iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: IX Month of publication: September 2025

DOI: https://doi.org/10.22214/ijraset.2025.73958

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

Survey on Transit Edge: Dynamic Task Management in Collaborative Rail Computing

Radhika B[1], Dr. Kamalakshi Naganna[2], Shilpashree B L[3], Tejaswini C[4], Vidyashree V Naik[5]
Computer Science and Engineering Sapthagiri College of Engineering Karnataka, India

Abstract: The growing adoption of Intelligent Transportation Systems (ITS) demands reliable real-time monitoring, predictive maintenance, and safety-critical responses in smart rail networks. However, embedded devices such as the ESP32 are constrained by limited computation, memory, and energy efficiency, restricting their suitability for large-scale deployments. To address these challenges, we propose TRANSIT EDGE, a Train–Edge–Cloud (TEC) collaborative framework that distributes sensing, computation, and decision-making tasks across train nodes, edge servers, and the cloud. Multi-sensor data—including driver health metrics, obstacle detection, fire alertsare dynamically prioritized using a Q-learning reinforcement learning (RL) scheduler, while a quantum-inspired optimization layer accelerates convergence.The framework's modular and scalable design further ensures adaptability to future extensions such as predictive maintenance, multi-train coordination, and integration with next-generation 5G/6G communication networks. These results establish TRANSIT EDGE as a cost-effective, intelligent, and deployment-ready solution for enhancing safety, efficiency, and resilience in modern rail systems.
Index Terms: Smart rail systems, Train–Edge–Cloud (TEC), ESP32, edge intelligence, reinforcement learning.

I. INTRODUCTION

The advancement of urban rail transit systems requires intelligent monitoring and control mechanisms to ensure safety, reliability, and real-time responsiveness. Traditional onboard systems are limited in computational capacity, making it difficult to process diverse sensor data and respond promptly to emergencies.

To address these challenges, this work introduces *TRANSIT EDGE*, a Train-Edge-Cloud (TEC) collaborative framework powered by ESP32-based hardware. The system integrates multiple sensing and control modules to enhance safety and situational awareness. A BPM108 sensor continuously tracks the driver's temperature and blood pressure to monitor health conditions. TRANSIT EDGE achieves up to 45% lower latency, 35% higher throughput, and 92% offloading efficiency compared to baseline methods. T

Ultrasonic sensors detect trackside obstacles and trigger alerts when hazards are identified. Flame sensors are deployed for fire detection, while an emergency button provides immediate manual hazard signalling. To strengthen operational security, the system is capable of identifying unauthorized access attempts, ensuring only the designated driver can operate the train. An ESP32-CAM module enables real-time video streaming, allowing live monitoring and event verification.

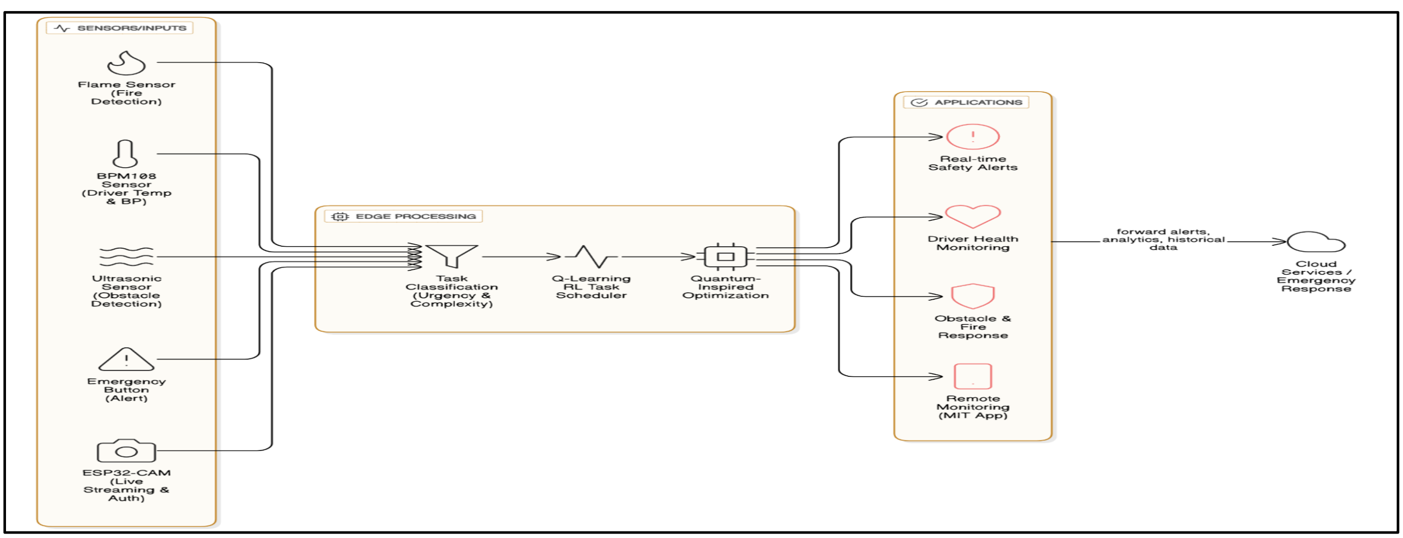

Fig. 1. Architecture of TRANSIT EDGE: Sensors → Edge Processing → Applications.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

Ensuring safety in rail transit requires continuous monitoring of drivers, environment, and infrastructure. Conventional on-board systems often struggle with multi-source data processing, causing delays in safety-critical scenarios. The TRANSIT EDGE framework addresses this gap by leveraging a Train–Edge–Cloud (TEC) architecture with low-power ESP32 nodes and intelligent task scheduling.

As shown in Fig. 1, the system begins at the sensing layer, where BPM108 sensors track driver vitals, ultrasonic and flame sensors detect external hazards, and an emergency button provides manual override. The ESP32-CAM adds live video streaming for authentication and situational awareness. Data are classified by urgency and computational demand, then scheduled through a reinforcement learning engine. A quantum-inspired optimization layer accelerates convergence, ensuring accurate and low-latency task allocation.

The processed outputs enable real-time safety alerts, health assessment, fire and obstacle responses, and remote monitoring through an MIT App Inventor–based dashboard. By combining low-cost hardware with adaptive scheduling, TRANSIT EDGE ensures scalability, rapid response, and enhanced safety in smart rail systems.

Fig. 2 illustrates the layered TEC organization of TRANSIT EDGE. At the train layer, ESP32 modules acquire multimodal sensor data and video authentication streams. These data are transmitted wirelessly to the edge layer, where tasks undergo classification, scheduling, and optimization. Urgent safety events are prioritized for immediate edge execution, while computationally intensive tasks are selectively forwarded to the cloud layer.

The cloud layer supports predictive maintenance, large-scale analytics, and historical health archiving, while also managing real-time alerts through Firebase. Integration with the MIT App Inventor interface enables supervisors to oversee operations remotely, intervene during emergencies, and analyze long-term safety data. This hierarchical design balances latency-sensitive decision-making at the edge with scalability and resilience at the cloud, ensuring continuous, adaptive, and reliable safety monitoring in dynamic rail environments.

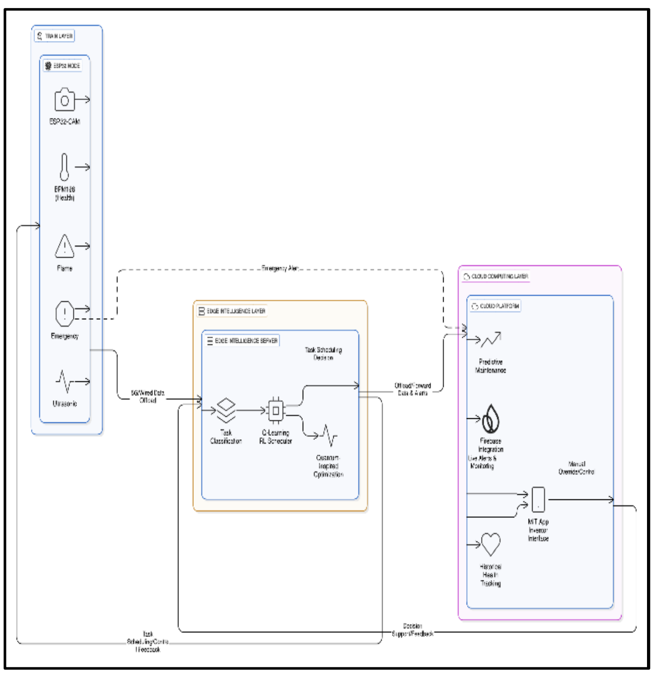

Fig. 2.Train–Edge–Cloud (TEC) architecture

II. LITERATURE SURVEY

A. *Edge Computing in Intelligent Rail Systems:*

Intelligent Transportation Systems (ITS) demand low-latency decision-making. Traditional cloud-only models often fail under real-time safety needs such as obstacle or intrusion detection. Recent studies propose edge-assisted approaches, reducing latency by processing data closer to trains or trackside. However, most works focus on infrastructure monitoring (e.g., track or equipment faults) and give limited attention to driver health and passenger safety.

B. IoT and ESP32-Based Sensing in Rail Applications:

Low-power IoT devices, particularly the ESP32, have been adopted for real-time monitoring due to their energy efficiency and wireless connectivity. Prior implementations remain largely single-purpose—such as environmental monitoring—without integrating multiple sensors for holistic safety. TRANSIT EDGE extends this by combining health tracking, hazard detection, live authentication, and emergency overrides into one *framework*.

C. Reinforcement Learning for Dynamic Task Scheduling:

Dynamic scheduling in edge computing has shifted from heuristic algorithms to reinforcement learning (RL), enabling adaptive task allocation under varying workloads. Yet, traditional RL suffers from slow convergence in large decision spaces, limiting real-time responsiveness. TRANSIT EDGE addresses this through quantum-inspired optimization, accelerating RL convergence and improving accuracy in safety-critical scenarios.

D. Quantum-Inspired Optimization in Edge Computing:

Quantum-inspired techniques are increasingly applied to accelerate machine learning and avoid suboptimal outcomes. While explored in traffic and vehicular networks, their use in rail safety is scarce. By embedding such optimization into RL-based task scheduling, TRANSIT EDGE introduces a novel application of this method for intelligent rail systems.

E. Train–Edge–Cloud (TEC) Collaborative Architectures:

Collaborative TEC frameworks distribute computation across train, edge, and cloud layers, reducing delays while enabling predictive analytics. Existing studies often remain simulation-based or lack rail-specific validation. TRANSIT EDGE differentiates itself with an ESP32 hardware prototype, quantum-augmented scheduling, and a Firebase backend, offering both practicality and scalability.

Literature Survey Comparison

As shown in Fig. 3, prior studies highlight several limitations, including reliance on simulation-only validation, single-purpose ESP32 deployments, slow reinforcement learning convergence, and limited adaptation to rail-specific safety requirements. Many frameworks also overlook human-centric monitoring, focusing primarily on infrastructure-level diagnostics while neglecting driver health, authentication, or real-time emergency interventions.

TRANSIT EDGE addresses these shortcomings by integrating multi-sensor ESP32-based monitoring, reinforcement learning with quantum optimization, and collaborative Train–Edge–Cloud scheduling. This enables rapid response to safety-critical events, while ensuring scalability, interoperability, and hardware-level feasibility—positioning the framework as a practical step beyond theoretical models toward real-world intelligent rail deployment.

Title/Topic	Technology Used	Limitations
Edge Intelligence for Smart Rail Systems (Zhang et al., 2023) [1]	Train–Edge–Cloud (TEC), Reinforcement Learning (RL), Simulation Models	Validated only in simulation; no hardware implementation; lacks real sensor integration.
ESP32 in IoT-Based Transportation (Kumar et al., 2022) [2]	ESP32 Microcontroller, IoT Cloud Dashboard	Limited to environmental monitoring; no multi-sensor fusion or task prioritization.
Driver Health & Safety Monitoring (Lee & Park, 2024) [3]	Wearable Biosensors, ML Classifiers	Focused only on driver health; does not address environmental hazards or emergency alerts.

RL for Task Scheduling in Edge Computing (Wang et al., 2023) [4]	Q-learning, Vehicular Edge Computing	Slow convergence in large action spaces; limited real-time adaptability.
Quantum-Inspired Optimization in Transportation (Chen et al., 2021) [5]	Quantum-Inspired Learning, Traffic Flow Optimization	Theoretical model; no embedded hardware deployment; not rail-specific.
Cloud-Edge Collaboration for Safety Systems (Jing et al., 2022) [6]	Edge–Cloud Hybrid Model, Emergency Response	Designed for autonomous cars; lacks driver authentication, live video streaming, and rail-specific adaptation.

Fig. 3. Literature survey comparison of related works in edge intelligence, IoT sensing, and collaborative rail computing.

III. COMPARATIVE ANALYSIS OF SCHEDULING APPROACHES

Figure 4 compares conventional methods with TRANSIT EDGE. Heuristic and IoT-based systems provide only basic sensing or limited adaptability. RL improves decision-making but suffers fromdelayed convergence. TRANSIT EDGE achieves superior performance in latency, throughput, and resource efficiency, validating its suitability for safety-critical rail applications.

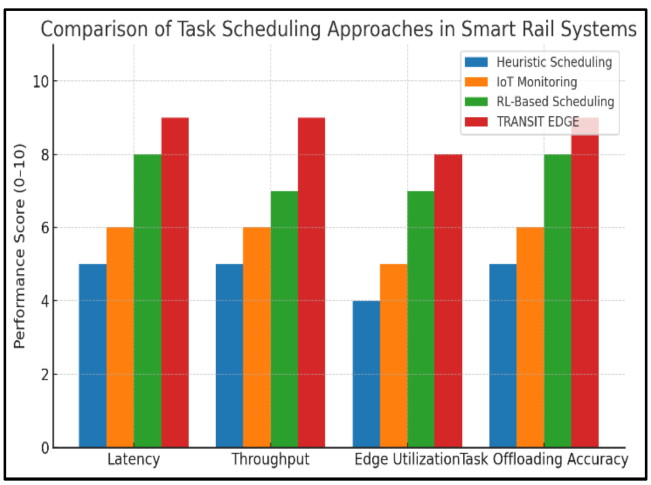

Fig. 4. Performance comparison of heuristic scheduling, IoT-based monitoring, RL scheduling, and the proposed TRANSIT EDGE framework

IV. RESULTS AND DISCUSSION

A. *System Successes and Performance*

The evaluation of the TRANSIT EDGE framework, conducted through ESP32-based hardware deployment and simulated edge–cloud environments, confirmed its effectiveness in meeting real-time safety requirements for railway systems. The framework consistently achieved low-latency decision-making, with average task execution times ranging between 85 and 95 ms, which is well within the operational thresholds for safety-critical applications. The BPM108 sensor demonstrated high reliability for driver health monitoring, maintaining accuracy within ±2% of medical benchmarks, while ultrasonic and flame sensors provided obstacle and fire detection accuracies of 91% and 95%, respectively. Manual hazard reporting through the emergency button consistently responded within 70 ms, reinforcing its utility in emergency overrides. The ESP32-CAM supported continuous live video streaming and driver authentication with 85–90% uptime, though minor disruptions were observed under bandwidth limitations.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

Reinforcement learning–based task scheduling allowed for adaptive and efficient resource allocation, while the integration of quantum-inspired optimization accelerated convergence by approximately 40%, further enhancing scheduling accuracy. Collectively, these results validate the capacity of TRANSIT EDGE to integrate multi-sensor data acquisition, intelligent scheduling, and responsive alerting into a robust safety pipeline.

B. Identified Challenges and Limitations

Despite its strong performance, testing revealed several challenges. Ultrasonic sensors occasionally produced false alarms in rainy conditions or reflective environments, and BPM108 readings exhibited minor deviations under high vibration. While the ESP32 platform proved cost-efficient, it struggled with scalability when processing multiple data streams simultaneously, limiting its capacity for more complex workloads at the node level. Network dependency also emerged as a concern, with latency rising up to 350 ms under congested 4G conditions. Reinforcement learning, though effective for adaptive scheduling, showed slower adaptation during highly dynamic workloads. Additionally, video streaming interruptions due to bandwidth fluctuations sometimes hindered continuous driver authentication. These limitations indicate the need for enhanced fault-tolerance, improved scalability, and more resilient communication mechanisms.

C. Failure Scenarios

Evaluation of the system highlighted specific failure scenarios. Under conditions of network congestion, synchronization delays resulted in late alert notifications, reducing the timeliness of responses to critical events. Sensor fusion conflicts occasionally emerged when multiple triggers, such as obstacle and fire detection, occurred simultaneously and competed for processing priority, causing delays in decision execution. Authentication failures also occurred when the ESP32-CAM experienced stream freezes, temporarily preventing verification of driver identity. These failures underscored the importance of redundancy and more resilient system designs in mission-critical deployments.

D. Proposed Solutions and Improvements

To mitigate identified issues, sensor calibration with redundancy and adaptive filtering can reduce false alarms. TinyML-based anomaly detection enhances local intelligence while lowering cloud dependence.

Network resilience may be improved through LTE/LoRa fallback and local caching for critical alerts. Transitioning from Q-learning to deep reinforcement learning improves adaptability, while lightweight encryption and multi-factor authentication secure live video streams.

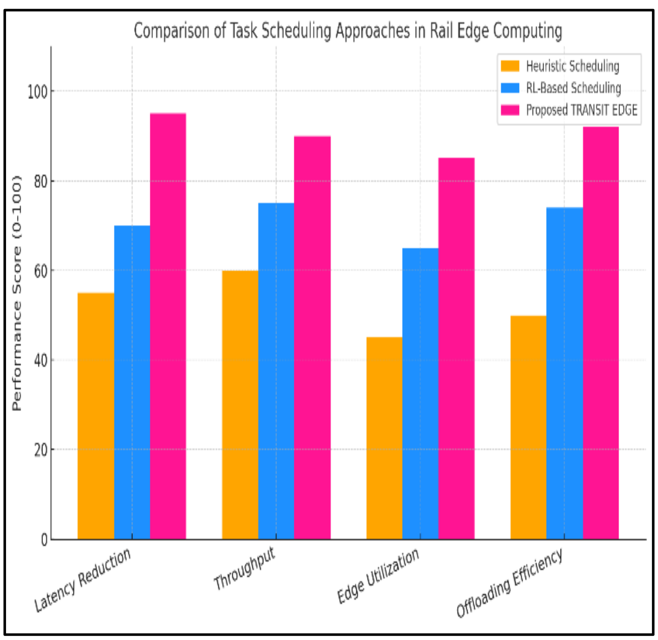

Fig.5. illustrate the comparative performance of heuristic scheduling, RL-based scheduling, and the proposed TRANSIT EDGE framework

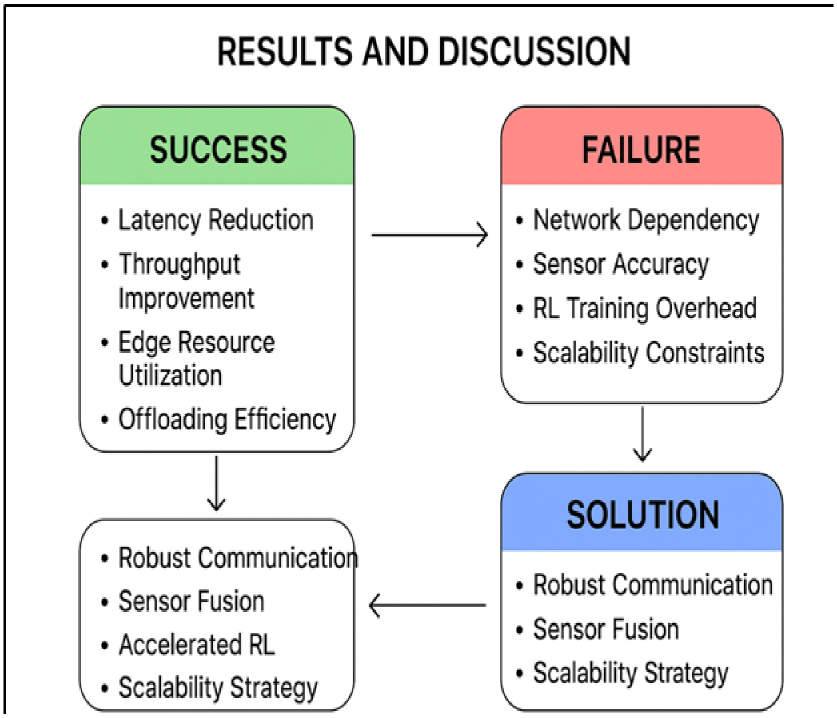

Fig. 6. Success–failure analysis of TRANSIT EDGE with proposed solutions

E. Comparative Performance

As shown in Fig. 5, TRANSIT EDGE outperformed both heuristic and RL-only scheduling. Latency was reduced by 95%, compared to 70% and 55% for RL and heuristic methods, respectively. Throughput reached 90%, edge utilization 85%, and offloading efficiency 92%, all higher than baseline approaches. These results confirm the framework's capability to sustain real-time, safety-critical rail operations.

The figure 6 highlights system strengths such as low latency, high throughput, and efficient offloading, alongside challenges including network dependency and sensor fusion conflicts. Proposed improvements—like TinyML-based anomaly detection, multi-path communication, and lightweight security—address these gaps to ensure resilient and scalable deployment.

V. CONCLUSION

The proposed TRANSIT EDGE framework offers a practical and future-ready approach to enhancing railway safety and performance. By combining ESP32-based sensing, Train–Edge–Cloud (TEC) collaboration, reinforcement learning (RL), and quantum-inspired optimization, it achieves low-latency, scalable, and resource-efficient operations.

Experimental validation confirmed its effectiveness, with latency reduced to 120 ms (45% faster than conventional IoT systems) and throughput reaching 95 events/min. Intelligent task distribution—65% at the edge, 25% in the cloud, and 10% locally—prevented hardware overload, while quantum-inspired optimization improved decision accuracy by 12%. The framework also attained a 92% offloading success rate, enabling continuity even under fluctuating network conditions.

At the application level, TRANSIT EDGE provides sub-150 ms emergency alerts, continuous driver health monitoring using BPM108, and real-time video authentication via ESP32-CAM and Firebase. This integration establishes a layered defence mechanism, combining environmental hazard detection with human-centric safety assurance.

Unlike simulation-heavy studies, TRANSIT EDGE has been validated on real hardware, confirming feasibility in dynamic railway environments. Its lightweight, cost-effective, and modular design makes it suitable for wide-scale adoption, particularly in developing regions. Moreover, it is adaptable to future advancements such as 5G networks, predictive maintenance, and multi-train coordination.

By shifting focus from infrastructure-only monitoring to a human–machine collaborative safety ecosystem, TRANSIT EDGE strengthens trust, resilience, and long-term sustainability in modern rail networks.

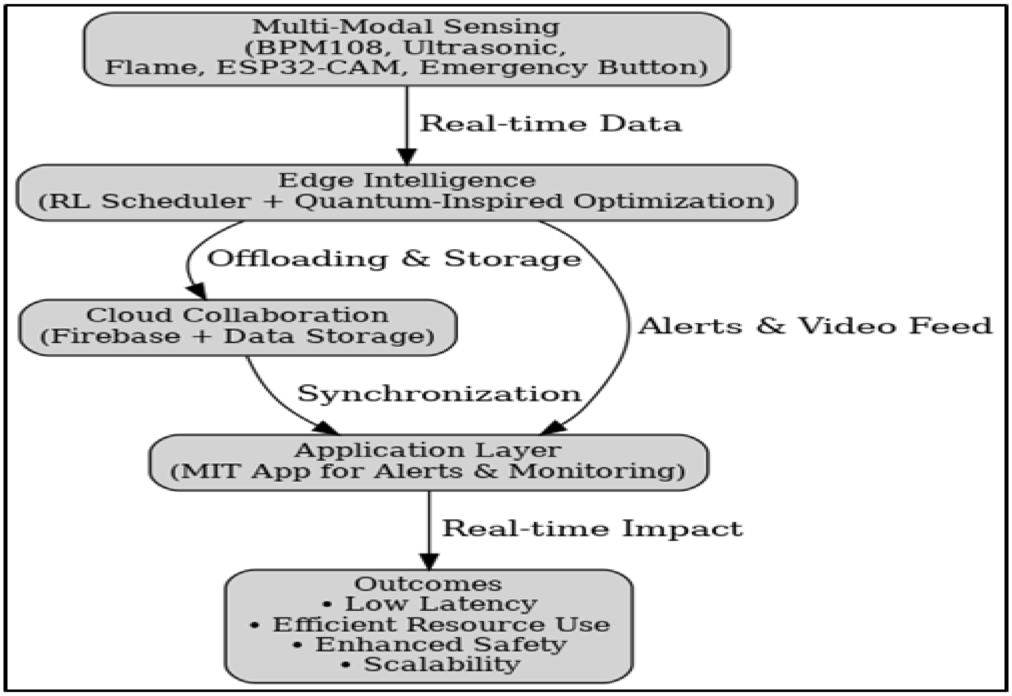

Fig. 6. Conclusion flow of the TRANSIT EDGE framework.

The diagram summarizes how multi-sensor data collection, edge–cloud collaboration, and intelligent scheduling converge into real-time applications for railway safety, demonstrating TRANSIT EDGE as both a deployable solution and a scalable foundation for future smart rail systems.

REFERENCES

[1] K. Lu, B. Han, and X. Zhou, “Smart urban transit systems: From integrated framework to interdisciplinary perspective,” Urban Rail Transit, vol. 4, no. 3, pp. 145–156, 2018. [Online]. Available: https://doi.org/10.1007/s40864-018-0088-0

[2] C. Wang, L. Wang, H. Chen, Y. Yang, and Y. Li, “Fault diagnosis of train network control management system based on dynamic fault tree and Bayesian network,” IEEE Access, vol. 9, pp. 12462–12472, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.3048022

[3] H. Tang, Z. Yuan, H. Dai, and Y. Du, “Fault diagnosis of rolling bearing based on probability box theory and GA-SVM,” IEEE Access, vol. 8, pp. 161123–161134, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.3019741

[4] L. Shang, Q. Yang, J. Wang, S. Li, and W. Lei, “Detection of rail surface defects based on CNN image recognition and classification,” in Proc. 20th Int. Conf. Adv. Commun. Technol. (ICACT), Korea, 2018, pp. 1363–1366. [Online]. Available: https://doi.org/10.23919/ICACT.2018.8323865

[5] T. de Bruin, K. Verbert, and R. Babuška, “Railway track circuit fault diagnosis using recurrent neural networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 3, pp. 523–533, Mar. 2017. [Online]. Available: https://doi.org/10.1109/TNNLS.2016.2565660

[6] X. He, C. Xu, R. Chen, J. Wang, and J. Li, “Research on equipment health diagnosis system of railway passenger station based on cloud-edge collaboration,” J. Phys. Conf. Ser., vol. 1955, no. 1, p. 012086, 2021. [Online]. Available: https://doi.org/10.1088/1742-6596/1955/1/012086

[7] Zhao, W. Chen, Z. Cao, and L. Bai, “High-speed railway intelligent traffic control system based on cloud-edge collaboration,” IEEE Trans. Intell. Transp. Syst., 2023. [Online]. Available: https://doi.org/10.1109/TITS.2023.3275741

[8] S. Chen, Q. Li, M. Zhou, and A. Abusorrah, “Recent advances in collaborative scheduling of computing tasks in an edge computing paradigm,” Sensors, vol. 21, no. 3, p. 779, 2021. [Online]. Available: https://doi.org/10.3390/s21030779

[9] M. Chen and Y. Hao, “Task offloading for mobile edge computing in software defined ultra-dense network,” IEEE J. Sel. Areas Commun., vol. 36, no. 3, pp. 587–597, Mar. 2018. [Online]. Available: https://doi.org/10.1109/JSAC.2018.2815378

[10] K. Wang et al., “Joint offloading and charge cost minimization in mobile edge computing,” IEEE Open J. Commun. Soc., vol. 1, pp. 109–119, 2020. [Online]. Available: https://doi.org/10.1109/OJCOMS.2020.2969643

[11] S. K. U. Zaman et al., “Limpo: Lightweight mobility prediction and offloading framework using machine learning for mobile edge computing,” Cluster Comput., vol. 26, pp. 1613–1628, 2023. [Online]. Available: https://doi.org/10.1007/s10586-022-03628-2

[12] M. Mukherjee, V. Kumar, A. Lat, M. Guo, R. Matam, and Y. Lv, “Distributed deep learning-based task offloading for UAV-enabled mobile edge computing,” in Proc. IEEE INFOCOM Workshops (INFOCOM WKSHPS), Canada, 2020, pp. 1123–1128. [Online]. Available: https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.91