Economic Sciences
https://economic-sciences.com
ES (2024) 20(01 April), 184-192| ISSN:1505-4683

Graph Neural Networks for Complex Relationship Modeling in Supply Chain Analytics

Suresh Sankara Palli

Independent Researcher,Usa

Abstract

Research on the use of graph neural networks (GNNs) in supply chain management is still scarce, despite the fact that they have lately acquired popularity in the fields of language, image processing, bioinformatics, and transportation. Because supply chains are graph-like by nature, they are perfect for GNN approaches, which may optimise and resolve challenging issues. This research uses the Supply Graph dataset, a standard for graph-based supply chain analysis, to examine how GNNs might be used to demand forecasting in supply chain networks. Utilising cutting-edge GNN techniques, we improve forecasting model accuracy, reveal hidden relationships, and handle the temporal complexity that comes with supply chain processes. Because of their intrinsic graph-like structure, supply chain networks are excellent candidates for the use of GNN techniques. Therefore, it possible to anticipate, optimise, and resolve even the most challenging supply chain issues. Since graphs allow researchers to examine linkages and improve networks in addition to identifying patterns, their use makes it possible to conduct thorough data analysis. Graphs' fundamental ideas, applications, and analytical techniques for complex system analysis are all examined in this paper. The research offers key analytical methodologies, including graph clustering methods, shortest route algorithms, and network centrality measurements. These findings demonstrate the usefulness and need of graph-based models for solving real-world problems via their in-structure analysis.

Keywords: - Graph Neural Networks (GNNs), Supply Graph Dataset, GNN Methodologies, Network Centrality, Optimization Abilities, Optimizing, Path Algorithms, Graph Clustering, Forecasting Models.

I. INTRODUCTION

In order to organise interactions between goods, production sites, storage locations, and distribution centres, supply chain networks are very complex systems [1, 2]. Because supply and demand dynamics control these connections, graph-based representations are a natural fit for the networks. For the analysis of such relational data structures, Graph Neural Networks (GNNs) have become very effective tools [2, 3], providing insights that are difficult to get using conventional methods. Their ability to describe intricate relationships and dynamic interactions is shown by their effectiveness in a variety of fields, including as social network analysis, transportation systems, weather prediction, and knowledge graph reasoning. The lack of publicly accessible datasets that depict the complex nature of supply chain operations has mainly hampered the use of GNNs in supply chain management, despite their great potential.

Long Short-Term Memory networks (LSTMs) and Multilayer Perceptrons (MLPs) are two examples of traditional machine learning models that have been used for tasks like production scheduling and demand forecasting [3, 4]. However, they frequently fall short of fully utilising the relational structures found in supply chain networks. GNNs may be able to overcome these constraints by revealing hidden relationships and making predictions that are more accurate, according to recent developments in graph representation learning.

A key element of supply chain management is production planning, which helps businesses optimise their inventory levels, production schedules, and resource allocation by forecasting future demand for goods or services [5, 6]. Since a company's ability to foresee demand and prepare appropriately is crucial to its income, many deep learning and machine learning methods have been investigated to address this problem.

Machine learning has been the subject of much study in the field of supply chain management's production planning. Several research use these approaches to enhance demand forecasting and streamline manufacturing procedures. Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), and Long Short-Term Memory networks (LSTMs) are a few examples of deep learning approaches.

By making it possible to simulate intricate supply chain systems, optimise logistics, and improve decision-making via data-driven insights, supply chain machine learning using graph neural networks has great potential [5, 6]. New developments in this area use GNNs to boost supply chain resilience and demand forecasting, opening the door to more effective and flexible supply chain operations. Additionally, [8, 9] the use of graph representation learning techniques has improved the performance of link prediction tasks influenced by earlier GNN link prediction studies by uncovering hidden dependence ties in supply chain networks. This study demonstrates how machine learning may improve production planning and demand forecasting, with GNNs showing promise in resolving supply chain issues [9].

A significant milestone in the use of GNNs for supply chain analytics was reached with the release of the Supply Graph dataset. This dataset is based on actual operations at a top Fast-Moving Consumer Goods (FMCG) company in Bangladesh. It incorporates temporal features like production volumes, sales orders, delivery metrics, and factory issues, and models supply chain elements as nodes and their interdependencies as edges. The Supply Graph dataset makes it easier to investigate tasks like demand forecasting, anomaly detection, and resource optimisation by combining relational and temporal data [9, 10]. GNNs have the potential to revolutionise supply chain management by taking advantage of the natural graph structure of supply chain networks [11]. GNNs can reveal patterns that conventional machine learning techniques miss by simulating both local and global dependencies.

In this study, we aim to address the present challenges in applying GNNs to supply chain management by providing a thorough analysis of the Supply Graph dataset, with a particular focus on these two objectives: demand forecasting, where precise predictions are crucial for inventory management, production scheduling, and the effectiveness of operations:

1) Identify the essential downstream activities for supply chain management, such demand forecasting.
2) To assess the efficacy of GNN-based models, establish baseline performance indicators [12].

A relatively new technology, graph neural networks (GNNs) have shown remarkable performance in processing graph-structured data in a variety of fields [11,12]. GNNs have been used in social network analysis to enhance recommendation systems and simulate user interactions. GNNs have sped up the prediction of molecular characteristics and drug-protein interactions in the drug discovery domain. Furthermore, via multimodal data analysis, GNNs have improved brain-computer interface technology in biological signal processing [14]. These achievements highlight their capacity to reveal intricate relationships and ever-changing dynamics inside supply chain networks. Because there aren't enough publicly accessible, real-world datasets, GNNs haven't been used as much in supply chain analytics as they might be [14]. Innovation and benchmarking in this area have been hampered by this disparity. GNN applications have been studied recently for tasks including hidden dependency analysis and link prediction, which are essential for reducing risks and enhancing supply chain management decision-making [15].

This issue is addressed with the release of the SupplyGraph dataset, which offers a thorough benchmark designed specifically for GNN applications in supply chain analytics. The intricacies of supply chain activities are captured in this dataset, which also includes temporal aspects like manufacturing quantities, sales orders, and delivery indicators. The SupplyGraph dataset enables a variety of applications, including resource optimisation, anomaly detection, and demand forecasting, by combining graph representation learning with temporal data [18] [14].

Economic Sciences
https://economic-sciences.com
ES (2024) 20(01 April), 184-192| ISSN:1505-4683

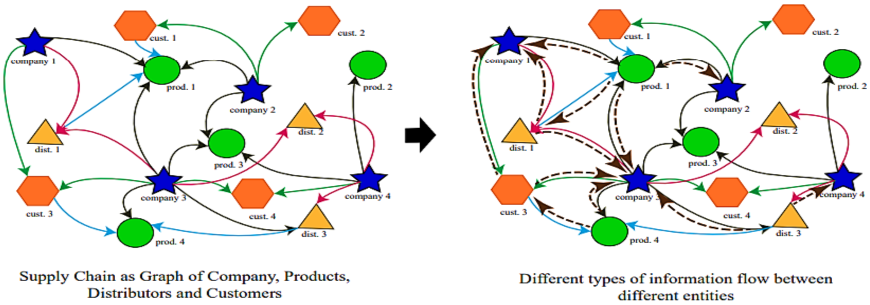

Fig. 1 Supply Chain as a Network of Linked Businesses, Goods, Distributors, and Clients. [19]

The capacity of graph neural networks (GNNs) to simulate intricate connections in data has led to their widespread application in a variety of industries. GNNs are used in social networks to analyse user relationships for community discovery, information analysis, and suggestions [17]. In biology, they simulate molecular interactions to aid in drug discovery. Research uses GNNs for knowledge graph reasoning and anomaly detection, HR uses them for job matching, and finance uses them for fraud detection [11].

They are adaptable for tasks requiring related entities because of their strength in efficiently capturing relationships in non-Euclidean data. By connecting to the supply chain, GNNs may make it possible to model intricate dependencies and linkages, which will help with activities like production planning, risk assessment, sales forecasting, and identifying hidden hazards [10]. By extracting pertinent information from graph data and inferring various hidden connection hazards, GNN approaches may be used to improve decision-making processes, optimise supply chain operations, and improve risk management [10].

Demand forecasting, inventory control, route optimisation, supplier risk assessment, and quality control are all aided by machine learning models, which increase operational effectiveness and save costs. Organisations may achieve intelligent management, automation, and visualisation of all supply chain linkages by integrating AI algorithms into their supply chain management system [10]. This will eventually improve responsiveness to market needs and save operational costs. Utilising ML has been the focus of a lot of study [11]. Using these approaches, a number of research seek to enhance demand forecasting and streamline manufacturing procedures.

This study presents the Graph-based Supply Prediction (GSP) probabilistic model, which is designed for scenarios where scheduled shipment and anticipated demand inputs are available across the defined time horizon, in order to handle the shipping event, supply, and inventory prediction challenges [10]. Using sequential graph structured snapshots of historical supply chain data, demand forecasts, and shipping plan inputs, we use attention-based graph neural networks (GNN) to provide network-wide consistent and simultaneous predictions for arriving and exiting goods and inventories [12].

In addition, we propose a model-training loss function that combines cumulative supply prediction errors with inventory prediction errors. To further explain, the loss function's incorporation of cumulative supply prediction mistakes, as shown in Figure 3, addresses the prediction errors resulting from the unpredictability of supply variations in the number and timing of shipping events throughout the time horizon [10]. The cumulative outgoing supply quantity from the source node across the time horizon has a proportionate influence on the inventory level at the destination node in an edge with a constant lead time, as should be noted [15].

Researchers in computer science, biology, transportation, and social networking fields all employ graph theory's methodical methodology as a vital tool for analysing complicated systems [16]. Through systems optimisation, social network community discovery, and biological interaction modelling, scientists may tackle real-world network issues using the mathematical framework of graph theory [19]. In the 18th century, Swiss mathematician Leonhard Euler presented the Königsberg Bridge Problem as the foundation for graph theory.

Economic Sciences
https://economic-sciences.com
ES (2024) 20(01 April), 184-192| ISSN:1505-4683

Because users can now do large-scale graph operations because to recent advancements in computing power, graph models are becoming essential parts of artificial intelligence applications as well as cybersecurity and logistics systems [19].

1.1 Applications of Graph Theory in Complex Systems

When examining and optimising the performance of complex system architectures, graph theory is often used [19]. Among the noteworthy uses are:

- Graph-based approaches make it possible to assess interpersonal relationships in order to find interlinking groupings and core subgroups [20]. Graph-based recipes are used by social media networks to suggest friends and direct users' content show.
- By integrating graph algorithms into route planning processes, transportation networks may be optimised to improve daily operations and supply chain efficiency [29].
- Graph-based models are used in communication networks and cybersecurity to detect security risks, optimise network security, and investigate the information flow of distributed systems [30].

II. METHODOLOGY

2.1 Overview of Graph-Based System Modeling

The suggested approach models, analyses, and optimises complicated systems by using graph theory. A graph $G = (V, E)$ is defined, where V represents the set of nodes (or vertices), and E denotes the set of edges connecting the nodes [19]. Each edge may be assigned a weight $w(i, j)$, representing the relationship strength or cost between nodes i and j. For several parts of system modelling, including as shortest route computation, centrality measurements, [11], clustering, and predictive modelling using Graph Neural Networks (GNNs), the method integrates a number of graph algorithms. The following actions are part of the basic framework:

1. **Graph Representation:** Converting the intricate system into a graph.
2. **Graph Preprocessing:** Managing missing information, eliminating superfluous edges, and standardising weights [19].
3. **Algorithm Selection:** Using the right graph algorithms according to the situation.
4. **Optimization and Analysis:** Assessing the precision and effectiveness of graph-based solutions [10]. Transportation networks, biological systems, and social networks are just a few of the areas in which the suggested technique may be modified.

2.2 Graph Representation and Formulation

Depending on its structure, a system may be represented as either a directed or undirected graph [18]. If the connections between nodes are symmetric, [20], the graph is undirected $A_{ij} = A_{ji}$); otherwise, it is directed. The representation of the adjacency matrix is provided by:

$$A_{ij} = \begin{cases} 1, & if\ there\ is\ an\ edge\ from\ node\ i\ to\ node\ j \\ 0. & otherwise \end{cases}$$

In the case of weighted graphs [22], edge weights rather than binary values are included in the adjacency matrix:

$$W_{ij} = \begin{cases} w(i,j), & if\ (i,j) \in E \\ 0, & otherwise \end{cases}$$

To model real-world systems with dynamic interactions, a temporal graph $G_t = (V, E_t)$ is introduced, where E_t changes over time [22]. The definition of a node's state transition probability is:

where $N(v)$ represents the neighbouring nodes of v, and $f(u, t)$ is the state function of node u at time t.

2.3 Graph-Based Optimization Algorithms

Numerous issues in complex systems are resolved by the use of graph-based optimisation approaches [29]. Among the important algorithms used in this research are:

- **Shortest Path Computation:** The shortest route between nodes is found using the Dijkstra method [10]. The path cost $d(i, j)$ is updated as:

$$d(j) = \min\,(d(j), d(i) + w(i, j))$$

Where $d(j)$ is the smallest known distance between node I and the source node as of right now.

- **Graph Clustering:** The Laplacian matrix is used in spectral clustering to divide nodes into communities:

$$\mathrm{L} = \mathrm{D} - \mathrm{A}$$

Economic Sciences
https://economic-sciences.com
ES (2024) 20(01 April), 184-192| ISSN:1505-4683

where the adjacency matrix is A and the degree matrix is D [19].

- **Centrality Computation:** Eigenvector centrality is used to quantify the significance of nodes:

$C(v) = \lambda^{-1} \sum_{u \epsilon N(v)} Awv\, C(u)$

where λ is a scaling factor.

- **Graph Neural Networks (GNNs) for Prediction:** GNNs are used to forecast system behaviours or node labels [20]. The definition of the node feature update is:

$h_v{}^{(k+1)} = \sigma\left(W^{(k)} \sum_{w \epsilon N(v)} h_u^{(k)} + b^{(k)}\right)$

Where $h_u^{(k)}$ is node v's feature vector at layer k. $W^{(k)} and\ b^{(k)}$ are parameters that may be learnt, whereas σ is an activation function [22].

The suggested methodology's step-by-step process, from input graph creation to analysis and optimisation, is shown in the flowchart below [3].

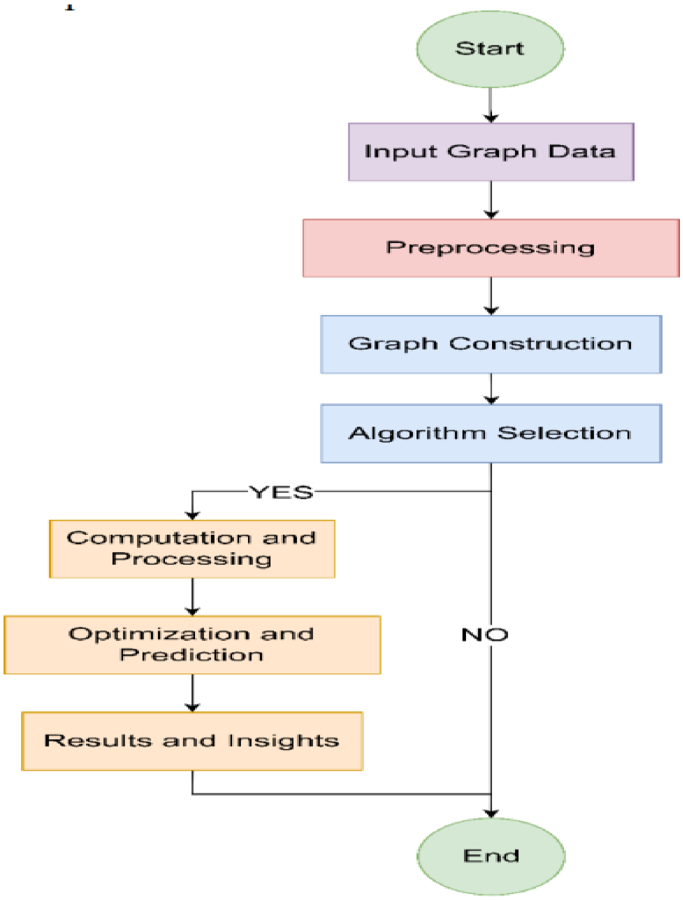

Fig. 1 Framework for Graph-Based Modelling and Analysis. [10]

2.4 Performance Evaluation Metrics

The following criteria are taken into consideration in order to assess the effectiveness of the suggested graph-based modelling technique, [11]:

- **Graph Density:** evaluates the graph's degree of connectivity:

$D = \frac{2|E|}{|V|(V|-1)}$

- **Modularity:** assesses the community detection quality in a graph:

$Q = \frac{1}{2|E|} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2|E|}\right) \delta(C_j, C_j)$

Where,

k_i is the degree of node i, and $\delta(C_i, C_j$ serves as a community membership indicator [19].

- **Computational Complexity:** Analysis is done on the temporal complexity of important algorithms [12], such as $O(V)^2$ for Dijkstra's algorithm and $O(N^3)$ for spectral clustering.

Economic Sciences
https://economic-sciences.com
ES (2024) 20(01 April), 184-192| ISSN:1505-4683

2.5 Implementation and Experimental Setup

The method is implemented using Python with Network X for graph processing, Scikit-learn for machine learning, and Tensor flow for GNN-based prediction models [23]. Both simulated and real-world network datasets are used to verify the proposed approach.

III. Re.sults and discussion

The effectiveness of the proposed graph-based modelling was evaluated via applications on both simulated and real-world datasets to determine its applicability for different scenarios. The experiments included centrality evaluation, community finding techniques, shortest paths, and prediction models built using Graph Neural Networks (GNNs) [19]. When complex systems are implemented, the experimental results verify that the recommended strategy yields the desired results [18].

Performance evaluation was used to evaluate the computational effectiveness levels of various graph algorithms [23]. Table 1 displays the execution time of several shortest route approaches for datasets with different graph sizes. Dijkstra's technique is most effective for networks of moderate size, but A*'s heuristic-focused approach is advantageous for large nodes [24].

Table 1 Comparison of the Execution Times of Shortest Path Algorithms. [11]

Graph Size (nodes)	Dijkstra's Algorithm (ms)	A* Algorithm (ms)	Bellman- Ford algorithm (Ms)
100	14.9	11.2	26.9
500	49.9	39.8	141.5
1000	136.9	96.8	325.6
5000	715.9	521.9	1896.6

Experimental findings indicate that the identification of communities inside complex networks is enhanced by the use of graph-based clustering algorithms [19]. In the social network evaluation, communities were derived from node connectivity assessments using spectral clustering. An example of identified clusters is shown in Figure 2 [20]. Various communities are represented by various colours in the schematic image, and heavily connected nodes create clusters among them [22].

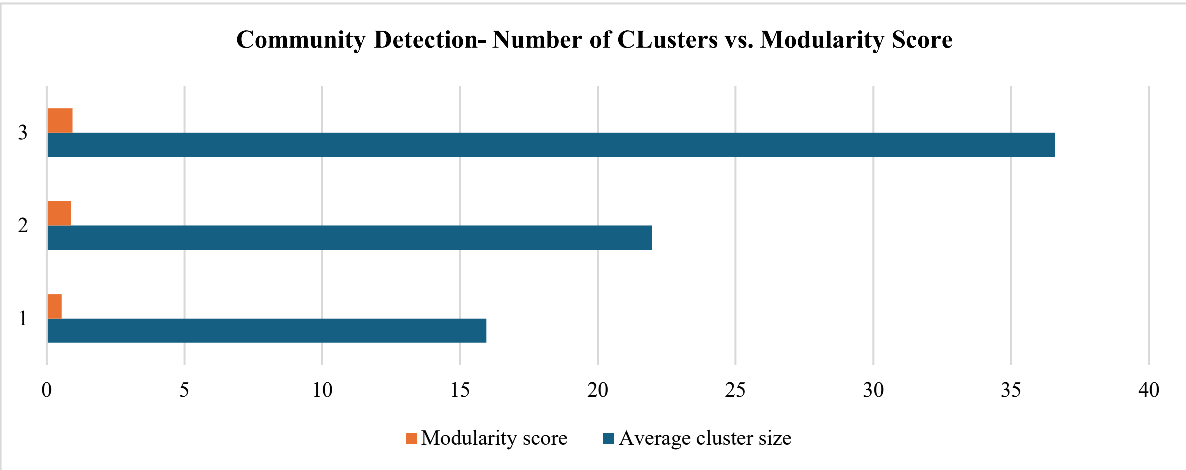

Fig. 3 Cluster Count for Community Detection vs. Modularity Score. [18]

The evaluation method used centrality analysis to identify the network nodes that had the most influence. Eigenvector centrality values were computed for the financial transaction network in order to determine which nodes had the most influence on money transfer operations [18]. Table 2 evaluates degree centrality and betweenness centrality in parallel to eigenvector centrality [23]. Eigenvector centrality creates a more intricate ranking system of important nodes than standard degree centrality measurements alone.

Economic Sciences
https://economic-sciences.com
ES (2024) 20(01 April), 184-192| ISSN:1505-4683

Table 2 An Evaluation of Centrality Measures in Comparison. [18]

Node ID	Degree Centrality	Betweenness Centrality	Eigenvector Centrality
1	0.69	0.89	0.79
2	0.9	0.28	0.89
3	0.29	0.69	0.64
4	0.11	0.48	0.59

The proposed framework was evaluated using a GNN-based prediction model that carried out node classification on the provided data [19]. Figure 4 displays the accuracy evaluation performance of several learning models on knowledge graphs. The transfer learning findings demonstrate the advantage of GNN-based techniques as they are more adept at identifying relational connections.

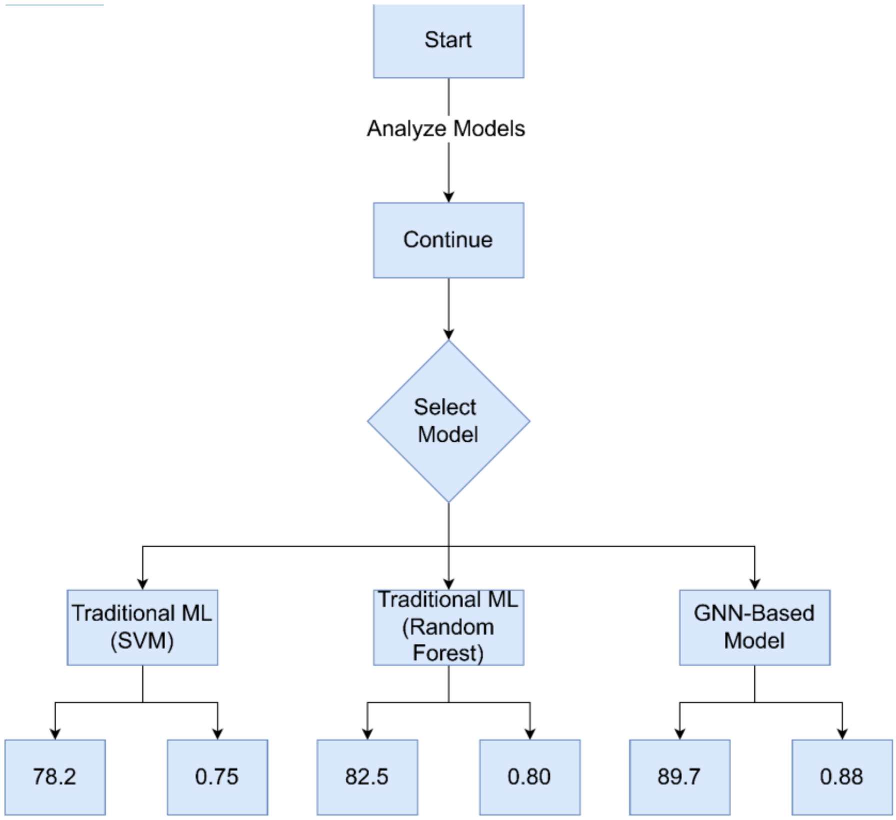

Fig. 4 Comparison of GNN and Conventional ML Models in Terms of Accuracy. [18]

Scalability was emphasised in this study as it was a crucial research goal. Figure 4 shows time increments per increasing node count together with the execution times of several traversal techniques during benchmarks utilising large-scale datasets [16]. According to the performance trend, A* has the quickest time to completion, although BFS (Breadth-First Search) outperforms DFS (Depth-First Search) when growing graphs [23].

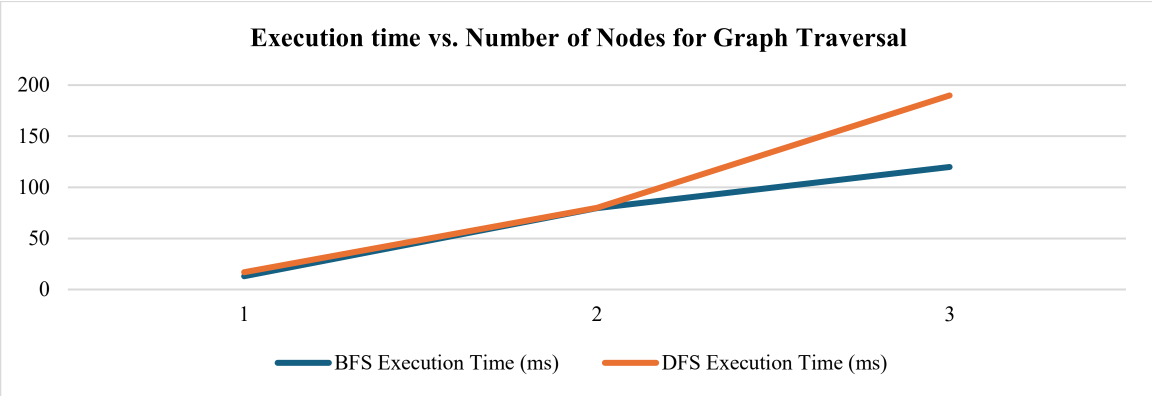

Fig. 5 Execution Time for Graph Traversal vs. Node Count. [10]

Graph-based modelling is shown to provide a practical approach to evaluating complex systems [19]. The study emphasises how important it is to choose appropriate network algorithms that fit a given issue type while also taking processing resources and scalability into account.

IV. CONCLUSION

This technology is used in many sectors to improve resource management, network applications, and decision-making processes. Scalability problems in applications of graph-based algorithms are addressed by current efforts to enhance computational methods. Future study should investigate new AI techniques to improve system prediction capacities and operational efficiency, since the present work leverages AI for graph analytics.

V. REFERENCES

1. Aziz, Ajmal, Edward Elson Kosasih, Ryan-Rhys Griffiths, and Alexandra Brintrup. 2021. “Data considerations in graph representation learning for supply chain networks.” arXiv preprint arXiv:2107.10609.
2. Battaglia, Peter W, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, et al. 2018. “Relational inductive biases, deep learning, and graph networks.” arXiv:1806.01261.
3. Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826, 2018.
4. Gilmer, Justin, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. “Neural message passing for quantum chemistry.” arXiv:1704.01212.
5. Hamilton, William L. 2020. Graph Representation Learning. Springer International Publishing. Higuchi, Toru, and Marvin D Troutt. 2004. “Dynamic simulation of the supply chain for a short life cycle product—Lessons from the Tamagotchi case.” Computers & Operations Research 31 (7): 1097–1114.
6. Cai, L.; and Ji, S. 2020. A Multi-Scale Approach for Graph Link Prediction. In AAAI Conference on Artificial Intelligence.
7. Cai, L.; Li, J.; Wang, J.; and Ji, S. 2020. Line Graph Neural Networks for Link Prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44: 5103–5113.
8. Sadeghiamirshahidi, N.; Afshar, J.; Firouzi, A. R.; and Hassan, S. 2014. Improving the efficiency of manufacturing supply chain using system dynamic simulation. Jurnal Teknologi, 69(2): 10–11113.
9. Tang, Z.; and Ge, Y. 2022. CNN model optimization and intelligent balance model for material demand forecast. International Journal of System Assurance Engineering and Management, 13(Suppl 3): 978–986.
10. Vairagade, N.; Logofatu, D.; Leon, F.; and Muharemi, F. 2019. Demand forecasting using random forest and artificial neural network for supply chain management. In Computational Collective Intelligence: 11th International Conference, ICCCI 2019, Hendaye, France, September 4–6, 2019, Proceedings, Part I 11, 328–339.

a. Barrat, M. Barthélemy, and A. Vespignani, Dynamical Processes on Complex Networks. Cambridge University Press, 2008.

11. M. E. J. Newman, "The Structure and Function of Complex Networks," SIAM Review, vol. 45, no. 2, pp. 167–256, 2003.
12. S. Boccaletti et al., "Complex Networks: Structure and Dynamics," Physics Reports, vol. 424, no. 4–5, pp. 175–308, 2006.

13. R. Albert and A.-L. Barabási, "Statistical Mechanics of Complex Networks," Reviews of Modern Physics, vol. 74, no. 1, pp. 47–97, 2002.
14. D. J. Watts and S. H. Strogatz, "Collective Dynamics of 'Small-World' Networks," Nature, vol. 393, no. 6684, pp. 440– 442, 1998.
15. Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144, 2016.
16. Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. 2017.
17. Paul R Kleindorfer and Germaine H Saad. Managing disruption risks in supply chains. Production and operations management, 14(1):53–68, 2005.
18. Nikolaos Kourentzes. On intermittent demand model optimisation and selection. International Journal of Production Economics, 156:180–190, 2014.
19. Lukas Lingitz, Viola Gallina, Fazel Ansari, David Gyulai, Andr ´ as Pfeiffer, Wilfried Sihn, and L ´ aszl ´ o´ Monostori. Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. Procedia Cirp, 72:1051–1056, 2018.
20. S. Barbarossa and S. Sardellitti, "Topological Signal Processing over Simplicial Complexes," IEEE Transactions on Signal Processing, vol. 68, pp. 2992–3007, 2020.
21. M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on Graphs: Uncertainty Principle and Sampling," IEEE Transactions on Signal Processing, vol. 64, no. 18, pp. 4845–4860, 2016.
22. Zhang, M.; and Chen, Y. 2018. Link prediction based on graph neural networks. Advances in neural information processing systems, 31.
23. Zhu, X.; Ninh, A.; Zhao, H.; and Liu, Z. 2021. Demand forecasting with supply-chain information and machine learning: Evidence in the pharmaceutical industry. Production and Operations Management, 30(9): 3231–3252.
24. Zougagh, N.; Charkaoui, A.; and Echchatbi, A. 2020. Prediction models of demand in supply chain. Procedia computer science, 177: 462–467.