



## Energy-Aware Training and Deployment of Large-Scale Machine Learning Models: A Review of Distributed Graph Data Science and Multi-Objective Resource Optimization

Abhirami S<sup>1\*</sup>, V.Anuratha<sup>2</sup>, M.Elamparithi<sup>3</sup>

<sup>1</sup>Research Scholar, Department of Computer Science, Kamalam College of Arts and Science, Udumalpet, Affiliated to Bharathiar University, Coimbatore

\* Corresponding Author Email: [abhiramisidharth09@gmail.com](mailto:abhiramisidharth09@gmail.com) - ORCID: 0000-0002-5667-7850

<sup>2</sup>Associate Professor, Department of Computer Science, Kamalam College of Arts and Science, Udumalpet, Affiliated to Bharathiar University, Coimbatore

Email: [profanuratha@gmail.com](mailto:profanuratha@gmail.com) - ORCID: 0000-0002-5247-3350

<sup>3</sup>Associate Professor, Department of Computer Science, Kamalam College of Arts and Science, Udumalpet, Affiliated to Bharathiar University, Coimbatore

Email: [profelamparithi@gmail.com](mailto:profelamparithi@gmail.com) - ORCID: 0000-0002-5247-1150

### Article Info:

DOI: 10.22399/ijcesen.5204

Received : 25 October 2025

Revised : 25 December 2025

Accepted : 27 December 2025

### Keywords

Machine learning  
Energy-Aware Training  
Resource Optimization

### Abstract:

The exponential growth of large-scale Machine Learning (ML) models, particularly Transformers and Graph Neural Networks (GNNs), has catalyzed advancements across various domains, yet it imposes a substantial environmental and computational cost. This review paper investigates energy-aware strategies for the training and deployment phases of large-scale ML systems through the lens of Graph Data Science and Distributed Computing. By synthesizing recent literature on GNNs, Deep Reinforcement Learning (DRL), and bio-inspired optimization techniques, this study explores novel methods to minimize resource usage. We analyze frameworks that optimize computation graphs, implement dynamic cache scheduling, and employ game-theoretic Nash equilibrium for task offloading in edge-cloud environments. The paper identifies critical research gaps in dynamic graph partitioning and energy profiling, proposing a multi-objective framework that balances accuracy, latency, and energy efficiency. Finally, we recommend a two-phase research trajectory to advance sustainable "Green AI," paving the way for scalable, environmentally responsible artificial intelligence.

## 1. Introduction

The advent of the "Big Data" era has been accompanied by a corresponding explosion in the size, complexity, and computational demands of Machine Learning (ML) models. From Natural Language Processing (NLP) transformers with hundreds of billions of parameters to massive Graph Neural Networks (GNNs) modelling biochemical and social networks, the capacity of AI has skyrocketed. However, this progress is underpinned by a hidden crisis: the exponential increase in energy consumption and carbon footprint [1]. A single training run of a large language model can emit over 626,000 pounds of CO<sub>2</sub>, equivalent to the lifetime emissions of five

standard cars [2]. This phenomenon, termed "Red AI," defines the current paradigm where state-of-the-art accuracy is bought through brute-force computational power. In stark contrast, "Green AI" is an emerging research field that seeks to decouple AI performance from energy consumption by treating "Efficiency" as a primary evaluation metric alongside accuracy [3]. The figure 1 illustrates the conceptual line graph comparing "Red AI" which shows exponential growth in energy consumption to achieve accuracy, versus "Green AI" which shows a balanced, energy-aware trajectory where accuracy is maintained while energy consumption plateaus through graph optimization.

As AI leaves the data center and enters the pervasive environment of the Internet of Things (IoT) and 6G networks, the energy crisis becomes a

bottleneck for deployment [4]. Edge devices—such as autonomous vehicles, smart city sensors, and mobile phones—are battery-constrained. They cannot run massive models locally. This limitation necessitates "Distributed Training" and "Task Offloading," where computation is divided among multiple devices or sent to the cloud. However, this introduces a new energy drain: Communication Overhead. The energy required to transmit data across a network often exceeds the energy required to compute it. This is particularly acute in Graph Neural Networks. GNNs aggregate information from neighboring nodes. In a distributed graph (like a billion-user social network spread across many servers), fetching data from "remote" neighbors creates massive bandwidth bottlenecks and energy spikes. As highlighted in recent studies, communication overhead can consume 50% to 90% of training time in distributed GNN settings [5]. To address these interconnected challenges, this research proposes a fundamental shift in how we view AI architecture. Instead of treating infrastructure (hardware) and ML models (software) as separate entities, we utilize **Graph Databases and Graph Theory** to unify them. By modeling both the computational network and the data dependencies as graphs, we can apply advanced graph algorithms to optimize the system. Specifically, this review investigates energy-aware strategies across two distinct phases of the ML lifecycle:

1. **The Training Phase:** We examine how frameworks like RapidGNN [6] use deterministic sampling and graph caching to minimize the communication energy wasted when different GPUs talk to each other. We also explore Graph Partitioning [7], where large datasets are strategically cut into subgraphs to balance the load and minimize inter-server communication.
2. **The Deployment Phase:** Once trained, how do we run these models efficiently? We review hybrid frameworks that combine GNNs with Deep Reinforcement Learning (DRL) [8] and Game Theory (Nash Equilibrium) [9]. These systems treat resource allocation as a multi-agent game, deciding in real-time whether a specific computation should be executed on a low-power edge device or sent to a high-power cloud server to minimize overall system energy.

Furthermore, this paper acknowledges that optimization is not limited to neural networks. Bio-inspired meta-heuristic algorithms, such as the Grey Wolf Optimizer (GWO) [10], offer lightweight alternatives for path planning and routing in edge

networks. Coupled with data cleaning techniques that utilize discrete optimization [11], we can ensure that computational cycles are not wasted on noisy, redundant data. By synthesizing these core base papers, this review establishes a foundation for a multi-objective optimization framework that paves the way for sustainable, energy-efficient, and highly scalable Artificial Intelligence.

## 2. Related works along with literature survey

The pursuit of energy efficiency in machine learning has bifurcated into two main streams: algorithmic optimization and system-level resource management. Early work in algorithmic optimization focused on model compression techniques, such as quantization and pruning, to reduce the memory footprint of neural networks [12]. However, as models scaled to graph datasets with billions of edges, system-level communication became the primary energy sink.

In Graph Data Science, traditional sampling methods like GraphSAGE [13] alleviate computational loads but suffer from redundant data fetching in distributed settings. To address this, Niam et al. [6] introduced RapidGNN, a deterministic sampling framework that builds a steady cache for highly accessed "hub" nodes, significantly reducing CPU and GPU energy consumption by 44% and 32%, respectively. Parallel to this, Long [7] tackled the scalability of combinatorial optimization by combining the Louvain graph partitioning algorithm with distributed GNN training, allowing ultra-large graphs to be processed in parallel without exceeding single-node memory.

The Figure 2: Taxonomy of Energy-Aware Distributed GNN Training Strategies describes a hierarchical tree diagram categorizing the current research methods. The root "Distributed GNN Training" branches into "Algorithmic Optimizations" (Sampling/Partitioning) and "System-Level Optimizations" (Caching/Pipeline). Terminal nodes highlight specific methods like Louvain Partitioning and RapidGNN Caching.)

In the deployment phase, Cloud and Edge resource management has transitioned from static heuristics to predictive AI-driven frameworks. Kumar et al. [14] modeled cloud infrastructure as a spatio-temporal graph, using GNNs to proactively allocate resources based on predicted workloads. Huang [8] extended this to vehicular networks, utilizing a hybrid GNN-DRL approach. Advancing to 6G networks, Al-Qudah et al. [9] identified that multi-agent DRL fails in highly competitive environments. They proposed a Heterogeneous Graph Transformer integrated with Nash

Equilibrium to coordinate task offloading among competing agents.

Data integrity also plays a crucial role in energy awareness. Training on incomplete data wastes energy. Smith and Climer [11] applied discrete optimization to maximize data retention during cleaning. Additionally, for mobile robotic agents, Chen et al. [10] improved the bio-inspired Grey Wolf Optimization (GWO) algorithm using reinforcement learning for energy-efficient path planning.

### 3. Methods incorporated

The reviewed literature employs a sophisticated convergence of Graph Data Science, Reinforcement Learning, and Heuristic Optimization to achieve energy efficiency.

#### 3.1 Graph Neural Networks (GNNs) and Transformers for System Modeling

Traditional resource allocation methods rely on time-series forecasting (e.g., ARIMA) to predict server loads. However, cloud-edge infrastructures are inherently relational. To capture this, recent methods employ GNNs to model the physical infrastructure as a topological graph [8][14]. Hardware devices are nodes, and bandwidth connections are edges. Spatio-temporal GNNs propagate feature vectors across this graph, predicting energy spikes. Furthermore, Al-Qudah et al. [9] advance this by integrating Heterogeneous Graph Transformers. Unlike standard GNNs, Transformers use "Attention Mechanisms" to dynamically weigh the importance of different edges, allowing the model to focus on critical bottlenecks in 6G networks.

#### 3.2 Distributed Graph Partitioning and Deterministic Sampling

Handling ultra-large-scale graphs requires sophisticated division mechanisms [17]. Paper [7] utilizes the **Louvain algorithm**, a well-known heuristic in graph theory, for structural partitioning. By maximizing the "modularity" of the graph, the algorithm divides the data such that nodes within a cluster are densely connected, while connections between different clusters are minimized. Complementing this, Niam et al. [6] introduce **Deterministic Sampling**. Instead of randomly fetching neighbor data for GNN aggregation—which causes unpredictable network traffic—this method pre-calculates the necessary "remote" nodes and loads them into a two-level cache system, reducing remote feature fetches by up to 15.39x.

#### 3.3 Deep Reinforcement Learning (DRL) and Game Theory Control

Once the graph infrastructure is modeled, dynamic control is required to make offloading decisions. Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO) are widely used [8][18]. The DRL agent observes the "state" of the system (battery levels, network latency) and takes an "action" (execute locally, offload to edge/cloud). The reward balances latency with minimal energy. In highly competitive environments, [9] incorporates **Nash Equilibrium** from non-cooperative Game Theory [19]. It ensures that the resource allocation reaches a stable state where no single device can improve its energy efficiency by unilaterally changing its strategy.

#### 3.4 Discrete Optimization and Bio-Inspired Heuristics

Smith & Climer [11] apply **Integer Linear Programming (ILP)** to the problem of missing data. Instead of arbitrarily deleting rows with missing values, their algorithm mathematically maximizes data retention. For dynamic deployment, Chen et al. [10] utilize the **Multi-Strategy Improved Grey Wolf Optimization (MSIAR-GWO)** algorithm. Mimicking the social hierarchy of wolves [20], this algorithm performs global searches for the most energy-efficient routing paths, modified with RL to avoid local optima.

The Figure 3: Architecture of the Multi-Objective Edge-Cloud Offloading Framework point up a technical block diagram showing component interaction. Graph A (ML Task Flow) and Graph B (Physical Infrastructure) feed into a GNN Predictor. This data passes to a DRL Agent which calculates the Nash Equilibrium to make the final energy-efficient task placement decision.)

### 4: Discussion

#### Advantages & problems

The integration of GNNs into resource allocation offers a significant advantage: **topology awareness**. Unlike traditional forecasting, GNN-based methods [14] understand that server loads are spatially correlated. Distributed training frameworks like RapidGNN [6] provide the advantage of **scalability**, allowing massive datasets to be trained on commodity hardware by optimizing memory access.

However, problems persist. **Complexity** is a major issue; implementing a distributed GNN training pipeline requires intricate synchronization between workers. Furthermore, "Graph Partitioning" [7] is an NP-hard problem; heuristics like Louvain are

fast but may cut critical edges, resulting in information loss and reduced model accuracy.

### Challenges

A primary challenge is **Heterogeneity**. In 6G and IoT networks [9], devices have vastly different compute capabilities and energy constraints. Designing a single "Energy-Aware" framework that works for a high-power GPU server and a low-power edge sensor is difficult.

**Data Dependency** is another challenge. Techniques like those in [6] rely on "static" graphs where the structure is known. In dynamic networks (e.g., vehicular networks [8]), the graph topology changes millisecond-to-millisecond. Re-training or re-partitioning the graph in real-time incurs a high computational penalty, potentially negating the energy savings.

### Limitations

Current approaches often suffer from **Simulation-to-Reality gaps**. Many of the reviewed papers [8][10] validate their energy savings in simulated environments (MATLAB/Python). Real-world hardware exhibits non-linear energy behaviors (leakage current, cooling overhead) that idealized models miss.

Additionally, **Over-Smoothing** in GNNs remains a limitation [21]. As GNNs go deeper to capture long-range dependencies in large networks, node representations tend to become indistinguishable, degrading performance. This limits the depth of reasoning possible in complex resource allocation graphs.

### Research gap and future directions

There is a clear gap in **Unified Holistic Energy Profiling**. While papers address energy in specific niches (training [6], offloading [8]), there is no holistic framework that tracks energy from "Data Collection" to "Deployment."

A significant future direction is **Dynamic Graph Partitioning for Green AI**. Existing partitioning [7] is static. Future work should develop "Learning-

to-Partition" algorithms that dynamically adjust graph cuts based on real-time network bandwidth and energy availability.

Another gap is the integration of **Knowledge Graphs (KGs)**. Current methods use structural graphs; augmenting these with semantic KGs could allow the system to "reason" about compatibility and security constraints.

### Recommendations

Based on the synthesized literature, to effectively proceed with the proposed Ph.D. research titled "Energy-Aware Training and Deployment of Large-Scale Machine Learning Models," a **Two-Phase Research Trajectory** is recommended:

#### Research Phase 1: Energy-Efficient Dynamic Graph Partitioning (Training Focus)

The research should extend the concept of RapidGNN [6] and distributed partitioning [7]. The goal is to develop an AI-driven Energy Profiling Tool and a dynamic Graph Database partitioner using RL. The agent will dynamically move graph nodes between workers during training to balance load and minimize communication energy.

The Figure 4: Workflow of the Proposed Two-Phase Ph.D. Research Trajectory points the flowchart illustrating the transition of the PhD project. It starts with Phase 1 (Dynamic Graph Partitioning to optimize training energy) and outputs a trained model. This model moves into Phase 2 (Game-Theoretic Edge-Cloud Allocation), which dynamically deploys the model to the physical world while minimizing carbon footprint.)

#### Research Phase 2: Green-MEC Game-Theoretic Resource Allocation (Deployment Focus)

Building on Huang [8] and Al-Qudah [9], the research focus the development of the "Multi-Objective Framework" for deployment. This phase will model the edge-cloud infrastructure as a Resource Graph. A Nash-regularized DRL agent will make real-time offloading decisions for deployed ML models, balancing latency, accuracy, and carbon footprint metrics across heterogeneous hardware.

*Table 1: Literature Survey of Recent Articles*

| Authors and Year           | Focus Area / Methodology               | Key Findings / Contribution   |
|----------------------------|--|---|
| Niam et al. (2025) [6]     | Distributed GNN Training / Caching     | RapidGNN reduces remote feature fetches by up to 15.39x; improves energy efficiency by 44% in large-scale graph training. |
| Long (2025) [7]            | Combinatorial Optimization / GNN + RL  | Integrates Louvain partitioning with distributed GNNs to solve Maximum Independent Set on million-node graphs.            |
| Kumar et al. (2025) [14]   | Cloud Resource Allocation / GNN        | Models cloud infrastructure as graphs; proactive scaling reduces operational costs and SLA violations.                    |
| Huang (2025) [8]           | Vehicular Task Offloading / GNN + DRL  | Captures inter-task dependencies using Graph-to-Sequence models; minimizes expected system energy in Edge computing.      |
| Al-Qudah et al. (2026) [9] | 6G Edge Computing / Graph Transformers | Uses Nash Equilibrium to coordinate 1000+ agents; reduces latency by 23.4% and improves energy efficiency by 31.7%.       |

|                                       |                                       |   |
|---------------------------------------|---------------------------------------|---|
| <b>Chen et al. (2025)</b> [10]        | Mobile Path Planning / MSIAR-GWO      | Combines Reinforcement Learning with Grey Wolf Optimization for adaptive, energy-efficient robotic routing.                     |
| <b>Smith &amp; Climer (2024)</b> [11] | Data Cleaning / Discrete Optimization | Formulates data deletion as an Integer Linear Program, maximizing valid element retention for downstream ML.                    |
| <b>Ganesan et al. (2024)</b> [15]     | Hierarchical Edge-Fog-Cloud / RL      | Optimizes task offloading in 6G architectures, demonstrating sub-10ms latency capabilities.                                     |
| <b>Shao et al. (2024)</b> [5]         | Distributed GNN Survey                | Identifies communication overhead as taking 50-90% of total training time, highlighting the need for efficient graph databases. |
| <b>Ma et al. (2024)</b> [16]          | Scalable RL Network Control           | Achieves large-scale network control using distributed reinforcement learning with minimal communication overhead.              |

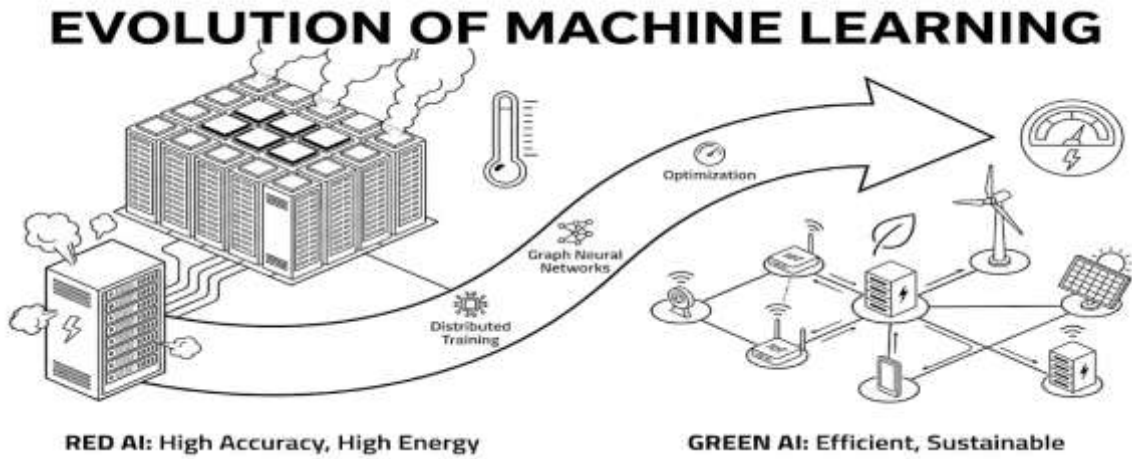


Figure 1: Evolution of Machine Learning: Transitioning from Red AI to Green AI.

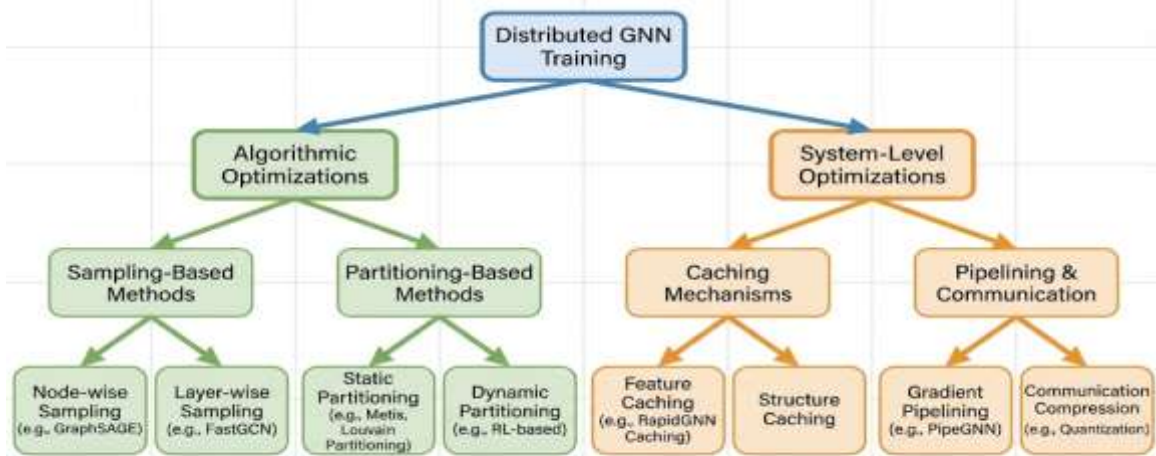


Figure 2: Taxonomy of Energy-Aware Distributed GNN Training Strategies.

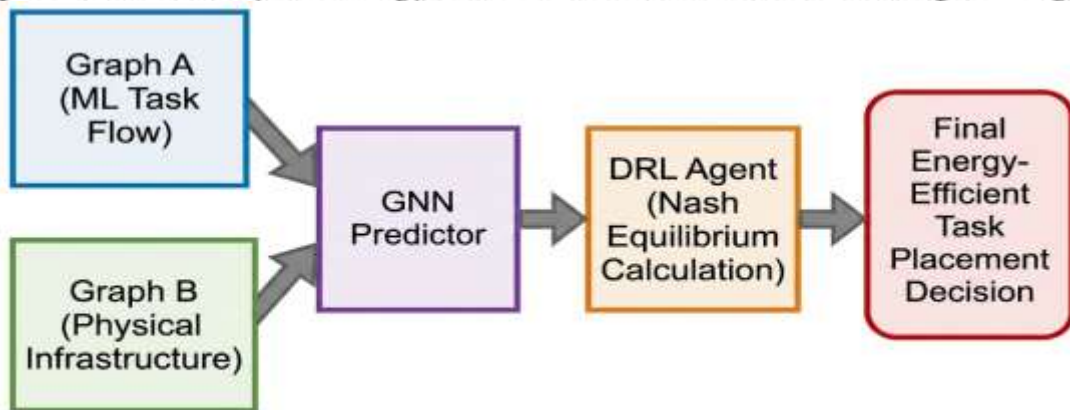
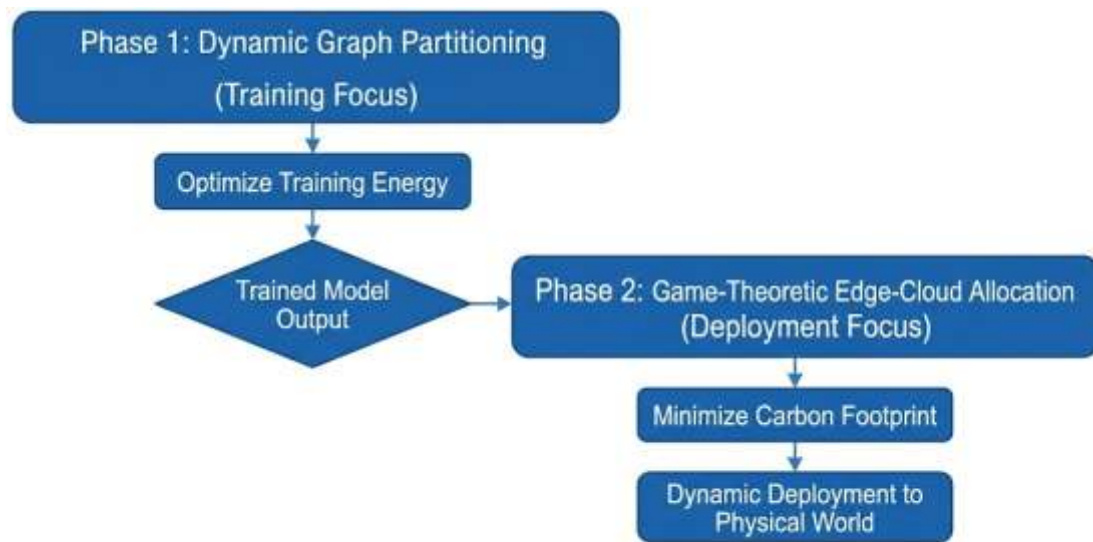


Figure 3: Architecture of the Multi-Objective Edge-Cloud Offloading Framework.



**Figure 4: Workflow of the Proposed Two-Phase Ph.D. Research Trajectory.**

## 5. Conclusions

The exponential growth of large-scale machine learning models has precipitated a critical need for sustainable, energy-efficient AI systems. This review paper has systematically analyzed the intersection of Graph Data Science, Distributed Computing, and Advanced Optimization to address the dual challenges of performance and environmental responsibility. By synthesizing recent core base papers, we have established that the transition from accuracy-centric "Red AI" to efficiency-centric "Green AI" requires a multi-layered approach.

During the training phase, the communication bottleneck in distributed Graph Neural Networks can be effectively mitigated through deterministic sampling, advanced graph partitioning algorithms, and intelligent two-level caching systems. Methods such as RapidGNN demonstrate that significant energy reductions—up to 44% in CPU and 32% in GPU usage—are achievable without sacrificing model accuracy [6].

In the deployment phase, the Edge-Cloud continuum presents a highly dynamic environment. The reviewed literature confirms that hybrid frameworks combining GNNs for topology modeling with Deep Reinforcement Learning for decision-making offer superior task offloading performance [8]. Furthermore, the integration of Game Theory, specifically Nash Equilibrium, is essential for coordinating resource allocation among thousands of competing agents in emerging 6G networks [9].

The proposed Ph.D. research on "Energy-Aware Training and Deployment of Large-Scale Machine Learning Models" is both timely and essential. By implementing the recommended two-phase

research trajectory—focusing on dynamic graph partitioning for training and game-theoretic resource allocation for deployment—this research will directly contribute to the realization of Green AI, paving the way for cost-effective, scalable, and environmentally responsible artificial intelligence.

## Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.
- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

## References

- [1] Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., ... & Dean, J. (2021). Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*.

- [2] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243*.
- [3] Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. *Communications of the ACM*, 63(12), 54-63.
- [4] Chowdhury, M. Z., et al. (2020). 6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions. *IEEE Open Journal of the Communications Society*, 1, 957-975.
- [5] Shao, Y., et al. (2024). Distributed graph neural network training: A survey. *ACM Computing Surveys*, 56(8), 1-39.
- [6] Niam, A., Kosar, T., & Nine, M. S. Z. (2025). RapidGNN: Energy and communication-efficient distributed training on large-scale graph neural networks. *arXiv preprint arXiv:2509.05207*.
- [7] Long, Y. (2025). A distributed training architecture for combinatorial optimization. *arXiv preprint arXiv:2511.09261*.
- [8] Huang, Y. (2025). Multimedia tasks-oriented edge computing offloading scheme based on graph neural network in vehicular networks. *IEEE Access*, 13, 9780-9791.
- [9] Al-Qudah, R., Al Tawil, A., AlMuhajiri, M., Almazaydeh, L., & Suen, C. Y. (2026). Nash-regularized heterogeneous graph transformer networks for strategic task offloading in 6G edge computing. *International Journal of Data and Network Science*, 10, 1-12.
- [10] Chen, D., Liu, J., Li, T., He, J., Chen, Y., & Zhu, W. (2025). Research on mobile robot path planning based on MSIAR-GWO algorithm. *Sensors*, 25(3), 892.
- [11] Smith, K., & Climer, S. (2024). Improving data cleaning using discrete optimization. *arXiv preprint arXiv:2405.00764*.
- [12] Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:1510.00149*.
- [13] Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30, 1024-1034.
- [14] Kumar, K. P., Lal, B., & Srinivasulu, B. V. (2025). Graph-neural-network-driven proactive resource allocation for cloud workloads: A predictive graph-modeling framework. *American Journal of AI Cyber Computing Management*, 5(4), 335-345.
- [15] Ganesan, M., & Chelliah, B. (2024). Optimizing task offloading and resource management in 6G networks through a hierarchical edge-fog-cloud architecture. *IEEE Access*, 12, 87452-87468.
- [16] Ma, C., et al. (2024). Efficient and scalable reinforcement learning for large-scale network control. *Nature Machine Intelligence*, 6(9), 1006-1020.
- [17] Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., ... & Leskovec, J. (2020). Open graph benchmark: Sets and benchmarks for machine learning on graphs. *Advances in neural information processing systems*, 33, 22118-22133.
- [18] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533.
- [19] Nash, J. (1951). Non-cooperative games. *Annals of mathematics*, 286-295.
- [20] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- [21] 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.
- [22] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- [23] Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer networks. *Advances in neural information processing systems*, 28.
- [24] 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.
- [25] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- [26] Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource management with deep reinforcement learning. *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, 50-56.
- [27] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks?. *arXiv preprint arXiv:1810.00826*.
- [28] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- [29] Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785-794.
- [30] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 265-283.
- [31] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- [32] Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., ... & Zhang, Z. (2019). Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.
- [33] Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*.
- [34] Karypis, G., & Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular

- graphs. *SIAM Journal on scientific Computing*, 20(1), 359-392.
- [35] Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10), P10008.
- [36] Dorigo, M., Maniezzo, V., & Coloni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1), 29-41.
- [37] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks*, 4, 1942-1948.
- [38] Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Vol. 200). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- [39] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. *Nature inspired cooperative strategies for optimization (NICSO 2010)*, 65-74.
- [40] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 95, 51-67.
- [41] Heidari, A. A., Mirjalili, S., Faris, H., & Aljarah, I. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, 97, 849-872.
- [42] Wang, G. G., Deb, S., & Cui, Z. (2019). Monarch butterfly optimization. *Neural computing and applications*, 31, 1995-2014.
- [43] Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., ... & Pande, V. (2018). MoleculeNet: a benchmark for molecular machine learning. *Chemical science*, 9(2), 513-530.
- [44] Dwivedi, V. P., Joshi, C. K., Luu, A. T., Laurent, T., Bengio, Y., & Bresson, X. (2020). Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*.
- [45] Jiang, W. (2022). Graph-based deep learning for communication networks: A survey. *Computer Communications*, 185, 40-54.
- [46] Rusek, K., Suárez-Varela, J., Almasan, P., Barlet-Ros, P., & Cabellos-Aparicio, A. (2020). RouteNet: Leveraging