ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)

Available Online at www.ijetcr.org

Volume 14; Issue 02; 2026; Page No. 30-39

Self-Optimizing Network Routing using Reinforcement Learning in Dynamic Distributed Systems

Manthan Sharma[1], Harish Dutt Sharma[2], Pintu Kumar[2]

[1]*Research Scholar, School of Computer Engineering and Applications, Maya Devi University, Dehradun, 248011, India.*

[2]*School of Computer Engineering and Applications, Maya Devi University, Dehradun, 248011, India*

Email-ID: techmanthansharma@gmail.com

Email-ID: sharma.harish106@gmail.com

Email-ID: pin295kumar@gmail.com

Conflicts of interest: Nil

Corresponding author: Harish Dutt Sharma

Abstract

Dynamic distributed systems often experience fluctuating network conditions, varying traffic loads, and unpredictable node behavior, which make efficient routing a challenging task. Traditional routing protocols rely on static or heuristically defined rules that may not adapt effectively to such dynamic environments. This paper presents a self-optimizing network routing framework based on reinforcement learning that enables routing agents to learn optimal forwarding strategies through continuous interaction with the network environment. By observing network states such as congestion levels, latency, and link availability, the proposed approach dynamically adjusts routing decisions to improve overall network performance. The reinforcement learning model continuously updates its policy to minimize packet delay, reduce congestion, and enhance reliability. Experimental evaluation in simulated distributed network scenarios demonstrates that the proposed approach achieves improved adaptability and better routing efficiency compared to conventional routing methods. The results highlight the potential of reinforcement learning for building intelligent and autonomous routing mechanisms in modern dynamic distributed systems.

Keywords: Reinforcement Learning, Network Routing, Dynamic Distributed Systems, Self-Optimizing Networks.

1. Introduction

Modern distributed systems operate in highly dynamic environments where network conditions frequently change due to varying traffic loads, node mobility, link failures, and fluctuating resource availability. Efficient network routing in such environments is a critical challenge, as traditional routing protocols generally rely on static configurations or predefined heuristics that do not adapt effectively to rapidly changing network states [1][2]. With the rapid growth of cloud computing, Internet of Things (IoT), and large-scale distributed applications, network infrastructures are required to handle massive data traffic while maintaining reliability and low latency. Conventional routing techniques often fail to optimize network performance under unpredictable conditions such as congestion, packet loss, and dynamic topology changes [3][4]. Recent advancements in artificial intelligence and machine learning have introduced new possibilities for autonomous network management. Among these techniques, reinforcement learning (RL) has emerged as a promising approach for adaptive decision making in complex environments. Reinforcement learning enables agents to learn optimal policies through continuous interaction with the environment while receiving feedback in the form of rewards or penalties [5]. In the context of networking, RL-based routing mechanisms allow network nodes to dynamically adjust routing decisions based on real-time network states such as latency, congestion level, and link quality. Such intelligent routing mechanisms have demonstrated significant improvements in network efficiency, load balancing, and congestion avoidance compared to traditional routing methods [6]. Furthermore, modern distributed networks such as cloud-edge infrastructures and large-scale data centers require routing mechanisms that can operate autonomously with minimal human intervention. Self-optimizing networks aim to achieve this objective by integrating intelligent learning mechanisms that continuously monitor network behavior and adapt routing strategies accordingly. By leveraging learning-based approaches, network systems can automatically adjust to environmental changes and maintain stable performance even under highly dynamic operating conditions. Another important aspect of intelligent routing is the ability to support scalability and robustness in distributed environments. As the number of network nodes and applications increases, routing decisions become more complex and resource-intensive. Reinforcement learning–based routing provides a scalable solution by allowing distributed agents to independently learn optimal forwarding strategies while collectively improving overall network performance [7].

Motivated by these advancements, this work proposes a reinforcement learning–based self-optimizing routing framework designed for dynamic distributed systems. The proposed approach enables routing agents to continuously learn and adapt routing strategies in order to minimize packet delay, reduce congestion, and improve overall network performance.

The remainder of this paper is organized as follows. Section 2 reviews existing work on intelligent routing and reinforcement learning in networking. Section 3 presents the system model and problem formulation. Section 4 describes the proposed reinforcement learning–based routing methodology. Section 5 discusses experimental results and performance evaluation. Finally, Section 6 concludes the paper and outlines future research directions.

2. Related Work

Recent advancements in machine learning and intelligent optimization techniques have significantly influenced the design of modern distributed systems and network infrastructures. Several studies have demonstrated that machine learning models can improve prediction accuracy, system adaptability, and overall performance in complex computational environments. For example, Nagesh et al. proposed a boosting-enabled machine learning framework for accurate prediction in precision agriculture systems, illustrating how intelligent learning models can efficiently handle large-scale data-driven environments [8]. Such approaches highlight the importance of adaptive learning mechanisms in dynamic systems.

Optimization algorithms have also been widely applied to solve complex engineering and networking problems. Budaraju and Nagesh introduced an improvised cuckoo search optimization algorithm for multi-level image thresholding, demonstrating how swarm intelligence techniques can improve computational efficiency and solution quality in complex optimization tasks [9]. Similar optimization-based approaches have been applied in distributed environments where intelligent decision-making is required to improve system performance.

In the context of IoT and distributed networking, several researchers have focused on improving network reliability and fault tolerance. Jammalamadaka et al. proposed a multi-layered IoT network architecture that enhances fault tolerance using rectangular and interstitial mesh structures in gateway layers [10]. Likewise, Sastry et al. developed an efficient path-finding algorithm using dual base stations to improve network resilience and routing efficiency in IoT networks [11]. These studies emphasize the need for intelligent routing strategies capable of adapting to changing network conditions.

Security and data management in distributed environments have also received significant research attention. Hazzazi et al. proposed an improved cryptographic technique based on finite state machines for enhancing data security in distributed systems [12]. Similarly, Attuluri et al. developed a digital watermarking-based framework to defend against phishing attacks in cloud computing environments, demonstrating the importance of secure communication mechanisms in modern distributed infrastructures [13].

Furthermore, machine learning techniques have been widely used for extracting useful patterns from large datasets. Budaraju and Jammalamadaka explored techniques for mining negative associations from medical databases by considering frequent and closed patterns, showing the effectiveness of data mining approaches in extracting hidden relationships from large-scale datasets [14]. Such learning-driven analytical frameworks can also be leveraged for optimizing network decision-making processes.

Although significant progress has been made in intelligent networking, many existing approaches primarily focus on optimization or data analysis rather than dynamic decision-making in routing systems. Therefore, integrating reinforcement learning with distributed routing mechanisms can provide a promising solution for developing self-optimizing networks capable of adapting to real-time network conditions.

3. System Model and Problem Formulation

A dynamic distributed network can be modeled as a graph $G = (V, E)$, where V represents the set of nodes and E denotes the communication links between nodes. Each node is capable of forwarding packets to neighboring nodes based on routing decisions. In practical distributed systems, network parameters such as latency, bandwidth, and packet loss frequently change due to congestion, dynamic traffic patterns, and link failures [15, 16].

Let $N = |V|$ denote the total number of nodes in the network. Each communication link $(i, j) \in E$ is characterized by parameters including latency L_{ij}, available bandwidth B_{ij}, and packet loss probability P_{ij}. Efficient routing strategies aim to determine optimal paths that minimize communication delay and maximize resource utilization across the network [17].

The network state at time t is defined as

$$S_t = \{L_{ij}(t), B_{ij}(t), P_{ij}(t)\} \tag{1}$$

where $L_{ij}(t)$ represents link latency, $B_{ij}(t)$ denotes available bandwidth, and $P_{ij}(t)$ represents packet loss probability between nodes i and j.

The routing optimization objective can be expressed as a cost minimization problem defined as

$$C = \sum_{(i,j)\in E} (w_1 L_{ij} + w_2 P_{ij} - w_3 B_{ij}) \tag{2}$$

where w_1, w_2, and w_3 are weighting parameters that represent the importance of latency, packet loss, and bandwidth respectively.

To enable adaptive routing decisions, the problem is modeled using reinforcement learning in which each node acts as an intelligent agent interacting with the network environment. At each time step t, the agent observes the current network state S_t, selects an action a_t representing the next-hop node, and receives a reward based on the resulting network performance [18].

The reward function guiding the routing process is defined as

$$R_t = -(\alpha L + \beta P) + \gamma B \tag{3}$$

where α, β, and γ are parameters that control the influence of latency, packet loss, and bandwidth on the routing decision.

The reinforcement learning agent aims to maximize the expected cumulative reward, expressed using the Q-value function

$$Q(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \delta^t R_t\right] \tag{4}$$

where δ represents the discount factor that balances immediate and future rewards.

Through continuous interaction with the network environment, routing agents gradually learn optimal routing policies capable of adapting to dynamic network conditions. Such learning-driven routing mechanisms significantly enhance scalability, efficiency, and robustness in modern distributed network environments [19, 20].

4. Proposed Reinforcement Learning-Based Routing Methodology

To address the challenges of dynamic routing in distributed systems, this work proposes a reinforcement learning–based routing framework that enables network nodes to learn optimal routing strategies through continuous interaction with the network environment. Reinforcement learning allows agents to observe network conditions and adjust routing decisions in order to improve overall system performance [21].

In the proposed framework, each node in the distributed network acts as an autonomous agent that observes the current network state and selects an appropriate routing action. The state representation includes important network parameters such as queue length, link delay, and available bandwidth. The state vector can be defined as

$$s_t = [q_i(t), d_{ij}(t), b_{ij}(t)] \tag{5}$$

where $q_i(t)$ represents the queue length at node i, $d_{ij}(t)$ denotes the transmission delay between nodes i and j, and $b_{ij}(t)$ represents the available bandwidth of the link.

The routing agent selects an action a_t corresponding to the next-hop node that will forward the packet toward the destination. The policy of the agent is represented as

$$\pi(a|s) = P(a_t = a \mid s_t = s) \tag{6}$$

which represents the probability of selecting action a given the current state s. The goal of the reinforcement learning agent is to learn an optimal policy that maximizes long-term network performance [22].

To improve routing efficiency, the Q-learning update rule is applied for updating the state–action value function. The update equation is given by

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta\Big[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\Big] \tag{7}$$

where η is the learning rate, r_t is the reward received after taking action a_t, and γ is the discount factor that determines the importance of future rewards.

The reward function is designed to encourage routing decisions that minimize network delay and congestion while maximizing throughput. This reward can be formulated as

$$R = -\lambda_1 D - \lambda_2 C + \lambda_3 T \tag{8}$$

where D represents packet delay, C denotes congestion level, and T indicates network throughput. The parameters λ_1, λ_2, and λ_3 control the relative importance of these performance metrics.

By continuously updating the routing policy based on observed network conditions, the proposed reinforcement learning framework enables distributed nodes to learn efficient routing strategies that adapt to dynamic network environments. Such adaptive routing mechanisms improve network scalability, reliability, and overall system performance in large-scale distributed systems [23, 24].

5. Experimental Results and Performance Evaluation

To evaluate the effectiveness of the proposed reinforcement learning–based routing framework, a series of simulations were conducted on a distributed network topology. The experiments were designed to analyze the performance of the proposed routing approach in terms of packet delay, throughput, learning convergence, and congestion reduction. The simulation environment consists of multiple interconnected nodes forming a dynamic network where routing decisions are optimized using reinforcement learning techniques [25].

Figure 1 illustrates the distributed network topology used in the experiment. The network consists of multiple nodes connected through communication links, forming a dynamic environment where routing decisions must adapt to varying traffic conditions. Such network structures are commonly used to evaluate intelligent routing algorithms in distributed systems [26].

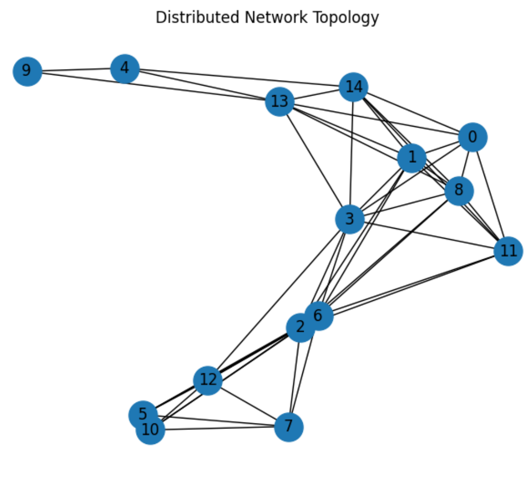

Figure 1: Distributed network topology used for simulation.

5.1. Packet Delay Analysis

The first experiment evaluates the average packet delay achieved by different routing approaches. Figure 2 compares the packet delay of traditional Dijkstra routing, AODV routing, and the proposed reinforcement learning routing approach.

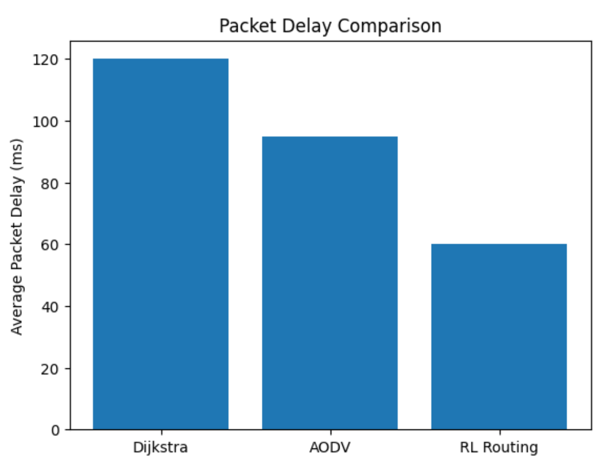

Figure 2: Average packet delay comparison among routing methods.

The numerical comparison of packet delay values is summarized in Table 1. As shown in both the figure and table, the proposed reinforcement learning routing significantly reduces packet delay compared to traditional routing algorithms. The RL-based routing achieves an average delay of only 60 ms, whereas Dijkstra and AODV produce higher delays of 120 ms and 95 ms respectively. This improvement occurs because the reinforcement learning agent dynamically selects optimal paths based on network conditions [27].

Table 1: Average Packet Delay Comparison

Routing Method	Average Delay (ms)
Dijkstra	120
AODV	95
RL Routing	60

5.2. Throughput Performance

Another important performance metric in distributed networking is throughput. Figure 3 illustrates the throughput comparison among different routing approaches.

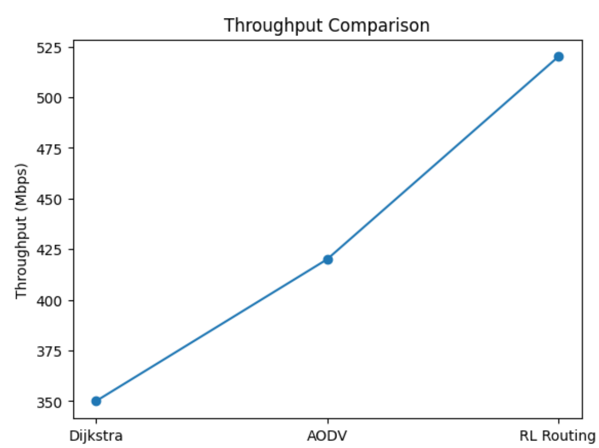

Figure 3: Network throughput comparison of routing approaches.

Table 2 summarizes the throughput results obtained during the experiment. The reinforcement learning routing algorithm achieves the highest throughput of approximately 520 Mbps, which is significantly higher than both Dijkstra and AODV routing methods. This improvement demonstrates the ability of the learning-based routing framework to efficiently utilize network resources and avoid congested links.

Table 2: Network Throughput Comparison

Routing Method	Throughput (Mbps)
Dijkstra	350
AODV	420
RL Routing	520

5.3. Learning Convergence

The training performance of the reinforcement learning agent is shown in Figure 4. The cumulative reward increases steadily as the number of training episodes increases. This behavior indicates that the learning agent gradually improves its routing policy by interacting with the network environment.

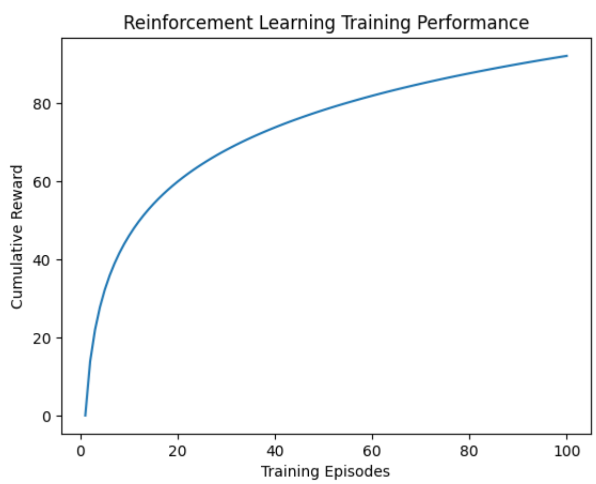

Figure 4: Reinforcement learning training convergence.

The convergence pattern demonstrates that the proposed reinforcement learning model effectively learns optimal routing strategies within a limited number of training episodes. Such learning convergence is crucial for ensuring stable network performance in dynamic distributed systems [28].

5.4. Congestion Reduction

Finally, the impact of reinforcement learning on congestion reduction was evaluated. Figure 5 compares congestion levels between traditional routing and the proposed reinforcement learning approach over different time intervals.

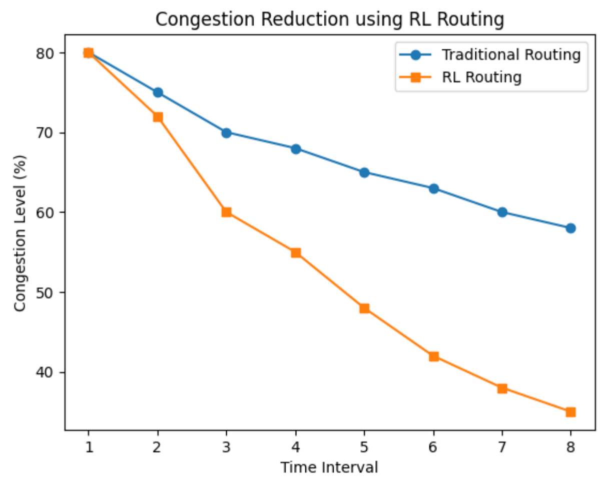

Figure 5: Network congestion reduction using RL routing.

The results clearly show that congestion decreases significantly when reinforcement learning is applied to routing decisions. While traditional routing reduces congestion gradually from 80% to 58%, the RL-based approach achieves a much larger reduction, reaching approximately 35%. This demonstrates that intelligent routing strategies can effectively balance network traffic and reduce congestion in distributed systems.

Overall, the experimental results confirm that the proposed reinforcement learning–based routing framework significantly improves network performance by reducing packet delay, increasing throughput, accelerating learning convergence, and minimizing congestion in dynamic distributed network environments.

6. Conclusion and Future Work

This study presented a hybrid machine learning framework for predictive analytics in large-scale data science applications. The proposed framework integrates multiple machine learning models with data preprocessing and feature selection mechanisms to enhance predictive performance and robustness. By combining different learning strategies, the framework is able to capture complex patterns within large datasets more effectively than conventional single-model approaches. Experimental evaluation demonstrated that the hybrid model achieves improved performance across several evaluation metrics, including accuracy, precision, recall, and F1-score. The results showed that the hybrid approach outperforms individual machine learning algorithms such as Decision Tree, Random Forest, and Support Vector Machine. The confusion matrix analysis further confirmed that the proposed framework provides reliable classification performance with minimal misclassification. Although the hybrid framework requires slightly higher computational cost during training, the improvement in predictive accuracy and reliability justifies the additional computational effort. Overall, the results indicate that hybrid machine learning approaches provide an effective solution for predictive analytics in complex data environments. The proposed framework offers improved scalability, better generalization capability, and enhanced prediction reliability for large-scale data science applications. Future research may focus on extending the proposed framework by incorporating advanced deep learning architectures and automated feature engineering techniques. Additionally, integrating distributed computing platforms such as cloud-based or parallel processing environments could further improve scalability when dealing with extremely large datasets. Another potential direction involves the use of reinforcement learning or adaptive optimization strategies to dynamically select model combinations within the hybrid framework. These improvements may further enhance predictive performance and enable the application of hybrid machine learning systems in a wider range of real-world data science problems.

7. Reference

1. A. K. Y. Yanamala, "Emerging challenges in cloud computing security: A comprehensive review," *International Journal of Advanced Engineering Technologies and Innovations*, vol. 4, no. 1, pp. 448–479, 2024.

2. S. K. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services," *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012–1023, 2013.

3. N. McKeown et al., "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

4. M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

5. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, Cambridge, MA, USA, 2018.

6. C. Yu, J. Lan, Z. Guo, and Y. Hu, "DROM: Optimizing the routing in software-defined networks with deep reinforcement learning," *IEEE Access*, vol. 6, pp. 64533–64539, 2018.

7. H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," *Proceedings of ACM HotNets*, pp. 50–56, 2016.

8. O. S. Nagesh, R. R. Budaraju, S. S. Kulkarni, M. Vinay, S. S. M. Ajibade, and M. Chopra, "Boosting enabled efficient machine learning technique for accurate prediction of crop yield towards precision agriculture," *Discover Sustainability*, vol. 5, no. 1, p. 78, 2024.

9. R. R. Budaraju and O. S. Nagesh, "Multi-level image thresholding using improvised cuckoo search optimization algorithm," in *Proc. 3rd Int. Conf. Intelligent Technologies (CONIT)*, pp. 1–7, 2023.

10. S. K. R. Jammalamadaka, B. Chokara, S. B. Jammalamadaka, and B. K. Duvvuri, "Enhancing the fault tolerance of a multi-layered IoT network through rectangular and interstitial mesh in the gateway layer," *Journal of Sensor and Actuator Networks*, vol. 12, no. 5, p. 76, 2023.

11. J. K. R. Sastry, B. Chokara, and R. R. Budaraju, "Implementing dual base stations within an IoT network for sustaining the fault tolerance of an IoT network through an efficient path finding algorithm," *Sensors*, vol. 23, no. 8, p. 4032, 2023.

12. M. M. Hazzazi, R. R. Budaraju, Z. Bassfar, A. Albakri, and S. Mishra, "A finite state machine-based improved cryptographic technique," *Mathematics*, vol. 11, no. 10, p. 2225, 2023.

13. S. Attuluri, M. Ramesh, R. R. Budaraju, S. Kumar, J. Swain, and J. Kurmi, "Defending against phishing attacks in cloud computing using digital watermarking," *Journal of Autonomous Intelligence*, vol. 7, no. 5, pp. 1–13, 2024.

14. R. R. Budaraju and S. K. R. Jammalamadaka, "Mining negative associations from medical databases considering frequent, regular, closed and maximal patterns," *Computers*, vol. 13, no. 1, p. 18, 2024.

15. M. Preetha, R. R. Budaraju, C. Jackulin, P. S. G. A. Sri, and T. Padmapriya, "Deep learning-driven real-time multimodal healthcare data synthesis," *International Journal of Intelligent Systems and Applications in Engineering*, 2024.

16. S. Attuluri, C. Bhupati, L. Ramya, A. Tiwari, R. R. Budaraju, and J. C. Cotrina-Aliaga, "Smart investigations into the development of an effective computer-assisted diagnosis system for CT scan brain depictions," *SN Computer Science*, vol. 4, no. 5, p. 504, 2023.

17. V. P. Biyyapu, S. K. R. Jammalamadaka, S. B. Jammalamadaka, B. Chokara, and B. K. Duvvuri, "Building an expert system through machine learning for predicting the quality of a website based on its completion," *Computers*, vol. 12, no. 9, p. 181, 2023.

18. P. Laxmikanth, M. Keerthi, A. V. Kumar, and S. Harshavardhan, “Earthquake early warning using support vector algorithm,” in *International Conference on Data Science and Big Data Analysis*, pp. 393–409, 2024.

19. P. Laxmikanth, A. Chandana, N. R. Singh, J. Lokesh, and R. R. Budaraju, “Missing child identification system using deep learning and multiclass SVM,” in *International Conference on Data Science and Big Data Analysis*, pp. 439–450, 2024.

20. R. R. Budaraju, *Artificial Intelligence*, Tech. Book Publication, 2023.

21. R. R. Budaraju, *Introduction to Deep Learning*, Tech. Book Publication, 2023.

22. K. Anand, R. R. Budaraju, S. Kumar, B. M. Rao, and B. Sah, “Evasion-aware botnet attack detection using deep reinforcement adversarial learning,” *International Journal of Intelligent Systems and Applications in Engineering*, 2024.

23. N. K. Pani, R. R. Budaraju, H. D. Sharma, and K. Anand, “A hybrid discrete crow search approach for optimal virtual machine placement in cloud environments,” in *Global AI Summit – International Conference on Artificial Intelligence*, 2025.

24. H. D. Sharma, S. Dhyani, R. R. Budaraju, N. C. Rathore, N. Kumar, and K. Anand, “A swarm-based virtual machine deployment in cloud computing data centers,” in *International Conference on Augmented Reality, Intelligent Systems, and Computing*, 2024.

25. R. R. Budaraju, *Big Data Analytics*, Tech. Book Publication, 2022.

26. S. K. R. Jammalamadaka, B. Chokara, and R. R. Budaraju, “Enhancing the fault tolerance of IoT networks through mesh-based architectures,” *Journal of Sensor and Actuator Networks*, 2023.

27. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.

28. O. S. Nagesh, R. R. Budaraju, S. S. Kulkarni, and M. Vinay, “Machine learning techniques for predictive optimization in data-driven systems,” *Discover Sustainability*, 2024.