Rolling Bearing Fault Diagnosis Based on Horizontal Visibility Graph and Graph Neural Networks

Chenyang Li
School of Instrument Science and Engineering
Southeast University
Nanjing, China
licy@seu.edu.cn

Lingfei Mo*
School of Instrument Science and Engineering
Southeast University
Nanjing, China
lfmo@seu.edu.cn

Ruqiang Yan*
School of Instrument Science and Engineering
Southeast University
Nanjing, China
ruqiang@seu.edu.cn

***Abstract*—The automatic extraction and learning features relying on artificial intelligence algorithms replace traditional manual features. More effective feature expression improves the performance of machine fault diagnosis with fewer requirements for labor and expertise. However, the present models only can process the data in Euclidean space. The relations between data points are ignored for a long time, which can play a significant role in distinguishing diverse faults patterns. To combat this issue, a novel model for bearing faults diagnosis is proposed by incorporating the horizontal visibility graph (HVG) and graph neural networks (GNN). In the proposed model, time series is converted to graph retaining invariant dynamic characteristics through the HVG algorithm, and the generated graphs are fed into a designed GNN model for feature learning and faults classification further. Finally, the proposed model is tested on two actual bearing datasets, and it shows state-of-the-art performance in the bearing faults diagnosis. The experimental results demonstrate that extracting relation information using HVG benefits bearing faults diagnosis.**

***Keywords*— rolling bearing, fault diagnosis, horizontal visibility graph (HVG), graph neural networks (GNN)**

I. Introduction

With mechanical equipment becoming larger, high-speed, more precise, and high automation, the impact and harm caused by equipment failure is getting more and more serious, such as equipment damage, system collapse, and production stagnation. Therefore, the health management of mechanical equipment in intelligent manufacturing has become one of the key problems to be solved urgently in the system intelligence.

Traditional mechanical fault signal processing methods mainly focus on the time domain, frequency domain, and time-frequency domain, which can combine with machine learning methods such as multi-layer perceptron (MLP), support vector machine (SVM), and Bayesian estimation to diagnose the data with obvious features and simple patterns. With the rapid development of artificial intelligence, intelligent algorithms represented by deep learning have gradually emerged in the area of fault diagnosis [1], for instance, convolutional neural networks (CNN) [2], [3], automatic encoders (AE) [4], generative adversarial networks (GAN) [5], etc. Current feature extraction algorithms are aimed at the data value itself, but the ability to extract relations and structure between the data is limited.

In recent years, there is a growing interest in analyzing time series using complex networks in some specific domains. The common methods for transforming time series into graph representation are recurrence networks, transition networks, and visibility graphs [6] in terms of different principles. The application of the graph model in fault diagnosis has just getting started. Gao et al. divided the non-stationary periodic signal into a single period and utilized the graph structure to detect the change of mechanical equipment's working states [7]. Gao et al. transformed the original vibration signal to graph domain, and used graph Fourier transform to extract the graph spectrum coefficients for fault diagnosis [8]. Furthermore, the total variation which reflects the smoothness of graph signals served as the indicator to distinguish different rolling bearing faults signals [9].

Unlike the data in Euclidean space, the data in non-Euclidean space contain the complex structure and abundant relation information, such as the physical system, molecular fingerprint, social networks, recommendation systems, etc. The relational data pose a great challenge for previous machine learning and deep learning algorithms. Graph neural networks (GNN) emerging as a specialized graph signal processing algorithm have gained extensive attention from researchers [10]. GNN can achieve more effective data relationship extraction and inference by aggregating the information of the node's neighbors at any depth. At present, the main GNN models can be classified into four categories: recurrent GNN, convolutional GNN, graph auto-encoders, and spatial-temporal GNN [11]. GNN has been successfully applied in the field of physical models [12], chemical structure [13], social networks [14], natural language processing [15], image classification [16].

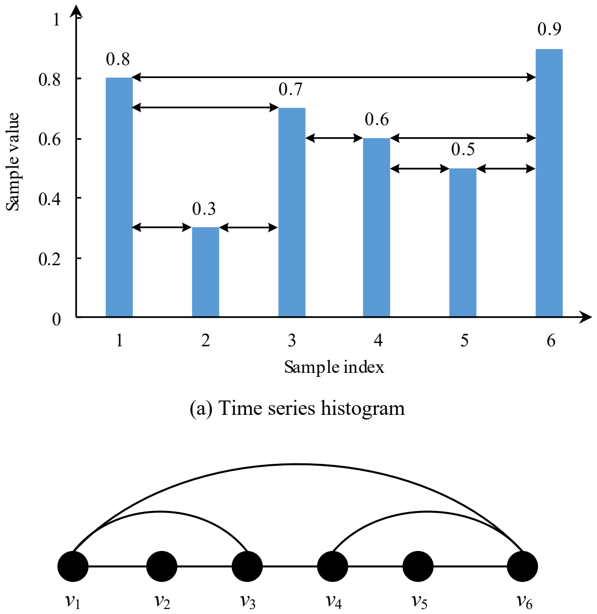
As a deep learning method based on graph domain, GNN has convincing performance and high interpretability in graph data learning, but there is barely any research about the application of GNN in intelligent diagnosis of mechanical equipment. In this paper, we devote to explore the feasibility of GNN in rolling bearing fault diagnosis. By the HVG algorithm, the original vibration data are converted to a whole graph composed of the numerical value and connections between the data. A GNN model is designed for the graph classification referring to the state-of-the-art model GIN. Experiments on two bearing datasets are conducted to verify the effectiveness of GNN in fault diagnosis. Besides, the comparison with MLP, recurrent neural networks (RNN), and

*Ruqiang Yan is the corresponding author. (e-mail: ruqiang@seu.edu.cn)

*Lingfei Mo is the co-corresponding author. (e-mail: lfmo@seu.edu.cn).

This work was supported by the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX20_0087.

978-1-7281-9277-2/20/$31.00 ©2020 IEEE



(a) Time series histogram

(b) HVG

Fig. 1. Illustration of HVG algorithm

its variants are given to demonstrate the improvement causing by the introduction of data relations.

II. Theoretical Background

The HVG algorithm proposed by Luque et al. is based on the visibility graph [17], [18]. The HVG algorithm maps the time series into graph in non-Euclidean space based on the horizontal relationship between data points. A graph G can be denoted as $G = (V, E)$, where $V=\{v_1, v_2, \dots, v_N\}$ is the set of N nodes, $E=\{e_{ij}| v_i$ and v_j are linked$\}$ is the edge set. Suppose a time series has N data points with values $X=\{x_1, x_2, \dots, x_N\}$, and the nodes are the discrete data points with the feature of corresponding data value. According to the horizontal visibility criterion, two nodes are considered linked only if a horizontal line connects x_i and x_j without intersecting by any intermediate data height, i.e., the following geometrical relationship is satisfied

$$x_i, x_j > x_n, \quad \text{all } n \in (i, j). \tag{1}$$

An illustration of the algorithm is given in Fig. 1. The top shows the histogram of a sampled time series {0.8, 0.3, 0.7, 0.6, 0.5, 0.9}. The edge exists when the bar heights of the two nodes are higher than any other data heights between them. By connecting the bars and the horizontal lines as the nodes and edges, the HVG can be generated as shown in Fig. 1(b). Naturally, the HVG possesses connectivity, invariance under affine transformations, and other characteristics. Graph neural networks is a feasible approach to deal with the HVG.

GNN, as a rising neural network inspired by Graph Embedding and CNN, shows a powerful ability to learn graph in the non-Euclidean space. Passing messages along edges can obtain representation and extract features of the nodes and graph [19]. A graph $G = (V, E)$ with N nodes has a node feature matrix $X \in \mathbb{R}^{N \times F}$, where F is the dimension of the node feature vector. GNNs embed the node feature representing as a hidden state h_v along the graph topology. The forward pass of current GNNs models follows the aggregation and readout strategy. In the first phase, the node features are updated by aggregating the information of its neighbors, the k-th propagation is formulated as [19], [20]:

$$m_v^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}(v)\right\}\right) \tag{2}$$

$$h_v^{(k)} = \text{COMBINE}^{(k)}\left(h_v^{(k-1)},\ m_v^{(k)}\right) \tag{3}$$

where $m_v^{(k)}$ is the passing message generated by adjacent nodes, $\mathcal{N}(v)$ is the neighbor set of node v, $h_v^{(k)}$ is the output feature of node v in the k-th layer, and $h_v^{(0)}$ is initialized with x_v. Through the k iterations of aggregation, the node feature can fuse the information within its k-hop neighbors, i.e., the receptive filed reaches to k-hop neighbors. For node-level tasks, the node label y_v can be inferred by the final node representation: $y_v = f(h_v)$. For the graph-level tasks, a graph representation h_G is further required, which will be computed in the second phase through a READOUT function:

$$h_G = \text{READOUT}\left(\left\{h_v^{(k)} \mid v \in G\right\}\right). \tag{4}$$

The choices of the AGGREGATE function, COMBINE function, and READOUT function have a great effect on the learning ability of GNNs. Apart from differentiable characteristic, the READOUT function also needs to be invariant to node permutations. Most representative GNNs models can be reformulated to the aggregation framework. The propagation rule of Graph Convolutional Networks (GCN) [21] utilizes the first-order approximation of spectral graph convolution:

$$h_v^{(k)} = \text{ReLU}\left(W \cdot \text{MEAN}\left\{h_u^{(k-1)}, \forall u \in N(v) \cup \{v\}\right\}\right) \tag{5}$$

where W is a learnable matrix, MEAN denotes the mean aggregator. GraphSAGE, an inductive framework, analyzed the Mean aggregator, LSTM aggregator, and Pooling aggregator [14]. The Pooling aggregator had the best performance in experiments, where a max-pooling operator served as the aggregator:

$$m_v^{(k)} = \text{MAX}\left(\left\{\text{ReLU}\left(W \cdot h_u^{(k-1)}\right), \forall u \in \mathcal{N}(v)\right\}\right). \tag{6}$$

Motivated by the Weisfeiler-Lehman (WL) graph isomorphism test, Xu et al. point out that the upper bound of the GNNs based on nodes aggregation is the WL test if the aggregator and graph-level readout functions are injective [20]. Based on this theorem, a simple but effective architecture, Graph Isomorphism Network (GIN) is developed to capture dependencies of isomorphic graphs. Considering the universal approximation of MLP, MLP is used to model the aggregation and combine functions:

$$h_v^{(k)} = \text{MLP}^{(k)}\left(\left(1+\varepsilon^{(k)}\right) \cdot h_v^{(k-1)} + \sum_{u \in N(v)} h_u^{(k-1)}\right) \tag{7}$$

where ε is a learnable or fixed parameter. Because the sum operator is injective, it is selected for aggregator instead of max or mean. GIN achieves state-of-the-art performance on several graph classification benchmarks.

III. Model Architecture

Normally, the working condition is monitored by sensors, thus, the potential fault information lies in the sampled time series. Through the HVG algorithm, we can transform a time series sample into a graph with condition-specific topology,

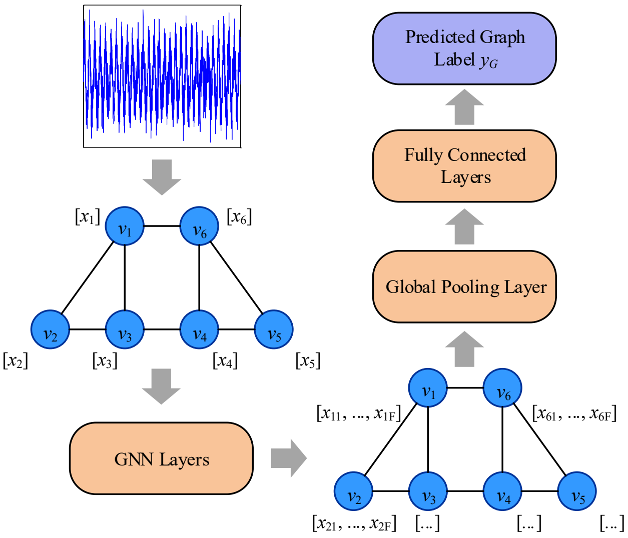


Fig. 2. The proposed GNNs architecture for fault diagnosis

and the sampling point value is treated as the corresponding node's feature. In this way, the fault diagnosis is converted to a graph-level classification task. To obtain the graph feature, we design a deep GNNs architecture based on the GIN model, which consists of three main parts: GNN layers, a global pooling layer, and two fully connected (FC) layers. The illustration of the GNNs architecture is shown in Fig. 2.

The GIN aggregator has been adjusted to better adapt to bearing faults data, but still employs injective functions. The MLP is replaced by 1-layer perceptron, i.e., a linear mapping function. While the representation ability of the 1-layer perceptrons is considered insufficient, it brings the equivalent performance as well as a stable training process on our bearing datasets. The sum aggregator is chosen the same as GIN and ε is fixed to 0. Thus, the aggregation of the GNN layer is defined as follows:

$$h_v^{(k)} = \text{Linear}^{(k)}\left(h_v^{(k-1)} + \sum_{u \in N(v)} h_u^{(k-1)}\right). \tag{8}$$

After passing the nonlinear activation function ReLU and a batch normalization layer, the aggregated graph is input into the next aggregation layer. By stacking multiple aggregation layers, the node feature can merge with the graph structure. Then, to get the entire graph representation r_G, the node features go to a global sum pooling layer:

$$r_G = \sum_{i=1}^{N} h_{v_i}. \tag{9}$$

After adding node features across the node dimension, one can get the entire graph feature to infer the fault type. The graph label is finally inferred by two FC layers with softmax function.

The negative log-likelihood (NLL) loss function calculates the negative log value of probability predicted by the softmax layer:

$$L(y_G) = -\log(y_G) \tag{10}$$

where y_G is the predicted label.

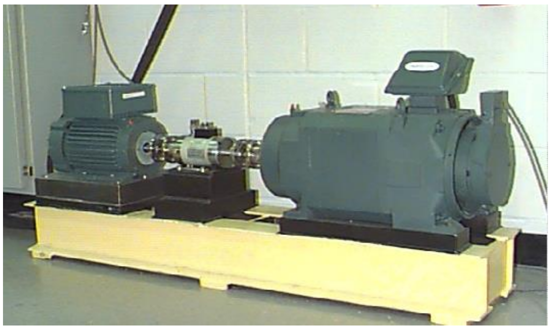


Fig. 3. CWRU bearing test rig [22]

TABLE I. CRWU BEARING FAULT TYPES DESCRIPTION

Types	Description	Sample size
Normal	Healthy bearing	100
BE	Ball element faults with 3 fault diameters	100*3
IR	Inner race faults with 3 fault diameters	100*3
OR	Outer race faults with 3 fault diameters	100*3

IV. EXPERIMENTS

To validate the effectiveness of the proposed method, we perform the experiments on two actual bearing datasets: Case Western Reserve University (CWRU) bearing dataset and Southeast University (SEU) bearing dataset. The comparative experiment with MLP is performed to highlight the influence of the latent structure of the time series. Moreover, we also analyze the feature extraction ability of different numbers of aggregation layers.

A. Datasets

1) CWRU bearing dataset

CWRU bearing dataset is a public dataset for bearing faults diagnosis, which has been tested a lot in previous research [22]. The bearing experiments were carried out on a 2 hp electric motor, and the vibration data were collected using accelerometers deployed on both the drive end and fan end of the motor housing as shown in Fig. 3. The faults were set manually by electro-discharge machining with different damage intensity. In our experiments, data in load 2 hp with three fault diameters: 0.1778 mm (0.007 inches), 0.3556 mm (0.014 inches), 0.5334 mm (0.021inches) are selected, and each fault diameter corresponds to three fault types. Thus, it becomes a ten-class classification task, and each class contains 100 samples. 1024 data points are partitioned to one sample with the sampling rate of 12000 Hz. The data adopted in the following experiments are listed in TABLE I.

2) SEU bearing dataset

The SEU bearing fault simulation data are collected on Drivetrain Dynamics Simulator (DDS). As depicted in Fig. 4, DDS consists of a motor, a planetary gearbox, a parallel gearbox, a brake, and controller modules, which can simulate different kinds of faults of gearboxes and bearings. The acceleration sensors are mounted on the drive motor, planetary gearbox, and parallel gearbox, respectively. Setting the rotating speed and load to 30 Hz and 2 V, the data of each bearing fault type are acquired at a sampling rate of 5120 Hz.

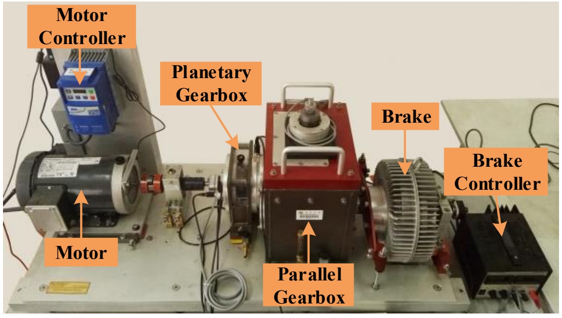


Fig. 4. SEU gearbox test rig

TABLE II. SEU BEARING FAULT TYPES DESCRIPTION

Types	Description	Sample size
Normal	Healthy bearing	800
Ball	Ball element fault	800
Inner	Inner ring fault	800
Outer	Outer ring fault	800
Combination	Inner and outer ring faults	800

The same as CRWU datasets, a sample contains 1024 sampling points, and four types of failures and one health condition constitute the five-class classification task. The detailed information of the fault types is listed in TABLE II.

B. Experiment Configuration

In the experiments, at most 4 aggregation layers are stacked, followed by one global sum pooling layer and 2 FC layers for feature mapping and probability calculation. To analyze the improvement brought by the graph structure, we use MLP with equal layers as the baseline. The layer type and numbers are used to identify the neural networks architecture, such as 3GNN-2FC denoting 3 GNN layers and 2 FC layers.

Batch normalization is incorporated after every hidden layer. The hyperparameters are tuned through repeated experiments. The hidden unit size in the GNN model is determined to be 16, while MLP is 128. Adam optimizer with learning rate 0.005 and batch size 32 remain unchanged in both two datasets. The data are split for training, validation, and test with the percentages of 0.8, 0.1, 0.1. The training process stops when the validation accuracy does not improve for 10 epochs. For an accurate comparison, the final results average the accuracies of 10 repeated experiments with random data splitting. The model is implemented using Pytorch Geometric libraries [23].

C. Results

Through transforming time series to graph using the HVG algorithm, the topology relation among the data can be introduced into the data. To investigated whether the HVG has extracted extra useful information, the proposed GNN designed for learning graph data is compared with the MLP model where only the samples' numerical values are input. The classification accuracies of the designed model and MLP are shown in Fig. 5 and Fig. 6 (CRWU and SEU bearing dataset). Intuitively, the proposed GNN model outperforms the MLP with equal layers. Stacking more MLP layers, there is no obvious increase in classification accuracy. However, the aggregator can capture longer dependency of nodes and obtain more helpful features using deeper GNN layers. The GNN model has a satisfying performance even only containing a single aggregation layer. The fault diagnosis ability is improved greatly because of the introduction of geometry between data.

RNN has been studied extensively as an effective solution to time series analysis [24]. Several variants are proposed based on RNN, such as gated recurrent units (GRU) networks, bidirectional gated recurrent units (BiGRU) networks, and local feature-based gated recurrent units (LFGRU) networks [25]. These models can capture the temporal dependency

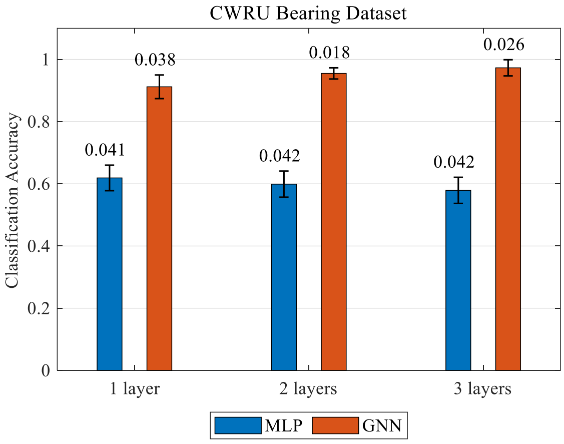


Fig. 5. Average accuracy and standard deviation of 10 repeated experiments on CRWU bearing dataset

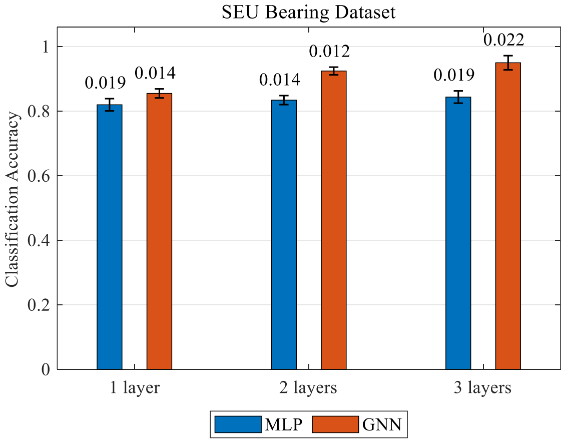


Fig. 6. Average accuracy and standard deviation of 10 repeated experiments on SEU bearing dataset

TABLE III. CLASSIFICATION ACCURACY COMPARISON ON SEU BEARING DATASET

Models	Accuracy
RNN [25]	0.920
GRU [25]	0.924
BiGRU [25]	0.936
LFGRU [25]	0.940
3GNN-2FC (Ours)[a]	**0.950**
4GNN-2FC (Ours)	**0.974**

[a] 3GNN-2FC denotes 3 GNN layers and 2 FC layers

between input data. Further, we compare the presented model with RNN and its variants, as exhibited in TABLE III. The proposed model performs better than RNN baselines when stacking 3 GNN layers and 4 GNN layers. The improvement attributes to the strong data representation ability of HVG and GNN.

V. Conclusion

A new model for bearing faults diagnosis based on HVG and GNN is presented in this paper. The original data series are first transformed to graphs in non-Euclidean space according to the visibility between data points. To extract the underlying features of the graph, GIN is modified to learn the graph feature generated from the vibration signal. Two bearing datasets are involved to validate the performance of the proposed method. The experimental results show that the transformation of HVG provides extra useful information for the classification comparing to pure numerical information. The GNN model also has superiority over RNN, and its variants. In future works, the weighted HVG and limited penetrable HVG will be investigated, and a more efficient GNN architecture will be explored to fit the characteristics of HVG.

Acknowledgment

This work was supported by the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX20_0087.

References

[1] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, “Deep learning and its applications to machine health monitoring,” Mech. Syst. Signal Process., vol. 115, pp. 213–237, 2019.

[2] L. Wen, X. Li, L. Gao, and Y. Zhang, “A New Convolutional Neural Network-Based Data-Driven Fault Diagnosis Method,” IEEE Trans. Ind. Electron., vol. 65, no. 7, pp. 5990–5998, 2018.

[3] Z. Zhu, G. Peng, Y. Chen, and H. Gao, “A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis,” Neurocomputing, vol. 323, pp. 62–75, 2019.

[4] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, “A sparse auto-encoder-based deep neural network approach for induction motor faults classification,” Measurement, vol. 89, pp. 171–178, 2016.

[5] S. Shao, P. Wang, and R. Yan, “Generative adversarial networks for data augmentation in machine fault diagnosis,” Comput. Ind., vol. 106, pp. 85–93, 2019.

[6] Z. Yong, D. R. V., M. Norbert, K. Jürgen, and D. J. F, “Nonlinear time series analysis by means of complex networks,” Sci. Sin. Phys. Mech. Astron., vol. 50, no. 1, p. 010509, Jan. 2020.

[7] Z. Gao, G. Lu, and P. Yan, “Graph-based change detection for condition monitoring of industrial machinery: an enhanced framework for non-stationary condition signals,” Meas. Sci. Technol., vol. 30, p. 115002, 2019.

[8] Y. Gao, D. Yu, and H. Wang, “Fault diagnosis of rolling bearings using weighted horizontal visibility graph and graph Fourier transform,” Measurement, vol. 149, pp. 1–12, 2020.

[9] Y. Gao and D. Yu, “Total variation on horizontal visibility graph and its application to rolling bearing fault diagnosis,” Mech. Mach. Theory, vol. 147, p. 103768, 2020.

[10] J. Zhou et al., “Graph Neural Networks: A Review of Methods and Applications,” pp. 1–20, Dec. 2018, ArXive ID: 1812.08434.

[11] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A Comprehensive Survey on Graph Neural Networks,” IEEE Trans. Neural Networks Learn. Syst., pp. 1–21, 2020.

[12] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, and K. Kavukcuoglu, “Interaction Networks for Learning about Objects, Relations and Physics,” in Advances in Neural Information Processing Systems (NIPS 2016), 2016, pp. 4502–4510.

[13] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, “Protein interface prediction using graph convolutional networks,” in Advances in Neural Information Processing Systems Conference (NIPS 2017), 2017, pp. 6530–6539.

[14] W. L. Hamilton, Z. Ying, and J. Leskovec, “Inductive Representation Learning on Large Graphs,” in Advances in Neural Information Processing Systems (NIPS 2017), 2017, pp. 1024–1034.

[15] Y. Zhang, Q. Liu, and L. Song, “Sentence-state LSTM for text representation,” in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), 2018, vol. 1, pp. 317–327.

[16] X. Wang, Y. Ye, and A. Gupta, “Zero-Shot Recognition via Semantic Embeddings and Knowledge Graphs,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 6857–6866.

[17] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuño, “From time series to complex networks: The visibility graph,” in Proceedings of the National Academy of Sciences of the United States of America, 2008, vol. 105, no. 13, pp. 4972–4975.

[18] B. Luque, L. Lacasa, F. Ballesteros, and J. Luque, “Horizontal visibility graphs: Exact results for random time series,” Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys., vol. 80, p. 046103, 2009.

[19] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural Message Passing for Quantum Chemistry,” in Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 1263–1272.

[20] K. Xu, S. Jegelka, W. Hu, and J. Leskovec, “How powerful are graph neural networks?,” in 7th International Conference on Learning Representations (ICLR 2019), 2019, pp. 1–17.

[21] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” in 5th International Conference on Learning Representations (ICLR 2017), 2017, pp. 1–14.

[22] “Case western reserve university bearing data center.” [Online]. Available: http://csegroups.case.edu/bearingdatacenter.

[23] M. Fey and J. E. Lenssen, “Fast Graph Representation Learning with PyTorch Geometric,” in 7th International Conference on Learning Representations (ICLR2019), 2019, pp. 1–9.

[24] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, “Recent Advances in Recurrent Neural Networks,” 2017, ArXive ID: 1801.01078v3.

[25] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, “Machine health monitoring using local feature-based gated recurrent unit networks,” IEEE Trans. Ind. Electron., vol. 65, no. 2, pp. 1539–1548, 2018.