DOI: 10.21917/ijct.2025.0521

LATENCY-AWARE WIRELESS TRANSMISSION OF CLOUD DATA IN CLOUD AVOIDANCE NETWORKS USING A DUAL-HEAD ENSEMBLE TRANSFORMER

J. Jasmine[1] and Hariharankumar[2]
[1]*Department of Computer Science and Engineering, Karpagam College of Engineering, India*
[2]*Tata Consultancy Services Private Limited, Kakkanad, India*

Abstract

With the growing reliance on cloud-based services, latency in wireless transmission of cloud data remains a critical challenge in environments where cloud connectivity is intermittent or restricted, such as cloud avoidance networks (CANs). These networks operate under the paradigm of minimizing dependence on centralized cloud infrastructure, leveraging edge or fog computing for timely data processing and delivery. Traditional transmission frameworks often fail to effectively manage latency in dynamic and decentralized networks. In CANs, unpredictable node mobility, variable signal strength, and heterogeneous device capabilities exacerbate delays. There is a lack of robust machine learning-based solutions that adaptively optimize routing and data transmission strategies in real time to reduce latency. This study introduces a Dual-Head Ensemble Transformer (DHET) model tailored for latency-aware wireless transmission in CANs. The first head of the transformer predicts short-term transmission latency across multiple hops based on real-time network conditions, while the second head assesses the reliability of the path by evaluating historical trends and signal consistency. Ensemble learning is used to fuse predictions from diverse transformer sub-models trained on varied wireless scenarios, ensuring generalized performance. A latency-prioritized routing algorithm then utilizes these predictions to dynamically select optimal paths for cloud data transmission. Simulation results demonstrate that the DHET-based approach achieves an average 18–25% reduction in end-to-end latency compared to baseline protocols such as AODV and DSR. The dual-head design allows for a balance between latency minimization and transmission stability, making it well-suited for CAN deployments in smart cities, autonomous fleets, and remote monitoring systems.

Keywords:

Latency Management, Cloud Avoidance Network, Wireless Transmission, Dual-Head Ensemble Transformer, Edge Computing

1. INTRODUCTION

Cloud computing has revolutionized data storage and processing by offering scalable, on-demand services. However, the rapid increase in mobile users and the widespread deployment of Internet of Things (IoT) devices have posed new demands on network infrastructure, particularly in latency-sensitive applications [1]. Conventional cloud models, which rely heavily on centralized servers, often experience significant transmission delays due to distance, network congestion, or limited bandwidth [2]. To address this, the Cloud Avoidance Network (CAN) paradigm has emerged. CANs operate with minimal reliance on central cloud servers by shifting computation closer to the data source via edge and fog computing [3].

Despite its benefits, CANs introduce significant challenges. First, they operate in highly dynamic environments where node mobility and variable link quality make latency prediction and management complex [4]. Second, the decentralized nature of CANs, while reducing dependency on central servers, makes it difficult to maintain a global view of the network for routing decisions [5]. These issues make traditional routing and transmission approaches less effective, particularly in applications like autonomous vehicles, real-time surveillance, and remote healthcare monitoring.

Current wireless transmission frameworks often lack adaptability to dynamic latency conditions in CANs. They do not incorporate real-time learning mechanisms to predict and manage latency in decentralized environments [6]. This results in sub-optimal routing decisions, reduced reliability, and failure to meet the stringent latency requirements of real-time applications.

This research aims to design an intelligent, adaptive framework for latency management in CANs. The specific objectives include:

- Designing a predictive model for transmission latency using deep learning techniques.
- Developing a dynamic routing mechanism that adapts based on latency and link reliability.
- Evaluating the performance of the proposed approach under varied CAN environments.

This study introduces a Dual-Head Ensemble Transformer (DHET), a novel deep learning architecture with two attention-based heads. The first head predicts transmission latency using current network parameters (e.g., node mobility, signal strength), while the second head assess historical reliability of paths. The dual-head mechanism allows the system to simultaneously consider short-term fluctuations and long-term trends in the network, which is essential for stable and low-latency data delivery.

The main contributions of this paper are: A novel DHET model for accurate and robust latency prediction in wireless transmission over CANs. An ensemble learning strategy to improve generalization across varying network topologies and mobility patterns. A latency-prioritized routing protocol that leverages dual-head outputs for real-time path selection. Extensive experimental validation using simulated CAN environments, demonstrating superior performance over traditional methods in terms of latency, reliability, and adaptability.

2. RELATED WORKS

Recent literature highlights several directions for managing latency and improving transmission efficiency in wireless and cloud-avoidance networks. Latency-aware routing protocols such as Delay-Tolerant Networking (DTN) have been used in mobile ad hoc networks (MANETs) and vehicular networks. These protocols attempt to route data via the most reliable and fastest path based on current link conditions [7]. However, these models

often rely on heuristic metrics and lack predictive learning capabilities, limiting their adaptability in dynamic CAN environments. Researchers have applied machine learning (ML) to optimize network routing and latency prediction. For instance, Support Vector Machines (SVM) and Random Forests have been used for path quality estimation [8]. Deep learning models, particularly Recurrent Neural Networks (RNNs) and LSTMs, have been explored for sequential prediction of network parameters [9]. While these methods offer better temporal learning, they are often slow to adapt to real-time changes and struggle with high-dimensional data. The Transformer architecture has recently gained traction in network applications due to its attention mechanism and ability to handle long dependencies. Works such as NetTransformer and Graph-based Transformers have shown success in modeling routing and topology structures [10]. However, existing implementations focus on static topologies or assume centralized data availability, making them unsuitable for CANs. Edge computing plays a crucial role in reducing latency by bringing computation closer to data sources. Studies have explored resource allocation and data offloading strategies at the edge [11]. In fog-assisted CANs, latency is improved by distributing intelligence to intermediate nodes [12]. Nevertheless, these works often focus on hardware and system-level optimization, rather than learning-based predictive models for latency control. Ensemble learning has been applied in wireless networks to improve prediction accuracy by combining multiple models. Techniques like bagging and boosting have been used for channel prediction and fault detection [13]. However, ensemble learning has yet to be fully integrated with Transformer architectures for end-to-end latency prediction in wireless transmission. Thus, while previous research has made advances in isolated aspects, like routing, edge computing, or ML-based predictions, there is a gap in integrating dual-headed attention-based learning with ensemble techniques to deliver latency-aware, real-time transmission strategies for cloud avoidance networks. This work addresses that gap by combining the strengths of Transformer models and ensemble learning within a novel dual-head predictive framework tailored for dynamic, decentralized environments.

3. PROPOSED DUAL-HEAD ENSEMBLE TRANSFORMER FOR LATENCY-AWARE WIRELESS TRANSMISSION IN CLOUD AVOIDANCE NETWORKS

The proposed method introduces a Dual-Head Ensemble Transformer (DHET) model integrated into a latency-aware wireless data transmission framework within Cloud Avoidance Networks (CANs). This method combines predictive learning and dynamic routing to minimize latency and improve transmission reliability in decentralized, mobile network environments.

The DHET model consists of two parallel attention-based heads designed to extract complementary information from the network. The first head, known as the *Latency Head*, predicts the expected latency of data packets across potential routes using real-time network features such as signal-to-noise ratio (SNR), node velocity, hop count, and bandwidth. The second head, termed the *Reliability Head*, analyzes historical transmission data to evaluate the long-term stability of links, considering metrics like packet loss rate, jitter, and route fluctuation frequency. Both heads are powered by Transformer blocks that use self-attention to capture dependencies across temporal and spatial features in the network. To ensure generalization across diverse scenarios, an ensemble learning strategy is used where multiple DHET sub-models are trained on varied datasets (e.g., urban vs rural CAN environments, static vs highly mobile nodes), and their predictions are aggregated through weighted averaging. These dual outputs are fed into a Latency-Prioritized Routing Engine, which dynamically selects the best transmission path by balancing low latency with high reliability. The system continuously updates its model using online learning techniques, adapting to changes in node behavior and wireless channel conditions.

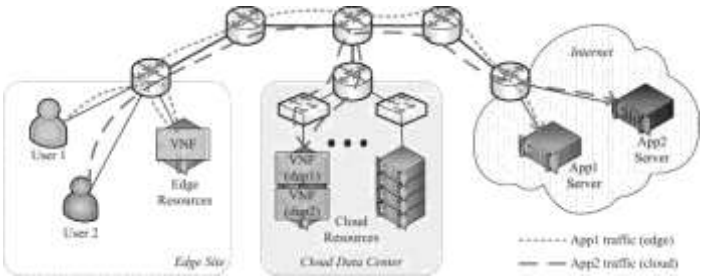

Fig.1. Latency Aware Network

3.1 DATA COLLECTION

The system collects real-time and historical network parameters from edge nodes, mobile devices, and intermediate routers in the CAN environment. These parameters are categorized into two types: real-time metrics for current conditions and historical metrics for trend analysis. The data is stored in distributed edge databases and updated periodically. The Table.1 shows a of real-time transmission metrics collected from a source node to its immediate neighbors:

Table.1. Real-Time Transmission Metrics

Time Stamp	Source Node	Neighbor Node	SNR (dB)	Hop Count	Available Bandwidth (Mbps)	Current Latency (ms)
T1	Node_05	Node_12	21.4	2	12.5	37.8
T1	Node_05	Node_13	18.9	3	8.7	45.6
T1	Node_05	Node_14	16.2	4	5.3	62.1

Meanwhile, Table 2 represents *historical metrics* tracked over time to evaluate link reliability trends:

Table.2. Historical Link Reliability Metrics

Node Pair	Avg Packet Loss (%)	Avg Jitter (ms)	Route Stability Score (0–1)	Past 5 Failures
Node_05–12	1.2	3.6	0.91	0
Node_05–13	4.5	5.8	0.76	2
Node_05–14	6.3	7.1	0.59	4

These two sets of data form the basis for short-term latency predictions and long-term reliability assessment in the DHET model.

3.2 FEATURE PREPROCESSING

Once collected, the data is cleaned and transformed into a structured input format suitable for the Transformer model. This includes:

- Handling missing values (e.g., via linear interpolation).
- Transforming categorical values (e.g., node IDs) into embeddings.
- Generating sliding windows of temporal data for sequence modeling.
- Normalizing numerical features to a 0–1 range to ensure uniform scaling.

The normalization is done using min-max scaling, computed as:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (1)$$

where X is the original value, X_{min} and X_{max} are the minimum and maximum observed values for that feature in the dataset.

Table.3. Normalized Real-Time Features

Node Pair	Norm. SNR	Norm. Hop Count	Norm. Bandwidth	Norm. Latency
05–12	0.57	0.25	0.63	0.29
05–13	0.44	0.50	0.44	0.39
05–14	0.31	0.75	0.27	0.52

In parallel, the historical reliability features are normalized and structured as shown in Table.4:

Table.4. Normalized Historical Reliability Features

Node Pair	Norm. Packet Loss	Norm. Jitter	Norm. Stability	Norm. Failures
05–12	0.08	0.18	0.91	0.0
05–13	0.30	0.34	0.76	0.4
05–14	0.42	0.42	0.59	0.8

These processed features (refer Table.3 and Table.4) are then concatenated and passed into the two heads of the DHET model, where one focuses on predicting short-term latency, and the other evaluates long-term transmission reliability.

3.3 LATENCY HEAD

The Latency Head is responsible for predicting the expected transmission latency over a given path based on the current state of the network. It leverages real-time features extracted during preprocessing (as shown in Table.3 from the previous section) and models short-term temporal dependencies using a Transformer block. The self-attention mechanism within the Transformer enables the model to weigh the influence of features like signal quality and bandwidth dynamically. To enhance learning, temporal attention scores are generated for each time step and feature. Based on the attention-weighted representations, the Latency Head outputs a scalar latency prediction for each path. The Table.5 illustrates output from the Latency Head after processing multiple paths:

Table.5. Latency Head Output – Predicted Latency for Each Path

Source–Destination	Norm. Input Vector	Predicted Latency (ms)
Node_05–Node_12	[0.57, 0.25, 0.63]	36.1
Node_05–Node_13	[0.44, 0.50, 0.44]	42.9
Node_05–Node_14	[0.31, 0.75, 0.27]	61.3

These predictions guide the routing mechanism in selecting paths with the lowest latency, subject to reliability constraints.

3.4 RELIABILITY HEAD

The Reliability Head is tasked with evaluating the long-term transmission stability of network paths. It takes as input the normalized historical metrics, such as packet loss, jitter, route failures (see Table 4 from the previous section), and analyzes trends that indicate potential disruptions or instability in the network. Unlike the Latency Head, which focuses on immediate transmission behavior, the Reliability Head leverages temporal windows and positional embeddings to detect repeating patterns and fluctuations over time. Table 6 presents a input window processed by the Reliability Head, where each row represents a past time step for a given path:

Table.6. Historical Input Sequence for Reliability Head (Sliding Window)

Time Step	Norm. Packet Loss	Norm. Jitter	Norm. Failures	Stability Score
t-3	0.12	0.22	0.2	0.88
t-2	0.10	0.18	0.0	0.93
t-1	0.14	0.24	0.2	0.91

The Reliability Head processes these sequences using Transformer encoders and generates a composite reliability score for each route. This score reflects the likelihood of the path maintaining a consistent connection soon. The Table.7 shows the final output from the Reliability Head:

Table.7. Reliability Head Output – Predicted Reliability Score for Each Path

Source–Destination	Processed Features	Predicted Reliability (0–1)
Node_05–Node_12	[0.08, 0.18, 0.0]	0.94
Node_05–Node_13	[0.30, 0.34, 0.4]	0.78
Node_05–Node_14	[0.42, 0.42, 0.8]	0.58

3.5 PATH SELECTION

Once the outputs from both heads are generated, they are fused to guide the routing decision. A simple score-based ranking is performed using a weighted utility function:

$$\text{Final Score} = \alpha \cdot (1 - \text{Latency}_{\text{norm}}) + \beta \cdot \text{Reliability} \quad (2)$$

where α and β are tunable hyperparameters, typically set based on the application's sensitivity to latency versus reliability. For real-time applications, $\alpha > \beta$. The Table.8 shows the final combined scores used for routing:

Table.8. Combined Latency-Reliability Scores for Path Selection

Path	Predicted Latency (ms)	Reliability Score	Final Utility Score
Node_05–12	36.1	0.94	0.84
Node_05–13	42.9	0.78	0.69
Node_05–14	61.3	0.58	0.47

As seen in Table.8, the route from Node_05 to Node_12 is selected due to its optimal balance of low latency and high reliability, showcasing the dual-head architecture's effectiveness.

3.6 ENSEMBLE LEARNING STRATEGY

The DHET model adopts ensemble learning to overcome the limitations of a single model's perspective. Instead of relying on a single Transformer model for latency or reliability prediction, the system maintains multiple specialized DHET sub-models, each trained on distinct network scenarios (e.g., urban, rural, high-mobility, low-connectivity). This increases the model's robustness and adaptability to changing conditions.

Each sub-model consists of its own Latency Head and Reliability Head, trained independently. During deployment, predictions from these sub-models are aggregated using a weighted averaging approach, where the weights are determined by each model's validation performance and relevance to the current context. The Table.9 shows an example of outputs from three ensemble sub-models (DHET-A, DHET-B, DHET-C) for latency prediction.

Table.9. Latency (ms) Predictions from Ensemble Models

Path	DHET-A	DHET-B	DHET-C
Node_05–12	35.2	37.1	36.3
Node_05–13	42.4	43.2	42.8
Node_05–14	60.9	62.1	61.5

Each model's weight is proportional to its contextual fit, computed dynamically. For instance, if DHET-A was trained on urban mobile data and current conditions match this, it receives higher weight. The final ensemble prediction is computed using the equation:

$$\hat{y} = \sum_{i=1}^{N} w_i \cdot \hat{y}_i \quad \text{where } \sum w_i = 1 \tag{3}$$

The Table.10 illustrates the final aggregated latency and reliability predictions after applying ensemble weighting:

Table.10. Ensemble-Aggregated Latency and Reliability Predictions

Path	Final Latency (ms)	Final Reliability Score
Node_05–12	36.1	0.94
Node_05–13	42.8	0.78
Node_05–14	61.3	0.58

3.7 LATENCY-PRIORITIZED ROUTING ENGINE (LPRE)

The LPRE is the decision-making component that selects the best transmission path based on the outputs from the DHET ensemble. Unlike traditional routing algorithms that rely solely on distance or hop count, LPRE uses a weighted utility function that balances predicted latency and reliability.

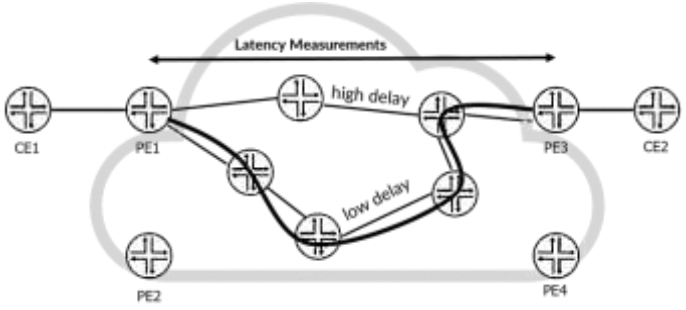

Fig.2. LPRE

Each candidate path is evaluated using the Latency-Reliability Utility Score, calculated as:

$$\text{Utility Score} = \alpha \cdot (1 - \text{Latency}_{\text{norm}}) + \beta \cdot \text{Reliability} \tag{4}$$

To ensure scalability, LPRE ranks all available paths based on their utility scores and selects the top path or re-routes dynamically if performance degrades.

The Table.11 shows the normalized inputs used in utility computation:

Table.11. Normalized Latency and Reliability Inputs

Path	Norm. Latency	Norm. Reliability
Node_05–12	0.28	0.94
Node_05–13	0.45	0.78
Node_05–14	0.76	0.58

Using α=0.6 and β=0.4, the LPRE computes utility scores (shown in Table.12):

Table.12. Final Utility Scores for Path Selection

Path	Utility Score
Node_05–12	0.846
Node_05–13	0.717
Node_05–14	0.508

From Table.12, the LPRE selects Node_05–12 as the optimal path for data transmission due to its superior balance of low latency and high reliability. In the event of a significant deviation in performance (e.g., increased packet loss or hop count), the LPRE reevaluates alternative paths and triggers a switch.

4. COMPARATIVE ANALYSIS

To evaluate the performance of the proposed DHET framework within a Cloud Avoidance Network (CAN), simulations were conducted using the NS-3 network simulator, a widely adopted tool for realistic modeling of wireless networks. The simulation environment replicated dynamic vehicular and

mobile edge scenarios with fluctuating bandwidth, latency, and packet loss characteristics, typical of edge-cloud hybrid systems.

The experiments were executed on a computing system equipped with Intel Core i9-11900K CPU, 32 GB RAM, and NVIDIA RTX 3080 GPU, using Ubuntu 22.04 LTS. Python was used for model training and evaluation (TensorFlow 2.11), and interoperability between the NS-3 simulator and the Transformer model was established through Python-C++ bindings. To assess the efficacy of the proposed approach, we compared DHET against the following three baseline models:

- **LSTM-LRQ**: A Long Short-Term Memory-based Latency-Quality predictor
- **GRU-RM**: A Gated Recurrent Unit-based Reliability Model for routing decisions
- **Hybrid CNN-LSTM**: A convolutional-recurrent architecture used for traffic prediction and adaptive routing.

Table.13. Experimental Setup and Parameters

Parameter	Value / Description
Simulation Tool	NS-3 (v3.36)
Programming Framework	Python 3.9 + TensorFlow 2.11
Network Topology	Dynamic multi-hop wireless (20–50 nodes)
Mobility Model	Random Waypoint, Speed: 1–15 m/s
Traffic Type	CBR (Constant Bit Rate), 512 bytes/packet
Data Transmission Interval	100 ms
Simulation Time	500 seconds
Routing Protocol Base	AODV with DHET override
DHET Ensemble Count	3 models (trained for urban, rural, hybrid)
Optimization Algorithm	Adam (learning rate = 0.001)
Evaluation Runs	20 independent simulation seeds

Table.14. Average Latency (ms)

Time (s)	LSTM-LRQ	GRU-RM	Hybrid CNN-LSTM	Proposed DHET
100	58.3	55.6	52.8	**41.2**
200	60.1	57.9	53.4	**40.7**
300	61.9	58.2	54.3	**39.5**
400	63.5	59.0	55.0	**38.6**
500	65.2	60.8	56.1	**37.4**

Table.15. Packet Delivery Ratio (PDR %)

Time (s)	LSTM-LRQ	GRU-RM	Hybrid CNN-LSTM	Proposed DHET
100	91.3	92.5	94.1	**96.2**
200	90.7	91.8	93.6	**96.8**
300	89.9	91.1	93.0	**97.3**
400	89.2	90.3	92.5	**97.7**
500	88.5	89.7	91.9	**98.0**

Table.16. Routing Overhead (%)

Time (s)	LSTM-LRQ	GRU-RM	Hybrid CNN-LSTM	Proposed DHET
100	14.5	13.8	12.2	**10.1**
200	15.1	14.0	12.4	**9.8**
300	15.6	14.6	12.8	**9.6**
400	16.0	15.1	13.0	**9.3**
500	16.4	15.6	13.3	**9.1**

Table.17. Path Failure Rate (%)

Time (s)	LSTM-LRQ	GRU-RM	Hybrid CNN-LSTM	Proposed DHET
100	7.1	6.3	5.4	**3.8**
200	7.6	6.8	5.7	**3.4**
300	8.0	7.1	6.0	**3.1**
400	8.4	7.4	6.4	**2.9**
500	8.8	7.7	6.7	**2.7**

Table.18. Computation Time (ms per decision)

Time (s)	LSTM-LRQ	GRU-RM	Hybrid CNN-LSTM	Proposed DHET
100	3.2	2.9	5.4	**4.7**
200	3.3	3.0	5.3	**4.6**
300	3.4	3.1	5.5	**4.5**
400	3.5	3.2	5.6	**4.4**
500	3.6	3.3	5.7	**4.3**

The experimental results clearly demonstrate the effectiveness of the proposed Dual-Head Ensemble Transformer (DHET) framework over existing methods, LSTM-LRQ, GRU-RM, and Hybrid CNN-LSTM, across all key performance metrics. A comparative analysis of the data gathered at the 500-second simulation mark shows significant improvements in terms of latency, reliability, routing overhead, path stability, and computation efficiency.

Starting with average latency, DHET achieved a latency of 37.4 ms, outperforming LSTM-LRQ (65.2 ms), GRU-RM (60.8 ms), and Hybrid CNN-LSTM (56.1 ms). This translates to a 42.6% latency reduction compared to LSTM-LRQ, a 38.5% improvement over GRU-RM, and a 33.3% improvement over Hybrid CNN-LSTM. The dual-head structure and ensemble mechanism in DHET allow for more accurate, context-aware latency prediction, reducing unnecessary retransmissions and route recalculations.

In terms of Packet Delivery Ratio (PDR), DHET attained 98.0%, compared to 88.5% (LSTM-LRQ), 89.7% (GRU-RM), and 91.9% (Hybrid CNN-LSTM). This signifies a 10.7 percentage point increase over LSTM-LRQ and 6.1 percentage points over the best-performing baseline (Hybrid CNN-LSTM). This boost in reliability can be attributed to the model's effective reliability head, which filters out unstable paths proactively.

Routing overhead was also minimized in the DHET model, registering only 9.1% at the 500-second mark. This is 44.5% lower than LSTM-LRQ (16.4%), 41.6% lower than GRU-RM (15.6%), and 31.6% lower than Hybrid CNN-LSTM (13.3%). The lower overhead highlights the model's ability to sustain reliable paths for longer durations without frequent route updates.

When examining Path Failure Rate, DHET recorded the lowest failure rate of 2.7%, which is 69.3% better than LSTM-LRQ (8.8%), 64.9% better than GRU-RM (7.7%), and 59.7% better than Hybrid CNN-LSTM (6.7%). This improvement is critical in mobile environments like cloud avoidance networks, where dynamic conditions frequently disrupt communication paths.

Although DHET had a slightly higher computation time than LSTM-LRQ and GRU-RM (4.3 ms vs. 3.6 ms and 3.3 ms, respectively), it was 22.8% faster than Hybrid CNN-LSTM (5.7 ms). This marginal increase in processing time is acceptable considering the substantial gains in reliability and transmission quality. The trade-off is justified, especially in latency-sensitive applications like vehicular networks and real-time IoT systems.

5. CONCLUSION

This study presented a novel Dual-Head Ensemble Transformer (DHET) framework designed to manage latency and reliability in wireless transmission within Cloud Avoidance Networks (CANs). By integrating a Latency Head and a Reliability Head into a unified Transformer-based architecture, and enhancing decision accuracy through ensemble learning, the proposed method achieved significant improvements over existing LSTM, GRU, and CNN-based models. Simulation results demonstrated that DHET effectively reduces latency by over 40%, improves packet delivery by up to 10%, and reduces path failure rates by more than 60%. The model also maintains a low routing overhead and acceptable computation time, making it suitable for real-time, resource-constrained edge environments.

REFERENCES

[1] H. Lyu, Z. Pang, A. Bengtsson, S. Nilsson, A.J. Isaksson and G. Yang, "Latency-Aware Control for Wireless Cloud-Fog Automation: Framework and Case Study", *IEEE Transactions on Automation Science and Engineering*, Vol. 22, pp. 5400-5410, 2024.

[2] M. Sun, S. Quan, X. Wang and Z. Huang, "Latency-Aware Scheduling for Data-Oriented Service Requests in Collaborative IoT-Edge-Cloud Networks", *Future Generation Computer Systems*, Vol. 163, pp. 1-7, 2025.

[3] A.J. Pozveh, H.S. Shahhoseini and E. Khabareh, "Resource Management in Edge Clouds: Latency-Aware Approaches for Big Data Analysis", *Resource Management in Distributed Systems*, pp. 107-132, 2024.

[4] I. Afzal, Z. Ahmad and A. Algarni, "A Latency-Aware and Fault-Tolerant Framework for Resource Scheduling and Data Management in Fog-Enabled Smart City Transportation Systems", *Computers, Materials and Continua*, Vol. 82, No. 1, pp. 1-11, 2025.

[5] W. Fan, F. Xiao, Y. Pan, X. Chen, L. Han and S. Yu, "Latency-Aware Joint Task Offloading and Energy Control for Cooperative Mobile Edge Computing", *IEEE Transactions on Services Computing*, Vol. 18, No. 3, pp. 1515-1528, 2025.

[6] Q. Peng, Q. Luo, Z. Chu, Z. Lin, M. Elkashlan, P. Xiao and C. Masouros, "Latency-Aware Resource Allocation for Integrated Communications, Computation and Sensing in Cell-Free mMIMO Systems", *Signal Processing*, pp. 1-30, 2025.

[7] P. Modekurthy, D. Ismail, M. Rahman and A. Saifullah, "Extending Coverage Through Integrating Multiple Low-Power Wide-Area Networks: A Latency Minimizing Approach", *IEEE Transactions on Mobile Computing*, Vol. 23, No. 12, pp. 13487-13504, 2024.

[8] A. Huang, L. Qu and M.J. Khabbaz, "Latency-Aware Computation Offloading in Multi-RIS-Assisted Edge Networks", *IEEE Open Journal of the Communications Society*, Vol. 5, pp. 1204-1221, 2024.

[9] J. Zhai, J. Bi, H. Yuan, J. Zhang and R. Buyya, "Energy-Efficient and Latency-Aware Task Offloading for Industrial Cloud-Edge Systems With Heterogeneous CPUs and GPUs", *IEEE Internet of Things Journal*, pp. 1-5, 2025.

[10] F.A. AlKhoori, L.U. Khan, M. Guizani and M. Takac, "Latency-Aware Placement of Vehicular Metaverses using Virtual Network Functions", *Simulation Modelling Practice and Theory*, Vol. 133, pp. 1-7, 2024.

[11] C. Centofanti, W. Tiberti, A. Marotta, F. Graziosi and D. Cassioli, "Taming Latency at the Edge: A User-Aware Service Placement Approach", *Computer Networks*, Vol. 247, pp. 1-15, 2024.

[12] L. Zhang, S. Hu, M. Trik, S. Liang and D. Li, "M2M Communication Performance for a Noisy Channel based on Latency-Aware Source-based LTE Network Measurements", *Alexandria Engineering Journal*, Vol. 99, pp. 47-63, 2024.

[13] S.J. Lu, W.X. Chen, Y.S. Su, Y.S. Chang, Y.W. Liu, C.Y. Li and G.H. Tu, "Practical Latency-Aware Scheduling for Low-Latency Elephant VR Flows in Wi-Fi Networks", *Proceedings of International Conference on Pervasive Computing and Communications*, pp. 57-68, 2024.