Online-Journals.org

iTDAF IETI Transactions on Data Analysis and Forecasting

iTDAF | eISSN: 2959-0442 | Vol. 1 No. 3 (2023) | OPEN ACCESS

https://doi.org/10.3991/itdaf.v1i3.44517

PAPER

A Review of Joint Applications of IoT and Deep Learning

Wenhao Tang[1](✉), **Jiaji Wang**[2]

[1]Henan Polytechnic University, Jiaozuo, Henan, PR China

[2]University of Leicester, Leicester, United Kingdom

wenhaotang@home.hpu.edu.cn

ABSTRACT

In recent years, graph convolutional networks (GCNs) have been widely used in image classification tasks. The combination of GCNs and the Internet of Things (IoT) has led to the development of some branches of the latter. This paper explores cases where convolutional neural networks and GCNs are combined with IoT to achieve better results. This paper also focuses on discussing the semi-supervised classification task of GCNs. The innovative approach explored for innovative GCNs dealing with semi-supervised classification tasks lies in optimizing the GCN topology and using graph convolutional operations in the topological space for better training of the model.

KEYWORDS

graph convolutional networks (GCNs), Internet of Things (IoT), convolutional neural networks, semi-supervised classification, network topology

1 INTRODUCTION

Image classification, object localization, semantic segmentation, and instance segmentation are the basic tasks of computer vision [1]. While the image classification task is the core task in computer vision, its goal is to distinguish different categories of images based on different features reflected in the image information. A category label is selected for a given input image from a known set of category labels.

Deep learning is a machine learning technique that has been developing rapidly in recent years and that realizes feature extraction and classification by constructing neural networks to simulate human brain functions. IoT technology image classification techniques are applied to face recognition, environmental monitoring [2], etc. K-nearest neighbor algorithms and support vector machines perform well on simple image classification but underperform on complex images. K-nearest neighbor algorithms are overly reliant on selecting the distance metric function and the parameter K [3], which leads to the degradation of classification performance. Although extreme learning machines have some advantages in image classification, such as fast training speed and simple implementation, they also have some drawbacks, including poor robustness to noise [4] and overfitting problems. The primary

Tang, W., Wang, J. (2023). A Review of Joint Applications of IoT and Deep Learning. *IETI Transactions on Data Analysis and Forecasting (iTDAF)*, 1(3), pp. 4–17. https://doi.org/10.3991/itdaf.v1i3.44517

Article submitted 2023-07-30. Revision uploaded 2023-09-14. Final acceptance 2023-09-20.

purpose of image preprocessing is to improve the reliability of feature extraction, image segmentation, matching, and recognition by eliminating irrelevant information from the image, recovering valuable and true information, enhancing the detectability of the information in question, and minimizing the data. In the model evaluation session, training is usually performed using the classifier trained on a training set [5, 6]. The validation set is generally used to predict performance and observe the model for overfitting or underfitting problems. The test set evaluates the performance with metrics such as accuracy, precision, specificity, recall, and F1-score. For cross-validation, no validation set is needed.

The joint application of deep learning and IoT has led to a broader scope of research and application of the former. In the development of cross-research between the two fields of deep learning and IoT, most cases have demonstrated the superiority of adapting to complex tasks compared to traditional methods. Lin et al., (2023) [7] pointed out that IoT suffers from the large size of assets, complex and diverse structures, and scarcity of computational resources compared with the characteristics of the traditional Internet. Moreover, they stated that IoT encounters many significant challenges in this era of vast amounts of data and information: large quantities of data redundancy, bottlenecks in cloud processing power, data security, and privacy. In the related research on intrusion detection techniques, it was found that most intrusion detection techniques fail to meet the actual demand standards of IoT, and there are also problems of poor detection of complex network intrusion methods [8].

Therefore, we focus on some of the classical model development cases of deep learning and the cross-study cases with IoT. The following is the structure of the paper: Section 2 first describes the basic concepts, advantages, and disadvantages of convolutional neural networks. Then, the classical convolutional neural network is used as an entry point and discusses the successful cases of combined application with IoT. Moreover, Section 3 focuses on application cases of graph convolutional networks in IoT and semi-supervised classification. Finally, Section 4 states our conclusions.

2 CONVOLUTIONAL NEURAL NETWORKS

2.1 Outline

Convolutional neural networks (CNNs) are a deep learning model based on feed-forward artificial neural networks, and the core idea is to use convolutional operations to extract features from the input data. CNNs consist of multiple layers, where convolutional, activation, and pooling layers map the input data to the feature space [9]. The fully connected layer convolves the feature mapping to obtain a one-dimensional vector [10]. Finally, it is generally normalized by the softmax function to get the predicted probability of each kind of distribution. The convolutional layer extracts local features from the input image by convolutional computation, which captures information such as local textures, edges, etc., in the image [11]. The activation layer introduces nonlinear factors to make the network more expressive and capable of more complex processing of the input data. The pooling layer, on the other hand, reduces the amount of computation and the risk of overfitting by decreasing the feature dimensions while retaining important feature information. Finally, the fully connected layer connects the outputs of the previous layers and realizes the final classification task. CNNs have achieved remarkable success in image recognition, natural language processing, etc., and have become one of the important research directions in deep learning. CNNs have the following advantages over traditional methods [12]: CNNs have a hierarchical feature representation and exponentially growing deep architecture representation; e.g., deep layers can extract richer semantic information than shallow layers.

2.2 Classic convolutional neural networks

A CNN [13] is a neural network specialized in processing data with a grid-like structure. It can also be described as a class of feed forward neural networks containing convolutional computation and a deep structure.

It was not until 2012 that Krizhevsky et al. (2012) [14] proposed AlexNet, as shown in Figure 1, a deeply scaled CNN that received first place in the ImageNet Large Scale Visual Recognition Challenge, that researchers began to pay attention to CNNs. AlexNet's innovations include the use of ReLU instead of the traditional saturated nonlinear function tanh, which reduces the computational complexity and improves the training speed; randomly discarding a portion of neurons by the dropout technique to improve the model's robustness and reduce the overfitting of the fully connected layer; and increasing the training samples by image panning, horizontal mirroring, and grayscale transformation to reduce the overfitting.

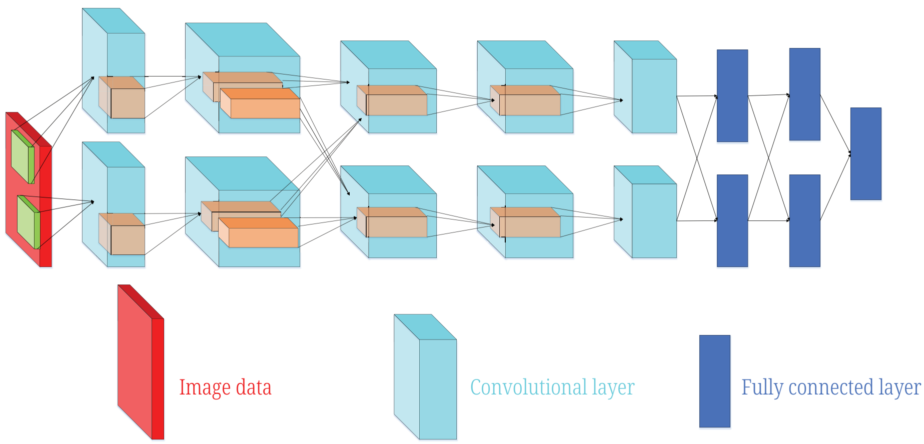

Fig. 1. A simple representation of the architecture of AlexNet

In 2014, Szegedy et al. (2014) [15] proposed GoogleNet. It is a CNN with more than 20 layers that employs convolutional operations with convolutional kernel sizes of 1×1, 3×3, and 5×5 to improve computational resource utilization. Moreover, the model has fewer parameters. In the same year, Simonyan and Zisserman (2014) [16] explored the importance of depth for CNNs. They constructed VGGNet by adding convolutional layers with 3×3 convolutional kernels to deepen the network. The performance is significantly improved when the number of layers reaches between 16 and 19. The VGG model replaces a single large convolutional kernel with multiple layers of small convolutional kernels to reduce the number of parameters.

In 2015, He et al. (2016) [17] introduced the residual network (ResNet) to solve the gradient vanishing problem. They introduced a shortcut connection technique across layers of information input and summed them with convolutional results, as shown in Figure 2, ResNet has only one pooling layer, allowing the underlying network to be fully trained and accuracy to increase significantly with depth. ResNet, with a depth of 152 layers, won first place in LSVRC-15. The depth of ResNet even reaches 1000 layers and is validated in the CIFAR-10 dataset.

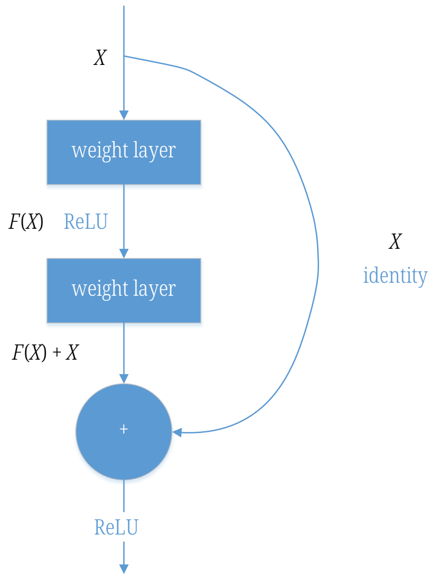

Fig. 2. Diagram of skip connection

The training of deep Convolutional Neural Networks (CNNs) poses challenges related to computational power, gradients, and activation functions [18, 19]. Additionally, the efficiency gains from adding convolutional layers are not significant [20]. Image classification tasks benefit from deep networks, but issues such as occlusion and blurring remain to be addressed [21]. Further, for most deep learning based on single image dehazing methods, the CNNs used for extracting features can only capture local features [22].

2.3 Newer development of convolutional neural networks

Convolutional neural networks are used in recent food image detection methods because they can effectively learn advanced features from data [23].

To cope with the problem of poor performance of CNNs in handling nonlinear data and sensitivity to local minima of training errors, Ahmed et al. (2022) used the MobileNetV3 network to extract high-quality features from food images. To select the best subset of features, they used the Shapley additive pre-planning (SHAP) algorithm, which effectively evaluates the prediction performance of different combinations of attributes. Finally, they further improved the classification performance by employing the powerful generalization capabilities and nonlinear decision bounds of the kernel extreme learning machine (KELM).

Kernel extreme learning machine is a state-of-the-art classification method for dealing with linearly indistinguishable features with good generalization ability. However, it is highly complex when dealing with large-scale datasets and requires complex inner product matrix computations. In contrast, the authors chose an approach that can benefit from the parallel and distributed environment of cloud computing.

Their proposed new classifier, called the multi-column kernel extreme learning machine (MCKELM), utilizes a distributed cloud environment to solve the dimensionality problem. In the food detection classification phase, MCKELM decomposes the network structure of KELM, uses an efficient neural network to extract reliable features, and utilizes SHAP values for feature selection to filter out irrelevant features. In addition, the visualization provided by Gradient Explainer highlights the image pixels and improves the interpretability of their proposed architecture. This approach performs better in terms of performance and interpretability than existing techniques.

Ben Atitallah et al. (2022) [24] proposed a new approach for detecting and multi-categorizing IoT malware by leveraging the power of deep migration learning. By utilizing pre-trained CNNs, including ResNet18, MobileNetV2, and DenseNet161, and employing ensemble strategies such as voting, stacking, and decision fusion, the proposed method improves detection accuracy without the need to create models from scratch. This approach capitalizes on the benefits of the deep TL methodology. They use an integrated learning strategy to fuse the outputs of the three CNN models mentioned above. The details of this integrated learning approach are exciting. It contains hard-voting, soft-voting, and stacking strategies that compensate each other for errors in classification results or reduce generalization errors. The MaleVis dataset, consisting of more than 14,000 RGB images from 26 different series, including various malware types and one benign category, was subjected to a rigorous performance evaluation that demonstrated the superiority of the methodology. A comparative analysis with recent research results on the same dataset further highlights the effectiveness of the proposed IoT malware detection and classification strategy.

Unconstrained by the traditional deep learning approach, where a complete model architecture contains only one CNN, Alabsi et al. (2023) [25] designed a combination of two CNNs to detect IoT cyber-attacks. The core idea of this approach is to select the most informative features as inputs to the second newly constructed CNN by the first CNN, calculating the average activation of each feature mapping across all instances in the test dataset. The most informative features are selected by sorting the average number of activations in descending order and selecting the top k feature maps to identify the most informative feature maps as inputs to the newly constructed CNN. After experiments on the BoT-IoT 2020 dataset, it was confirmed that CNN-CNN has strong robustness and high accuracy.

The joint application of deep learning and IoT in analyzing and identifying potential malware attacks has also been successful. Conventional CNN-LSTM methods cannot provide sufficient useful features for classifiers. The novel CNN-LSTM proposed by Akhtar and Feng (2022) [26] is innovative in that it combines the temporal and geographical interactions of the CNN-LSTM. Using the CNN-LSTM in the dataset experiments used for malware detection confirms that the CNN-LSTM has the highest accuracy by comparing the experimental data with decision trees, support vector machines, and LSTMs.

Convolutional neural networks are widely used in the field of biological image segmentation [27] and have also significantly improved the performance of semantic image segmentation [28]. However, the presence of feature extraction by CNNs may lead to the loss of contextual and spatial information. To overcome these problems of CNN for semantic image segmentation, Jiang et al. (2021) proposed a multilevel graph convolutional recurrent neural network (MGCRNN). This network combines CNN and graph neural networks (GNN) to fuse multilevel features. Building on the success of the GCRNN, the MGCRNN can obtain a holistic view of an image and converge multilevel contextual and structural information. Experiments confirm that MGCRNN ensures no loss of spatial data and achieves flexible sensory field and structural feature learning capabilities.

3 GRAPH CONVOLUTIONAL NETWORKS

3.1 Outline

A GCN is a method that enables deep learning of graph data [29, 30]. GCNs analyze nodes by combining the features of the nodes and their neighbors. The core idea of GCNs is to update a node's features by aggregating the neighboring nodes' information with the current node through a graph convolution operation. GCNs are generally used to process non-Euclidean data [31] to compensate for the lack of CNNs. From the perspective of convolutional methods, GCNs can be categorized into two types: spectral and null domains. Kipf and Welling (2016) [32] proposed spectral convolution, whereby the filter of the convolutional network is converted to the Fourier domain for processing along with the graph signal. Niepert et al. (2016) [33] proposed null-domain convolution, which constructs a hierarchy by connecting nodes in the graph in the spatial domain and then performs a convolution operation.

Suppose there is an undirected graph G. Denote the set of nodes by G, and $G = \{V, E, X\}$. E, X denotes the set of edges between nodes and the matrix of node features, respectively. If each node has a feature vector of dimension d, then X will be an $n \times d$ matrix, where n is the number of nodes.

The propagation formula for the improved GCNs is as follows:

$$H^{(l+1)} = \sigma(\tilde{A}H^{(l)}W^{(l)}). \tag{1}$$

where, $H^{(l)}$ represents the node features of l-th layer of GCN. σ denotes certain activation functions.

$$\tilde{A} = \tilde{D}^{-\frac{1}{2}}\hat{A}\tilde{D}^{-\frac{1}{2}}, \tag{2}$$

where $\tilde{A}$ is a sparse matrix, and D is a diagonal matrix representing the degree of the nodes. $\hat{A}$ is a node self-connected adjacency matrix, and

$$\hat{A} = [A_{ij}] \in \mathbb{R}^{N\times N}. \tag{3}$$

and

$$\hat{A} = A + I_N, \tag{4}$$

where I_N are identity matrices, and A is an adjacency matrix [34]. $W^{(l)}$ is a learnable weight matrix.

The graph convolutional layer consists of two phases: feature aggregation and feature extraction [35, 36]. Feature aggregation controls the attributes of local neighboring nodes to enhance similarity. Feature extraction follows feature aggregation and helps extract common features between neighboring nodes.

Before the improvement of Eq. 1,

$$\tilde{A} = \tilde{D}^{-\frac{1}{2}}\hat{A}\tilde{D}^{-\frac{1}{2}} + I_N, \tag{5}$$

where, $\hat{A} = A$, the range of eigenvalues in the Eq. 5 was small. When this operator is applied multiple times in a deep neural network model, it may lead to unstable values, triggering the problem of exploding or vanishing gradients. To mitigate this problem, Kipf and Welling (2016) redefined the matrix as shown in the following equation,

$$\hat{A} = A + I_N, \tag{6}$$

and the sparse matrix $\tilde{A}$ then changes as follows

$$\tilde{A} = \tilde{D}^{-\frac{1}{2}}\hat{A}\tilde{D}^{-\frac{1}{2}}. \tag{7}$$

3.2 Applications in the Internet of Things

The GCNs have been widely applied and studied in the IoT field in recent years. The IoT generates heterogeneous data and complex relationships as a network connecting various physical devices and sensors. Traditional machine learning methods are often tricky to handle these data effectively. GCNs, as a deep learning method with the ability to handle graph-structured data, provide new ideas for data modeling and analysis in IoT.

Ma et al. (2023) [37] proposed an innovative approach for addressing the challenges posed by the demands of IoT applications in shared networks. Combining IoT devices' topology and node features achieves more accurate and efficient node classification. Specifically, the method uses GCN to capture the relationship between topology and node features and utilizes this information to update the embedded representation of the nodes. The GCN-VNE method efficiently embeds the virtual network into the physical infrastructure. The policy network in the GCN-VNE consists of two hidden layers of GCNs. In each layer, the adjacency matrix A stays static while the input matrix X and the convolution kernel matrix W change dynamically. Each embedding result is randomly labeled, and all virtual nodes are manually defined as an entity with an n-dimensional label vector y, where n equals the number of nodes in the underlying network. The method combines the advantages of graph convolutional networks and virtual network embedding to classify and recognize IoT devices effectively. Through many experiments, they demonstrated that the GCN-VNE algorithm performs well.

The application of GCNs in IoT can be categorized into several aspects. First, GCNs perform well in the task of node classification in IoT. By taking the devices, sensors, etc., in IoT as nodes of the graph and constructing the connection relationship between the devices as edges of the graph, GCNs can effectively learn the interactions and relationships between the nodes to realize the classification and recognition of the nodes. For example, traffic flow prediction [38], transportation trajectory prediction [39], environmental monitoring [40], and other fields in smart cities can use GCNs to classify nodes and achieve accurate forecasts and analyses.

The method proposed by Li and Li (2021) [41] introduces a GPFS system to predict human poses in smart homes through graph modeling effectively. Unlike traditional methods, the system adopts an online learning technique that allows real-time updating of the prediction model to improve the accuracy and usefulness of the prediction gradually. The GPFS system has the advantage of real-time learning by utilizing continuously accumulated pose data, and it has achieved remarkable results in smart home environments. The system can capture the complex relationship between human posture and the environment by constructing a graph structure and connecting various data points to achieve more accurate predictions.

Online learning techniques further enhance the system's adaptability, allowing it to continuously adjust the prediction model to accommodate changes in the environment and data. Their approach has significant potential to improve the accuracy and utility of predictions and positively impact the development of the smart home and human behavior analytics fields.

Graph neural networks also play an essential role in anomalous alert identification in cyber security. Intrusion detection systems (IDSs) check for intrusions and try to generate a series of alerts [42]. An alert graph is constructed by using alerts as nodes, and the presence of edges is measured by the similarity of the alerts [42]. Alert-GCN [42] performs node classification in the alert graph, thus enabling attack scenario discovery. Alert-GCN outperforms traditional models because it can better capture the implicit relationships between different alerts from the alert graph, which outperforms traditional models. However, their shallower models fail to overcome the over-smoothing problem of graph neural networks.

Graph neural networks can also be used for recommend graph data in the Internet of Vehicles (IoV). Jiang et al. (2022) [43] further explored the relationship between the two entities and other potential interests of users by constructing a graph structure and generating a knowledge graph with users and vehicles as entities. They studied recommendation techniques based on knowledge graph-KGCN using different aggregators to collect neighbourhood information. The aggregation is repeated H times for the obtained user-vehicle entity pairs in a sensory field of size H. The aggregation is done in the same way for the user-vehicle entity pairs. The features of each entity in the neighbourhood are computed once after each of the above processes. Next, the features obtained by aggregating H times are aggregated with themselves to get a feature representation for the next iteration. Their method performs better compared to traditional methods. However, their improved method requires a larger dataset and higher cost.

Further, GCNs are also applied in intrusion detection tasks for the IoT. The core idea of the NIDS approach, flow topology-based graph convolutional network (FT-GCN), proposed by Deng et al. (2023) [44], is to learn combinatorial representations from the topological and statistical features of the flow and provide classification results to determine whether the flow is malicious or not. They designed node-level spatial attention (NLS), an approach inspired by the squeeze and excitation network [45]. The attention mechanism can be further applied to non-Euclidean structured data. Different aspects of the data are used as nodes in the graph. These nodes can be represented as feature vectors, meaning that attention can also be applied to domains using graph-structured data [46]. During an NLS run, the input feature graph X first undergoes global average pooling, which can express information about the global sensory field. The above process is shown in the Eq. 8,

$$Z = avgpool\left(\frac{1}{N}\sum_{i=0}^{N-1} X_i\right), \tag{8}$$

where *avgpool* refers to the global average pooling. After that, the first fully connected layer reduces the feature map's size and is activated using the ReLU function. Then, the feature map undergoes a second fully connected layer for upscaling and is activated using the sigmoid function. The above process is shown in the Eq. 9,

$$S = \sigma\left\{W_2\left[\delta(W_1 \cdot Z)\right]\right\}, \tag{9}$$

where S refers to the feature map obtained by the attention mechanism. And, W_1, W_2 refer to the weights of the first and second fully connected layers.

Finally, the feature map weight matrix S obtained from the above process is multiplied by the matrix of the original feature map X, to get the output $\hat{X}$ of the node-level spatial attention.

$$\hat{X} = S \cdot X. \tag{10}$$

Input $\hat{X}$ to the topologically adaptive graphic convolutional network (TAGCN) and use it to feature learning. Experiments on three datasets show that FT-GCN has high detection accuracy.

3.3 Commonly used citation network datasets

CiteSeer, CORA, PubMed [47, 48], and BlogCatalog [49] are datasets commonly used for GCN semi-supervised node classification. The first three datasets above contain sparse bag-of-words feature vectors for each document and a list of citation links between documents [32]. The following is a description of these datasets (Table 1):

Table 1. Description of the three common datasets

Dataset	Number of Nodes	Dimension of Node Features	Classes of Nodes	Number of Edges
CORA	2708	1433	7	5429
CiteSeer	3312	3703	6	4732
PubMed	19717	500	3	44338
BlogCatalog	5196	8189	6	343486

The CORA dataset has a total of 2708 sample points, each of which is a scientific paper, all categorized into seven classes. A 1433-dimensional word vector represents each paper, so each sample point has 1433 features. Each paper cites or is cited by at least one other paper. If the sample points are viewed as points in a graph, this is a connected graph with no isolated points.

The CiteSeer dataset contains 3312 scientific publications in six categories. The citation network consists of 4732 edges. Each publication in the dataset is described by a 0/1-valued word vector that indicates whether the corresponding word exists in the dictionary. The dictionary contains 3703 unique words.

The PubMed dataset contains 19717 nodes and 44338 edges organized into three categories. A 500-dimensional feature descriptor represents each node and edge.

The BlogCatalog dataset is a dataset that deals with social networks and user behavior, with a wide range of social network analysis and community discovery applications. For studying social network structure, user interests, and community characteristics, BlogCatalog provides researchers with a rich data resource.

3.4 Application to semi-supervised classification

Wang et al. (2020) [50] noted that the success of GCNs is partly attributed to the fact that they provide a fusion strategy of topology and node features to learn node representations. They proposed adaptive multi-channel graph convolutional networks (AM-GCN). The core idea of the model is to simultaneously extract special embeddings and common embeddings from node features, topology, and their combinations and use the attention mechanism to learn the adaptive fusion weights of the embeddings. The advantages of AM-GCN include the following: the AM-GCN performs graph convolution operations on topology and feature space and combines with the attention mechanism so that the different information can be adequately fused; AM-GCN extracts the most relevant information from node features and topology and performs challenging classification tasks well. Compared with traditional GCNs, the main advantage of AM-GCN is its adaptive multi-channel learning ability, i.e., it propagates the node features in the topology space and the feature space and adaptively extracts and fuses the relevant information in these two spaces. Specifically, the core framework of AM-GCN is divided into two main parts: the construction of the feature graph and the propagation process of GCN. First, for the construction of the feature graph, AM-GCN takes the k-nearest neighbor graph of the nodes as the structure of the feature graph and uses a shared parameter strategy to design the co-convolution module to extract the embeddings that are shared on both the feature graph and the topology graph. This approach capitalizes on the similarity between features and the similarity inferred from the topology and can adaptively fuse this information. Second, during GCN propagation, AM-GCN allows node features to propagate not only in topological space but also in feature space. In this way, each node can update its representation according to its features and topology. In addition, AM-GCN introduces a difference constraint to ensure independence between embeddings extracted in the two spaces. Because of the adaptive multi-channel propagation and the difference constraint, AM-GCN can achieve excellent performance in semi-supervised node classification tasks. It can fully utilize the data's topological information and node features and adaptively fuse them to derive more accurate classification results. AM-GCN is proven to perform well on multiple datasets.

Heidari and Iosifidis (2021) [51] found that existing methods employ a user-defined network structure to obtain node embeddings by experimenting with a fixed number of layers and neurons per layer using a hierarchical propagation rule. They designed an automated process to define the architecture of graph-convolutional networks for solving specific problems, as shown in Figure 3, which can help reduce the cost of manually designing model structures.

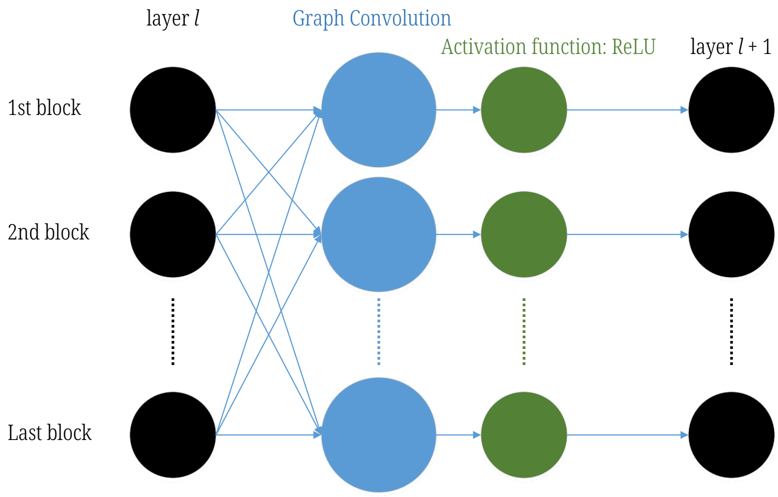

Fig. 3. The chart is an example of a novel GC layer design scheme for PGCN. In this network architecture, graph convolutional feature aggregation occurs for each existing block. Each time, it accepts the graph characterization information from the output of all the pre-existing blocks in the previous layer

They proposed progressive graph convolutional networks (PGCN), a type of GCN that gradually increases each layer's size and hence the entire network's size. PGCN has a compact number of parameters and good performance. It also makes the GCN topology guaranteed to converge to a (local) minimum, which helps improve performance. Their asymptotic learning approach can be summarized as follows: Before stopping the fine-tuning optimization, each layer is trained with at least one block, and the end of learning at each layer is marked by the failure of the training on the next block to achieve better results. When the next layer does not achieve better results, the model stops fine-tuning the optimization, removes this last layer, and stops growing the network's topology. Predictions are made directly with the saved output layer weights of the best results. The design concept of Progressive Graph Convolutional Networks exploits the features of graph structure and the challenges of semi-supervised learning to improve the performance of semi-supervised node classification through a novel graph convolutional network structure and training strategy. This strategy enables the model to utilize unlabeled data better and improves generalization performance and stability. Therefore, progressive graph convolutional networks are a promising semi-supervised node classification method. Although PGCN performs better, the training process has to deal with large-scale graphs with millions of nodes and edges, which requires a large amount of memory.

4 CONCLUSION

This paper focuses on applying CNNs and GCNs to IoT and semi-supervised classification. First, the basic concepts, advantages, and disadvantages of CNNs are described. Then, the classical CNNs are used as an entry point, and the successful cases of combined application with IoT are discussed. The subsequent section focuses on the application cases of GCNs in IoT and semi-supervised classification. In the case of IoT, GCNs have been developed in several domain branches. Topological map studies are significant in facilitating the development of joint applications. As for the semi-supervised classification task, relevant cases have been studied and analyzed, highlighting the importance of improving the topology of traditional networks and utilizing the topology to train models better. The combination of GCNs with IoT has potential for development, and it is believed that GCNs can achieve better results in semi-supervised classification tasks as well. In the future, we will pay more attention to the joint application of deep learning and IoT. And we will continue to summarize our research on its development.

5 REFERENCES

[1] B. Yang, Y. Kang, L. Zhang, and H. Li, "GGAC: Multi-relational image gated GCN with attention convolutional binary neural tree for identifying disease with chest X-rays," *Pattern Recognition*, vol. 120, p. 108113, 2021. https://doi.org/10.1016/j.patcog.2021.108113

[2] Y. R. Wu, X. Y. Zhang, Y. Xiao, and J. Feng, "Attention neural network for water image classification under IoT environment," *Applied Sciences*, vol. 10, no. 3, p. 909, 2020. https://doi.org/10.3390/app10030909

[3] Q. H. Wang, J. W. Liu, and L. L. Zhang, "Study on the classification of K-nearest neighbor algorithm," *Journal of Xi'an Technological University*.

[4] J. Cao, K. Zhang, M. Luo, C. Yin, and X. Lai, "Extreme learning machine and adaptive sparse representation for image classification," *Neural Networks*, vol. 81, pp. 91–102, 2016. https://doi.org/10.1016/j.neunet.2016.06.001

[5] K. Hirose, K. Miura, and A. Koie, "Hierarchical clustered multiclass discriminant analysis via cross-validation," *Computational Statistics & Data Analysis*, vol. 178, p. 107613, 2023. https://doi.org/10.1016/j.csda.2022.107613

[6] L. Du *et al.*, "Development and external validation of a machine learning-based prediction model for the cancer-related fatigue diagnostic screening in adult cancer patients: A cross-sectional study in China," *Supportive Care in Cancer*, vol. 31, no. 2, p. 106, 2023. https://doi.org/10.1007/s00520-022-07570-w

[7] H. Lin, Q. Xue, J. Feng, and D. Bai, "Internet of Things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine," *Digital Communications and Networks*, vol. 9, no. 1, pp. 111–124, 2023. https://doi.org/10.1016/j.dcan.2022.09.021

[8] Y. Wang, J. Wang, H. Jin, and I. S. Ansari, "Network intrusion detection method based on improved CNN in Internet of Things environment," *Mobile Information Systems*, vol. 2022, p. 10, 2022. https://doi.org/10.1155/2022/3850582

[9] Y.-D. Zhang, "Twelve-layer deep convolutional neural network with stochastic pooling for tea category classification on GPU platform," *Multimedia Tools and Applications*, vol. 77, no. 17, pp. 22821–22839, 2018. https://doi.org/10.1007/s11042-018-5765-3

[10] Z. Zhu, S. Wang, and Y. Zhang, "A survey of convolutional neural network in breast cancer," *Computer Modeling in Engineering & Sciences*, vol. 136, no. 3, pp. 2127–2172, 2023. https://doi.org/10.32604/cmes.2023.025484

[11] K. Muhammad, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3613–3632, 2019. https://doi.org/10.1007/s11042-017-5243-3

[12] C. M. He, H. Y. Kang, T. Yao, and X. R. Li, "An effective classifier based on convolutional neural network and regularized extreme learning machine," *Mathematical Biosciences and Engineering*, vol. 16, no. 6, pp. 8309–8321, 2019. https://doi.org/10.3934/mbe.2019420

[13] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. https://doi.org/10.1162/neco.1989.1.4.541

[14] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, 2012.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, and A. Rabinovich, "Going deeper with convolutions," *IEEE Computer Society*, pp. 1–9, 2015. https://doi.org/10.1109/CVPR.2015.7298594

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Science*, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Las Vegas, NV, USA, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

[18] S.-H. Wang, "Unilateral sensorineural hearing loss identification based on double-density dual-tree complex wavelet transform and multinomial logistic regression," *Integrated Computer-Aided Engineering*, vol. 26, pp. 411–426, 2019. https://doi.org/10.3233/ICA-190605

[19] A. K. Sangaiah, "Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization," *Neural Computing and Applications*, vol. 32, pp. 665–680, 2020. https://doi.org/10.1007/s00521-018-3924-0

[20] J. Wang, S. Satapathy, S. Wang, and Y. Zhang, "LCCNN: A lightweight customized CNN-based distance education app for COVID-19 recognition," *Mobile Networks and Applications*, 2023. https://doi.org/10.1007/s11036-023-02185-9

[21] R. N. Wessner, R. Frozza, D. Duarte da Silva Bagatini, and R. F. Molz, "Recognition of weeds in corn crops: System with convolutional neural networks," *Journal of Agriculture and Food Research*, vol. 14, p. 100669, 2023. https://doi.org/10.1016/j.jafr.2023.100669
[22] B. Hu, Z. Z. Yue, M. C. Gu, Y. Zhang, Z. Xu, and J. H. Li, "Hazy removal via graph convolutional with attention network," *Journal of Signal Processing Systems for Signal Image and Video Technology*, vol. 95, no. 4, pp. 517–527, 2023. https://doi.org/10.1007/s11265-023-01863-x
[23] G. Ahmed, T. Chu, and K. Loo, "Novel multicolumn kernel extreme learning machine for food detection via optimal features from CNN," arXiv preprint arXiv:2205.07348, 2022.
[24] S. Ben Atitallah, M. Driss, and I. Almomani, "A novel detection and multi-classification approach for IoT-malware using random forest voting of fine-tuning convolutional neural networks," *Sensors*, vol. 22, no. 11, p. 4302, 2022. https://doi.org/10.3390/s22114302
[25] B. A. Alabsi, M. Anbar, and S. D. A. Rihan, "CNN-CNN: Dual convolutional neural network approach for feature selection and attack detection on Internet of Things networks," *Sensors*, vol. 23, no. 14, p. 6507, 2023. https://doi.org/10.3390/s23146507
[26] M. S. Akhtar and T. Feng, "Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time," *Symmetry*, vol. 14, no. 11, p. 2308, 2022. https://doi.org/10.3390/sym14112308
[27] H. Xue, Z. Hu, S. Wang, and Y. Zhang, "A survey on deep learning in COVID-19 diagnosis," *Journal of Imaging*, vol. 9, p. 1, 2023. https://doi.org/10.3390/jimaging9010001
[28] D. C. Jiang, H. Qu, J. H. Zhao, J. L. Zhao, and W. Liang, "Multi-level graph convolutional recurrent neural network for semantic image segmentation," *Telecommunication Systems*, vol. 77, no. 3, pp. 563–576, 2021. https://doi.org/10.1007/s11235-021-00769-y
[29] S.-H. Wang, "Covid-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network," *Information Fusion*, vol. 67, pp. 208–229, 2021. https://doi.org/10.1016/j.inffus.2020.10.004
[30] D. S. Guttery, "Improved breast cancer classification through combining graph convolutional network and convolutional neural network," *Information Processing and Management*, vol. 58, p. 2, 2021. https://doi.org/10.1016/j.ipm.2020.102439
[31] X. Z. Xu, X. Y. Zhao, M. Wei, and Z. N. Li, "A comprehensive review of graph convolutional networks: Approaches and applications," *Electronic Research Archive*, vol. 31, no. 7, pp. 4185–4215, 2023. https://doi.org/10.3934/era.2023213
[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, ICLR, 2016.
[33] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," *JMLR.org*, 2016.
[34] A. Apicella, F. Isgrò, A. Pollastro, and R. Prevete, "Adaptive filters in graph convolutional neural networks," *Pattern Recognition*, vol. 144, p. 109867, 2023. https://doi.org/10.1016/j.patcog.2023.109867
[35] B. Y. Chen, Y. Ma, J. Wang, T. Jia, X. Liu, and W. H. K. Lam, "Graph convolutional networks with learnable spatial weightings for traffic forecasting applications," *Transportmetrica A Transport Science*, 2023. https://doi.org/10.1080/23249935.2023.2239377
[36] D. Li, W. Zhao, J. Hu, S. Zhao, and S. Liu, "A long-term water quality prediction model for marine ranch based on time-graph convolutional neural network," *Ecol. Indic.*, vol. 154, p. 110782, 2023. https://doi.org/10.1016/j.ecolind.2023.110782
[37] S. H. Ma *et al.*, "Graph convolutional network aided virtual network embedding for Internet of Thing," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 1, pp. 265–274, 2023. https://doi.org/10.1109/TNSE.2022.3207205
[38] F. Liang, C. Qian, W. Yu, D. Griffith, and N. Golmie, "Survey of graph neural networks and applications," *Wireless Communications & Mobile Computing*, vol. 2022, p. 28, 2022. https://doi.org/10.1155/2022/9261537

[39] R. W. Liu *et al.*, "STMGCN: Mobile edge computing-empowered vessel trajectory prediction using spatio-temporal multigraph convolutional network," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7977–7987, 2022. https://doi.org/10.1109/TII.2022.3165886

[40] C. H. Yang, H. H. Shuai, C. Y. Shen, and M. S. Chen, "Learning to solve task-optimized group search for social Internet of Things," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5429–5445, 2022. https://doi.org/10.1109/TKDE.2021.3057361

[41] X. Li and D. W. Li, "GPFS: A graph-based human pose forecasting system for smart home with online learning," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, 2021. https://doi.org/10.1145/3460199

[42] Q. M. Cheng, C. M. Wu, and S. Y. Zhou, "Discovering attack scenarios via intrusion alert correlation using graph convolutional networks," *IEEE Communications Letters*, vol. 25, no. 5, pp. 1564–1567, 2021. https://doi.org/10.1109/LCOMM.2020.3048995

[43] X. L. Jiang, Y. Fu, and C. C. Dong, "Recommendation method for fusion of knowledge graph convolutional network," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, 2022. https://doi.org/10.1186/s13634-022-00854-7

[44] X. H. Deng, J. C. Zhu, X. J. Pei, L. Zhang, Z. Ling, and K. P. Xue, "Flow topology-based graph convolutional network for intrusion detection in label-limited IoT networks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 684–696, 2023. https://doi.org/10.1109/TNSM.2022.3213807

[45] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020. https://doi.org/10.1109/TPAMI.2019.2913372

[46] G. Brauwers and F. Frasincar, "A general survey on attention mechanisms in deep learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3279–3298, 2023. https://doi.org/10.1109/TKDE.2021.3126456

[47] Prithviraj *et al.*, "Collective classification in network data," *Ai Magazine*, vol. 29, no. 3, p. 93, 2008. https://doi.org/10.1609/aimag.v29i3.2157

[48] B. Jiang, Z. Zhang, D. Lin, J. Tang, and B. Luo, "Semi-supervised learning with graph learning-convolutional networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. https://doi.org/10.1109/CVPR.2019.01157

[49] Z. Meng, S. Liang, H. Bao, and X. Zhang, "Co-embedding attributed networks," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, Melbourne VIC, Australia, 2019. https://doi.org/10.1145/3289600.3291015

[50] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, "AM-GCN: Adaptive multi-channel graph convolutional networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020. https://doi.org/10.1145/3394486.3403177

[51] N. Heidari and A. Iosifidis, "Progressive graph convolutional networks for semi-supervised node classification," *IEEE Access*, vol. 9, pp. 81957–81968, 2021. https://doi.org/10.1109/ACCESS.2021.3085905

6 AUTHORS

Wenhao Tang is a Computer Science and Technology student at the Henan Polytechnic University (E-mail: wenhaotang@home.hpu.edu.cn).

Jiaji Wang is presently a Computer Science M.Phil., student at the University of Leicester (E-mail: jw933@le.ac.uk).