


# Chapter 6

## Fundamental Concepts in Graph Attention Networks

**R. Soujanya**

*Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India*

**Ravi Mohan Sharma**

 <https://orcid.org/0000-0001-5750-0450>

*Makhanlal Chaturvedi National University of Journalism and Communication, Bhopal, India*

**Manish Manish Maheshwari**

*Makhanlal Chaturvedi National University of Journalism and Communication, Bhopal, India*

**Divya Prakash Shrivastava**

*Higher Colleges of Technology, Dubai, UAE*

### ABSTRACT

*Graph attention networks, also known as GATs, are a specific kind of neural network design that can function on input that is arranged as a graph. These networks make use of masked self-attentional layers in order to compensate for the shortcomings that were present in prior approaches that were based on graph convolutions. The main advantage of GAT is its ability to model the dependencies between nodes in a graph, while also allowing for different weights to be assigned to different edges in the graph. GAT is able to capture both local and global information in a graph. Local information refers to the information surrounding each node, while global information refers to the information about the entire graph. This is achieved through the use of attention mechanisms, which allow the network to selectively focus on certain nodes and edges while ignoring others. It also has scalability, interpretability, flexibility characteristics. This chapter discusses the fundamental concepts in graph attention networks.*

DOI: 10.4018/978-1-6684-6903-3.ch006

## INTRODUCTION

Graph Attention Networks, also known as GATs, focus on graph data in their analysis. The GAT is constructed using graphs of increasing attention levels that are stacked one over the other. The input for each graph attention layer is the node embeddings, while the layer's output is an updated version of the original node embeddings. While determining how the node should be embedded, the embeddings of the other nodes to which it is linked are considered (Velickovic et al., 2018).

It is possible to explain what a graph attention network is by saying that it makes use of the attention mechanism that is present in graph neural networks in order to address some of the flaws that are present in graph neural networks. Because of their skills of learning via graph data and producing more accurate results, graph neural processing is now one of the most popular study areas in the fields of data science and machine learning. A graph neural network and an attention layer have been combined to create what is known as a graph attention network.

The graph neural networks do quite well when it comes to categorising nodes based on the graph-structured data. Because of the way that graph structure aggregates information, graph convolutional networks may be reducing the generalizability of data that is arranged in a graph, which is one of the numerous shortcomings that we may uncover while investigating many of the difficulties. The use of a graph attention network to such issues can modify the way information is aggregated, which is one of the benefits of doing so.

The Graph Attention Network, also known as GAT (Velickovic et al., 2018), is a design for graph neural networks that makes use of the attention mechanism to learn the weights that are associated with linked nodes. In contrast to GCN, which employs weights that have already been calculated for the neighbours of a node that correspond to the normalisation coefficients, BCN uses weights that are randomly generated. The aggregation process of GCN (Zhou et al., 2020) is altered as a result of GAT's ability to understand, via the attention mechanism, the strength of the link that exists between surrounding nodes.

Instead of computing that coefficient directly, as GCNs do, the key concept behind GAT is that it should be done implicitly instead.

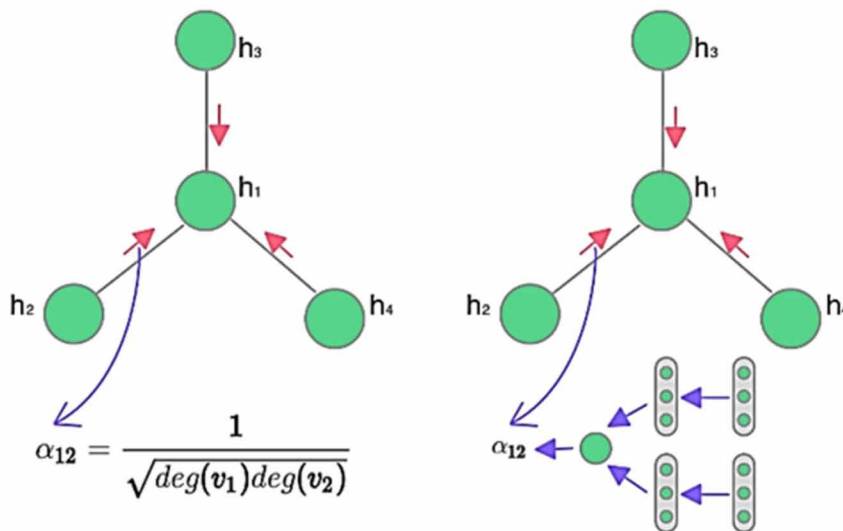
An operation that is statically normalised and convolutional can be provided by the attention, just as it is in GCN. As consideration is given to the network, the weights assigned to the more significant nodes during the neighbourhood aggregation process are increased.

Graph Attention Networks (GATs) have shown great promise in the field of graph representation learning, and there are several potential research directions that could further advance the state of the art (Verma, 2021):

1. **Incorporating Heterogeneous Graph Structures:** Most existing GAT models assume homogeneous graphs, where all nodes and edges have the same type. However, many real-world graphs are heterogeneous, with nodes and edges of different types. Future research could explore ways to extend GATs to heterogeneous graphs, allowing them to model more complex relationships between nodes.
2. **Handling Dynamic Graphs:** Many real-world graphs are dynamic, where nodes and edges are added or removed over time. Current GAT models are designed to work with static graphs, and it remains an open research question how to effectively model dynamic graphs.

3. **Scaling to Large Graphs:** GATs can become computationally expensive when applied to large graphs with millions of nodes and edges. Future research could explore ways to scale GATs to such large graphs, either by developing more efficient algorithms or by using parallel computing.
4. **Incorporating Graph Context Into Attention Mechanisms:** While GATs use attention mechanisms to weight the contributions of neighboring nodes, they do not explicitly consider the larger graph structure. Future research could explore ways to incorporate graph-level context into the attention mechanism, allowing GATs to better capture the overall structure of the graph.
5. **Transfer Learning Across Graphs:** GATs are typically trained on a single graph, but in many real-world scenarios, there may be multiple related graphs that share some common structure. Future research could explore ways to transfer knowledge learned from one graph to another, allowing GATs to more effectively generalize to new graphs.

Figure 1. Difference between standard GCN and GAT



## BACKGROUND

The graph attention network (Velickovic et al., 2018) is a combination of a graph neural network and an attention layer.

### Graph Neural Network

Graph neural networks are so-called because they are able to operate with information or data that is laid down in the form of a graph and are therefore referred to by that name. When it comes to modelling, graph neural networks use graph data, which can be thought of as the structural relationship that already existing between the items in the dataset. There is also the possibility of using graph data to explain the data (Scarselli et al., 2008).

## Fundamental Concepts in Graph Attention Networks

The data is stored in a graph-structured fashion, with the vertices and nodes of the graph serving as the storage locations for the information. For neural networks, this makes it very easy to interpret and learn the data points that are present in the graph or three-dimensional structure. In the data, the information and labels that are related with a classification problem can be correspondingly represented as nodes and vertices (Tran & Niedereée, 2018).

In this, we will explore how to design and carry out modelling using graph neural networks by developing and implementing them ourselves. Graph neural networks are a type of artificial neural network (Kumar & Thakur, 2017). The following items are examples of what can make up a simple graph's data:

1. **Node Features:** This element displays the total number of nodes and features that are contained inside an array. The dataset that we are utilising for this post contains information on papers that may be utilised as nodes, and the characteristics of the nodes are the word-presence binary vectors of each paper.
2. **Edges:** This is a sparse matrix of links between the nodes that represent the number of edges in both dimensions.
3. **Edge Weights:** This is a non-mandatory element that takes the form of an array. The number of edges, which may be thought of as a quantification between nodes, is represented by these values below the array. Let's check out the several ways we can make them.

## The Architecture of Graph Attention Network

In this part of the article, we will investigate the structure of a graph attention network, which we may utilise to construct one. In most cases, we have discovered that such networks maintain the layers in the network in a stacked manner. By gaining a grasp of the functions performed by the network's three primary levels, we may gain comprehension of the network's design.

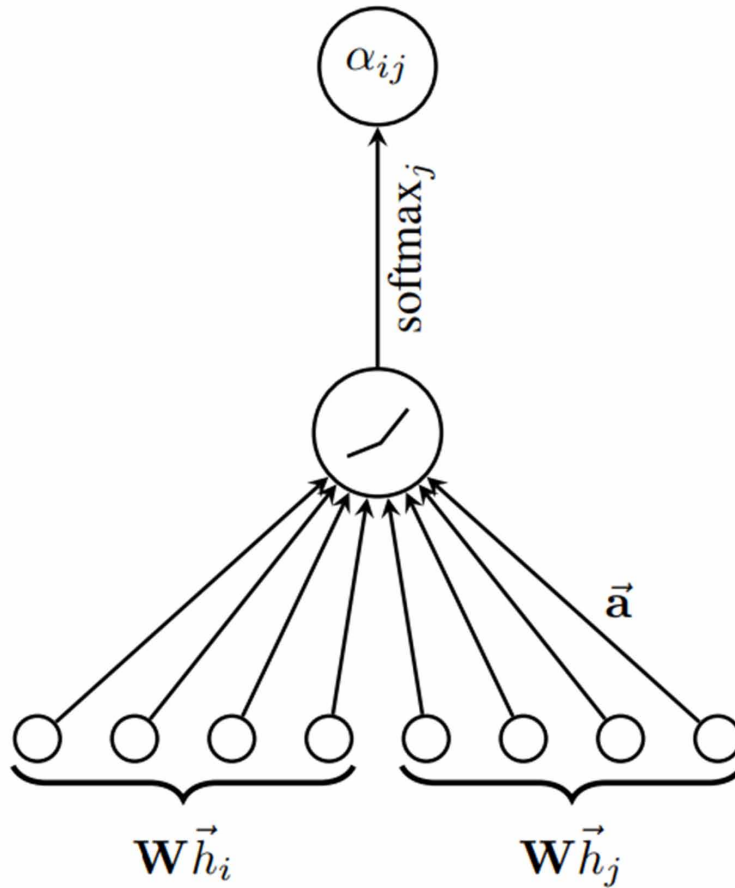
**Input Layer:** It is possible to construct the input layer such that it is composed of utilising a set of node features, and it should be able to produce a new set of node features as the output of the system. In addition to this, these layers may be able to convert the characteristics of the input nodes into linear features that can be learned.

The input to the layer is a set of node features,  $h = [h_1, h_2, \dots, h_N]$ ,  $h_i \in \mathbb{R}^F$ , where  $N$  is the number of nodes, and  $F$  is the number of features in each node.

The layer produces a new set of node features (of potentially different cardinality  $F'$ ),  $h' = [h'_1, h'_2, \dots, h'_N]$ ,  $h'_i \in \mathbb{R}^{F'}$ , as its output.

**Attention Layer:** When the features have been transformed, an attention layer may be added to the network. The operation of the attention layer can be parameterized by the output of the input layer using a weight matrix, and this can be done before or after the features have been transformed. We may give each node its own attention by first applying this weight matrix to each of the nodes in the network. In a purely mechanical sense, we may assume that our attention layer is a single-layer feed-forward neural network, and this will allow us to get a normalised attention coefficient (Zangari et al., 2021).

Figure 2. Representation of the attention layer applied to the GCN

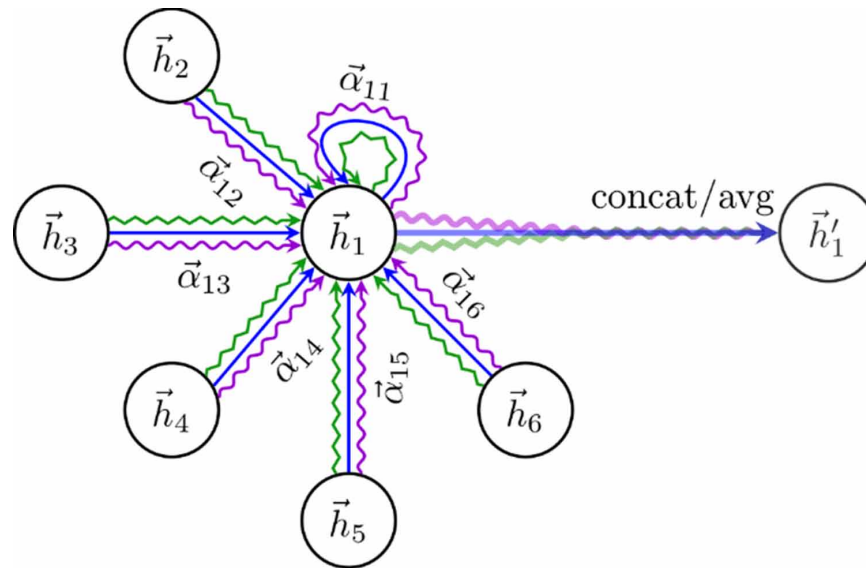


Where  $\alpha_{ij}$  is the Attention coefficient. Figure 2 is a representation of the attention layer applied to the GCN.

- **Output Layer:** Since we have the normalised attention coefficient, we can use it to compute the set of features that correspond to the coefficient, and then we can utilise those features as the final features that come from the network. In order to maintain control over the attention process, we may make use of multi-head attention. This allows for many types of independent attention to be applied in order to carry out transformations and concatenate output features.

Figure 3 is a depiction of the multi-head attention that was applied in order to stabilise the process of self-attention, which computes attention and concatenates aggregated data.

Figure 3. Multi-head attention



## IMPLEMENTING THE GRAPH NEURAL NETWORK

In order to construct a network that is compatible with the graph data. In order to accomplish this, we need to create a layer that is capable of operating on the graph data (Wu, et al., 2020).

### Graph Layer

In the next section of the article, we are going to discuss the duties that are necessary for a simple graph layer to fulfil in order for it to be operational. Instead, we are going to talk about the work at hand and the functionality that is provided by the layer. This is due to the fact that the amount of code is so vast, and we are not going to push it here. This location contains the whole of the implementation. Let's get started with the very first assignment.

1. The purpose of this task is to prepare the input nodes that will be used in the feed-forward neural network that we have created. This network will generate a message in order to facilitate the processing of input node representations.
2. The following step is utilising the edge weights to do an aggregate of the messages that have been sent from the node to its neighbouring node. In this particular application of mathematics, permutation invariant pooling techniques are being utilised. These procedures provide a single aggregated message for each and every node in the network.
3. The development of a new state for the node representations is the next job that has to be completed. At this phase of the project, we will be fusing the representation of the nodes with the collected messages. Generally speaking, if the combination is of the GRU type, then the node representations and aggregated messages may be stacked to generate a sequence, which can then be processed by a GRU layer.

In order to carry out these responsibilities, we designed a graph convolutional layer in the form of a Keras layer that is comprised of functions that prepare, aggregate, and update data.

- **Graph Neural Node Classifier**

When we have completed the layer, we will go on to creating a network neural node classifier. The following methodologies may be applied to this classifier:

The process of generating the node representation begins with the preprocessing of the node characteristics.

1. Implementing graph layers in the design.
2. The post-processing of the node representation, which results in the generation of the final node representations.
3. Producing the predictions based on the node representation by using a softmax layer.

## **GRAPH ATTENTIONAL LAYER**

The conventional neural networks do not have the capacity to retain and process information that is both lengthy and extensive. The attention layer of a neural network can assist the network in learning to remember extensive data sequences (Velickovic et al., 2017; Yadav et al., 2023). We are able to create a neural network that is capable of remembering lengthy sequences of information thanks to the attention layer, which is a layer in the neural network.

If we give the learning model a massive dataset to work with, there is a chance that it will disregard certain key aspects of the data. If the dataset is large enough, however, the models should be able to handle it. It is crucial to pay attention to the key facts, and doing so can lead to improvements in the performance of the model. This may be accomplished by including a supplementary attention component in the various models. This feature may be simply included into neural networks that have been constructed using many layers by employing one of those layers (Kumar et al., 2022; Lalotra et al., 2022). Neural Network architecture makes use of the attention layer to help increase the performance of the network.

It is now plainly evident to us that traditional neural networks are not able to retain and analyse lengthy and extensive quantities of information in the bulk of these cases. This realisation came as a complete and utter surprise to us. Let's talk about seq2seq models, which are a sort of neural network and are frequently employed for modelling language. These models are quite popular. It is common knowledge that these models are successful in what they set out to do. On a more technical level, we may say that the seq2seq models are intended to conduct the translation of sequential information into sequential information, and both kinds of information can be of arbitrary form. This is because both kinds of information are sequential. This is due to the fact that both forms of information are presented in sequential order. When we discuss the tasks that are performed by the encoder, we can state that it converts the sequential information into an embedding, which is another name for a context vector that has a predetermined number of elements. The fact that the network is unable to recall the longer phrases is a significant drawback of the context vector design that is fixed in length. After digesting the entire series of information, we can run into the issue of forgetting the initial portion of the sequence, even if

## ***Fundamental Concepts in Graph Attention Networks***

we might regard it to be the sentence. This is a common problem. We are therefore able to rectify the situation by introducing an appropriate attention mechanism to the network.

### **Attention Mechanism**

It is possible to refer to a system that supports a neural network in memorising extensive sequences of information or data as the attention mechanism, and usually speaking, this mechanism is utilised in the process of neural machine translation (NMT), a system for focussing attention that creates a shortcut linking the whole input to the context vector and enables the weights of the connection between the two to be changed in a manner that is distinct for each output and can be customised as needed. It is possible to alleviate some of the problems that are associated with forgetting lengthy sequences as a result of the relationship that exists between the input and the context vector (Luong et al., 2015). The context vector is able to have access to the entirety of the input as a result of this relationship, which makes it possible to do so.

Depending on how a network's attention mechanism works, a context vector may contain the following information:

1. Encoder hidden states
2. Decoder hidden states
3. Alignment between source and target

We can categorize the attention mechanism into the following ways:

1. Self-Attention Attention Mechanism
2. Global/Soft Attention Mechanism
3. Local/Hard Attention Mechanism

### **Self-Attention Mechanism**

When an attention mechanism is applied to a network in order for it to be able to relate to different positions within a single sequence and be able to compute the representation of the same sequence, this type of attention may be referred to as self-attention or intra-attention, depending on the context. Inside an LSTM network, we are able to see the functions of self-attention processes.

### **Soft/Global Attention Mechanism**

Since the attention that is being applied in the network is for the goal of learning, every patch or sequence of the data may be regarded a Soft or global attention mechanism. Two domains, namely image processing and language processing, have the potential to gain advantages from this concentration of attention.



## Local/Hard Attention Mechanism

It is possible to refer to the attention mechanism as the Local/Hard attention mechanism when it is applied to specific parts of the data, such as sequences or patches. This particular form of attention is focused primarily on the network that is responsible for the image processing task.

An attention layer is a method that is used to aid in the extraction of only the information that is of the utmost significance from lengthy or comprehensive data sets. A graph neural network is the strategy that should be utilised when dealing with data that has extensive structural information since it is the most effective way. When these two things are connected to one another, a new entity is produced, which may be referred to as a graph attention network.

The attention mechanism in GAT involves the following steps:

For each node in the graph, a linear transformation is applied to its feature vector to obtain a query vector.

Similarly, a linear transformation is applied to the feature vectors of all its neighbors to obtain a set of key vectors.

The query vector is multiplied element-wise with each key vector, and the resulting vectors are passed through a softmax function to obtain the attention coefficients.

The attention coefficients are used to compute a weighted sum of the neighbor feature vectors, which are then concatenated with the node's own feature vector.

The concatenated vector is passed through a feedforward neural network to obtain the updated representation of the node.

## Combination of GNN and Attention Layer

An approach that assists in the extraction of only the most significant information from lengthy or extensive data sets is known as an attention layer. When dealing with data that consists of lengthy structural information, a graph neural network is the superior method to use. The resulting object, which may be referred to as a graph attention network, is formed when these two items are connected together.

## APPLICATIONS OF GAT

Graph Attention Networks (GATs) are a type of neural network that can be applied to problems involving graph-structured data. They were first introduced in 2018 by Veličković et al. and have since gained popularity in various domains. Here are some applications of GATs (zhou et al., 2020; Sharma et al., 2022):

**Social Network Analysis:** GATs can be used to analyze social networks, where each node represents a person and edges represent relationships between them. GATs can identify important nodes and communities within the network (Bai et al., 2020; Shaik et al., 2023).

**Recommender Systems:** GATs can be used to build personalized recommender systems. Nodes in the graph can represent items, users, or both, and the edges can represent ratings, purchases, or other interactions between them. GATs can learn the relationships between nodes and predict which items a user is likely to be interested in.

## **Fundamental Concepts in Graph Attention Networks**

**Natural Language Processing (NLP):** GATs can be used to model sentence or document-level representations. Each node in the graph can represent a word or phrase, and edges can represent syntactic or semantic relationships between them. GATs can learn to capture the meaning of a sentence or document by attending to important words or phrases.

**Drug Discovery:** GATs can be used to model molecular structures and predict their properties. Each node can represent an atom, and edges can represent bonds between atoms. GATs can learn to predict the activity of a molecule by attending to important atoms and their relationships.

**Computer Vision:** GATs can be used for tasks such as image segmentation or object detection. Each node can represent a pixel or a region of interest, and edges can represent spatial relationships between them. GATs can learn to attend to important regions of an image to perform the task at hand (Wang et al., 2020).

**Pandemic Forecasting:** Aims to predict the spread of a disease within a country in terms of time and space.

Overall, GATs are a powerful tool for modeling relationships between entities in a graph, and can be applied to a wide range of domains.

## **CONCLUSION**

Graph Attention Networks (GATs) are a type of neural network designed for processing graph-structured data. Unlike traditional graph convolutional networks, which apply the same transformation to all nodes in the graph, GATs allow each node to learn a different linear transformation. This is accomplished by using an attention mechanism, which assigns a weight to each neighbor of a node based on its importance to that node. The GAT architecture consists of several layers of graph convolutions, each of which applies the attention mechanism to update the node features. The final layer produces the output of the network, which can be used for tasks such as node classification or graph classification. GATs have been shown to outperform previous state-of-the-art methods on a variety of graph-based tasks, including citation network classification, protein function prediction, and traffic prediction. They are also highly interpretable, as the attention weights can be used to identify which neighbours are most important for a given node.

## **FUTURE RESEARCH DIRECTIONS**

Future research directions for GATs include incorporating heterogeneous graph structures, handling dynamic graphs, scaling to large graphs, incorporating graph context into attention mechanisms, and exploring transfer learning across graphs.

## **REFERENCES**

Bai, T., Zhang, Y., Wu, B., & Nie, J. Y. (2020). Temporal graph neural networks for social recommendation. In *2020 IEEE International Conference on Big Data (Big Data)* (pp. 898-903). IEEE. 10.1109/BigData50022.2020.9378444

- Kumar, V., Lalotra, G. S., & Kumar, R. K. (2022). Improving performance of classifiers for diagnosis of critical diseases to prevent COVID risk. *Computers & Electrical Engineering*, *102*, 108236. doi:10.1016/j.compeleceng.2022.108236 PMID:35915590
- Kumar, V., & Thakur, R. S. (2017). Jaccard similarity based mining for high utility webpage sets from weblog database. *Int J Intell Eng Syst*, *10*(6), 211–220. doi:10.22266/ijies2017.1231.23
- Lalotra, G. S., Kumar, V., Bhatt, A., Chen, T., & Mahmud, M. (2022). iReTADS: An intelligent real-time anomaly detection system for cloud communications using temporal data summarization and neural network. *Security and Communication Networks*, *2022*, 1–15. doi:10.1155/2022/9149164
- Luong, M. T., Pham, H., & Manning, C. D. (2015). *Effective approaches to attention-based neural machine translation*. doi:10.18653/v1/D15-1166
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80. doi:10.1109/TNN.2008.2005605 PMID:19068426
- Shaik, C. M., Penumaka, N. M., Abbireddy, S. K., Kumar, V., & Aravinth, S. S. (2023, February). Bi-LSTM and Conventional Classifiers for Email Spam Filtering. In *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)* (pp. 1350-1355). IEEE. 10.1109/ICAIS56108.2023.10073776
- Sharma, R. M., Agrawal, C., Kumar, V., & Mulatu, A. N. (2022). lou, V., & Mulatu, A. N. (2022). CFS-BFDroid: Android Malware Detection Using CFS+ Best First Search-Based Feature Selection. *Mobile Information Systems*, *2022*, 1–15. doi:10.1155/2022/6425583
- Tran, N. K., & Niedereée, C. (2018). Multihop attention networks for question answer matching. In *The 41st international ACM SIGIR conference on research & development in information retrieval* (pp. 325-334). 10.1145/3209978.3210009
- Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *Stat*, *1050*, 20.
- Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., ... Yu, J. (2020). Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020* (pp. 1082-1092). 10.1145/3366423.3380186
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24. doi:10.1109/TNNLS.2020.2978386 PMID:32217482
- Yadav, A., Kumar, V., Joshi, D., Rajput, D. S., Mishra, H., & Paruti, B. S. (2023). Hybrid Artificial Intelligence-Based Models for Prediction of Death Rate in India Due to COVID-19 Transmission. *International Journal of Reliable and Quality E-Healthcare*, *12*(2), 1–15. doi:10.4018/IJRQEH.320480
- Yugesh Verma. (2021). *A beginners guide to using attention layer in neural networks*. <https://analytic-sindiamag.com/a-beginners-guide-to-using-attention-layer-in-neural-networks/>

Zangari, L., Interdonato, R., Calió, A., & Tagarelli, A. (2021). Graph convolutional and attention models for entity classification in multilayer networks. *Applied Network Science*, 6(1), 87. doi:10.100741109-021-00420-4

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1, 57–81.

## **KEY TERMS AND DEFINITIONS**

**Attention Layer:** In a Graph Attention Network (GAT), the attention layer computes a weighted sum of the neighboring node features to update the representation of each node in the graph. The attention mechanism allows the model to learn to assign different weights to the neighboring nodes based on their relevance to the current node and the task at hand.

**Graph Attention Network:** GATs leverage the attention mechanism to compute a weighted sum of the neighboring nodes' features, enabling them to learn a representation of each node by aggregating information from its neighbors. The attention mechanism allows the model to attend to different parts of the input graph, giving more weight to more relevant nodes for a given task.