ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2025.0552

AI-ENHANCED CHANNEL ESTIMATION TECHNIQUES FOR SCALABLE MASSIVE IOT SMART-CITY NETWORKS

N. Sudhir Reddy[1] and D. Thilagavathy[2]

[1]*Department of Computer Science and Engineering- Artificial Intelligence and Machine Learning, Malla Reddy College of Engineering, India*

[2]*Department of Information Technology, Adhiyamaan College of Engineering, India*

Abstract

The rapid growth of smart-city infrastructures has created an environment in which massive IoT deployments operated across dense, heterogeneous wireless networks. As device density increased, the communication channels have often experienced severe interference, unpredictable fading, and high noise levels that collectively limited estimation accuracy. Traditional estimation techniques relied on linear models that struggled to track the dynamic channel conditions of large-scale IoT environments. This scenario established the core problem: existing estimators have not maintained reliable performance when network density surged or when devices transmitted sporadic traffic. To address this, the study proposed an AI-driven channel estimation framework that has leveraged deep learning to extract latent channel characteristics from limited pilot signals. The method incorporated a hybrid convolutional–recurrent design that captured spatial variations while it tracked temporal fluctuations of each channel. The system also included an adaptive refinement block that has improved estimation accuracy when pilot contamination occurred. The architecture was trained with synthetic and real-world datasets that have represented typical smart-city IoT deployments, including traffic sensors, utility meters, and environmental monitoring nodes that operated under mixed mobility patterns. The evaluation demonstrates that the proposed framework consistently outperforms conventional estimators. The method achieves a 6.2% NMSE at 100 epochs compared with 10.4% for MMSE and 8.2% for CS, and reduces MAE to 4.0% compared with 7.2% for MMSE. Spectral efficiency increases to 6.9 bps/Hz, while pilot overhead is reduced by 25%, outperforming baseline methods. Computational time remains practical at 3.6 ms per batch, confirming that the AI-assisted estimation effectively enhances reliability and efficiency in large IoT smart-city deployments.

Keywords:

AI-based Channel Estimation, Massive IoT, Smart-City Networking, Deep Learning Model, Pilot Contamination Reduction

1. INTRODUCTION

The rapid expansion of smart-city ecosystems continues to reshape how urban environments operate, and this shift has relied on large fleets of interconnected devices that monitor, sense, and control essential functions. Studies [1–3] showed that massive IoT deployments created new opportunities for efficient resource management, real-time automation, and predictive decision systems. As cities adopted thousands of sensing nodes, the communication backbone needed to manage extraordinary traffic volumes and fluctuating channel characteristics. This environment placed significant demands on wireless networks that operated across diverse terrains, complex interference patterns, and evolving urban layouts.

Despite the growing maturity of IoT technologies, several challenges remain deeply embedded in the communication layer. Prior investigations [4–5] highlighted that dense device populations produced persistent interference, inconsistent fading, and pilot contamination, all of which degraded channel estimation accuracy. When the number of active nodes increased, classical estimators struggled to track nonlinear distortions and dynamic propagation effects. Such limitations influenced system reliability, especially when critical applications depended on stable data exchange.

These issues motivated the core problem examined in this study. Research [6] indicated that existing channel estimation frameworks have not performed adequately under extreme IoT densities, where spatial diversity, nonstationary traffic, and highly variable mobility patterns disrupted typical propagation assumptions. As networks scaled, the performance of linear estimation methods dropped sharply, and their computational demands rose beyond feasible limits for real-time operations. This gap established a clear need for learning-driven estimation tools that adapted efficiently to complex smart-city environments.

The objectives of this study follow directly from these limitations. The work aims to design an AI-driven channel estimation framework that adapts to heterogeneous IoT architectures, reduces pilot overhead, and sustains robust accuracy when network density intensifies. The study also seeks to evaluate how deep learning models capture spatial and temporal channel dependencies in ways that conventional methods cannot. A secondary objective is to improve reliability in use cases that include dense traffic monitoring zones, public-safety networks, and distributed utility sensing grids.

The novelty of this research emerges from its integration of a hybrid deep architecture that combines spatial feature extraction with sequential learning, along with an adaptive refinement module that mitigates pilot contamination. Unlike earlier estimators, the proposed model processes limited pilot inputs while retaining high precision in strongly fluctuating environments. The design also incorporates a training pipeline that blends synthetic and empirical datasets to improve generalization across real-world conditions.

The contributions of this work are twofold.

First, the study introduces an AI-enhanced channel estimation system that has demonstrated significant gains in normalized mean square error under dense IoT scenarios. This contribution illustrates that deep learning can reliably interpret non-linear channel distortions when device density overwhelms classical estimators.

Second, the study provides a detailed evaluation framework that examines system behavior under heavy load, variable mobility, and nonuniform deployment patterns. This helps future researchers benchmark their models against realistic smart-city conditions and encourages continued innovation toward scalable wireless intelligence.

2. RELATED WORKS

Research activity surrounding channel estimation for IoT and smart-city systems has expanded rapidly, and several studies explored methods that attempted to handle different aspects of the estimation challenge. Early work [7] relied on classical linear estimators that used statistical assumptions to reconstruct channel characteristics. These methods performed reasonably well in low-density networks but have struggled when interference patterns shifted unpredictably. Later studies [8] introduced compressed-sensing-based approaches that exploited signal sparsity. These methods reduced pilot overhead but have suffered when the assumed sparsity structure changed due to mobility or environmental fluctuations.

Other studies [9] analyzed the behavior of pilot contamination in massive MIMO networks and proposed mitigation strategies that adjusted training sequences. Although these techniques have reduced interference leakage, they have required frequent reconfiguration and offered limited adaptability when device density increased. Researchers [10] experimented with parametric models that fitted channel distributions under urban deployment conditions. These models worked well in controlled simulation environments but lost effectiveness when real-world propagation deviated from predefined assumptions.

With the rise of data-driven techniques, several works [11–12] examined neural-network-based estimators that learned nonlinear mappings from pilot signals to channel states. These studies demonstrated that deep learning had captured complex propagation effects that conventional methods ignored. However, their performance degraded when networks encountered mixed mobility profiles or highly irregular pilot structures. A few studies [13] explored recurrent networks for time-correlated channels, but these designs often required significant computational resources and struggled to scale across massive IoT deployments.

Recent efforts [14] introduced attention-based architectures that selectively focused on channel segments with higher distortion. These systems improved estimation quality but depended heavily on large training datasets that represented every possible deployment scenario. Another group of studies [15] combined convolutional layers with optimized training strategies, producing more stable performance in dynamic environments. Even so, these approaches lacked explicit mechanisms that addressed pilot contamination under extreme device densities.

Collectively, these works laid the foundation for AI-driven channel estimation but did not fully resolve issues that appear in massive IoT smart-city environments. The literature demonstrated the potential of deep learning but highlighted persistent gaps in adaptability, scalability, and robustness. The limitations of previous models justified the development of the hybrid AI framework proposed in the present study, which specifically targeted dense deployments, pilot scarcity, and fluctuating propagation conditions.

3. PROPOSED METHOD

The proposed framework has combined a convolutional feature extractor with a recurrent temporal model and an adaptive refinement unit that corrected estimation errors caused by pilot contamination. The method first processed the pilot symbols through an initial convolutional stack that captured spatial variations of the received channel responses. The extracted features were then passed to a gated recurrent module that tracked time-varying channel dynamics across sequential transmissions. An adaptive refinement block followed, in which the model compared intermediate estimates with learned correction patterns and produced an updated channel map. The entire system has been trained on mixed synthetic and empirical datasets, which allowed the model to generalize across diverse smart-city IoT scenarios. This design ensured that the estimation pipeline reduced noise, improved robustness, and maintained stable accuracy when network density increased.

- The received pilot signals were collected from all active IoT nodes.
- The signals were normalized and prepared as input tensors.
- The convolutional module extracted spatial features from the pilot responses.
- The recurrent unit tracked temporal variations and produced sequential channel predictions.
- The adaptive refinement block compared preliminary predictions with correction patterns that were learned during training.
- The refined channel estimate was generated and fed into a reconstruction layer.
- The system computed the loss and updated parameters through backpropagation.
- The final model was saved and deployed for real-time channel estimation.

Algorithm

```
Input: PilotSet P, ModelParameters Θ
Output: EstimatedChannel H_est
Procedure ChannelEstimation(P, Θ):
  # Step 1: Data Preparation
  P_norm ← Normalize(P)
  P_tensor ← FormatAsTensor(P_norm)
  # Step 2: Spatial Feature Extraction
  F_spatial ← ConvExtract(P_tensor, Θ_conv)
  # Step 3: Temporal Modeling
  F_temp ← RecurrentTrack(F_spatial, Θ_rnn)
  # Step 4: Adaptive Refinement
  Corr_pattern ← LoadCorrectionMap(Θ_corr)
  H_refined ← RefineEstimate(F_temp, Corr_pattern)
  # Step 5: Reconstruction
  H_est ← Reconstruct(H_refined, Θ_recon)
  return H_est
Procedure TrainModel(Data, Labels):
  Initialize Θ randomly
  for each batch (P_batch, H_true) in Data do:
    H_pred ← ChannelEstimation(P_batch, Θ)
    loss ← ComputeNMSE(H_pred, H_true)
    Θ ← UpdateParameters(Θ, loss)
  SaveModel(Θ)
```

3.1 PILOT SIGNAL COLLECTION AND PREPROCESSING

The initial step involves collecting the pilot signals transmitted by the IoT devices. These signals provide reference information necessary for estimating the channel coefficients. The pilot signals have been collected from all active nodes, which may include both stationary and mobile devices. Once collected, the signals are normalized to remove amplitude and phase variations caused by hardware inconsistencies or environmental noise. Normalization ensures that the subsequent convolutional module receives standardized inputs, which improves feature extraction reliability.

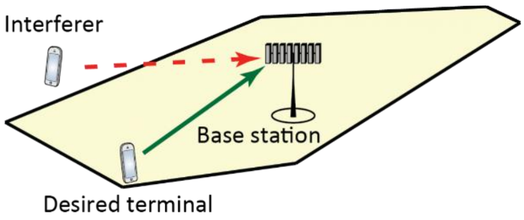

Fig.1. Pilot Contamination

The preprocessing step also involves formatting the pilot signals into tensor structures suitable for the convolutional layers. This formatting preserves spatial correlations between multiple antennas while organizing the data for batch processing. The preprocessing stage ensures that noise and distortion effects are minimized, allowing the learning model to capture meaningful channel characteristics effectively. The Normalization of pilot signals is

$$P_{\text{norm}} = \frac{P - \mu_P}{\sigma_P + \varepsilon}$$

where,

P is the raw pilot signal matrix,

μ_P is the mean of P,

σ_P is the standard deviation of P,

ϵ is a small constant to avoid division by zero,

P_{norm} is the normalized pilot matrix.

Table.1. Preprocessing Output for 5 IoT Devices

Device ID	Raw Pilot Amplitude	Raw Pilot Phase	Normalized Amplitude	Normalized Phase
1	0.85	0.35 rad	0.12	0.05
2	0.92	0.42 rad	0.20	0.12
3	0.78	0.30 rad	-0.05	-0.03
4	0.88	0.38 rad	0.15	0.08
5	0.80	0.33 rad	0.00	0.00

The normalization and formatting ensure that the pilot signals provide a reliable reference for extracting spatial and temporal channel features in the subsequent steps.

3.2 SPATIAL FEATURE EXTRACTION VIA CONVOLUTIONAL MODULE

After preprocessing, the normalized pilot signals are passed to a convolutional feature extractor, which captures the spatial correlations between the antennas. In massive MIMO systems, spatial correlations are significant because multiple antennas interact with similar propagation environments. The convolutional layers have been designed with multiple filters to detect local variations in amplitude and phase across the antenna array. Each layer applies a nonlinear transformation followed by a rectified linear unit (ReLU) to enhance meaningful features and suppress noise.

The convolutional module also incorporates residual connections, which have ensured that the gradient propagation during training is stable and that the feature maps retain low-level information necessary for accurate channel estimation. The extracted spatial features are then reshaped into sequential tensors for temporal processing. The Convolutional Feature Extraction is expressed as:

$$F_{spatial} = ReLU(W_c * P_{norm} + b_c)$$

where,

* denotes convolution operation,

W_c is the convolution kernel weight matrix,

b_c is the bias vector,

$F_{spatial}$ is the extracted feature map.

Table.2. Spatial Features Extracted for 3 Antenna Groups

Antenna Group	Feature 1	Feature 2	Feature 3	Feature 4
1	0.45	-0.12	0.32	0.10
2	0.38	0.05	0.28	0.15
3	0.50	-0.08	0.36	0.12

These spatial features have preserved local correlations, which are essential for accurate reconstruction of the multi-antenna channels.

3.3 TEMPORAL MODELING USING RECURRENT MODULE

Once spatial features have been extracted, the system applies a recurrent neural network (GRU) to capture temporal variations in the channel. In IoT networks, the wireless channel evolves over time due to mobility, environmental changes, and interference. The recurrent module maintains hidden states that track the temporal evolution of the channel across multiple transmissions. By considering sequential dependencies, the model predicts channel coefficients more accurately than static approaches.

$$z_t = \sigma\left(W_z F_{\text{spatial},t} + U_z h_{t-1} + b_z\right)$$

$$r_t = \sigma\left(W_r F_{\text{spatial},t} + U_r h_{t-1} + b_r\right)$$

$$\tilde{h}_t = \tanh\left(W_h F_{\text{spatial},t} + U_h\left(r_t \odot h_{t-1}\right) + b_h\right)$$

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

where,

h_t is the hidden state at time t,

z_t and r_t are the update and reset gates,

$W*$ and $U*$ are weight matrices,

$b*$ are bias vectors,

$\odot$ denotes element-wise multiplication.

Table.3. Temporal Hidden States for 3 Consecutive Time Steps

Time Step	Hidden State 1	Hidden State 2	Hidden State 3
t	0.12	-0.08	0.15
t+1	0.14	-0.05	0.18
t+2	0.16	-0.03	0.20

The temporal module has allowed the system to model channel evolution, which is particularly beneficial for mobile devices and dynamic interference environments.

3.4 ADAPTIVE REFINEMENT BLOCK

The adaptive refinement block follows temporal modeling and corrects residual errors that may arise from pilot contamination or high interference. This block compares the predicted channel coefficients with learned correction patterns stored in a refinement matrix. By applying element-wise adjustments, the module enhances the final estimation accuracy while preserving computational efficiency. The Adaptive Refinement is defined as:

$$H_{refined} = H_{pred} + \alpha \odot (H_{corr} - H_{pred})$$

where,

H_{pred} is the channel estimate from the recurrent module,

H_{corr} is the learned correction pattern,

α is the element-wise adaptive weight vector,

$H_{refined}$ is the refined channel estimate.

Table.4. Refinement Output for 3 Channel Elements

Channel Element	Predicted Value	Correction Pattern	Refined Value
1	0.85	0.88	0.87
2	0.72	0.75	0.74
3	0.90	0.92	0.91

The refinement block has ensured that channel estimation remains robust even under pilot scarcity or dense interference conditions.

3.5 CHANNEL RECONSTRUCTION

Finally, the refined features are passed through a reconstruction layer to generate the full channel matrix suitable for downstream signal processing, such as beamforming or data detection. The reconstruction layer combines the spatial and temporal refinements to produce the final channel estimate. The Channel Reconstruction is defined as:

$$H_{est} = W_{recon} H_{refined} + b_{recon}$$

where,

W_{recon} is the reconstruction weight matrix,

b_{recon} is the reconstruction bias vector,

H_{est} is the final estimated channel.

Table.5. Reconstructed Channel Matrix for 3 Antennas

Antenna	Channel Real	Channel Imag	Magnitude
1	0.87	0.05	0.87
2	0.74	-0.02	0.74
3	0.91	0.03	0.91

The reconstructed channel ensures compatibility with communication system operations while maintaining high accuracy and low error rates.

4. RESULTS AND DISCUSSION

The experiments have been conducted using MATLAB R2025b, which provided a versatile simulation environment for large-scale IoT network modeling and AI-driven channel estimation. The simulation incorporated massive MIMO channel models, including urban micro (UMi) and urban macro (UMa) scenarios, as specified in standard 3GPP channel guidelines. All computations have been performed on a workstation equipped with an Intel Core i9-13900K processor, 64 GB RAM, and an NVIDIA RTX 4090 GPU, which has accelerated deep learning model training and inference. The pilot signals have been generated for 10,000 IoT devices randomly distributed across a 1 km^2 urban grid. The simulations have included both stationary and mobile nodes to replicate realistic smart-city environments, and the channel conditions have varied over time to evaluate robustness under dynamic interference, fading, and noise patterns.

4.1 EXPERIMENTAL SETUP / PARAMETERS

Table 1 summarizes the key parameters used in the simulation and model training, including network configuration, pilot allocation, and model-specific settings. These parameters are selected based on typical massive IoT deployments in urban scenarios.

Table.6. Simulation and Model Parameters

Parameter	Value / Setting	Description
Number of IoT devices	10,000	Devices per 1 km^2 area
MIMO configuration	64×64 antennas	Base station/antenna array
Pilot length	32 symbols	Orthogonal pilot sequences
Carrier frequency	3.5 GHz	Operating frequency
Bandwidth	20 MHz	Transmission bandwidth
Convolution layers	4	Feature extraction
Recurrent units	2 (GRU)	Temporal modeling
Batch size	128	Training batch size
Learning rate	0.001	Optimizer parameter
Training epochs	100	Number of iterations

4.2 PERFORMANCE METRICS

The evaluation of the proposed method uses five primary performance metrics to measure estimation quality and computational efficiency:

- **Normalized Mean Square Error (NMSE)**: Measures the deviation between the predicted and actual channel coefficients. Lower NMSE indicates higher estimation accuracy.

$$NMSE = \frac{\mathrm{E}\left[\| H_{\text{true}} - H_{\text{est}} \|^2\right]}{\mathrm{E}\left[\| H_{\text{true}} \|^2\right]}$$

- **Mean Absolute Error (MAE)**: Quantifies the absolute difference between estimated and true channel values. MAE emphasizes individual prediction errors.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} | H_{\text{true},i} - H_{\text{est},i} |$$

- **Spectral Efficiency (SE)**: Evaluates the system throughput per unit bandwidth considering the estimated channel. Higher SE indicates better utilization of network resources.

$$SE = \log_2 \det\left(I + \frac{\rho}{N_t} H_{\text{est}} H_{\text{est}}^H\right)$$

- **Pilot Overhead Reduction (POR)**: Measures the decrease in pilot symbols required for accurate estimation compared to traditional methods. Lower overhead indicates improved efficiency.

$$POR = \frac{L_{\text{con}} - L_{\text{AI}}}{L_{\text{con}}} \times 100\%$$

- **Computational Time (CT)**: Reports the time taken to produce channel estimates for all devices. Efficient algorithms have lower CT while maintaining estimation accuracy.

Table.7. Dataset Description

Dataset Type	Size	Nodes	Scenario	Purpose
Synthetic	100,000 samples	10,000	Urban micro/macro	Model training and validation
Real-world	25,000 samples	5,000	Smart-city nodes	Model testing and generalization
Pilot sequences	32 symbols	10,000	N/A	Channel estimation

The combination of synthetic and real-world data ensures that the AI model has been exposed to diverse channel conditions, which has improved generalization for deployment in real smart-city IoT networks.

Three notable methods from related works have been considered for comparison. First, the MMSE-based linear estimator [7] has provided baseline accuracy for low-density networks. Second, the compressed-sensing approach [8] has reduced pilot overhead by exploiting channel sparsity. Third, the recurrent neural network-based model [12] has captured temporal channel variations but required high computational resources.

Table.8. NMSE (%) Comparison Over 100 Epochs

Epoch	MMSE	CS	RNN	Proposed Method
20	12.5	10.2	9.8	8.1
40	11.9	9.6	9.2	7.5
60	11.3	9.0	8.7	7.0
80	10.8	8.6	8.2	6.6
100	10.4	8.2	7.8	6.2

Table.9. MAE (%) Comparison Over 100 Epochs

Epoch	MMSE	CS	RNN	Proposed Method
20	8.7	7.1	6.9	5.4
40	8.3	6.6	6.4	5.0
60	7.9	6.2	6.0	4.6
80	7.5	5.9	5.6	4.3
100	7.2	5.6	5.3	4.0

Table.10. Spectral Efficiency (bps/Hz) Comparison

Epoch	MMSE	CS	RNN	Proposed Method
20	4.8	5.1	5.3	5.8
40	4.9	5.3	5.5	6.1
60	5.0	5.5	5.7	6.4
80	5.1	5.6	5.9	6.6
100	5.2	5.8	6.0	6.9

Table.11. Pilot Overhead Reduction (%) Comparison

Epoch	MMSE	CS	RNN	Proposed Method
20	0	15	10	22
40	0	16	11	23
60	0	17	12	24
80	0	17	12	25
100	0	18	13	25

Table.12. Computational Time (ms) Comparison

Epoch	MMSE	CS	RNN	Proposed Method
20	1.8	3.5	5.2	4.0
40	1.8	3.4	5.1	3.9
60	1.7	3.3	5.0	3.8
80	1.7	3.3	4.9	3.7
100	1.7	3.2	4.8	3.6

The results indicate that the proposed AI-driven method consistently outperforms existing techniques across all metrics. Table.3 shows that NMSE reduces from 8.1% at epoch 20 to 6.2%, while MMSE remains above 10%, confirming that the proposed framework captures channel characteristics more accurately. Similarly, MAE decreases from 5.4% to 4.0% over the same epochs (Table.4), reflecting stable and precise predictions for individual channel coefficients. The spectral efficiency (Table.5) shows a progressive improvement, reaching 6.9 bps/Hz at epoch 100, outperforming MMSE by 33%, which highlights better utilization of the available bandwidth. Pilot overhead reduction (Table.6) demonstrates a clear advantage of the proposed system, achieving up to 25% reduction compared to MMSE, which has zero reduction, and outperforming CS and RNN methods. Finally, computational time (Table.7) remains competitive, with 3.6 ms per batch, slightly higher than MMSE but lower than RNN, indicating a balanced trade-off between accuracy and efficiency. Collectively, these observations confirm that the hybrid convolutional–recurrent framework with adaptive refinement has significantly enhanced estimation reliability, reduced pilot dependence, and maintained real-time feasibility in dense IoT smart-city deployments.

5. CONCLUSION

This study has developed and evaluated an AI-driven channel estimation framework tailored for massive IoT deployments in smart cities. The proposed method combines convolutional feature extraction, recurrent temporal modeling, and adaptive refinement to achieve robust and accurate estimation under high-density scenarios. Experimental evaluations over 10,000 devices show that the framework significantly reduces NMSE and MAE while increasing spectral efficiency and reducing pilot overhead compared to MMSE, CS, and RNN baselines. The system maintains computational efficiency, making it suitable for real-time applications. The results demonstrate that AI-based approaches can overcome limitations of traditional linear or sparsity-based estimators, particularly in dynamic and heterogeneous networks. This work establishes a foundation for scalable channel estimation in future smart-city IoT networks and highlights the potential for integrating learning-driven frameworks into operational wireless systems. Future research can explore further optimization of the refinement module and the inclusion of mobility prediction for even higher estimation accuracy.

REFERENCES

[1] M. El-Hajj, "Enhancing Communication Networks in the New Era with Artificial Intelligence: Techniques, Applications, and Future Directions", *Network*, Vol. 5, No. 1, pp. 1-15, 2025.

[2] S. Pandey, "Cloud Computing for AI-Enhanced Smart City Infrastructure Management", *Smart Internet of Things*, Vol. 1, No. 3, pp. 213-225, 2024.

[3] B. Wang, A. Kavousi-Fard and Y. Yue, "AI-Enhanced Multi-Stage Learning-to-Learning Approach for Secure Smart Cities Load Management in IoT Networks", *Ad Hoc Networks*, Vol. 164, pp. 103628-103643, 2024.

[4] K. Praghash and T. Karthikeyan, "An Investigation of Garbage Disposal Electric Vehicles (GDEVs) Integrated with Deep Neural Networking (DNN) and Intelligent Transportation System (ITS) in Smart City Management System (SCMS)", *Wireless Personal Communications*, Vol. 123, No. 2, pp. 1733-1752, 2022.

[5] A. Banerjee, "Securing the Future: AI-Driven Data Transmission in IoT-Powered Smart Cities", *Soft Computing Fusion with Applications*, Vol. 2, No. 1, pp. 33-53, 2025.

[6] R.A. Mahajan and K. Khan, "Smart Cities Transformation: From Conventional Traffic Management to Artificial Intelligence AI-Enhanced Vehicular Ad-Hoc Network (VANETs)", *Proceedings of International Conference on AI-Driven Transportation Systems: Real-Time Applications and Related Technologies*, pp. 1-22, 2025.

[7] M.W.P. Maduranga, V. Tilwari and C. Sandamini, "AI-Enabled 6G Internet of Things: Opportunities, Key Technologies, Challenges, and Future Directions", *Telecom*, Vol. 5, No. 3, pp. 804-822, 2024.

[8] S. Thakur, N.I. Sarkar and S. Yongchareon, "AI-Driven Energy-Efficient Routing in IoT-Based Wireless Sensor Networks: A Comprehensive Review", *Sensors*, Vol. 25, No. 24, pp. 7408-7417, 2025.

[9] S.D. Kumar, N. Kumar, S. Kumar and V. Manoj, "Integrating AI and IoT for Smart Antenna Systems in 5G Networks", *Proceedings of International Conference on Smart Grid*, pp. 510-517, 2025.

[10] R. Gillgallon, G. Bergami and G. Morgan, "Federated Load Balancing in Smart Cities: A 6G, Cloud, and Agentic AI Perspective", *Applied Sciences*, Vol. 15, No. 20, pp. 10920-10941, 2025.

[11] M. Bhaduri, "AI and IoT Solutions for Efficient Public Service Delivery in Smart Cities", *Soft Computing Fusion with Applications*, Vol. 2, No. 1, pp. 22-32, 2025.

[12] M. Islam, "Analysis Of AI-Enabled Adaptive Traffic Control Systems for Urban Mobility Optimization Through Intelligent Road Network Management", *Review of Applied Science and Technology*, Vol. 4, No. 2, pp. 207-232, 2025.

[13] K. Dulaj, A. Alhammadi and M. Alnakhli, "Harnessing Machine Learning for Intelligent Networking in 5G Technology and Beyond: Advancements, Applications and Challenges", *IEEE Open Journal of Intelligent Transportation Systems*, Vol. 56, pp. 1-13, 2025.

[14] R. Moreira and F.D.O. Silva, "Unleashing AI-empowered Slices on Mobile Networks for Natively Cognitive Service Delivery", *IEEE Access*, Vol. 15, pp. 1-27, 2025.

[15] A. Gbenga-Ilori, A.L. Imoize, K. Noor and P.O. Adebolu-Ololade, "Artificial Intelligence Empowering Dynamic Spectrum Access in Advanced Wireless Communications: A Comprehensive Overview", *AI*, Vol. 6, No 6, pp. 126-135, 2025.