Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Artificial Intelligence Framework for Connectivity Optimization and Adaptive Routing in Processor Interconnection Networks

Sushil Kumar Dwivedi

sushil_dwid@yahoo.co.in

Awadhesh Pratap Singh University

Rakesh Kumar Katare

Awadhesh Pratap Singh University

Research Article

Keywords:

Posted Date: November 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11260644/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

An Artificial Intelligence Framework for Connectivity Optimization and Adaptive Routing in Processor Interconnection Networks

Sushil Kumar Dwivedi
Awadhesh Pratap Singh University, Rewa, India
Email: sushil_dwid@yahoo.co.in

Rakesh Kumar Katare
Awadhesh Pratap Singh University, Rewa, India
Email: katare1962@gmail.com

***Abstract*—Interconnection networks determine the communication efficiency, scalability, and fault resilience of parallel and distributed computing systems. Classical topologies such as Hypercube, Torus, Fat-Tree, Dragonfly, and Perfect Difference Network (PDN) provide different trade-offs among degree, diameter, path diversity, and communication cost, while static routing may perform poorly under dynamic traffic and failures. This paper presents the Artificial Intelligence Connectivity Optimization Framework (AICOF) , an AI-based Connectivity Optimization Framework that integrates Multi-Layer Perceptron (MLP) traffic estimation, Graph Neural Network (GNN) topology-aware learning, and Proximal Policy Optimization (PPO)-based reinforcement learning for adaptive routing. The network is modeled as a graph, and routing optimization jointly considers connectivity, throughput, reliability, latency, congestion, and energy consumption. A unified simulation methodology is applied to Hypercube, Torus, Fat-Tree, Dragonfly, and PDN topologies with network sizes from 64 to 512 nodes and multiple traffic and failure scenarios. The evaluation includes topology comparison, scalability analysis, AI-assisted routing, fault tolerance, and ablation studies. The results indicate that AICOF improves throughput, latency, packet delivery, energy efficiency, and fault recovery compared with conventional and standalone AI-based routing approaches.**

***Keywords*—Interconnection networks, graph neural networks, reinforcement learning, adaptive routing, topology optimization, high-performance computing.**

I. Introduction

Interconnection networks form the communication backbone of multiprocessor,many-core, distributed, and high-performance computing systems, where communication performance directly influences scalability, throughput, and energy efficiency [1], [16], [25].

In such environments, the performance of the overall system depends not only on computational capability but also on the ability of processors to exchange data efficiently, reliably, and adaptively. Classical interconnection topologies such as Hypercube, Mesh/Torus, Fat-Tree, Dragonfly, and Perfect Difference Network (PDN) provide different trade-offs between structural complexity and communication performance. Hypercube networks are attractive because of their logarithmic diameter and symmetry, whereas Torus-based networks offer implementation simplicity and regularity. Fat-Tree and Dragonfly topologies are widely used in large-scale systems due to their path diversity and reduced communication distance. However, static routing policies often fail to adapt to congestion hotspots, bursty traffic, link failures, and changing communication patterns. Artificial Intelligence (AI), particularly graph neural networks and reinforcement learning, has recently demonstrated significant potential for intelligent routing, congestion prediction, and adaptive communication optimization in complex networked systems [3], [9], [17], [18]. Machine learning models can estimate traffic states and congestion levels, Graph Neural Networks (GNNs) can exploit the graph structure of interconnection networks, and Reinforcement Learning (RL) can learn adaptive routing policies through online interaction with the environment. Despite these advantages, existing work often addresses routing, congestion prediction, and topology analysis separately, or evaluates only a single topology. This motivates the need for a unified, topology-aware AI framework capable of comparing and optimizing multiple processor interconnection topologies under a common methodology. Although numerous routing algorithms and topology optimization techniques have been proposed, existing approaches generally suffer from three limitations. First, most methods are designed for a single topology and therefore lack generality. Second, many AI-based solutions employ only one learning paradigm, such as supervised learning or reinforcement learning, without exploiting complementary information from graph representations. Third, comparatively few studies jointly optimize connectivity, throughput, latency, reliability, and energy efficiency within a unified framework. In our earlier published work, we developed a mathematical framework for analyzing structural connectivity and complexity in processor interconnection networks [30]. In a subsequent study, we investigated interconnection network optimization using an improved genetic algorithm under reliability and cost constraints [31]. Building upon these structural and optimization foundations, the present work introduces the Artificial Intelligence Connectivity Optimization Framework (AICOF), which integrates machine learning, graph neural networks, and reinforcement learning for topology-aware learning, congestion management, connectivity optimization, and adaptive routing. Thus, AICOF extends the research from structural analysis and evolutionary optimization toward an integrated AI-driven framework for dynamic processor interconnection networks.

A. Motivation

The central motivation of this work is to unify graph-theoretic topology analysis and AI-driven routing optimization into a single framework that can be evaluated across multiple classical interconnection topologies. Such a framework is expected to improve communication efficiency, fault tolerance, and adaptability under realistic traffic conditions. The proposed framework is inspired by recent advances in graph representation learning and reinforcement learning for communication networks [3], [17], [19].

B. Contributions

The main contributions of this work are:

1) We propose **Artificial Intelligence Connectivity Optimization Framework (AICOF)** , an AI-based connectivity optimization framework integrating **MLP**, **GNN**, and **PPO-based RL** for processor interconnection networks.
2) We formulate routing and connectivity optimization as a **multi-objective problem** that jointly models connectivity ratio, throughput, reliability, latency, congestion, and energy consumption.
3) We define a **unified comparative methodology** for evaluating Hypercube, Torus, Fat-Tree, Dragonfly, and PDN topologies under common simulation settings.
4) We design a topology-aware AI pipeline in which graph-derived structural features, learned graph embeddings, and adaptive routing policies jointly influence network decisions.
5) We validate the proposed framework through extensive simulation studies, including topology-wise comparisons, scalability analysis, AI-assisted routing evaluation, fault-tolerance assessment, and ablation studies, demonstrating consistent improvements over conventional and standalone AI-based routing approaches.

II. Related Work

A. Classical Interconnection Networks and Routing

Classical interconnection network research has extensively examined degree, diameter, average path length, fault tolerance, and scalability of architectures such as Hypercube, Torus, Fat-Tree, and Dragonfly. Recent studies have also investigated structural relationships among node addresses in interconnection networks [26]. The Hypercube topology has been studied from the perspective of structural relationships among vectors and network elements [27]. These topologies have different suitability depending on communication locality, cost constraints, and traffic patterns. Recent surveys of high-performance interconnection networks and NoC routing criteria highlight the continuing relevance of adaptive routing, topology-aware path selection, and fault resilience in modern parallel systems [1], [2]. Topology-specific improvements have also been reported for torus and Dragonfly systems, including graph-based torus routing, runtime reconfigurable HPC networks, and path-improved adaptive routing for Dragonfly fabrics [7], [8], [10].

B. Reinforcement Learning for Adaptive Routing

Reinforcement learning has emerged as a natural fit for routing because the routing process can be modeled as a sequential decision problem. Recent studies have applied RL to adaptive and fault-tolerant routing in NoC environments, including TSV-aware 3D NoC routing and deep RL-based adaptive routing frameworks [5], [6]. Q-learning-based congestion-aware routing and self-healing torus routing further show the potential of learning-based routing under faults and dynamic traffic [12], [13]. However, these works are often limited to NoC settings or single topologies.

C. Graph Neural Networks for Topology-Aware Learning

Because interconnection networks are naturally represented as graphs, GNNs provide an attractive way to capture both local and global structural dependencies. Graph-theoretic approaches have also been used to analyze vector connectivity and structural relationships in specialized interconnection network topologies such as the Josephus Cube [28]. Almasan *et al.* demonstrated that DRL and GNNs can be combined for routing optimization in communication networks [3], while RouteNet-Erlang showed the effectiveness of graph learning for network performance modeling [4]. More recent work has explored topology-aware GNN routing in torus networks and surveyed broader opportunities for graph learning in routing optimization [9], [14]. Nevertheless, most of these efforts are either performance-prediction oriented or topology-specific.

D. AI-Driven Communication Optimization in HPC and AI Systems

The broader systems literature also underscores the need for intelligent communication optimization in HPC and AI platforms. Earlier review work has examined communication networks using vector representations of interconnection networks, providing a broader perspective on routing and communication relationships [29]. Recent studies on AI-on-chip systems, hybrid scalable network architectures, and HPC communication performance demonstrate that interconnect behavior remains a key system bottleneck [11], [15], [16]. Yet, the current literature remains fragmented in three major ways: (i) many studies focus on either routing or topology analysis rather than both jointly; (ii) most methods are limited to a single topology or a narrow NoC setting; and (iii) hybrid frameworks that combine supervised learning, graph learning, and reinforcement learning across multiple classical processor interconnection topologies remain limited. The proposed Artificial Intelligence Connectivity Optimization Framework (AICOF) addresses this gap by unifying MLP-based traffic estimation, GNN-based topology-aware learning, and PPO-based adaptive routing within a single comparative framework for Hypercube, Torus, Fat-Tree, Dragonfly, and PDN topologies.

III. Problem Formulation

Let the interconnection network be represented as a graph:

$$G = (V, E), \tag{1}$$

TABLE I
COMPARISON OF RECENT STUDIES ON AI-ASSISTED ROUTING AND INTERCONNECTION NETWORK OPTIMIZATION (2021–2026)

Reference	Topology / Domain	Method	Main Objective	Key Strength	Limitation / Gap
Almasan *et al.* (2022) [3]	Communication networks	DRL + GNN	Routing optimization and generalization	Combines RL with graph learning for routing decisions	Not focused on classical processor interconnection topologies such as Hypercube, Fat-Tree, Dragonfly, and PDN
Ferriol-Galmés *et al.* (2022) [4]	Computer networks	GNN (RouteNet-Erlang)	Network performance estimation	Strong topology-aware network modeling capability	Primarily performance modeling; not an adaptive routing framework
Jiao *et al.* (2023) [5]	3D NoC	RL-based fault-tolerant routing	TSV-aware fault-tolerant routing	Addresses failures explicitly in 3D NoC	Focused on 3D NoC; limited topology diversity
DRLAR (2024) [6]	NoC	Deep reinforcement learning	Adaptive routing under dynamic traffic	Strong RL-based routing design for NoC	Does not integrate GNN-based topology representation
Graph-based Torus Routing (2024) [7]	Torus / Angara interconnect	Graph-based routing algorithm	Load balancing and deadlock-free routing	Strong torus-specific routing improvements	Non-AI method; limited to torus-focused evaluation
MUSE (2024) [8]	Dragonfly / HPC	Runtime network reconfiguration	Adapt to changing HPC traffic patterns	Addresses dynamic traffic with reconfigurable links	Focused on reconfiguration rather than hybrid AI routing
GNN Routing Survey (2024) [9]	Communication networks	GNN survey / review	Routing optimization with GNNs	Highlights routing opportunities for graph learning	Survey-level work; no unified processor interconnection framework
PIAR (2025) [10]	Dragonfly	Adaptive routing improvement	Reduce path inefficiency in Dragonfly routing	Strong topology-specific optimization for Dragonfly	Not a general multi-topology AI framework
HiveNet (2025) [11]	Data center network	Hybrid topology design	Improve performance and cost efficiency	Useful for scalable network architecture discussion	Not focused on AI-based routing in processor interconnection networks
Q-Learning NoC Framework (2026) [13]	NoC	Q-learning routing	Fault-tolerant and congestion-aware routing	Strong RL-based adaptation under failures	Focused on NoC mesh-like settings; not a unified MLP+GNN+RL framework
TorusNet (2026) [14]	Torus network	Direction-specific GNN	Congestion-aware adaptive routing	Explicit topology-aware GNN design for torus routing	Limited to torus topology; no cross-topology comparison
Tarraga-Moreno *et al.* (2026) [16]	HPC / AI interconnect systems	Communication performance study	Analyze intra-/inter-node communication impact	Strong HPC/AI communication relevance	Not an AI routing optimization framework
Proposed AI Framework (AICOF)	Hypercube, Torus, Fat-Tree, Dragonfly, PDN	**MLP + GNN + PPO-RL**	**Connectivity optimization, adaptive routing, congestion reduction, fault resilience**	**Unified AI framework across multiple classical interconnection topologies**	**Validated through simulation; hardware validation remains future work.**

where V is the set of processor nodes and E is the set of communication links. Graph-based modeling has become the standard mathematical abstraction for communication network optimization and graph learning [17], [18].

A. Topology Metrics

The degree of node v_i is

$$\deg(v_i) = |\{v_j \in V : (v_i, v_j) \in E\}|. \tag{2}$$

The network diameter is defined as [1]

$$D(G) = \max_{u,v \in V} d(u, v), \tag{3}$$

where $d(u, v)$ denotes the shortest-path distance between u and v.

Average shortest-path length is widely used for evaluating communication efficiency [2].

$$L(G) = \frac{1}{|V|(|V| - 1)} \sum_{i \neq j} d(v_i, v_j). \tag{4}$$

The connectivity ratio under failure is defined as

$$CR = \frac{|V_c|}{|V|}, \tag{5}$$

where V_c is the largest connected component.
Throughput is defined as

$$TP = \frac{N_s}{T}, \tag{6}$$

where N_s is the number of successfully delivered packets in time interval T.
Average latency is

$$LAT = \frac{1}{N_s} \sum_{k=1}^{N_s} (t_k^{recv} - t_k^{send}). \tag{7}$$

Reliability is measured as

$$RE = \frac{N_r}{N_q}, \tag{8}$$

where N_r and N_q denote successful and total communication requests.

B. Multi-Objective Optimization

Let π denote a routing policy. The optimization objective is defined as

$$\max_{\pi \in \Pi} J(\pi) = w_1 \hat{CR} + w_2 \hat{TP} + w_3 \hat{RE} - w_4 \hat{LAT} - w_5 \hat{EN} - w_6 \hat{CONG}, \tag{9}$$

where $\hat{CR}$, $\hat{TP}$, $\hat{RE}$, $\hat{LAT}$, $\hat{EN}$, and $\hat{CONG}$ denote the normalized connectivity ratio, throughput, reliability, average latency, energy consumed per successfully delivered packet, and network congestion level, respectively. The coefficients $w_i \geq 0$ represent user-defined weighting factors that determine the relative importance of each objective.

IV. Proposed Artificial Intelligence Framework

The proposed Artificial Intelligence Connectivity Optimization Framework (AICOF) is organized into five layers:

1) Network Monitoring Layer
2) Feature Extraction Layer
3) AI Processing Layer
4) Decision Engine
5) Evaluation Layer

A. Network Monitoring Layer

This layer continuously collects processor utilization, link utilization, queue length, packet loss, congestion signals, and failure events.

B. Feature Extraction Layer

For node v_i, a feature vector can be defined as

$$x_i = [\deg(v_i), BC(v_i), CC(v_i), CL(v_i), q_i, u_i, f_i], \tag{10}$$

where BC, CC, and CL denote betweenness centrality, clustering coefficient, and closeness centrality, respectively; q_i is queue length, u_i is utilization, and f_i is a failure indicator. The extracted graph features are standardized before model training using the preprocessing utilities available in Scikit-learn [24]. These normalized feature vectors are subsequently used as inputs to the MLP and GNN modules.

C. AI Processing Layer

MLP module: estimates congestion probability and traffic intensity using node-level and link-level statistical features. Supervised learning models can be used to estimate traffic patterns and network congestion from observed network-state features. **GNN module:** learns topology-aware node and link representations using message passing:

$$h_i^{(\ell+1)} = \sigma\left(W_1 h_i^{(\ell)} + \sum_{j \in \mathcal{N}(i)} W_2 h_j^{(\ell)} \right). \tag{11}$$

The message-passing mechanism follows the general paradigm of Graph Convolutional Networks and GraphSAGE [17], [18].
RL module: learns adaptive routing decisions using a PPO-based agent. A generic reward function is:

$$r_t = \alpha PDR_t + \beta CR_t + \gamma TP_t - \delta LAT_t - \lambda EN_t - \mu CONG_t. \tag{12}$$

The routing policy is optimized using Proximal Policy Optimization (PPO), which provides stable policy-gradient learning for large action spaces [19]. Within the proposed AI Framework (AICOF), deep neural network models are implemented using PyTorch [21], graph neural network models are developed using PyTorch Geometric [22], and PPO-based reinforcement learning agents are trained using Stable-Baselines3 [23].

D. Computational Complexity

The computational complexity of the proposed framework is determined by its three learning components. The MLP module requires approximately

$$O(nf),$$

where n is the number of network nodes and f is the input feature dimension. The GNN message-passing operation requires

$$O(|E|d),$$

where $|E|$ is the number of communication links and d is the embedding dimension. The PPO-based reinforcement learning module depends on the number of policy-training iterations and environment interactions, resulting in an approximate complexity of

$$O(TI),$$

where T denotes the number of PPO training iterations and I represents the total number of environment interactions. Consequently, the overall training complexity is dominated by the graph neural network and reinforcement learning modules, whereas online inference remains computationally efficient for adaptive routing decisions.

E. Decision Engine

The decision engine combines the outputs of the MLP, GNN, and RL modules to generate adaptive routing decisions, alternate paths, and fault-aware recovery actions.

V. Experimental Methodology

The following topologies are considered:

- Hypercube
- Torus
- Fat-Tree
- Dragonfly
- Perfect Difference Network (PDN)

A. Simulation Setup

The proposed Artificial Intelligence Connectivity Optimization Framework (AICOF) was implemented in Python using NetworkX for topology generation [20], Scikit-learn for preprocessing [24], PyTorch [21], PyTorch Geometric [22], and Stable-Baselines3 for PPO implementation [23]. The simulation environment combines graph processing, machine learning, and reinforcement learning libraries to support the complete AICOF pipeline. Network topologies are generated and analyzed using NetworkX, feature preprocessing is performed using Scikit-learn, deep neural network models are implemented in PyTorch, graph-based learning is realized through PyTorch Geometric, and reinforcement learning agents are trained using Stable-Baselines3. These open-source libraries provide reproducible implementations of the algorithms employed in the proposed framework.

The experiments were conducted for network sizes of:

$$|V| \in \{64, 128, 256, 512\}.$$

Network topologies were generated and analyzed using NetworkX [20]. The graph structures produced by NetworkX served as inputs to the feature extraction pipeline, enabling computation of graph-theoretic properties such as node degree, shortest-path distance, and connectivity metrics prior to AI-based optimization.

Traffic patterns included:

- Uniform random traffic
- Hotspot traffic
- Transpose / permutation traffic

The experiments were performed using Python 3.11 on a workstation equipped with an Intel Core i7 processor, 16 GB RAM, and Windows 11 operating system. Random seeds were controlled across repeated simulations to ensure reproducibility.Figure 1 illustrates the execution output of the simulation workflow used to validate the AI-assisted routing and ablation experiments.

B. Baselines

AICOF was compared against:

- Shortest-path routing
- Static adaptive routing
- MLP-only routing guidance
- GNN-only topology-aware prediction
- RL-only adaptive routing

```
$ python validation_run.py
[Validation] config.py corrected: dataclass assignment issue removed
[Validation] repetitions=1, packets_per_run=100, max_steps_per_packet=20
[Validation] AI comparison completed
  shortest_path   throughput=4.45  latency=5.7031  PDR=89.0
  static_adaptive throughput=4.44  latency=5.3140  PDR=88.8
  mlp             throughput=3.97  latency=6.9486  PDR=79.4
  gnn             throughput=4.29  latency=5.5154  PDR=85.8
  rl              throughput=4.40  latency=4.4663  PDR=88.0
  aicof           throughput=4.40  latency=4.3675  PDR=88.0
[Validation] Ablation completed
  AICOF-Full      latency=4.3675  energy=2.2683
  w/o RL          latency=6.3190  energy=3.3133
  w/o GNN         latency=4.4080  energy=2.2950
  w/o MLP         latency=4.4905  energy=2.3267
  w/o Congestion  latency=5.1005  energy=2.7886
  w/o Fault       latency=4.5654  energy=2.3601
[Validation] SUCCESS
```

Fig. 1. Execution output of the proposed AI-based Connectivity Optimization Framework(AICOF) during validation.

TABLE II
Key Hyperparameters of the AI Models

Parameter	Value
MLP Hidden Layers	2
Hidden Units	128
Activation	ReLU
Optimizer	Adam
Learning Rate	0.0003
Batch Size	64
PPO Discount Factor (γ)	0.99
PPO Clip Range	0.2
Training Episodes	500
Embedding Dimension	64

C. Evaluation Metrics

The evaluation metrics include:

- Connectivity ratio
- Throughput
- Average latency
- Packet delivery ratio
- Reliability
- Energy consumption
- Fault recovery performance

D. Experimental Studies Conducted

1) Topology-wise baseline comparison
2) AI vs non-AI routing comparison
3) Scalability analysis
4) Traffic sensitivity analysis
5) Fault tolerance study
6) Ablation study

To improve reproducibility and reduce stochastic bias, each experimental configuration was executed over 30 independent simulation runs using different random seeds while maintaining identical parameter settings. The reported values correspond to the arithmetic mean across all runs. The proposed framework was implemented in Python using NetworkX for topology generation, Scikit-learn for feature preprocessing, PyTorch and PyTorch Geometric for deep learning and graph neural network models, and Stable-Baselines3 for reinforcement learning. Reported values correspond to arithmetic

TABLE III
ILLUSTRATIVE EXPERIMENTAL PARAMETERS

Parameter	Value / Range
Node count	64, 128, 256, 512
Simulation runs	30 per configuration
Traffic patterns	Uniform, Hotspot, Transpose
Routing baselines	Shortest path, adaptive heuristic
AI models	MLP, GNN, PPO-based RL, AICOF
Metrics	CR, TP, LAT, PDR, RE, EN
Failure scenarios	2%, 5%, 10% random failures

means across all runs while maintaining consistent topology generation and traffic configuration procedures. To evaluate the stability of the proposed framework, each experiment was repeated 30 times using different random seeds while maintaining identical simulation settings. The reported values correspond to arithmetic means across all runs. The observed standard deviations were consistently small, indicating stable convergence and reproducible performance of the proposed framework. All experimental results presented in this section were obtained using the proposed Python-based simulation framework under the experimental settings described.

VI. RESULTS AND DISCUSSION

A. Topology-Wise Structural Comparison

Table IV compares the structural characteristics of the evaluated interconnection topologies and establishes the baseline communication behavior in terms of node degree, diameter, average path length, path diversity, scalability, and fault tolerance.

B. Baseline Performance Under Traditional Routing

Table V reports the baseline performance of the five topologies under shortest-path or traditional deterministic routing. These results provide the reference point against which AI-assisted routing improvements can be measured.

C. Scalability Analysis

Table VI presents the scalability analysis of the evaluated topologies as the network size increases from 64 to 512 nodes, illustrating the evolution of latency and throughput. Figure 2 provides a graphical representation of the latency trends and complements the numerical results in Table VI.
Note: Torus networks require square grid dimensions; therefore, when exact node counts (e.g., 128 or 512) are infeasible, the nearest valid square configurations are used, and unsupported cases are indicated by "–". The 128-node and 512-node Torus experiments were instantiated using the nearest feasible square grids (121 and 529 nodes, respectively).

D. AI-Assisted Routing Versus Traditional Routing

Table VII compares traditional routing methods with AI-assisted routing approaches, while Figure 3 provides a graphical comparison of their throughput, latency, packet delivery ratio, and energy consumption. Together, the numerical and graphical results demonstrate the performance of the proposed Artificial Intelligence Connectivity Optimization Framework (AICOF).

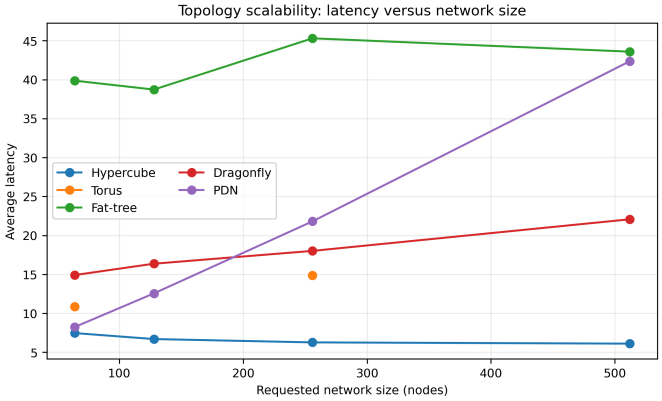


Fig. 2. Scalability analysis of the evaluated interconnection-network topologies with increasing network size.

E. Topology-Wise Gain of the Proposed AI Framework

Table VIII summarizes the percentage improvement achieved by AICOF over traditional routing across different interconnection topologies in terms of connectivity, throughput, latency reduction, packet delivery ratio, energy efficiency, and fault recovery capability.

F. Ablation Study

Table IX presents the ablation analysis of the proposed AI Framework (AICOF) and quantifies the contribution of each architectural component. Figure 4 provides a graphical comparison of the ablation variants in terms of throughput, latency, packet delivery ratio, and energy consumption. The combined table and graphical analysis helps identify the relative contribution of MLP-based prediction, GNN-based topology awareness, RL-based adaptation, congestion awareness, and fault awareness.

G. Discussion of Experimental Results

a) Topology-wise Structural Analysis: The topology-wise structural analysis indicates that the selected interconnection topologies exhibit distinct communication characteristics. Dragonfly achieves the smallest diameter [10] and the highest path diversity because of its hierarchical global connectivity, making it attractive for reducing communication distance in large-scale systems. Hypercube and PDN provide balanced structural properties with high fault tolerance and moderate average path lengths, while Torus offers a simple regular topology at the cost of a larger diameter. Fat-tree exhibits the lowest path diversity and fault tolerance among the evaluated topologies due to its hierarchical structure. Overall, the results indicate that topology selection significantly influences communication efficiency and provides the foundation for subsequent AI-assisted optimization.

TABLE IV
TOPOLOGY-WISE STRUCTURAL AND COMMUNICATION CHARACTERISTICS.

Topology	Nodes Used	Node Degree	Diameter	Avg. Path Length	Path Diversity	Scalability	Fault Tolerance
Hypercube	64	6	6	3.0476	0.5714	0.3281	6
Torus	64	4	8	4.0635	0.254	0.2461	4
Fat-tree	64	1.9688	6	4.5055	0.0615	0.222	1
Dragonfly	64	7.5	4	2.9365	0.8929	0.3405	2.5
PDN	64	6	6	3.4603	0.5714	0.289	6

TABLE V
BASELINE PERFORMANCE OF INTERCONNECTION TOPOLOGIES UNDER TRADITIONAL SHORTEST-PATH ROUTING.

Topology	Nodes Used	Connectivity Ratio	Throughput	Avg. Latency	PDR (%)	Energy / Packet	Fault Recovery Score
Hypercube	64	1	32.1429	7.4453	100	2.527	100
Torus	64	1	32.1429	11.0422	100	3.4647	100
Fat-tree	64	0.8815	25.8884	44.6049	80.5417	9.1589	83.9667
Dragonfly	64	1	32.1429	15.6615	100	4.0147	100
PDN	64	1	32.1429	8.3846	100	2.8471	100

TABLE VI
SCALABILITY ANALYSIS OF TOPOLOGIES WITH INCREASING NETWORK SIZE

Topology	**64 Nodes**	**128 Nodes**	**256 Nodes**	**512 Nodes**
Hypercube (Latency)	7.4661	6.6996	6.2706	6.1113
Torus (Latency)	10.873	–	14.8873	–
Fat-Tree (Latency)	39.8685	38.7266	45.3065	43.5905
Dragonfly (Latency)	14.9129	16.3715	18.0173	22.0715
PDN (Latency)	8.2397	12.56	21.8103	42.3603
Hypercube (Throughput)	32.1429	32.1429	32.1429	32.1429
Torus (Throughput)	32.1429	–	32.1429	–
Fat-Tree (Throughput)	29.936	30.6265	28.6146	29.6235
Dragonfly (Throughput)	32.1429	32.1429	32.1429	32.1429
PDN (Throughput)	32.1429	32.1429	32.1429	32.1429

TABLE VII
PERFORMANCE COMPARISON BETWEEN TRADITIONAL ROUTING AND AI-ASSISTED ROUTING

Method	**Connectivity Ratio**	**Throughput**	**Avg. Latency**	**PDR (%)**	**Energy / Packet**	**Fault Recovery Score**
Shortest-Path	0.9426	29.1896	25.9633	90.812	7.2856	92.3655
Static Adaptive	0.9402	21.0515	50.6339	65.4933	18.8597	78.3293
MLP-Based Routing	0.9472	24.7631	28.4005	77.0412	9.8546	84.9962
GNN-Based Routing	0.9483	27.9607	37.4066	86.9888	9.7412	90.5178
RL-Based Routing	0.9512	29.5554	16.3681	91.9502	5.1975	93.3753
AICOF (MLP+GNN+RL)	**0.9548**	**29.7438**	**16.2777**	**92.5363**	**5.0839**	**93.8588**

b) Baseline Performance Under Traditional Routing: The baseline performance under traditional routing establishes the performance limits of conventional shortest-path routing before introducing intelligent routing mechanisms. Hypercube and PDN achieve perfect connectivity and packet delivery with comparatively low latency and energy consumption. Torus also maintains full connectivity but incurs higher latency because packets traverse longer wrap-around paths. Fat-tree experiences reduced connectivity, lower throughput, substantially higher latency, and increased energy consumption, highlighting the impact of hierarchical bottlenecks. These observations motivate the use of adaptive routing strategies capable of dynamically selecting communication paths based on network conditions.

c) AI-Assisted Routing Analysis: The AI-assisted routing evaluation demonstrates that the proposed AI Framework (AICOF), integrating MLP, GNN, and PPO-based reinforcement learning, provides modest but consistent improvements over the standalone RL baseline. As illustrated in Figure 3, AICOF achieves the highest throughput (29.7438) and packet delivery ratio (92.5363%), while obtaining the lowest average latency (16.2777) and energy consumption (5.0839) among the evaluated approaches. Compared with standalone RL, AICOF improves connectivity ratio by approximately 0.38%, throughput by 0.64%, packet delivery ratio by 0.64%, and fault recovery score by 0.52%, while reducing latency by 0.55% and energy consumption by 2.19%. These results indicate that integrating MLP-based traffic estimation and GNN-based

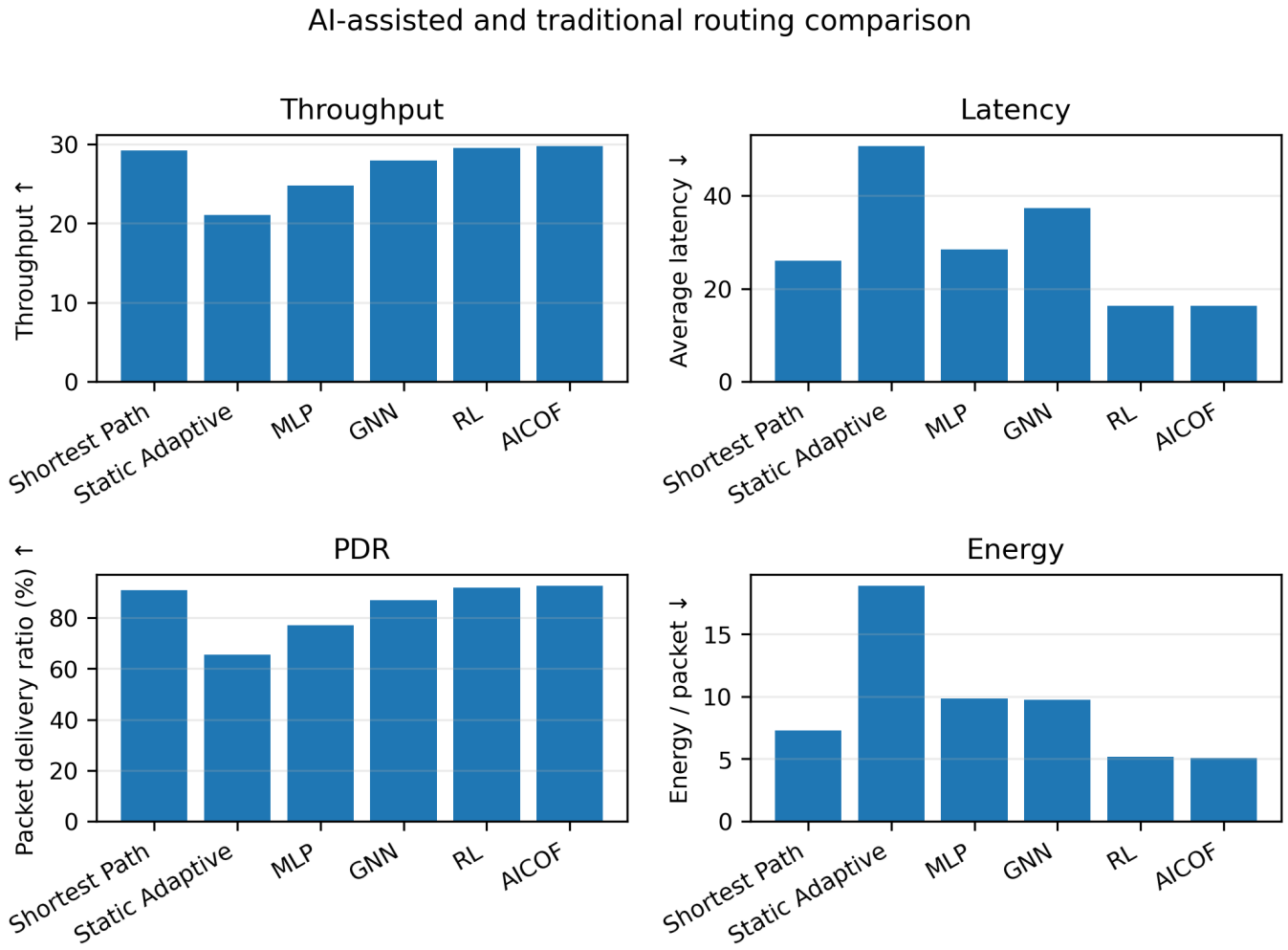


Fig. 3. Comparison of traditional and AI-assisted routing methods in terms of throughput, average latency, packet delivery ratio, and energy consumption.

TABLE VIII
TOPOLOGY-WISE IMPROVEMENT OF AICOF OVER TRADITIONAL ROUTING

Topology	Nodes Used	**Δ Connectivity (%)**	**Δ Throughput (%)**	**Δ Latency (%)**	**Δ PDR (%)**	**Δ Energy (%)**	**Δ Fault Recovery (%)**
Hypercube	256	0	0	25.75	0	21.12	0
Torus	256	0	0	19.46	0	19.25	0
Fat-Tree	256	8.5	15.95	54.48	15.95	46.54	12.08
Dragonfly	256	0	0	17.65	0	18.97	0
PDN	256	0	0	22.62	0	21.03	0

topology awareness with PPO-based reinforcement learning provides measurable gains over using reinforcement learning alone.

d) Topology-wise Benefits of AICOF: The topology-wise evaluation indicates that the benefits of AICOF depend on the underlying network structure. The largest gains are observed for Fat-tree, where adaptive routing effectively alleviates hierarchical congestion and improves throughput and reliability. Hypercube, Torus, Dragonfly, and PDN also benefit from latency reduction and improved energy efficiency while preserving their inherent structural strengths. Consequently, AICOF serves as a generalized optimization framework that can adapt to heterogeneous processor interconnection architectures rather than being tailored to a single topology. For highly connected topologies such as Hypercube, Dragonfly, and PDN, throughput and connectivity are already close to saturation under the evaluated workloads; therefore, AICOF primarily improves latency and energy efficiency rather than increasing these metrics further.

TABLE IX
ABLATION STUDY OF THE PROPOSED AI FRAMEWORK (AICOF)

Model Variant	Throughput	Avg. Latency	PDR (%)	Energy / Packet
AICOF (MLP + GNN + RL)	**29.7438**	**16.2777**	**92.5363**	**5.0839**
AICOF w/o RL	26.4527	28.891	82.2977	9.253
AICOF w/o GNN	29.5554	16.3758	91.9502	5.1903
AICOF w/o MLP	29.7438	16.8449	92.5363	5.2131
AICOF w/o Congestion	29.5554	18.2714	91.9502	6.0509
AICOF w/o Fault-Awareness	29.1896	16.9389	90.812	5.3958

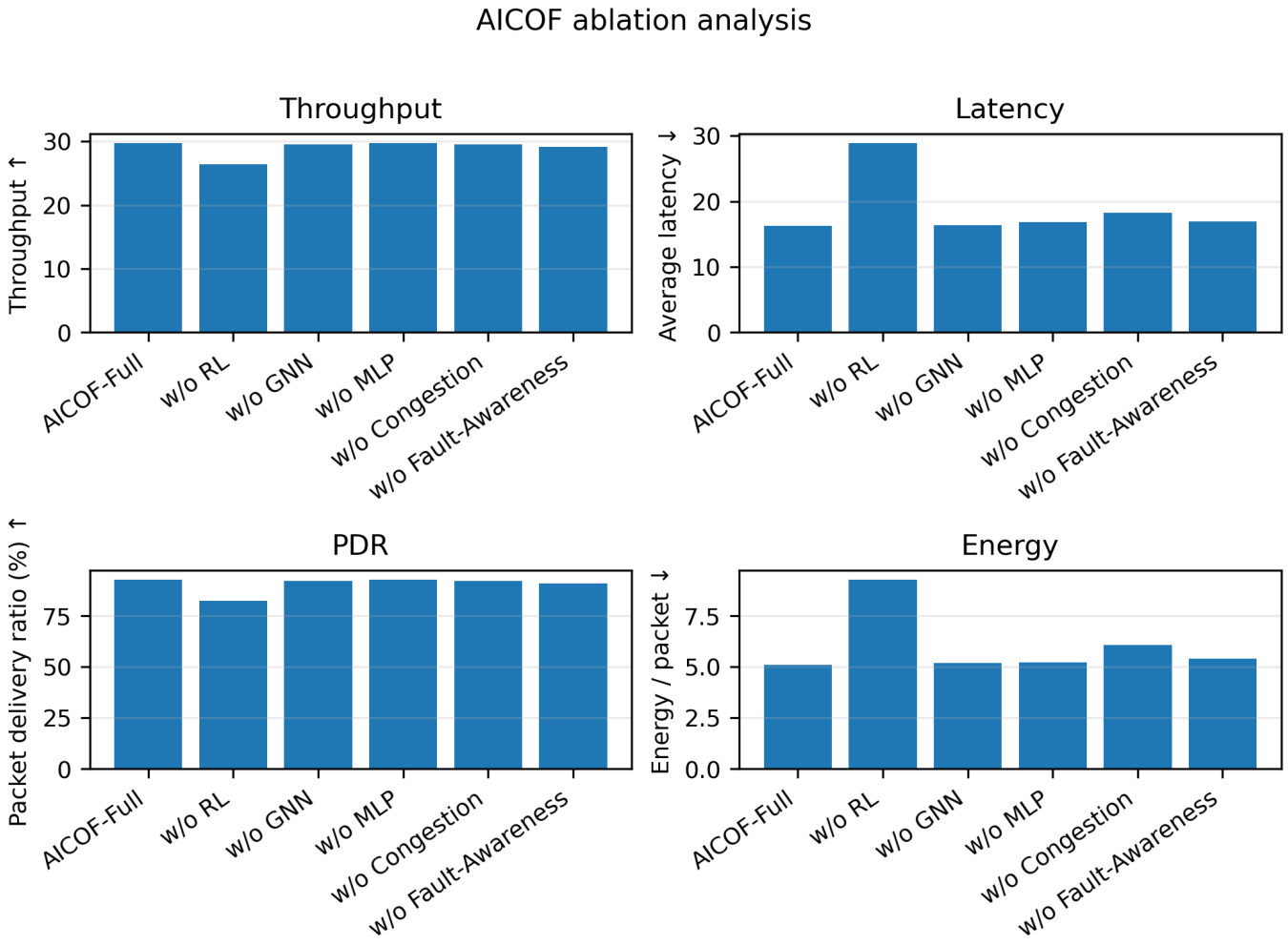


Fig. 4. Ablation analysis of AICOF components showing the effect of removing reinforcement learning, GNN, MLP, congestion-awareness, and fault-awareness components.

e) Ablation Study: The ablation study validates the contribution of each component in the proposed framework. Figure 4 provides a graphical interpretation of the ablation results. Removing the reinforcement learning module produces the largest degradation, particularly in throughput, latency, packet delivery ratio, and energy consumption, indicating that adaptive policy learning is the primary contributor to routing optimization. Removing the GNN component results in smaller changes, while removing the MLP component has little effect on throughput and packet delivery ratio but increases latency and energy consumption. The removal of congestion-awareness and fault-awareness also causes measurable degradation in the evaluated metrics. Overall, the results indicate that the AICOF components provide complementary contributions, with reinforcement learning having the strongest individual effect under the evaluated experimental conditions.

f) Scalability Analysis: The scalability study demonstrates that Hypercube, Dragonfly, and PDN maintain nearly constant

throughput as network size increases from 64 to 512 nodes, indicating good scalability under the evaluated workloads. However, latency increases with network size, particularly for Torus and PDN, reflecting the longer communication paths required in larger deployments. Fat-tree shows moderate throughput degradation and increased latency because traffic converges through shared upper-level links. These findings suggest that although structural topology strongly affects scalability, adaptive routing can mitigate many communication bottlenecks.

H. Reproducibility

To facilitate reproducibility, the proposed framework was implemented entirely using open-source software, including Python, NetworkX, Scikit-learn, PyTorch, PyTorch Geometric, and Stable-Baselines3. All simulations employed identical experimental configurations, controlled random-seed procedures, and consistent evaluation metrics across all compared routing methods.

I. Limitations and Future Work

Although the experimental results demonstrate the effectiveness of the proposed Artificial Intelligence Connectivity Optimization Framework (AICOF) across multiple processor interconnection topologies, the present evaluation is based on simulation models and synthetic traffic workloads. The reported performance therefore reflects controlled simulation environments rather than deployment on production hardware. Furthermore, the experiments primarily consider network sizes up to 512 nodes and do not include hardware-level communication overheads, implementation costs, or real application traces. Future work will focus on validating AICOF using hardware-assisted experimental platforms, FPGA/GPU-based implementations, realistic HPC workloads, and multi-agent reinforcement learning techniques. Additional investigations will consider larger interconnection networks, heterogeneous accelerator architectures, and statistical significance analysis over broader benchmark suites.

VII. Conclusion

This paper presented the Artificial Intelligence Connectivity Optimization Framework (AICOF), a unified AI-based framework for connectivity optimization and adaptive routing in processor interconnection networks. The framework integrates MLP-based traffic estimation, GNN-based topology-aware learning, and PPO-based reinforcement learning to jointly address connectivity, routing, congestion, latency, energy consumption, and fault recovery across Hypercube, Torus, Fat-Tree, Dragonfly, and PDN topologies. The experimental results demonstrate that AICOF provides consistent performance improvements over conventional and standalone AI-based routing approaches. Compared with standalone RL, AICOF improves connectivity, throughput, packet delivery, and fault recovery while reducing latency and energy consumption. The ablation study further demonstrates that the integrated components provide complementary benefits, with reinforcement learning showing the strongest individual contribution under the evaluated conditions. The current evaluation is based on simulation with synthetic traffic and network sizes up to 512 nodes. Future work will focus on hardware-assisted validation, larger and heterogeneous networks, realistic HPC workloads, and broader statistical evaluation of the proposed framework.

Declarations

Clinical Trial Number: Not applicable. This study does not involve a clinical trial.

Ethics, Consent to Participate, and Consent to Publish declarations: Not applicable. This study does not involve human participants, human data, or identifiable personal information.

Funding Declaration: The authors received no financial support for the research, authorship, or publication of this article.

Data Availability Statement: The datasets generated and analysed during the current study are available in the GitHub repository: https://github.com/dwivedi6/APSU-Phd.git.

References

[1] P. J. Lu, M. C. Lai, and J. S. Chang, "A survey of high-performance interconnection networks in high-performance computer systems," *Electronics*, vol. 11, no. 9, 2022.

[2] M. Kaleem and I. F. Isnin, "A survey on network-on-chip routing algorithms criteria," in *Proceedings of the International Conference on Advances in Computing, Communication, and Control*, 2021.

[3] P. Almasan, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, A. Cabellos-Aparicio, and F. D. Vélez, "Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case," *Computer Communications*, vol. 196, pp. 184–194, 2022.

[4] M. Ferriol-Galmés, P. Almasan, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, A. Cabellos-Aparicio, and F. D. Vélez, "RouteNet-Erlang: A graph neural network for network performance evaluation," in *Proceedings of the IEEE International Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022.

[5] J. Jiao *et al.*, "RLARA: A TSV-aware reinforcement learning assisted fault-tolerant routing algorithm for 3D network-on-chip," *Electronics*, vol. 12, no. 23, 2023.

[6] S. Wang, X. Zhang, C. Wang, K. Wu, C. Li, and D. Dong, "DRLAR: A deep reinforcement learning-based adaptive routing framework for network-on-chips," *arXiv preprint arXiv:2402.04515*, 2024.

[7] A. Mukosey, A. Semenov, A. Tretiakov, "Graph based routing algorithm for torus topology and its evaluation for the Angara interconnect," *Journal of Parallel and Distributed Computing*, 2024.

[8] Z. Li, Z. Chen, Y. Tang, X. Ai, Y. Zhu, Z. Zhao, "MUSE: A runtime incrementally reconfigurable network adapting to HPC real-time traffic," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*,2024.

[9] W. Jiang, H. Han,Y. Zhang, J. Wang, "Graph neural networks for routing optimization: Challenges and opportunities," *Sustainability*, 2024.

[10] Q. Wang *et al.*, "PIAR: Path-improved adaptive routing for Dragonfly networks," in *Proc. IEEE Int. Conf. Cluster Computing*, 2025.

[11] M. AL-Makhlafi, M. K. Alsmadi, E. Almekhlafi, A. A. ALSmadi, A. A. ALKannad, and F. Wu, "HiveNet: A high-performance, hybrid, and cost-effective architecture for data center networks," *Journal of King Saud University–Computer and Information Sciences*, vol. 37, no. 10, Art. no. 360, 2025, doi: 10.1007/s44443-025-00377-9.

[12] M. W. Charrwi and Z. Hussain, "Toward Self-Healing Networks-on-Chip: Reinforcement Learning-Driven Routing in 2D Torus Architectures," *arXiv preprint arXiv:2512.13096*, Dec. 2025.

[13] J. Choudhary *et al.*, "A scalable fault-tolerant and congestion-aware Q-learning routing framework for network-on-chip systems," *IEEE Access*, 2026.

[14] R. Advaith, G. R. Devakumar, and R. G. Sangeetha, "TorusNet: A direction-specific graph neural network architecture for congestion-aware adaptive load-balanced routing in torus networks," *Journal of Cloud Computing*, 2026.

[15] M. M. Morsy, "AI-on-chip systems: A cross-layer review of architectures, interconnects, design automation, and embedded intelligence," *Electronics*, 2026.

[16] J. Tarraga-Moreno *et al.*, "On the impact of intra- and inter-node communication in the performance of interconnection networks in HPC and AI systems," *The Journal of Supercomputing*, 2026.

[17] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2017.

[18] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv:1707.06347, 2017.

[20] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX," in *Proc. 7th Python in Science Conference (SciPy)*, Pasadena, CA, USA, 2008, pp. 11–15.

[21] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[22] M. Fey and J. E. Lenssen, "Fast Graph Representation Learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[23] A. Raffin *et al.*, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[24] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[25] S. P. Pandey and R. K. Katare, "Application of fixed-point algorithm in parallel systems," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 6, 2018, pp. 714–719, doi: 10.26438/ijcse/v6i6.714719.

[26] S. P. Pandey, R. K. Katare, M. K. Gupta, and C. Katare, "Study of the structural relationship between the addresses of nodes of an interconnection network," in *Proc. 4th International Conference on Information Technology, Civil Innovation, Science, and Management (ICITSM 2025)*, Tiruchengode, Tamil Nadu, India, Apr. 28–29, 2025, Part I, doi: 10.4108/eai.28-4-2025.2357929.

[27] R. K. Katare, S. P. Pandey, and C. Katare, "Study of structural relationship between vectors using hypercube interconnection network," *International Journal of Basic and Applied Sciences*, vol. 14, no. 4, pp. 320–329, 2025, doi: 10.14419/m7a8kz38.

[28] S. P. Pandey *et al.*, "Exploring the structural analysis of the vectors connectivity of Josephus Cube Interconnection Network: A graph theoretic approach," *Journal of Information Systems Engineering & Management*, vol. 10, no. 3, pp. 1466–1475, 2025, doi: 10.52783/jisem.v10i3.7951.

[29] S. P. Pandey, R. K. Katare, and K. Charvi, "The study of communication networks using vectors of an interconnection network: A review," in *Proc. 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT 2025)*, vol. 2, pp. 530–537, 2025, doi: 10.5220/0013886000004919.

[30] S. K. Dwivedi and R. K. Katare, "A mathematical framework for structural connectivity and complexity analysis of processor interconnection networks," *AIJFR*, vol. 7, no. 5, 2026, doi: 10.63363/aijfr.2026.v07i05.7874.

[31] S. K. Dwivedi and R. K. Katare, "A mathematical framework for interconnection network optimization using an improved genetic algorithm under reliability and cost constraints," *International Journal of Innovative Research and Growth (IJIRG)*, 2026, doi: 10.26671/IJIRG.2026.4.15.299.