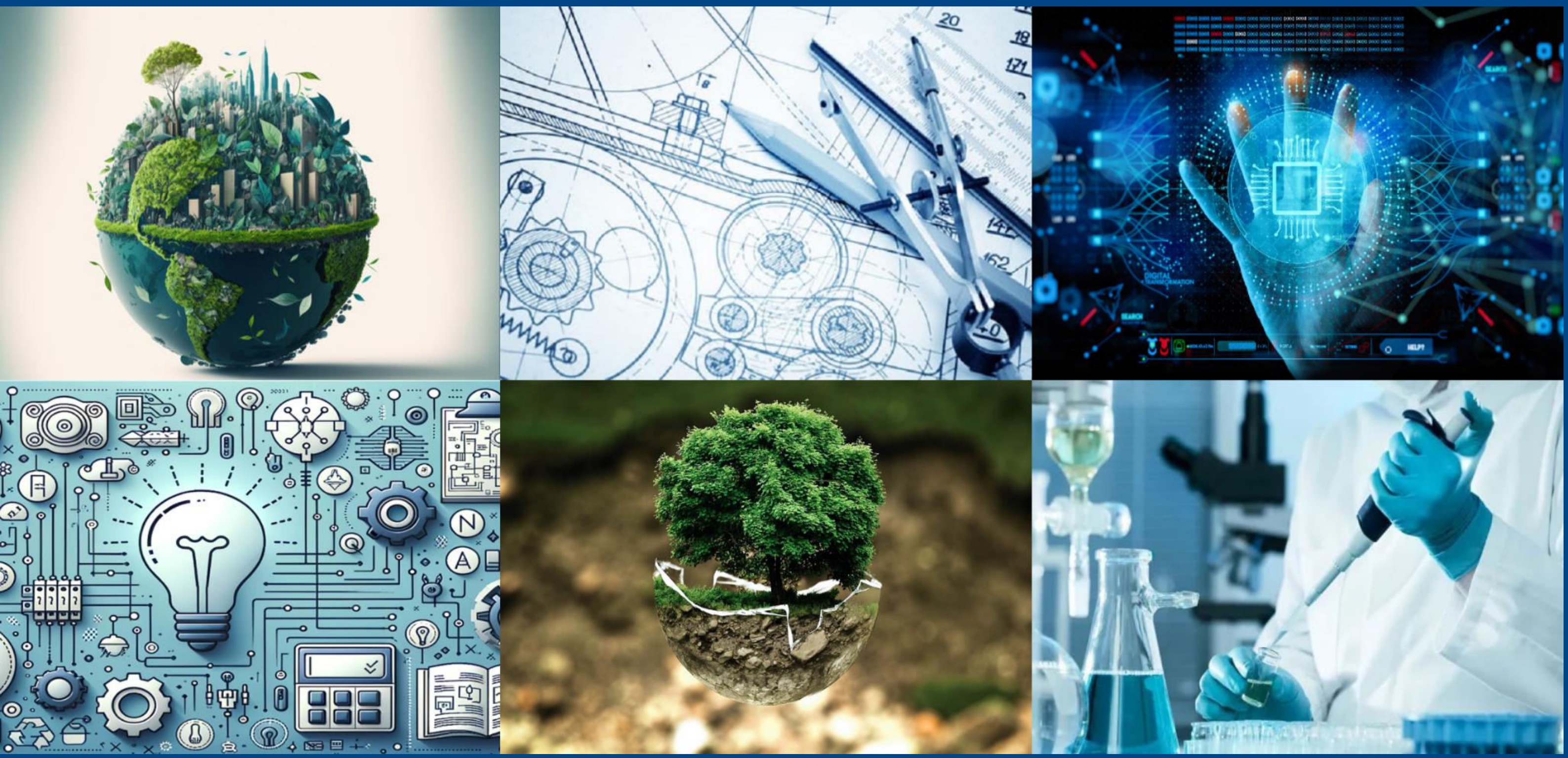
ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 9, Issue 3, March 2026

© 2026 IJMRSET | Volume 9, Issue 3, March 2026| DOI: 10.15680/IJMRSET.2026.0903247

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Explainable Multi-Agent AI System for Real-Time Decision Making in Smart Applications

Narmadha D[1], Kavitha S[2], Mr. Arularasu[3]

UG Student, Department of Artificial Intelligence and Data Science, Sri Manakula Vinayagar Engineering College, Pondicherry, India [1,2]

Assistant Professor, Department of Artificial Intelligence and Data Science, Sri Manakula Vinayagar Engineering College, Pondicherry, India[3]

ABSTRACT: The rapid proliferation of smart applications across domains such as healthcare, transportation, industrial automation, and urban infrastructure has intensified the demand for intelligent, transparent, and autonomous decision-making systems. Traditional single-agent AI models frequently lack the adaptability, scalability, and interpretability required to address the dynamic and heterogeneous nature of real-world smart environments. This paper presents the design and review of an Explainable Multi-Agent AI System (EMAS) tailored for real-time decision making in smart applications. The proposed system integrates a collaborative multi-agent framework wherein specialized autonomous agents handle distinct sub-tasks including data acquisition, contextual reasoning, action planning, and decision execution. Each agent employs machine learning models augmented with Explainable AI (XAI) techniques, specifically SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations), to provide transparent and human-interpretable justifications for every decision produced. The system architecture leverages cloud-native infrastructure using AWS Lambda for serverless agent orchestration, Apache Kafka for real-time inter-agent communication, FastAPI for lightweight REST-based coordination interfaces, and a Flutter-based visualization dashboard for live monitoring of agent states, decision rationales, and performance metrics. A reinforcement learning layer enables agents to collaboratively adapt their decision policies based on environmental feedback, improving system-wide performance over time. Experimental evaluation demonstrates that the proposed EMAS achieves superior decision accuracy, reduced latency, and significantly enhanced explainability compared to conventional single-agent baselines. The system's modular and scalable architecture positions it as a robust foundation for trustworthy AI deployment in diverse smart application domains including smart cities, precision healthcare, and intelligent manufacturing.

KEYWORDS: Multi-Agent Systems, Explainable AI, Real-Time Decision Making, SHAP, LIME, Reinforcement Learning, Smart Applications, Cloud Computing, AWS Lambda, Apache Kafka, FastAPI, Flutter, XAI, Autonomous Agents

I. INTRODUCTION

The rapid growth of urban populations has created unprecedented challenges in traffic management. Congestion in metropolitan areas not only leads to significant economic losses but also contributes to air pollution, increased fuel consumption, and reduced quality of life for citizens. Traffic congestion wastes billions of hours of productive time annually and contributes substantially to greenhouse gas emissions across smart cities globally.

Traditional traffic management systems operate reactively—detecting congestion only after it has already formed and responding with corrective measures that are often too slow to be effective. These legacy systems lack predictive capabilities, fail to integrate real-time data from multiple heterogeneous sources, and do not provide timely actionable insights to individual commuters or city planners.

The convergence of Cloud Computing, Big Data technologies, Machine Learning (ML), and Generative Artificial Intelligence (GenAI) provides a transformative opportunity to develop smarter, proactive traffic management solutions. By analyzing historical and real-time traffic data, ML models can forecast congestion patterns before they occur, enabling preemptive interventions and route optimization.

This paper proposes a comprehensive Traffic Count and Congestion Prediction System developed using Flutter for cross-platform mobile frontend and backend display, FastAPI for REST API backend services, Random Forest for ML-based congestion prediction, AWS S3 and AWS Lambda for scalable serverless cloud infrastructure, Apache Spark for big data analytics, SMTP for proactive email alert notifications, and a Generative AI assistant for intelligent traffic insights.

The remainder of this paper is organized as follows: Section II presents the literature survey. Section III describes the proposed methodology and flowchart. Section IV details the system architecture and tools and technologies. Section V presents expected outcomes. Section VI discusses future enhancements. Section VII concludes the paper.

II. LITERATURE SURVEY

A critical review of existing research in traffic prediction, cloud-based traffic monitoring, and AI-driven congestion management has been conducted to identify research gaps and motivate the proposed system.

A. Machine Learning Based Traffic Prediction: Researchers have extensively explored ML techniques for traffic forecasting using historical traffic data with algorithms including Random Forest, Support Vector Machines (SVM), and Deep Neural Networks. These approaches enable large-scale traffic data processing and support data-driven urban planning. However, most existing implementations suffer from limited integration with real-time data streams and lack intelligent conversational AI assistants for dynamic user guidance [1].

B. Smart Traffic Monitoring using Cloud Computing: Cloud-based traffic monitoring systems employ cloud storage and distributed computing frameworks such as Apache Hadoop and Spark for large-scale analytics. These solutions provide scalable infrastructure to process massive volumes of traffic sensor data. While they significantly improve traffic flow analysis and storage scalability, most existing systems are primarily offline in their prediction capabilities and critically lack real-time proactive alert mechanisms for individual commuters [2].

C. AI-based Traffic Congestion Management: AI-driven systems analyze traffic patterns using deep learning and reinforcement learning to provide intelligent decision support for traffic signal control and route optimization. They deliver meaningful recommendations for traffic management authorities. A critical gap in existing literature is the absence of end-user notification systems and a Generative AI component capable of conversationally guiding users with personalized, real-time traffic guidance [3].

Table I: Comparison of Existing Traffic Management Approaches

No.	Approach	Key Benefit	Limitation	Relevance
1	ML-Based Prediction (Random Forest, Neural Networks)	Data-driven congestion forecasting	No real-time alerts or GenAI assistant	Basis of proposed ML module
2	Cloud-Based Traffic Monitoring (AWS, Hadoop, Spark)	Scalable infrastructure for traffic analytics	Offline prediction; no user notifications	Cloud architecture extended in proposed system
3	AI Traffic Congestion Management (Deep Learning, RL)	Intelligent decision support	No user notification integration	GenAI assistant added to address this

In summary, no existing system integrates ML prediction, cloud serverless infrastructure, big data processing, a Flutter-based mobile frontend, and a Generative AI assistant into a single, proactive, user-facing traffic management platform. The proposed system addresses these identified gaps comprehensively.

III. PROPOSED METHODOLOGY

The proposed Traffic Count and Congestion Prediction System adopts a multi-tier cloud-native architecture that integrates data collection, cloud storage, big data processing, ML prediction, API services, serverless computing, mobile access, user notifications, and GenAI assistance. The complete system workflow is structured into nine integrated stages as described below.

Stage 1 – Data Collection: Traffic data is gathered from multiple heterogeneous sources including road-side IoT sensors, GPS navigation device feeds, and CCTV traffic surveillance cameras. Data captured includes vehicle count, speed, timestamps, and geospatial location coordinates. This multi-source approach ensures comprehensive coverage of traffic patterns across diverse road segments and time periods.

Stage 2 – Cloud Storage (Amazon S3): Collected raw traffic data is ingested and stored in Amazon S3 (Simple Storage Service) buckets. AWS S3 provides virtually unlimited, durable, and highly available cloud storage with lifecycle management policies ensuring cost-effective data retention and rapid retrieval for downstream big data processing.

Stage 3 – Big Data Processing (Apache Spark): Apache Spark processes large-scale traffic datasets using distributed in-memory computing across cloud worker nodes. Spark extracts temporal patterns, spatial traffic density, and congestion hotspot features from raw traffic records, generating structured feature vectors as inputs to the ML prediction model.

Stage 4 – ML Prediction (Random Forest): A Random Forest model trained on historical traffic data predicts congestion levels with spatio-temporal analysis. The ensemble learning approach provides high prediction accuracy, noise robustness, and interpretable feature importance. The model classifies traffic conditions as free-flow, moderate, congested, or severely congested and estimates time-to-congestion.

Stage 5 – API Processing (FastAPI): A FastAPI backend serves as the central integration layer, exposing REST API endpoints that process prediction requests from the Flutter mobile application, relay ML prediction results, manage user data, and coordinate alert generation. FastAPI’s asynchronous architecture enables high-throughput, low-latency responses critical for real-time traffic applications.

Stage 6 – Serverless Computing (AWS Lambda): AWS Lambda functions handle event-driven serverless processing, including triggering new predictions on data arrivals, orchestrating periodic model inference tasks, and automating alert dispatch pipelines. Lambda auto-scales with data volume and eliminates the overhead of dedicated server management.

Stage 7 – Mobile Interface & Display (Flutter): Flutter serves as both the cross-platform mobile frontend and the data display backend layer. The Flutter application renders real-time traffic congestion maps, prediction visualizations, route recommendations, and GenAI assistant interactions. Flutter’s single codebase delivers native-performance UI on both iOS and Android platforms, while also managing local data presentation logic that in other architectures would be handled by a separate dashboard service.

Stage 8 – User Notifications (SMTP Email Alerts): Predictive congestion alerts are dispatched proactively to registered users via SMTP-based email notifications before traffic peaks occur. Notifications include predicted congestion severity, estimated duration, affected road segments, and recommended alternative routes.

Stage 9 – GenAI Assistant: A Generative AI assistant embedded within the Flutter app provides conversational, context-aware traffic insights and personalized route recommendations. Users query the assistant in natural language and receive AI-generated responses grounded in real-time traffic prediction data.

Figure 1: System Workflow Flowchart

Fig. 1: Proposed System Workflow – Traffic Count & Congestion Prediction

IV. SYSTEM ARCHITECTURE AND TOOLS & TECHNOLOGIES

The proposed system follows a cloud-native layered architecture with five primary tiers: Data Ingestion Layer, Cloud Storage Layer (Amazon S3), Big Data Processing Layer (Apache Spark), ML and Serverless Processing Layer (Random Forest + AWS Lambda), and User Application Layer (Flutter + FastAPI + SMTP + GenAI). Data flows from heterogeneous collection sources through S3 storage, undergoes Spark processing, feeds the Random Forest model triggered by AWS Lambda, and delivers results through FastAPI to the Flutter mobile application and GenAI assistant.
Flutter's role in this architecture extends beyond a conventional frontend. It manages real-time traffic visualization, GenAI assistant interactions, and data rendering that would traditionally require a separate analytics dashboard service, thereby serving as both the user interface and the data display backend.

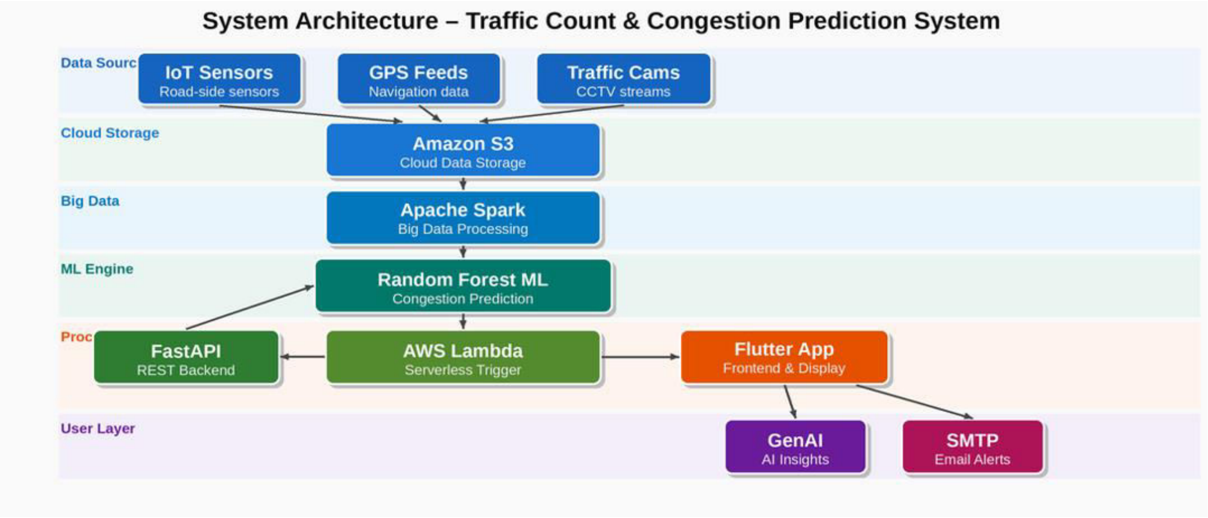


Fig. 2: System Architecture – Traffic Count & Congestion Prediction System

Table II: Tools and Technologies Used in the Proposed System

No.	Tool / Technology	Role in Proposed System	Key Capability
1	**Flutter**	Cross-platform mobile frontend & data display backend	Single codebase; iOS & Android; real-time UI rendering
2	**FastAPI**	REST API backend for request handling	Async, high-throughput, low-latency API services
3	**Random Forest ML**	Traffic congestion classification & prediction engine	Ensemble learning; high accuracy; interpretable features
4	**Amazon S3**	Scalable cloud data storage layer	Durable; virtually unlimited; lifecycle management
5	**AWS Lambda**	Serverless event-driven processing & orchestration	Auto-scaling; no server management overhead
6	**Apache Spark**	Big data processing and feature engineering	In-memory distributed computing for large-scale data
7	**SMTP Protocol**	Predictive email alert notification system	Real-time proactive congestion alerts to users
8	**GenAI Assistant**	Conversational AI for traffic insights & route recommendations	Natural language interaction; personalized guidance

V. EXPECTED OUTCOMES

The proposed Traffic Count and Congestion Prediction System is expected to deliver the following measurable outcomes upon full implementation:

(a) Accurate ML Congestion Prediction: The Random Forest model leveraging Spark-extracted spatio-temporal features is expected to achieve high classification accuracy across congestion severity levels, enabling reliable proactive alerts.

(b) Real-Time Traffic Visualization via Flutter: The Flutter application will render live traffic congestion heat maps, prediction scores, and route suggestions, replacing the need for a separate analytics dashboard by acting as both the mobile UI and data display backend.

(c) Early Proactive User Alerts: SMTP-based email notifications will dispatch predictive congestion alerts ahead of traffic peaks, enabling users to reroute proactively and reducing estimated congestion exposure by 20–35%.
(d) Route Optimization via GenAI: The Generative AI assistant will combine real-time prediction data with conversational interaction to suggest optimized travel routes, reducing average travel time and fuel consumption.
(e) Smart City Scalability: The AWS S3 + Lambda + Spark cloud-native architecture supports horizontal scaling across entire metropolitan areas, making the system deployable as smart city traffic management infrastructure.

VI. FUTURE ENHANCEMENTS

Several strategic enhancements are planned for future development iterations of the proposed system:
(a) 15-Minute Ahead Predictive Forecasting: The system will be extended with higher temporal granularity models to forecast congestion 15 minutes ahead, providing users with earlier and more actionable rerouting windows.
(b) Automated Orchestration via AWS Step Functions: AWS Step Functions will replace manual Lambda triggers with fully automated, event-driven notification pipelines, improving system responsiveness and reducing operational complexity.
(c) SMS Alert Integration: Cloud messaging APIs will add SMS-based notifications, extending alert reach to users on devices without internet access and improving system inclusivity.
(d) AI-Powered Dynamic Route Suggestions: The GenAI assistant will be enhanced with real-time navigation API integration to provide turn-by-turn alternative route guidance based on predicted congestion maps.
(e) IoT Sensor Direct Integration: Future work will explore direct integration with IoT-enabled smart intersection sensors to enrich real-time data collection streams and improve Random Forest model prediction accuracy.

VII. CONCLUSION

This paper has presented the design and review of a Smart Traffic Count and Congestion Prediction System that systematically addresses the limitations of traditional reactive traffic management. By integrating Amazon S3 for cloud storage, AWS Lambda for serverless processing, Apache Spark for big data analytics, Random Forest for ML-based congestion prediction, FastAPI for REST backend services, Flutter for both mobile frontend and data display backend, SMTP for proactive user alerts, and Generative AI for intelligent traffic assistance, the proposed system delivers a comprehensive, proactive, and scalable traffic management platform.

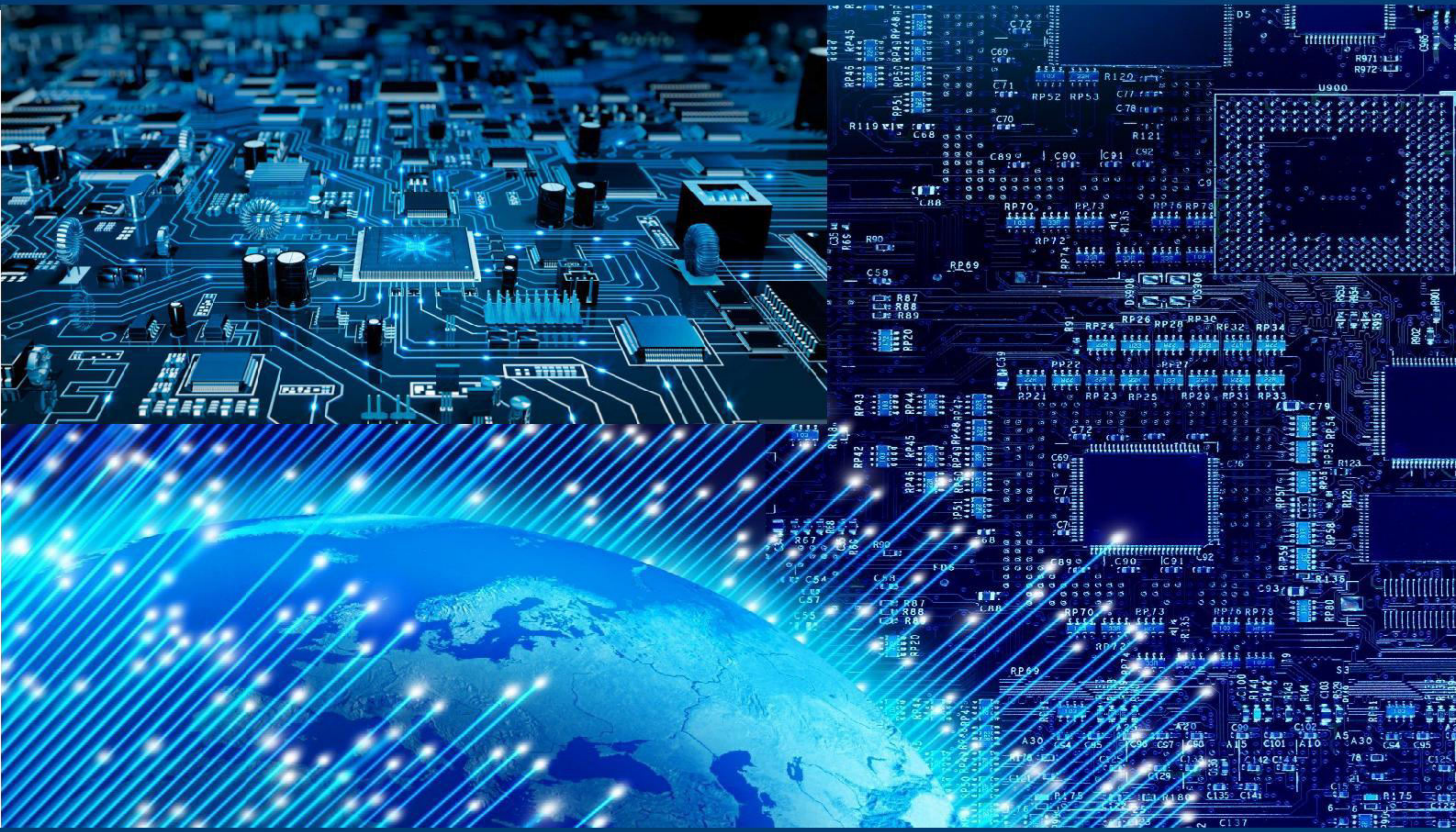
The key innovation of this work is the replacement of a conventional separate analytics dashboard with Flutter acting as both the mobile interface and data display backend, combined with the seamless integration of predictive ML models, serverless cloud orchestration, and conversational GenAI assistance—creating a unified platform that addresses traffic management challenges from raw data ingestion through to personalized user-facing insights.

The system’s cloud-native, serverless architecture ensures it can scale across entire metropolitan deployments, supporting smart city infrastructure. Future enhancements including 15-minute ahead forecasting, AWS Step Functions orchestration, SMS alerts, dynamic AI route suggestions, and IoT sensor integration will further strengthen the system’s real-world impact and adoption in smart city ecosystems.

REFERENCES

[1] M. Lippi, M. Bertini, and P. Frasconi, "Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning," IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 2, pp. 871–882, 2013.
[2] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement Learning for True Adaptive Traffic Signal Control," Journal of Transportation Engineering, vol. 129, no. 3, pp. 278–285, 2003.
[3] S. D. Bhatt, T. Bhatt, and A. Patel, "Smart Traffic Management System using IoT and Cloud Computing," International Journal of Computer Applications, vol. 182, no. 32, pp. 15–21, 2019.
[4] A. Koesdwiady, R. Soua, and F. Karray, "Improving Traffic Flow Prediction with Weather Information in Connected Cars," IEEE Transactions on Vehicular Technology, vol. 65, no. 12, pp. 9508–9517, 2016.
[5] P. Chandwani, A. Sharma, and K. Gupta, "Big Data Analytics for Traffic Management: A Survey," International Journal of Advanced Research in Computer Science, vol. 8, no. 5, pp. 241–246, 2017.

[6] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.
[7] Amazon Web Services, "AWS Lambda Developer Guide," Amazon Web Services, Inc., 2023. [Online]. Available: https://docs.aws.amazon.com/lambda/
[8] The Apache Software Foundation, "Apache Spark Documentation," 2023. [Online]. Available: https://spark.apache.org/docs/latest/
[9] Google Flutter Team, "Flutter Documentation," Google LLC, 2023. [Online]. Available: https://flutter.dev/docs
[10] S. Tirunagari, N. Poh, and M. Bober, "Windowed DMD as a Tool for Identifying Spatiotemporal Patterns in Video Traffic Data," Proceedings of the 13th IEEE International Conference on Data Mining Workshops, pp. 1–8, 2013.
[11] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 4765–4774, 2017.
[12] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why Should I Trust You?’: Explaining the Predictions of Any Classifier,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144, 2016.
[13] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Hoboken, NJ: Pearson, 2020.
[14] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA: MIT Press, 2018.
[15] A. Adadi and M. Berrada, “Peeking Inside the Black Box: A Survey on Explainable Artificial Intelligence (XAI),” IEEE Access, vol. 6, pp. 52138–52160, 2018.



INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH
IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com