambigai, S. Ganapathy, L. S. Ramesh, H. K. Nehemiah, and K. Arputharaj, "Fuzzy Temporal Approach for Energy Efficient Routing in WSN," presented at the Proc. Int. Conf. Inform. Anal., Pondicherry, India, 2016, doi: 10.1145/2980258.2982109.

[18] V. V. Mandhare, V. R. Thool, and R. R. Manthalkar, "QoS routing enhancement using Metaheuristic approach in mobile ad-hoc network," *Comput. Netw.*, vol. 110, pp. 180–191, Dec. 2016.

[19] X. Deng, Q. Peng, L. He, and T. He, "Interference-aware QoS routing for neighbourhood area network in smart grid," *IET Commun.*, vol. 11, no. 5, pp. 756–764, Mar. 2017.

[20] Z. Li, X. Fang, and O. R. L. Sheng, "A survey of link recommendation for social networks: Methods, theoretical foundations, and future research directions," *ACM Trans. Manage. Inf. Syst. (TMIS)*, vol. 9, no. 1, pp. 1–26, 2017.

[21] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proc. IEEE*, vol. 104, no. 1, pp. 11–33, Jan. 2016.

[22] H. Hu, C. Zhu, H. Ai, L. Zhang, J. Zhao, Q. Zhao, and H. Liu, "LPI-ETSLP: LncRNA–protein interaction prediction using eigenvalue transformation-based semi-supervised link prediction," *Mol. BioSyst.*, vol. 13, no. 9, pp. 1781–1787, 2017.

[23] D. K. Duvenaud, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2224–2232.

[24] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.

[25] C. Fu, M. Zhao, L. Fan, X. Chen, J. Chen, Z. Wu, Y. Xia, and Q. Xuan, "Link weight prediction using supervised learning methods and its application to yelp layered network," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1507–1518, Aug. 2018.

[26] D. M. Dunlavy, T. G. Kolda, and E. Acar, "Temporal link prediction using matrix and tensor factorizations," *ACM Trans. Knowl. Discovery from Data*, vol. 5, no. 2, pp. 1–27, Feb. 2011.

[27] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 390, no. 6, pp. 1150–1170, Mar. 2011.

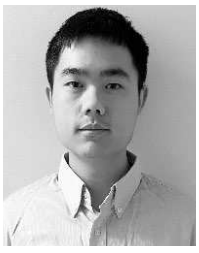
[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[29] H. Shi, Y. Zhang, Z. Zhang, N. Ma, X. Zhao, Y. Gao, and J. Sun, "Hypergraph-induced convolutional networks for visual classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 2963–2972, Oct. 2019.

[30] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Adv. neural Inf. Process. Syst.*, vol. 2016, pp. 3844–3852.

[31] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell, "Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching," in *Proc. CVPR*, Jun. 2011, pp. 2705–2712.
[32] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018.
[33] M. Zhang and Y. Chen, "Weisfeiler-lehman neural machine for link prediction," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, 2017, pp. 575–583.
[34] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels," *J. Mach. Learn. Res.*, vol. 12, pp. 2539–2561, Sep. 2011.
[35] F. Y. Zhou, L.-P. Jin, and J. Dong, "Review of convolutional neural network," *Chin. J. Comput.*, vol. 40, no. 6, pp. 1229–1251, 2017.
[36] X. Yang, C. Liu, Z. Wang, J. Yang, H. L. Min, L. Wang, and K.-T. T. Cheng, "Co-trained convolutional neural networks for automated detection of prostate cancer in multi-parametric MRI," *Med. Image Anal.*, vol. 42, pp. 212–227, Dec. 2017.
[37] P. Yang, P. D. Yoo, J. Fernando, B. B. Zhou, Z. Zhang, and A. Y. Zomaya, "Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications," *IEEE Trans. Cybern.*, vol. 44, no. 3, pp. 445–455, Mar. 2014.
[38] F. Zou, L. Shen, Z. Jie, W. Zhang, and W. Liu, "A sufficient condition for convergences of adam and RMSProp," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11127–11135.
[39] J.-N. Hwang and Y. H. Hu, *Handbook of Neural Network Signal Processing*. Boca Raton, FL, USA: CRC Press, 2001.
[40] K. C. Das, "The Laplacian spectrum of a graph," *Comput. Math. Appl.*, vol. 48, nos. 5–6, pp. 715–724, Sep. 2004.
[41] O. Mülken, A. Volta, and A. Blumen, "Asymmetries in symmetric quantum walks on two-dimensional networks," *Phys. Rev. A, Gen. Phys.*, vol. 72, no. 4, Oct. 2005, Art. no. 042334.
[42] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve.," *Radiology*, vol. 143, no. 1, pp. 29–36, Apr. 1982.
[43] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst. (TOIS)*, vol. 22, no. 1, pp. 5–53, Jan. 2004.
[44] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, "Bipartite network projection and personal recommendation," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 4, Oct. 2007, Art. no. 046115.
[45] *GitHub*. Accessed: Mar. 29, 2020. [Online]. Available: https://github.com/malei666666/WL-DCNN_link_based/

RU HUANG received the B.S. degree from Nanjing University, Nanjing, China, in 1999, and the Ph.D. degree in circuit and system from Shanghai Jiao Tong University, Shanghai, China, in 2008. He was a Visiting Scholar with the University of Wisconsin-Madison, WI, USA, from March 2015 to March 2016. He is currently an Associate Professor of electronics and communication engineering with the East China University of Science and Technology, Shanghai. His current research interests include wireless sensor networks, complex networks, and software-defined networks.



LEI MA received the B.S. degree in information engineering from the University of East China University of Science and Technology, in 2018. He is currently pursuing the M.S. degree in information and communication engineering with the East China University of Science and Technology. His research directions comprises complex networks, deep learning, and data mining, specializing at building deep learning model for data mining tasks in complex networks.

GUANGTAO ZHAI (Senior Member, IEEE) received the B.S. degree from Shandong University, China, in 2001, and the Ph.D. degree from Shanghai Jiao Tong University, China, in 2009. He is currently a Professor with the Department of Electronics Engineering, Shanghai Jiao Tong University. He has published over 280 international journal and conference papers in topics of multimedia signal processing and perceptual signal processing with Google Scholar citation number of 4800. His research interests are in the fields of multimedia and perceptual signal processing. He is a member of IEEE CAS VSPC TC and MSA TC and is on the editorial board of *Digital Signal Processing* (Elsevier), IEEE Access, and *Science China Information Science* (Springer).

JIANHUA HE (Senior Member, IEEE) received the Ph.D. degree from Nanyang Technological University, Singapore, in 2002. He is currently a Reader with the University of Essex, U.K. He has authored or coauthored over 100 technical articles in major international journals and conferences. His research interests include 5G and beyond technologies, ADAS, the Internet of Things, machine learning, and big data analytics. He is a TPC member of many international conferences, including IEEE Globecom and IEEE ICC. He acted as the TPC Chair for international conference ICAIT'2010 and ICONI'2010. He is an Editor or a Guest Editor of several international journals, including *Wireless Communications and Mobile Computing*, IEEE Access, *International Journals of Communication Systems*, *International Journal of Distributed Sensor Networks*, and *KSII Transactions on Internet and Information Systems*.

XIAOLI CHU (Senior Member, IEEE) received the B.Eng. degree in electronic and information engineering from Xi'an Jiao Tong University, in 2001, and the Ph.D. degree in electrical and electronic engineering from the Hong Kong University of Science and Technology, in 2005. From September 2005 to April 2012, she was with the Centre for Telecommunications Research, King's College London. Her research interest includes modeling, analysis, and algorithm design for improving the performance and efficiency of wireless communication systems.

HUAICHENG YAN (Member, IEEE) received the B.S. degree in automatic control from the Wuhan University of Technology, China, in 2001, and the Ph.D. degree in control theory and control engineering from the Huazhong University of Science and Technology, China, in 2007. From 2007 to 2009, he was a Postdoctoral Fellow with The Chinese University of Hong Kong. In 2011, he was a Research Fellow with The University of Hong Kong. In 2012, he was a Research Fellow with the City University of Hong Kong. He is currently a Professor with the School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China. His research interests include networked control systems, multiagent systems, smart grids, and robotics.