Metaversalize

www.meta.reapress.com

Metaverse Vol. 1, No. 3 (2024) 139–149.

Paper Type: Original Article

AI-Powered Routing Strategies for Low-Latency IoT Networks

Sanchita Roy*

School of Computer Science Engineering, KIIT University, Bhubaneswar, India; 2229197@kiit.ac.in.

Citation:

Received: 14 January 2024 *Revised: 28 March 2024* *Accepted: 27 May 2024*	Roy, S. (2024). AI-powered routing strategies for low-latency IoT networks. *Metaversalize, 1*(3), 139-149.

Abstract

With the swift expansion of Internet of Things (IoT) networks, it has become crucial to ensure real-time, low-latency communication, especially in vital areas such as autonomous vehicles, industrial automation, and healthcare. Conventional routing protocols, like Ad hoc On-Demand Distance Vector (AODV) and Destination Sequenced Distance Vector (DSDV), often fail to remain efficient in the dynamic and distributed nature of IoT, leading to considerable communication delays. This paper explores AI-enhanced routing strategies that dynamically optimize data paths by evaluating real-time network conditions and learning from past data. The study utilizes essential AI techniques, including machine learning-driven decision-making, Reinforcement Learning (RL) for adaptive route management, and AI-assisted congestion management, to improve real-time routing choices. Our results indicate that these AI-based methods successfully reduce latency and enhance network performance, rendering them suitable for latency-critical IoT applications. Furthermore, AI-enabled routing shows potential for adjusting to network changes and device mobility, thus ensuring sustained low latency. Future studies will aim at expanding these approaches to larger networks and strengthening their security to ultimately meet the increasing requirements of real-time IoT systems.

Keywords: AI-powered routing, Low-latency, Internet of things, Machine learning, Reinforcement learning, Route management.

1 | Introduction

The rapid growth of Internet of Things (IoT) networks has reshaped device communication, connecting billions of devices across applications like smart homes, healthcare, and industrial automation [1], [2]. This connectivity demands low-latency networks to ensure efficient data flow and responsiveness. Traditional routing methods struggle to meet these real-time requirements as IoT networks continuously adapt to devices joining or leaving.

Corresponding Author: 2229197@kiit.ac.in
https://doi.org/10.22105/metaverse.v1i3.69

Consequently, researchers are exploring AI-driven routing solutions that can dynamically optimize paths, minimize delays, and support the responsive interactions that IoT applications need for seamless performance in diverse environments [3]. *Fig. 1* shows IoT network architecture.

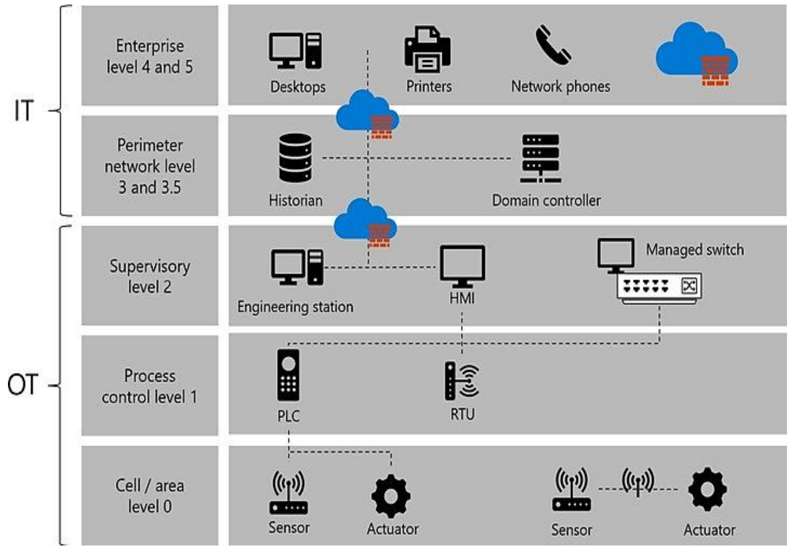

Fig. 1. IoT architecture.

Despite the promise of IoT, traditional routing protocols, such as Ad hoc On-Demand Distance Vector (AODV) [4] and Optimized Link State Routing (OLSR) [5], face substantial limitations in dynamic environments where devices frequently join or leave the network. These protocols often struggle to maintain optimal routing efficiency under varying network conditions, leading to increased latency and degraded performance. These limitations become more pronounced as IoT applications grow in scale and complexity, especially when supporting real-time services like remote monitoring and automated controls. The static nature of traditional protocols limits their adaptability, making it difficult to respond promptly to sudden changes in network topology. Moreover, these protocols cannot effectively anticipate and address congestion or link failures without predictive capabilities. To overcome these challenges, researchers are adopting AI-driven routing methods that can dynamically adapt, learning from network patterns to deliver low-latency, reliable communication across ever-changing IoT landscapes [6], [7].

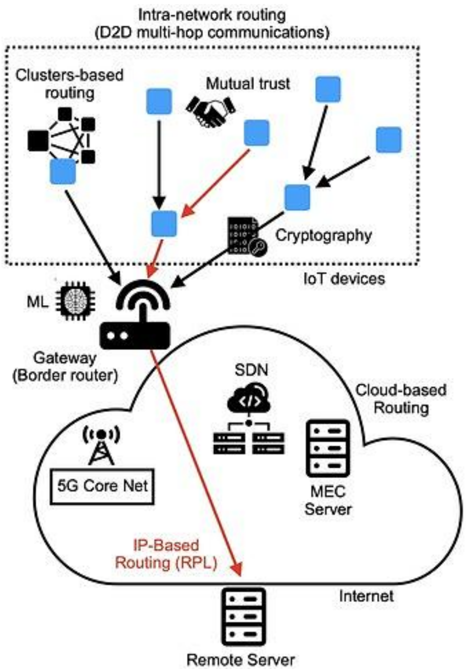

Fig. 2. Routing strategies in IoT networks.

Table 1. Issues associated with routing strategies.

Challenges	Description
Power and payload	IoT devices often lack the processing power and storage to run complex AI algorithms for low-latency routing.
Dynamic network	IoT networks in mobile environments are highly dynamic, complicating AI model applications that depend on stable structures for routing.
Energy efficiency	IoT devices have limited power, and continuous AI processing for routing can quickly drain energy, reducing device lifespan and network reliability.
Data Security	Routing algorithms rely on real-time IoT data, posing security and privacy risks.

The paper is structured as follows. Section 2 provides a literature review of different routing strategies. Section 3 discusses the challenges associated with routing strategies, while Section 4 highlights the limitations of various routing strategy algorithms. Section 5 explores potential improvements in these algorithms to enhance efficiency and fault tolerance. Finally, Section 6 summarizes the key findings, future research directions, and references.

2 | Literature Review

The concept of AI-driven routing in IoT networks is rooted in the growing complexity of managing large-scale device connections while maintaining low-latency communication. Traditional routing protocols such as AODV and Destination Sequenced Distance Vector (DSDV) have served well in static or less dynamic environments. Still, they are increasingly inadequate for IoT networks' highly dynamic, resource-constrained nature.

Table 2. Routing strategies classification.

ML-based routing uses machine learning to predict optimal paths, reducing latency and conserving energy. This makes it ideal for dynamic IoT networks.	Continuously learns and adjusts to changing network conditions for optimal performance. It predicts efficient routes, reduces latency, and conserves energy, making it ideal for resource-constrained IoT devices.	Examples of machine learning: supervised and unsupervised.
Reinforcement Learning (RL)** is a machine learning method where an agent learns by interacting with an environment, aiming to maximize rewards over time.	Aims to maximize long-term rewards, achieving efficient outcomes. Handles complex, high-dimensional problems well with appropriate algorithms.	Examples: Q learning, deep Q-networks, SARSA

2.1 | AI-Powered Routing Strategies

2.1.1 | Reinforcement learning

Reinforcement Learning (RL) is a strategic approach encompassing various algorithms, including Q-learning and Deep Q-learning Networks (DQN) [8], employed to optimize routing paths in dynamic IoT networks. Here's a breakdown of the steps involved in implementing one of these algorithms.

Algorithm 1. Q learning.

1. Initialize Q-Table: Set up a Q-table with initial values for each state-action pair.
2. Set Parameters: Define the learning rate α, discount factor γ, and exploration rate ϵ.
3. Loop Until Convergence: Repeat the following steps for each episode.
4. Observe Current State: Begin in an initial state.
5. Choose Action: Select an action using either exploration (random choice) or exploitation (highest Q- value).
6. Take Action and Observe Reward: Execute the chosen action, move to a new state ', and receive a reward r.
7. Update Q-Value: Update Q(s, a) using the Q-learning formula based on the reward and future Q-values.
8. Set New State: Update the current state to s' for the next iteration.
9. Decay Exploration Rate (optional): Gradually reduce ϵ to increase exploitation over time.
10. Repeat: Continue looping until Q-values converge to stable values.

Fig. 3 shows the flow chart of the Q-learning algorithm.

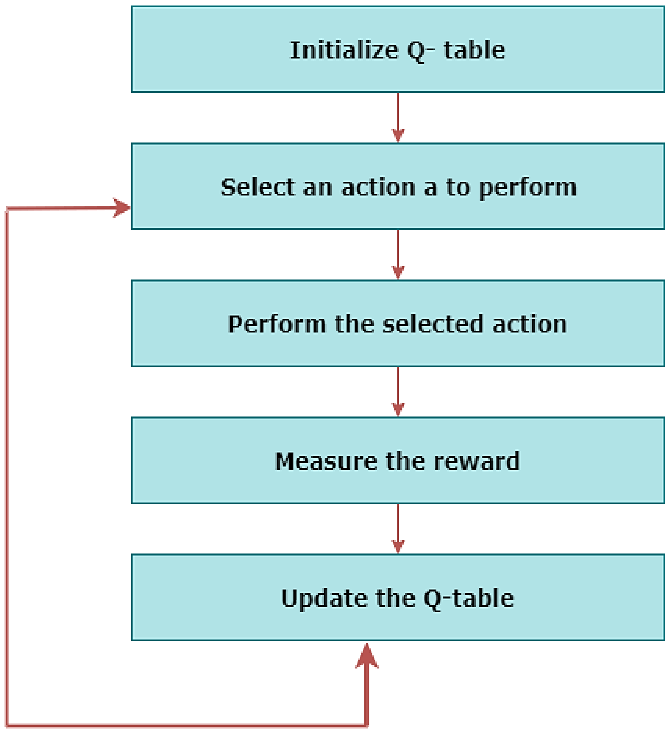

Fig. 3. Q-learning algorithm flow-chart.

2.1.2 | Deep Q-network algorithm

RL is a strategic approach encompassing various algorithms, including Q-learning and DQN, employed to optimize routing paths in dynamic IoT networks. Here's a breakdown of the steps involved in implementing one of these algorithms.

Algorithm 2. Deep Q-networks initialize replay memory D to capacity N.

Initialize action-value function Q with random weights

for episode = 1, M do

Initialise sequence s1 = {x1} and preprocessed sequenced φ1 = φ(s1)

for t = 1, T do

With probability e, select a random action ot;otherwise, select at = max Q∗(φ(st), a; θ)

Execute action in the emulator and observe reward rt and image xt+1

Set st+1 = st, at, xt+1 and preprocess φt+1 = φ(st+1) Store transition (φt, at, rt, φt+1) in D. Sample random minibatch of transitions (φj , aj , rj , φj+1) from D Set yj = { rj for terminal φj+1

rj + γ maxa' Q(φj+1, a'; θ) for non-terminal φj+1

Perform a gradient descent step on (yj − Q(φj , aj ; θ))2 according to equation 3

end for

end for

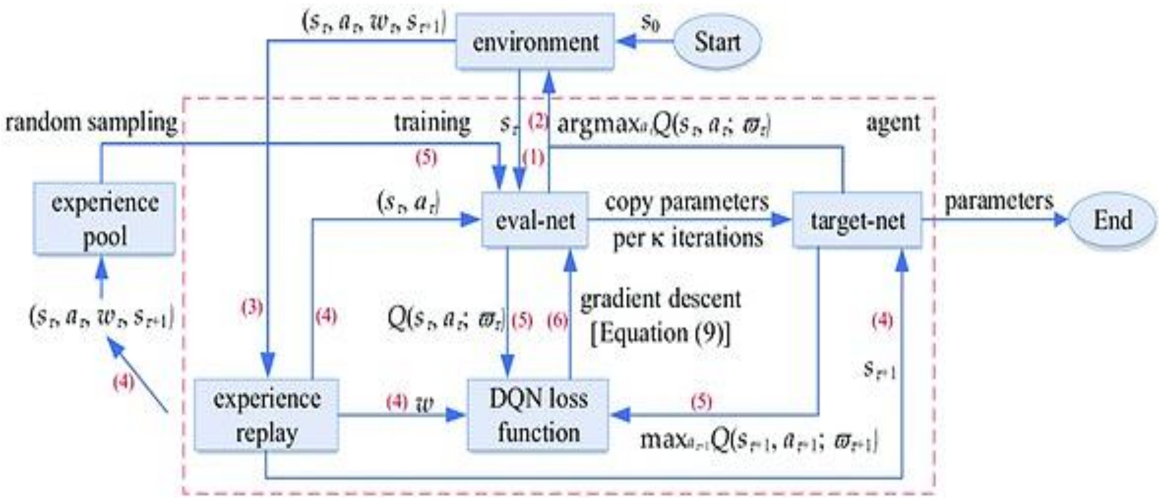

Fig. 4. Deep Q-network algorithm flow chart.

2.1.3 | SARSA algorithm

The SARSA algorithm is an on-policy RL approach for training agents to make sequential decisions [9]. Unlike Q-learning, which learns from actions outside the agent's current policy, SARSA updates its values based on its actions, leading to safer learning in uncertain environments. The agent selects an action, observes the resulting reward and next state, then chooses the following action based on its policy and updates its value estimates. This makes SARSA especially suited for situations where safe exploration and adapting to real-time feedback are essential.

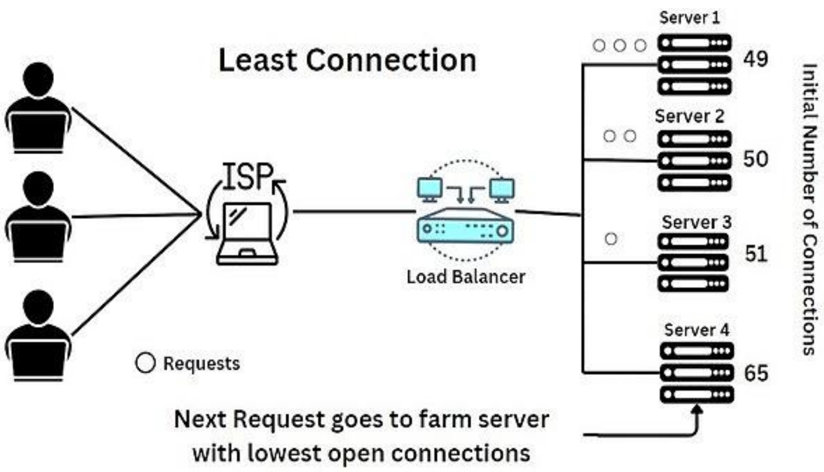

Fig. 5. Least connection algorithm.

SARSA is used in scenarios like autonomous navigation, gameplay, and agent training in dynamic environments where safety or penalties need to be considered carefully.

I. Initialize: Initialize Q-values for all state-action pairs (often randomly).

II. Select action (A): Choose an action based on the current state S, using a policy derived from Q-values, such as the ε-greedy policy.

III. Take action and observe reward: Take action A, observe the reward R, and the new state S'. Choose next action A': Based on the new state S', select the following action A' using the same policy. Update Q-value: Update the Q-value for the state-action pair (S, A) using the following SARSA update rule:

$$Q(S,A) \leftarrow Q(S,A) + \alpha(R + \gamma Q(S',A') - Q(S,A))$$

IV. α: Learning rate controls how much new information overrides old information.

V. γ: Discount factor represents the importance of future rewards.

VI. Repeat: Set $S \leftarrow S'$ and $A \leftarrow A'$, and repeat steps 3 to 5 until the episode ends.

Example

An agent might use SARSA to learn the safest path to a goal in a grid-world navigation task. Each movement earns a small negative reward (e.g., -1) while reaching the goal provides a positive reward (e.g., +10). Over time, SARSA will guide the agent toward the path that minimizes the steps to the goal and maximizes the cumulative reward, even if it means occasionally exploring alternative routes to improve its policy.

Table 3. Routing strategies algorithm analysis.

Algorithm	Convergence Speed	Advantages	Disadvantages	Scalability
Q Learning		Simple and easy to implement, it guaranteed convergence to an optimal policy with enough exploration	It is unsuitable for large/continuous state spaces; it needs discretization.	Limited Scalability
Deep Q-Network		Handles large and continuous state spaces; effective in complex environments.	It can be unstable; it requires a large replay memory and a target network to stabilize learning.	Improve scalability by employing deep neural networks
SARSA		is Safer and more conservative than Q-learning avoids risky paths by updating based on the current policy.	Converges to suboptimal policies if exploration is limited; slower convergence than Q- learning	Shares similar scalability constraints with Q- Q-learning

Recommendation for low-latency IoT networks

In the context of low-latency IoT networks, Q-learning and SARSA are the most suitable choices, as they are easy to implement and demonstrate the lowest latency in discrete environments. SARSA excels in this area due to its on-policy learning approach, which allows it to adaptively modify task policies in response to dynamic conditions and potential hazards. This adaptability ensures safer decision-making, critical in environments where timely and accurate responses are essential.

While DQN can tackle complex problems characterized by high-dimensional state spaces and numerous degrees of freedom, it often introduces additional latency due to the computational demands of deep learning architectures. The process of action selection in DQNs, coupled with the need for extensive training and hyperparameter tuning, can detract from the immediate responsiveness required in low-latency applications. Therefore, in scenarios where minimizing latency is paramount, Q-learning and SARSA are more effective options, providing reliable performance without compromising speed.

3 | Challenges Associated with Routing Strategies

IoT routing strategies face challenges related to scalability, energy efficiency, and low latency. Dynamic network conditions, security threats, and limited device resources add complexity, and privacy concerns further complicate decentralized routing.

3.1 | Power and PayLoad

IoT devices are equipped with finite battery power; thus, energy efficiency is an important prerequisite for successful routing. The size of the data traffic could be very large, which causes an overloading problem on network resources; thereby, delays and packet loss are inevitable.

3.2 | Dynamic Network

Moving networks encounter challenges due to device mobility, which complicates routing by altering network topology. Adaptive routing can dynamically adjust connections to address these changes. Moreover, environmental factors like interference and congestion require intelligent mechanisms to optimize routes and maintain performance quickly.

3.3 | Energy Efficiency

Energy efficiency is crucial in IoT networks due to the limited battery capacities of many devices. Increased energy consumption can lead to frequent battery replacements, limiting device lifespan and reliability. Therefore, routing strategies must minimize energy use by balancing data transmission and node power to enhance overall network performance.

3.4 | Data Security

Data security is crucial in IoT networks due to the sensitive information transmitted. Device vulnerabilities can lead to unauthorized access and data breaches. Therefore, robust security measures, such as encryption and secure authentication, are essential to protect data integrity while maintaining network performance and minimizing latency.

4 | Limitations of AI-Powered Routing Strategies

4.1 | Limitations of Q-Learning

Q-learning has limitations in vast state-action spaces, where prolonged exploration results in sluggish learning, notwithstanding its effectiveness in learning optimal policies. Because it treats each state separately, it lacks generalization and has trouble with continuous state spaces that aren't modified, such as function approximations. These problems limit its scalability and efficiency for high-dimensional, complex situations.

4.1.1 | Scalability

Q-learning struggles with large state-action spaces, as significant memory and computational resources are required to store and update the Q-values for every possible state-action pair.

4.1.2 | Incompatibility with continuous spaces

Q-learning requires discrete states, making it challenging to use directly in environments with continuous states without additional modifications like function approximations.

1.4.3 | Lack of generalization

Q-Learning assigns a unique Q-value to each specific state-action pair, treating them as separate experiences. This lack of generalization means that similar states are not recognized as related, preventing the agent from applying knowledge gained in one area to another. As a result, the agent often re-learns similar information multiple times, leading to redundant computations and slower convergence, especially in environments with numerous or structurally similar states.

4.2 | Limitations of Deep Q-Networks

DQNs improve on standard Q-Learning by using deep neural networks to approximate Q-values, but they still have notable limitations:

4.2.1 | Sample inefficiency

DQNs require large amounts of training data to learn effectively, as neural networks need numerous samples to approximate Q-values accurately, making training slow and resource-intensive.

4.2.2 | Stability and convergence issues

DQNs can be unstable and struggle to converge, especially in environments with sparse rewards or delayed rewards. To improve stability, techniques like experience replay and target networks are often required.

4.2.3 | Poor generalization in complex environments

DQNs can overfit to specific states seen during training, which limits their ability to generalize well in more complex or dynamic environments and reduces performance outside of training conditions.

4.3 | Limitations of SARSA Algorithm

The SARSA (State-Action-Reward-State-Action) algorithm, commonly used in RL, has demonstrated effectiveness across various domains, particularly in managing dynamic environments and making sequential decisions. However, like any algorithm, SARSA has limitations that affect its scalability, reliability, and efficiency in complex systems. This section examines the challenges faced when applying SARSA in diverse applications.

4.3.1 | Vulnerability to high variance in reward function

SARSA's reliance on observed rewards can lead to significant fluctuations in performance, particularly in environments with high variability in reward structures. High reward variance may cause SARSA to converge slowly or to a suboptimal policy due to the inconsistent reward feedback loop, leading to prolonged training times and unpredictable performance in real-world applications. This phenomenon is similar to the "credit assignment problem" in RL, where the agent struggles to identify actions that contribute to delayed rewards correctly.

4.3.2 | Dependence on exploration policy

SARSA's on-policy nature means it evaluates and improves the policy it uses to make decisions, relying heavily on a balanced exploration strategy. If the exploration policy is not well-designed, SARSA may excessively explore suboptimal regions of the state-action space, potentially resulting in slower convergence or convergence to a suboptimal solution. This issue, akin to "policy oscillation," occurs when the agent repeatedly visits states without improving its understanding of optimal actions.

4.3.3 | Sensitivity to environment dynamics

SARSA is sensitive to environmental changes since it updates the policy based on observed states and actions. This can lead to instability in highly dynamic environments, as SARSA may struggle to adapt quickly enough to changing conditions. This limitation is similar to the "catastrophic forgetting" problem, where an algorithm loses information about previously learned patterns due to new, conflicting information, often resulting in degraded performance over time.

5 | Proposed Work

The proposed work focuses on developing AI-powered routing strategies to optimize latency, efficiency, and adaptability within IoT networks. By addressing the unique challenges of IoT, such as scalability, dynamic network conditions, and low-power constraints, this work aims to enhance real-time data transmission, making IoT systems more responsive and reliable.

5.1 | Optimization of AI-Driven Routing Protocols

I. Develop advanced AI-based routing algorithms that dynamically adjust to varying network conditions, minimizing latency and maximizing data throughput in IoT networks.

II. Implement RL and predictive modeling techniques to enhance route decision-making, enabling optimized packet delivery paths and reducing bottlenecks in dense IoT environments.

5.2 | Adaptability and Low-Power Management

I. Introduce energy-efficient, AI-based routing mechanisms that prioritize low-latency paths while conserving device battery life, which is crucial in resource-constrained IoT applications.

II. Explore adaptive protocols that adjust routing decisions based on node availability, mobility patterns, and energy levels, supporting prolonged network functionality and resilience in dynamic IoT ecosystems.

5.3 | Enhancement of Security in AI-Powered Routing

I. Strengthen security to counter DDoS attacks and IP spoofing by integrating advanced authentication protocols and robust access control to ensure secure IoT data transmission.

II. Explore adaptive routing approaches that address challenges posed by dynamic IPs, improve security against spoofing, and enhance routing efficiency in variable network conditions.

5.4 | Refinement of Bandwidth and Network Resource Allocation

I. Develop proactive algorithms for dynamic bandwidth allocation, preventing network congestion and enhancing data transmission speeds in bandwidth-limited environments.

II. Investigate adaptive routing methods to manage network speed variations, incorporating intelligent protocols that optimize performance and maintain low-latency transmission across changing network conditions.

6 | Conclusion

In conclusion, this paper has undertaken a detailed study of AI-powered routing strategies for low-latency IoT networks, analyzing the current challenges and future opportunities within this evolving field. By examining issues such as dynamic network conditions, resource constraints, security vulnerabilities, and scalability, we have gained insights into the limitations of traditional IoT routing methods and identified areas ripe for innovation.

We aim to address these challenges through the proposed work by developing AI-enhanced routing protocols that adapt dynamically to fluctuating network environments. The proposed strategies focus on optimizing latency, conserving power in resource-limited devices, and implementing enhanced security measures. Additionally, adaptive routing techniques and scalable bandwidth allocation approaches aim to improve both the resilience and performance of IoT networks.

This research contributes to the ongoing development of intelligent routing solutions, emphasizing the need for secure, efficient, and highly responsive networks in IoT applications. By addressing key obstacles and implementing adaptive routing mechanisms, we strive to unlock the full potential of IoT technology in various industries, paving the way for more resilient, scalable, and efficient IoT infrastructures capable of supporting next-generation applications and services.

Author Contributions

Sanchita Roy.

Funding

This research received no external funding.

Data Availability

The data utilized and analyzed in this study are available from the corresponding author, Ayush Singh, upon reasonable request.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Mohapatra, H., Rath, A. K., & Panda, N. (2022). IoT infrastructure for the accident avoidance: An approach of smart transportation. *International journal of information technology (Singapore)*, *14*(2), 761–768. https://doi.org/10.1007/s41870-022-00872-6

[2] Iqubal, S., Khan, S., Pant, N., Sarkar, S., Rey, T., Mohapatra, H., & Rey, T. (2025). A study on IoT enabled smart bed with brain- computer interface for elderly and paralyzed individuals. In *Future innovations in the convergence of AI and internet of things in medicine* (pp. 61–87). IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3693-7703-1.ch004

[3] Umoga, U. J., Sodiya, E. O., Ugwuanyi, E. D., Jacks, B. S., Lottu, O. A., Daraojimba, O. D., & Obaigbena, A. (2024). Exploring the potential of AI-driven optimization in enhancing network performance and efficiency. *Magna scientia advanced research and reviews*, *10*(1), 368-378. https://doi.org/10.30574/msarr.2024.10.1.0028

[4] Pandey, P., & Singh, R. (2022). Efficient ad hoc on demand distance vector routing protocol based on route stability in MANETs. *International journal of wireless information networks*, *29*(3), 393–404. https://doi.org/10.1007/s10776-022-00570-x

[5] Dafalla, M. E. M., Mokhtar, R. A., Saeed, R. A., Alhumyani, H., Abdel-Khalek, S., & Khayyat, M. (2022). An optimized link state routing protocol for real-time application over vehicular ad-hoc network. *Alexandria engineering journal*, *61*(6), 4541-4556. https://doi.org/10.1016/j.aej.2021.10.013

[6] Tshakwanda, P. M., Arzo, S. T., & Devetsikiotis, M. (2024). Advancing 6G network performance: AI/ML framework for proactive management and dynamic optimal routing. *IEEE open journal of the computer society*, *5*, 303–314. https://doi.org/10.1109/OJCS.2024.3398540

[7] Agbon, E. E., Muhammad, A. C., Alabi, C. A., Adikpe, A. O., Tersoo, S. T., Imoize, A. L., & Sur, S. N. (2025). AI-driven traffic optimization in 5g and beyond: Challenges, strategies, solutions, and prospects. *Lecture notes in electrical engineering* (Vol. 1243 LNEE, pp. 491–510). Springer. https://doi.org/10.1007/978-981-97-6465-5_40

[8] Fu, F., Jiao, Q., Yu, F. R., Zhang, Z., & Du, J. (2021, June). Securing UAV-to-vehicle communications: A curiosity-driven deep Q-learning network (C-DQN) approach. *2021 IEEE international conference on communications workshops (ICC Workshops)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICCWorkshops50388.2021.9473714

[9] Yao, G., Zhang, N., Duan, Z., & Tian, C. (2025). Improved SARSA and DQN algorithms for reinforcement learning. *Theoretical computer science*, *1027*, 115025. https://doi.org/10.1016/j.tcs.2024.115025