

Supporting the Design of Electronic Circuits by Predicting Links in a Graph Structure [†]

Malinka Ivanova ^{1,*}  and Mariana Durcheva ^{1,2} 

¹ Department of Informatics, Faculty of Applied Mathematics and Informatics, Technical University of Sofia, 1797 Sofia, Bulgaria; mdurcheva66@gmail.com

² Mathematics Department, Sami Shamoon College of Engineering, Ashdod 77245, Israel

* Correspondence: m_ivanova@tu-sofia.bg

[†] Presented at the 15th International Scientific Conference TechSys 2026—Engineering, Technologies and Systems, Plovdiv, Bulgaria, 14–16 May 2026.

Abstract

Graph structures can be used to represent and explain data about connections and interactions between certain objects that form network systems. Graphs are applied in various scientific areas, and this paper explores their potential to support the design process of electronic circuits. Experimentations for predicting links in a heterogeneous graph structure are performed, which are built on the basis of schematic variants of electronic circuits and their elements. A graph neural network approach and the PyG library are used. The predictive model is evaluated, and the obtained results are promising.

Keywords: link prediction; graph neural network; electronic circuit

1. Introduction

Graph theory is widely used in various fields to represent and explain the relationships and interactions between objects within a given system or network. Tamura et al. study applications of graph theory for solving different problems in communications and examine whether there are relevant algorithms for these problems [1]. Göke relies on graph theory to model energy systems at the macro level, highlighting the ability to detail some processes related to energy consumption, transmission, storage, and conversion [2]. Han et al. use the theory of graphs in the field of design and optimization of manufacturing processes of contemporary equipment, propose an algorithm based on Hamiltonian graphs and, after verification, show the effectiveness of the solution [3]. Godquin et al. apply graph theory to model constraints in security for the development of an IoT network for smart cities [4]. More specifically, a weighted graph is used to find the appropriate locations for secure services. Wasik et al. discuss the utilization of graph theory in risk assessment during the supply chain management process to can support decision making [5].

Regardless of the different types of graph structures, each graph is characterized by nodes and edges. By definition, a graph is a nonempty finite set of nodes and a finite set of edges [6]. As a rule, the nodes are connected to each other by one or more edges. When the nodes and edges in a graph structure are of the same type, then the graph is homogeneous. In practice, heterogeneous graphs, formed by different types of nodes and edges, are more commonly used. Thus, each type of node or edge is described by a different data vector. Nowadays, problems with studying these complex heterogeneous graphs have arisen, particularly the challenge of computation limitations. The goal is to study the representation of an n -dimensional graph, including a description of its structure,



Academic Editors: Nikola G. Shakev,
Valyo N. Nikolov, Nikolay
R. Kakanakov and Sevil
A. Ahmed-Shieva

Published: 21 July 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

semantics, and attributes. To address this problem, deep learning techniques and a suitable graph neural network (GNN) architecture can be used [7]. Such an approach provides an opportunity to transform the complex graph structure into a low-dimensional vector space and to solve tasks such as link prediction [8], classification of nodes [9], implementation of recommendation systems [10], and others.

In the electronics industrial sector, graph theory and GNNs are also chosen by researchers as suitable approaches for modeling, design, presentation, and explanation of different processes and systems, as evidenced by recently published scientific works. Dong et al. propose a GNN-based method to automate the design of analog circuits that utilizes designed subgraphs corresponding to different circuit variants [11]. The method of Yamakaji et al. is also intended to assist the design process of electronic circuits, which are transformed into graph structures that are then studied using GNNs [12]. The effectiveness of the method is based on the conversion of heterogeneous graphs to homogenous ones. Hakhamaneshi et al. use GNNs to predict the voltages of a node, and in this way construct circuit variants that are presented through graphs [13].

The aim of this paper is, based on a summary and analysis of applications of GNNs in electronics, to explore an approach that supports the automated design of electronic analog circuits through the usage of graph theory and deep learning.

2. Graph Neural Networks in Circuit Design

Our investigation into facilitating and automating the process of designing electronic circuits using GNNs is carried out through a bibliometric analysis and a detailed analysis of articles indexed in the scientific database Scopus. The query submitted is “graph neural network” AND “circuit design”, and the returned result includes 77 documents. It is noteworthy that the topic has attracted the attention of researchers in the last few years, with the earliest document indexed in Scopus dating from 2019 (Figure 1). A significant rise in the number of publications is observed in 2023, indicating the rapidly growing interest in this research area. The annual growth rate of publications is estimated at 67.03%, highlighting the rapid development of GNN-related studies and their application in the electronic circuit design process. To perform the bibliometric analysis and extract information from the bibliometric data, the R (4.3.3) software environment and Biblioshiny application are used [14]. Figure 2 shows that the articles are mainly published in eight sources, which can be divided into two main categories: scientific journals and conference proceedings. There are four scientific journals, each with an impact factor/impact rank, and four conference proceedings. China and the USA have the highest number of publications on the subject (Figure 3). Despite having significantly fewer publications, Austria, Korea, and the UK remain in the top five in terms of contribution and scientific achievements. Authors from the USA are the most cited, followed by those from China (Figure 4). Interestingly, Georgia is not among the top five countries in terms of the number of published articles, but it achieves a high number of citations. This indicates that the scientific output from Georgia addresses significant contemporary issues that have been noticed by researchers and require further exploration.

In the built co-occurrence network, four clusters stand out, which unite the terms (key-words plus) associated with a specific topic (Figure 5). The first cluster relates to the study of GNN algorithms, including their parameters, architectures, and performance (in red). The second cluster focuses on the study of integrated circuit (IC) design, field-programmable gate arrays (FPGAs), high-level circuit synthesis approaches and development of predictive models through GNNs (in blue). The third cluster pertains to hardware and hardware security modeling and analysis utilizing GNNs (in green). The fourth cluster relates to

using GNNs in the computer-aided design and manufacturing process of integrated circuits (in lilac).

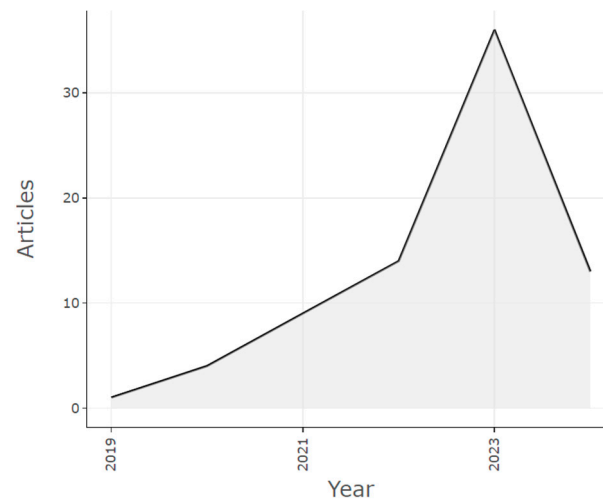


Figure 1. Annual scientific production.

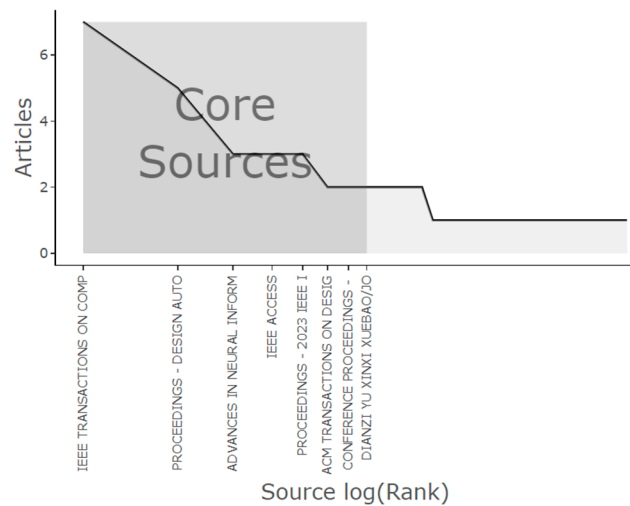


Figure 2. Source clustering through Bradford's Law.

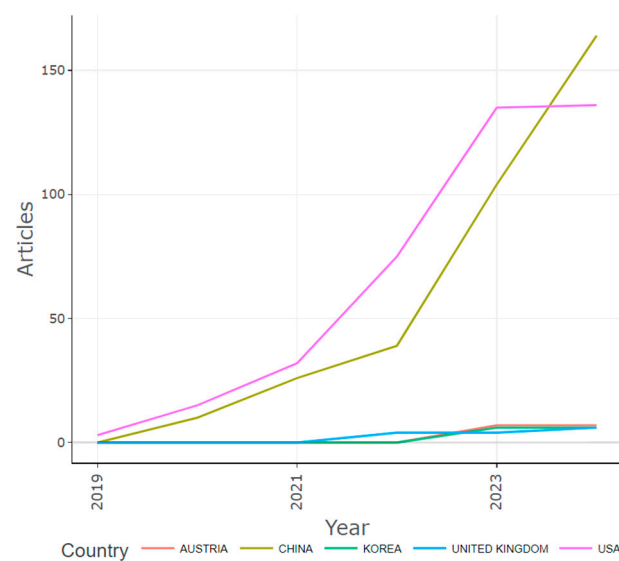


Figure 3. Production of countries over time.

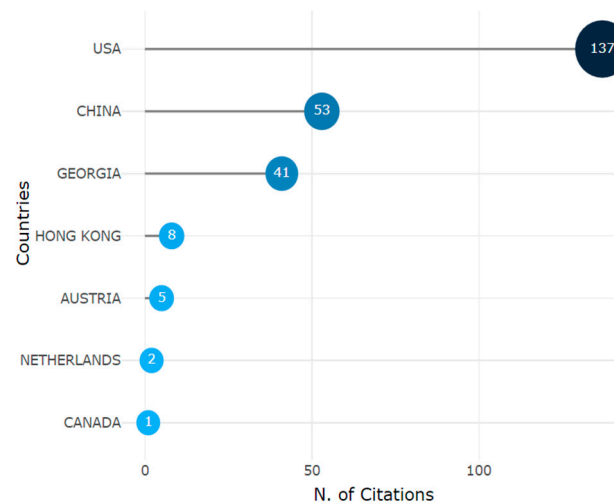


Figure 4. Most cited countries.

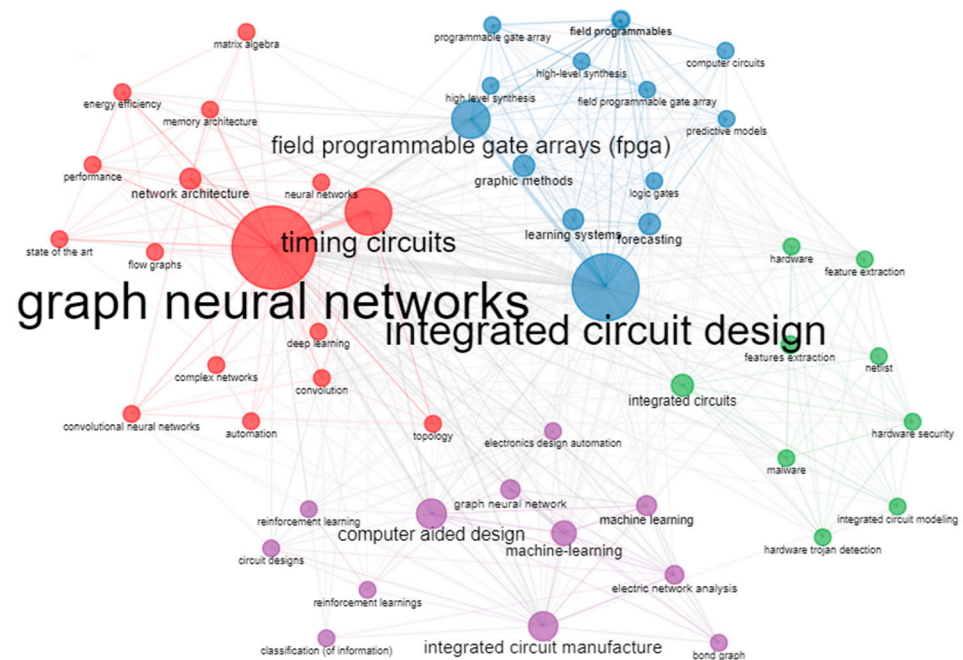


Figure 5. Keywords plus co-occurrence network.

Trending topics are visualized in Figure 6, showing the most frequently researched problems that are obviously related to the development and investigation of GNN architectures, as well as their application in the study of fields like the design of programmable gate arrays, integrated circuits and timing circuits.

The content of the most relevant and open-access articles is examined in more detail to see and understand exactly what scientific problems are under investigation.

The work of Abi-Karam and Hao [15] presents a framework, called GNNBuilder, designed to accelerate the generation and simulation of GNNs in an end-to-end manner. The framework is notable for its general applicability, as it is not limited to optimizing a specific GNN architecture but is capable of generating and optimizing a wide range of GNN models.

Jamal et al. propose a framework based on GNNs to predict the quality of results in the hardware design process during high-level synthesis [16]. Evaluating the designed product enables the designer to understand how well the design specifications have been met, while simultaneously reducing design effort and development time.

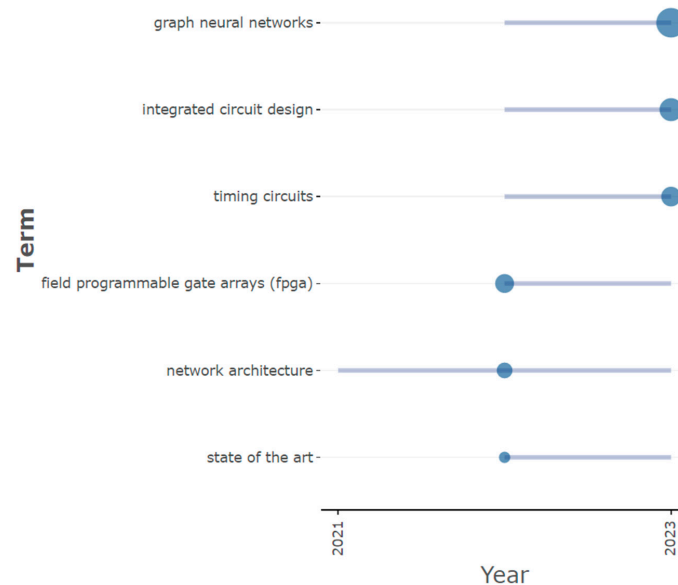


Figure 6. Trend topics utilizing keywords plus.

In the method proposed by Wang et al. for automated transistor sizing and for transferring knowledge across different technologies and circuit topologies, a convolutional GNN-based approach is used to analyze circuit topology represented as a graph [17]. The resulting tool, called GCN-RL (Graph Convolutional Network—Reinforcement Learning) Circuit Designer, achieves optimized design performance through the integration of reinforcement learning.

Dutta et al. deal with the problem of increased power consumption when using multi-core processor configurations, which is especially relevant in our time when huge amounts of data must be processed [18]. To optimize the performance of many core processors and bring their consumed energy in line with certain requirements and constraints, a GNN-based solution is developed, the efficacy of which is proven after a number of experiments.

Khamis and Agamy investigate the representation of electronic circuits by means of graph structures, proposing a methodology for the general case which can create graph representations regardless of the circuit design [19]. GNN is chosen to conduct the experiments as the input is homogenous graphs. The verification of the proposed methodology is performed under several circuit scenarios as the achieved accuracy in identifying the electric circuit type is high.

It can be summarized that in recent years, various GNN and GCN architectures have been intensively developed, and their capacity to be successfully applied in the electronics industrial sector have been evaluated, in particular when designing and studying the properties of FPGAs, ICs and various other hardware solutions. Numerous software tools have been developed to accelerate GNN generation and simulation, synthesize high-level hardware, model and study circuit topology, support IC design, and automate design and analysis tasks through the important role of GNNs. These advancements aim to reduce effort and experimentation time while achieving efficient design and manufacturing processes.

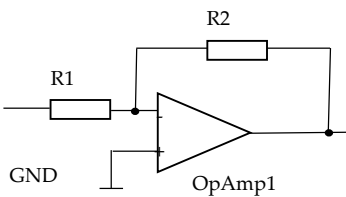
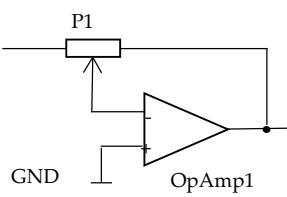
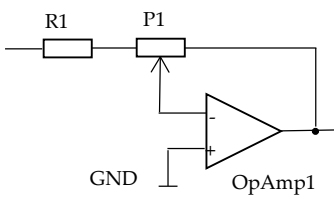
3. Used Method

The aim of the method is to predict the existence of links between elements in possible configurations of an electronic circuit through applying the concepts of heterogeneous graphs and a GNN-based link prediction algorithm. Existing links between nodes (elements) are referred to as *positive edges*, whereas the absence of a connection between nodes is denoted as a *negative edge*.

The applied method involves the following pipeline:

1. Preparation of a dataset that contains descriptions of the elements composing each electronic circuit. An example of the used electronic circuits and their corresponding element descriptions is shown in Table 1.
2. Creating a predictive model for solving the link prediction task.
3. Evaluating the performance of the developed model.

Table 1. Example of electronic circuits and their description.

	Circuit 1	Circuit 2	Circuit 3
Electronic Circuit			
Description	R1, R2, OpAmp1, GND	P1, OpAmp1, GND	R1, P1, OpAmp1, GND

To generate the GNN, train it on the dataset and evaluate its performance, the PyG library is used.

4. Experimentation and Results

The dataset created includes 883 records and contains data for 150 electronic circuits. Each circuit is composed of one or more elements. Each element possesses one of seven types: (1) resistor (R1, R2, R3, R4, R5, R6), (2) capacitor (C1, C2, C3), (3) operational amplifier (OpAmp1, OpAmp2, OpAmp3), (4) potentiometer (P1, P2), (5) break (B1, B2), (6) diode (D1, D2, D3) and (7) ground (GND).

The electronic circuits are presented via a heterogeneous graph (that consists of a set of subgraphs) and characterized by two node types, “circuit” and “element”, as well as one edge type, that captures the connection between circuits and elements (Figure 7). The “circuit” node type contains one or more elements, while the “element” node type represents an individual component belonging to a specific circuit. A GNN is used to predict the links between each circuit and its potential elements, thereby forming possible schematic configurations. According to the ability of the GNN, predictions are made as to which elements are best suited to which electronic circuit. In this way, the approach enables the automation of the process of designing electronic circuits by identifying and recommending possible elements for a certain configuration of the circuit.

The created heterogeneous GNN consists of two SAGEConv layers, and its purpose is to learn representations of nodes taking into account the surrounding subgraphs [20]. This information is essential for performing edge-prediction tasks. The input to the GNN is a set of subgraphs generated through a mini-batch loader.

For the purpose of GNN training, the edge data in the heterogeneous graph structure is divided into 80% for training, 10% for validation, and 10% for testing. For message passing and supervision, the test edges are further split in a 70%/30% ratio, respectively. During the evaluation phase, the proportion of positive to negative edges is chosen to be 2/1. Model tuning is conducted using optimization techniques like stochastic gradient decent and back-propagation.

The predictive model performance is evaluated through the loss function and the Area Under Curve (AUC).

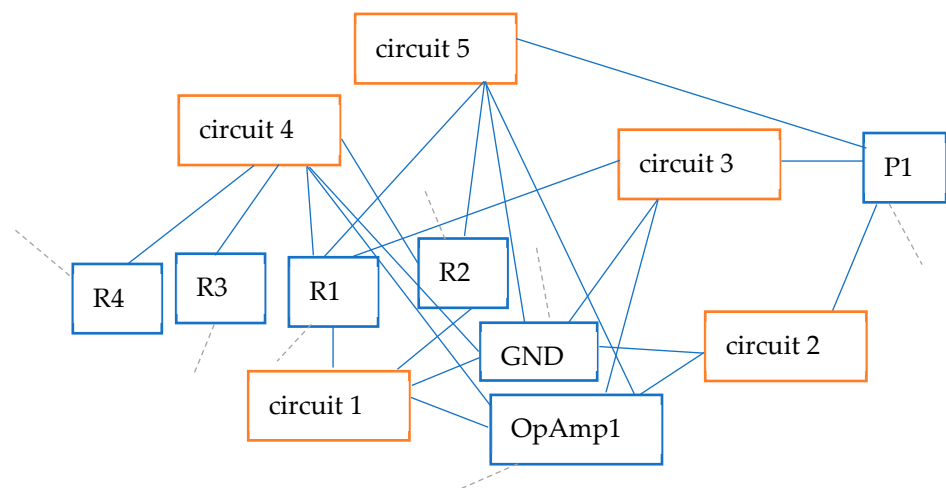


Figure 7. Heterogeneous graph structure.

The computed loss function (binary cross-entropy loss) over 200 epochs is presented in Figure 8 and outlines how accurately the GNN makes predictions. The loss values range between 0.1 and 0.23, indicating a low model error. Comparable loss behavior is observed at 100 and 300 epochs.

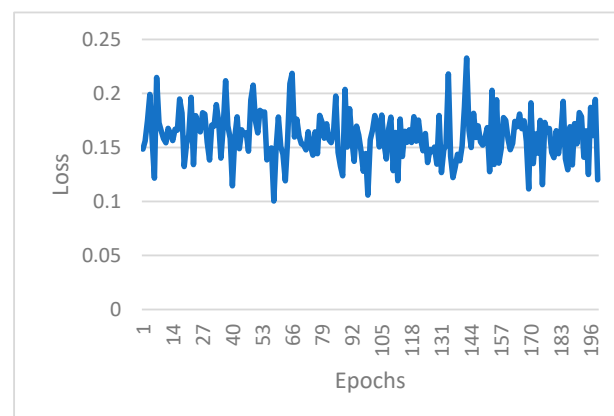


Figure 8. Loss function at 200 epochs.

AUC measures the model's ability to distinguish between different classes, with values closer to 1.0 indicating better performance. The AUC values at 100, 200 and 300 epochs are presented in Figure 9, with the highest value of 0.8 achieved at 300 epochs.

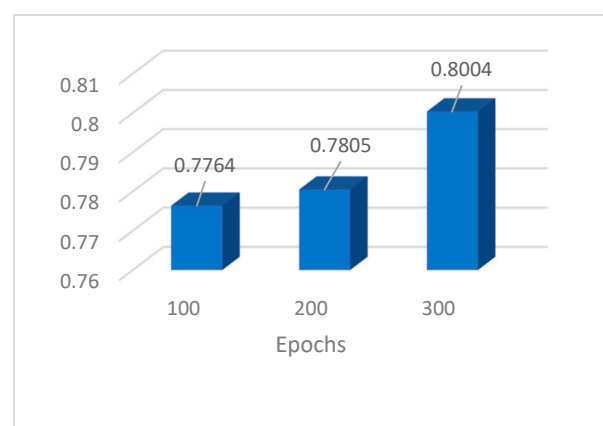


Figure 9. Validation AUC.

5. Conclusions

The paper explores and summarizes the importance of graph theory in representing and modeling electronic circuits and modules with graph structures. It also examines how recent advancements enable the use of GNNs for tasks such as prediction, classification, and recommendation. These developments enhance software tools for designing and simulating electronic circuits and hardware, reducing engineer effort, and increasing the speed of execution of these activities, as well as improving the design quality. It can be seen that this topic has experienced significant development since 2019, with major research efforts focused on inventing various GNN and GCN architectures and their use to support various real-world problems related to the design, acceleration and optimization of electronic circuits and hardware. The countries with the most contributions to the topic are China, the USA, Austria, Korea and the UK, with China and the USA leading. The identified trend topics are related to the development, optimization, and evaluation of GNN architectures, as well as their usage in the design of programmable gate arrays, integrated circuits, timing circuits and a wide variety of hardware systems.

This investigation and study addresses the application of heterogeneous GNNs for link prediction to support and automate the design of electronic circuits. The experiments conducted show that the created predictive model is characterized by good performance. This also confirms the capacity of GNNs as a possible approach in supporting the design of circuits and various hardware implementations.

Author Contributions: Conceptualization, M.I. and M.D.; methodology, M.I.; software, M.I.; validation, M.I. and M.D.; formal analysis, M.I.; investigation, M.I.; writing—original draft preparation, M.I.; writing—review and editing, M.I. and M.D.; visualization, M.I.; funding acquisition, M.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Bulgarian National Science Fund in the scope of the project “Exploration the application of statistics and machine learning in electronics” under contract number KPII-06-H42/1.

Data Availability Statement: The authors used data that are not publicly available.

Acknowledgments: The authors would like to thank the Bulgarian National Science Fund for the financial support in the scope of the project “Exploration the application of statistics and machine learning in electronics” under contract number KPII-06-H42/1.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tamura, H.; Nakano, K.; Sengoku, M.; Shinoda, S. On Applications of Graph/Network Theory to Problems in Communication Systems. *ECTI-CIT Trans.* **2016**, *5*, 15–21. [\[CrossRef\]](#)
2. Göke, L. A graph-based formulation for modeling macro-energy systems. *Appl. Energy* **2021**, *301*, 117377. [\[CrossRef\]](#)
3. Han, Z.; Cheng, L.; Xing, L.; Tian, L. A Graph Theory-Based Optimization Design for Complex Manufacturing Processes. *IEEE Access* **2020**, *8*, 95547–95558. [\[CrossRef\]](#)
4. Godquin, T.; Barbier, M.; Gaber, C.; Grimault, J.-L.; Le Bars, J.-M. Applied graph theory to security: A qualitative placement of security solutions within IoT networks. *J. Inf. Secur. Appl.* **2020**, *55*, 102640. [\[CrossRef\]](#)
5. Wasik, E.; Sidor, T.; Wolowiec, T.; Piwkowski, J.; Jasienski, M. Supporting Supply Chain Risk Management: An Innovative Approach Using Graph Theory and Forecasting Algorithms. *Eur. Res. Stud. J.* **2024**, *XXVII*, 25–37. [\[CrossRef\]](#)
6. Grinberg, D. An Introduction to Graph Theory. 2023. Available online: <https://www.cip.ifi.lmu.de/~grinberg/t/22s/graphs.pdf> (accessed on 25 October 2023).
7. Dong, Y.; Hu, Z.; Wang, K.; Sun, Y.; Tang, J. Heterogeneous Network Representation Learning. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, Yokohama, Japan, 7–15 January 2021; pp. 4861–4867. [\[CrossRef\]](#) [\[PubMed\]](#)

8. Li, J.; Ma, Y.; Mao, H.; Shah, N.; Shomer, H.; Tang, J.; Yin, D.; Zeng, S. Evaluating graph neural networks for link prediction: Current pitfalls and new benchmarking. In Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '23), New Orleans, LA, USA, 10–16 December 2023; pp. 3853–3866.
9. Tang, J.; Liao, R. Graph Neural Networks for Node Classification. In *Graph Neural Networks: Foundations, Frontiers, and Applications*; Wu, L., Cui, P., Pei, J., Zhao, L., Eds.; Springer: Singapore, 2022. [\[CrossRef\]](#)
10. Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; et al. A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. *ACM Trans. Recomm. Syst.* **2023**, *1*, 1–51. [\[CrossRef\]](#)
11. Dong, Z.; Cao, W.; Zhang, M.; Tao, D.; Chen, Y.; Zhang, X. CktGNN: Circuit Graph Neural Network for Electronic Design Automation. In Proceedings of the International Conference on Learning Representations (ICLR 2023), Kigali, Rwanda, 1–5 May 2023; pp. 1–20. [\[CrossRef\]](#)
12. Yamakaji, Y.; Shouno, H.; Fukushima, K. Circuit2Graph: Circuits with Graph Neural Networks. *IEEE Access* **2024**, *12*, 51818–51827. [\[CrossRef\]](#)
13. Hakhamaneshi, K.; Nassar, M.; Phielipp, M.; Abbeel, P.; Stojanovic, V. Pretraining Graph Neural Networks for Few-Shot Analog Circuit Modeling and Design. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *42*, 2163–2173. [\[CrossRef\]](#)
14. Aria, M.; Cuccurullo, C. bibliometrix: An R-tool for comprehensive science mapping analysis. *J. Informetr.* **2017**, *11*, 959–975. [\[CrossRef\]](#)
15. Abi-Karam, S.; Hao, C. GNNBuilder: An Automated Framework for Generic Graph Neural Network Accelerator Generation, Simulation, and Optimization. In Proceedings of the 33rd International Conference on Field-Programmable Logic and Applications (FPL 2023), Gothenburg, Sweden, 4–8 September 2023; pp. 212–218.
16. Jamal, M.U.; Li, Z.; Lazarescu, M.T.; Lavagno, L. A Graph Neural Network Model for Fast and Accurate Quality of Result Estimation for High-Level Synthesis. *IEEE Access* **2023**, *11*, 85785–85798. [\[CrossRef\]](#)
17. Wang, H.; Wang, K.; Yang, J.; Shen, L.; Sun, N.; Lee, H.-S.; Han, S. GCN-RL circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning. In Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC 2020), San Francisco, CA, USA, 20–24 July 2020. [\[CrossRef\]](#)
18. Dutta, A.; Choi, J.; Jannesari, A. Power Constrained Autotuning using Graph Neural Networks. In Proceedings of the 2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS) 2023, St. Petersburg, FL, USA, 15–19 May 2023; pp. 535–545. [\[CrossRef\]](#)
19. Khamis, A.K.; Agamy, M. Comprehensive Mapping of Continuous/Switching Circuits in CCM and DCM to Machine Learning Domain Using Homogeneous Graph Neural Networks. *IEEE Open J. Circuits Syst.* **2023**, *4*, 50–69. [\[CrossRef\]](#)
20. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017; pp. 1025–1035.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.