Computer Communications 250 (2026) 108433

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Agentic latency control and cooperative vehicle coordination in 5G-MEC: A prescriptive and explainable AI framework

Sheikh Muhammad Saqib [a], Tehseen Mazhar [b,j,*], Muhammad Usman Tariq [c], Tariq Shahzad [d], Asem Ibrahim AlAlwan [e], Habib Hamam [f,g,h,i]

[a] Department of Computing and Information Technology, Gomal University, Dera Ismail Khan, 29050, Pakistan
[b] School of Computer Science, National College of Business Administration and Economics, Lahore, 54000, Pakistan
[c] Department of Marketing, Operations, and Information Systems, College of Business, Abu Dhabi University, Abu Dhabi, United Arab Emirates
[d] Department of Computer Engineering, COMSATS University Islamabad, Sahiwal Campus, Sahiwal, 57000, Pakistan
[e] College of Engineering, Alfaisal University, Riyadh, 11533, Saudi Arabia
[f] Faculty of Engineering, Uni de Moncton, Moncton, NB, E1A3E9, Canada
[g] School of Electrical Engineering, University of Johannesburg, Johannesburg, 2006, South Africa
[h] International Institute of Technology and Management (IITG), Av. Grandes Ecoles, Libreville, BP 1989, Gabon
[i] Bridges for Academic Excellence - Spectrum, Tunis, Centre-ville, Tunisia
[j] Department of Computer Science and Information Technology, School Education Department, Government of Punjab, Layyah 31200, Pakistan

ARTICLE INFO

Keywords:
5G-MEC
SVM
XAI
LIME
Agentic AI

ABSTRACT

The problem of supporting ultra-low latency in 5G-MEC vehicular networks requires moving beyond passive forecasting toward real-time, proactive traffic management. Current solutions often rely on computationally expensive deep learning predictors or centralized network optimization, yet they frequently stop at prediction and do not complete the vehicle-level prescriptive action loop. This paper proposes an Agentic AI Coordination Framework that links latency-state prediction to immediate, explainable, and cooperative mitigation actions at the vehicle edge. At the core of the framework is a lightweight Support Vector Machine (SVM) trained on the 5G-V2N communication dataset, which enables highly reliable latency_category classification (Accuracy = 0.97; Weighted F1-score = 0.97; High-latency class Precision = 1.00 and Recall = 1.00). The predicted latency state then activates an in-vehicle Agentic AI module that generates dashboard-level prescriptive outputs, including status alerts, action plans, and optimized routing guidance. In addition, the framework leverages 5G-MEC cooperative broadcast to disseminate rerouting alerts to neighboring vehicles, supporting distributed congestion avoidance. To ensure transparency and controllability, the system integrates LIME-based explanations and actionable counterfactual analysis to indicate the minimal feature-level adjustments required to transition between High, Medium, and Low latency states. Overall, the framework forms a unified perception–decision–action pipeline that connects low-overhead inference, rule-based prescriptive control, and cooperative vehicle coordination. To align with sustainable computing and the Special Issue focus on energy-efficient routing in 5G-enabled VANETs, the framework couples low-complexity on-board inference with MEC-assisted broadcast to support real-time coordination while constraining communication and compute overhead.

1. Introduction

1.1. Context

The integration of 5G cellular communication and Multi-access Edge Computing (MEC) has provided a transformative opportunity to enhance connected vehicle safety applications, thereby complementing established technologies such as Dedicated Short Range Communication (DSRC) and Cellular Vehicle-to-Everything (C-V2X) direct communication [1]. Connected vehicle technologies, relying on embedded or tethered hardware for Wi-Fi and cellular connectivity [2], are paramount for creating safer, more connected transportation systems. These systems have a critical impact on reducing accidents, improving traffic

* Corresponding author. School of Computer Science, National College of Business Administration and Economics, Lahore, 54000, Pakistan.
E-mail addresses: saqibsheikh414@gmail.com (S.M. Saqib), tehseenmazhar719@gmail.com (T. Mazhar), Muhammad.kazi@adu.ac.ae (M.U. Tariq), aialalwan@alfaisal.edu (A.I. AlAlwan), Habib.Hamam@umoncton.ca (H. Hamam).

https://doi.org/10.1016/j.comcom.2026.108433
Received 14 November 2025; Received in revised form 17 January 2026; Accepted 23 January 2026
Available online 6 February 2026

List of Acronyms

MEC	(Multi-access Edge Computing)
SVM	(Support Vector Machine)
XAI	(Explainable Artificial Intelligence)
LIME	(Local Interpretable Model-agnostic Explanations)
DiCE	(Diverse Counterfactual Explanations)
VANET	(Vehicular Ad-hoc Network)
V2X	(Vehicle-to-Everything communication)
DSRC	(Dedicated Short-Range Communications)
C-V2X	(Cellular Vehicle-to-Everything)
ROC	(Receiver Operating Characteristic), AUC (Area Under the Curve), PR (Precision–Recall), AP (Average Precision)
SMOTE	(Synthetic Minority Over-sampling Technique), RBF (Radial Basis Function)

flow, and mitigating congestion by facilitating bidirectional data exchange within a range of approximately 1000 m between neighboring vehicles, smart devices, and roadside infrastructure (RSU) [3]. While incidents in various environments are often attributed to human error, technical faults, or environmental conditions [4], the stability and reliability of the underlying communication network remain a core challenge. Existing research has extensively utilized deep learning architectures, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Graph Neural Networks (GNN), to address traffic analysis. These efforts primarily focus on predictive tasks, including extracting spatiotemporal features, forecasting traffic anomalies, and predicting vehicle trajectories [5]. The GCN-DML framework, for instance, emphasizes high F1 scores by combining GCN for spatial learning and LSTM for temporal forecasting, with the explicit goal of optimizing prediction accuracy and reducing latency [6]. Similarly, models applied to Vehicular Ad-Hoc Networks (VANETs) prioritize prediction accuracy, adaptability, and data privacy through techniques like transfer learning and federated learning [7].

In line with the journal's emphasis on sustainable computing, we explicitly consider energy-aware operation alongside latency: our agentic, on-board predictor reduces round-trip communication and compute overhead, while MEC-assisted cooperative broadcast confines control signaling, supporting energy-efficient, real-time vehicle coordination [8,9].

However, a critical research gap persists in the transition from pure prediction to prescriptive, coordinated action. The current literature largely focuses on training high-accuracy models to predict latency or traffic flow, stopping short of integrating this predictive intelligence into an executable control loop for real-time risk mitigation. There is a clear need for a framework that not only accurately predicts network state but also utilizes this prediction to drive tangible, immediate, and cooperative actions for vehicle coordination and resource management.

Importantly, the proposed contribution is not a juxtaposition of independent components, but a single closed-loop perception–decision–action pipeline. The SVM predictor provides real-time latency-state awareness; the rule-based agentic layer converts this state into executable actions; the dashboard operationalizes driver-facing guidance; the MEC-assisted broadcast enables cooperative, network-wide mitigation; and the XAI/counterfactual modules provide transparent "why" and "what-to-change" justification for each prescribed action. Together, these elements function as one coherent method for vehicle-level prescriptive latency control rather than a collection of standalone techniques.

Moreover, recent edge-native, agentic pipelines demonstrate how lightweight perception fused with rule-based decision layers can close the perception-to-actuation loop under tight resource constraints—evidence that strengthens our choice of a lean, in-vehicle prescriptive module for sustainable, low-latency control [8]. Likewise, cooperative information-centric designs in 5G show how coordinated caching/routing can reduce access latency and network load, complementing our MEC-assisted broadcasts for energy-efficient, real-time VANET routing [9].

To address this gap, this study presents a comprehensive framework leveraging a trained, lightweight Support Vector Machine (SVM) model for highly accurate latency_category prediction, utilizing the recent "5G-Enabled Vehicle-to-Network Communication Data" dataset. Our contribution is multi-faceted: First, we integrate this predictive model into an Agentic AI module deployed on the vehicle, which utilizes continuous sensor data to predict the latency_category. Second, we propose a novel set of decision-making rules based on the interplay of all independent and predicted variables. This allows the Agentic AI to generate a full suite of prescriptive outputs, including Status Alerts, Action Plans, System Suggestions, and an Optimized Path, displayed on the driver dashboard. Third, this module is engineered to prepare and Broadcast alerts via 5G-MEC to nearby vehicles for immediate cooperative rerouting and distributed latency mitigation. Finally, the module incorporates a novel application of Counterfactual Analysis to provide actionable intelligence for risk mitigation (High to Low/Medium), stability assessment (Low to Medium/High), and transitional management (Medium to Low/High) by defining the precise modification required for sensor-read features.

Collectively, these choices align the contribution with the Special Issue mandate on energy-efficient routing for real-time communication in 5G-enabled VANETs, by tightly coupling low-overhead prediction, prescriptive control, and cooperative broadcast to jointly minimize delay and energy expenditure [8,9].

Accordingly, the investigation goal of this work is to bridge the prevailing gap between prediction-centric latency modeling and actionable vehicle coordination by designing an integrated agentic framework that (i) predicts latency states with low overhead, (ii) prescribes immediate mitigation actions through explicit decision rules, (iii) supports cooperative rerouting through MEC-assisted broadcasts, and (iv) ensures transparency through explainability and actionable counterfactual guidance.

1.2. Key contributions

Building on the identified gap between standalone prediction and system-level optimization, this work contributes an integrated, vehicle-edge prescriptive framework that couples lightweight prediction with explainable, cooperative control in 5G-MEC VANETs. Specifically, we.

- Design and validate a compact machine-learning model for accurate latency_category prediction tailored for in-vehicle execution, minimizing computational overhead while preserving real-time responsiveness.
- Embed this predictor within an Agentic AI module that converts predicted latency states into explicit, rule-based, prescriptive actions suitable for immediate driver-dashboard delivery.
- Engineer a MEC-assisted cooperative broadcast mechanism that disseminates actionable alerts (e.g., rerouting and mitigation messages) to neighboring vehicles to achieve network-wide, low-latency coordination.
- Incorporate Actionable Counterfactual Analysis to prescribe the minimal, feature-level adjustments required to transition among High, Medium, and Low latency states, providing auditable "what-to-change" guidance.
- Demonstrate, via simulation, the end-to-end pipeline that transforms predictive intelligence into real-time, corrective, and cooperative actions at the vehicle edge, thereby operationalizing sustainable, low-latency control in 5G-MEC environments.

2. Literature review

2.1. Current research

The current studies on vehicular communications have been mostly based on advanced predictive modeling or system level optimization, and it appears that there is a huge gap in one regard linking these two aspects to reach real time prescriptive vehicle coordination. A number of studies demonstrate such interests in prediction. As one example, a study of Dynamic Graph Convolutional Networks with Temporal Representation Learning (DGCN-TRL) focuses on the ability to learn the complex dynamic temporal relationships in traffic to make predictions [10]. On the same note, the application of Spatiotemporal Synchronous Graph architecture is specifically devoted towards prediction of traffic in a high-accurate manner through the modeling of dynamic correlation at both spatial and temporal levels. Although these models have been shown to have strong predictive capability, their complexity and centralized training needs pose a clear constraint to their practical application as a lightweight, in-vehicle, and low-latency agent [11]. Our model is solving this issue through using a lightweight Support Vector Machine (SVM) that is combined with explicit decision rules which significantly reduces the computation load which allows our model to provide immediate dashboard and broadcast output. Complementary evidence from edge/agentic AI in IoT shows that embedding lightweight perception with a rule-based agentic layer at the network edge can close the perception-to-actuation loop under tight latency and resource constraints, reinforcing our design choice for in-vehicle prescriptive control [8]. Recent sustainable-computing research in 5G VANETs, such as the MEDALS framework, similarly integrates AI-driven routing with explicit energy-efficiency objectives, confirming the importance of balancing low latency with green-network goals in next-generation vehicular systems [12].

In a similar continuation of the trend of prediction-centricity, the GCN-DML model, which is spread across edge devices, emphasizes high-accuracy traffic forecasting on systems such as NGSIM, although its emphasis is predominantly prediction-based, without a showing of prescriptive behavior inside the vehicle itself [6]. The LSTM-based Graph Attention hybrid model is engineered to encode vehicle interactions and predict trajectories, yet its high model complexity and reliance on centralized training pose limitations for edge evaluation [13]. Moreover, studies on Long-term Spatio-Temporal Graph Attention Networks (LSTGAN) focus on modeling long-term traffic patterns for forecasting, but they are not designed for immediate, per-vehicle actionable control in critical, short-horizon latency scenarios [14]. Even forecasting research focused on improving safety, such as the work on Forecasting & Improving Latency in 5G V2X, often remains confined to prediction with limited integration into a prescriptive or cooperative control loop. Our work directly converts this necessary prediction into rule-based outputs, such as alerts and optimized reroutes, for immediate cooperative mitigation via 5G-MEC [15]. The effort to improve Urban Traffic Forecasting by integrating travel time and uncertainty estimates also excels in prediction quality but similarly fails to deliver concrete, vehicle-level prescriptive actions [16]. By contrast, the proposed agentic framework operationalizes prediction at the edge and merges it with prescriptive control logic, advancing toward the sustainable, real-time coordination envisioned in recent green-AI vehicular studies [12].

The second major research stream focuses on optimization and infrastructure planning. Works like Rethinking the mobile edge for vehicular services concentrate on developing realistic MEC placement and system configurations to optimize resource use and reduce latency at a system level, but they do not demonstrate a closed-loop, vehicle-level decision-making process [17]. Related advances in 5G information-centric networking demonstrate cooperative caching and content routing that improve cache-hit ratios and reduce access latency at scale, yet they still stop short of providing an on-board, prescriptive control layer for per-vehicle coordination [9]. Similarly, DEEPCO-RIS optimization for energy-efficient vehicular networks introduces intelligent physical-layer control (RIS configuration and hybrid NOMA/OMA access) to sustain ultra-reliable low-latency links [18]; however, it remains focused on signal-level efficiency rather than in-vehicle prescriptive decisioning. At the architectural level, the energy-efficient cloud–edge collaborative framework based on digital-twin feedback demonstrates that coupling edge forecasting with cloud-scale coordination can halve latency in distributed networks [19]; our design extends this principle by embedding the forecasting and decision modules directly within vehicles and coordinating them through 5G-MEC for real-time, cooperative latency control. In parallel, the VANET Edge Computing Model focuses on offloading algorithms to minimize delay in smart-city applications, yet it often overlooks the integration of Counterfactual Explainable AI (XAI) and cooperative broadcast mechanisms crucial for large-scale coordination [20]. Similarly, research on MEC-enabled resource allocation in Internet of Vehicles explores joint task sequencing and resource balancing but lacks the integrated vehicle-level predictive decisioning and cooperative broadcast logic found in our work [21]. Our framework embeds a lightweight prediction and agentic decision module directly onto vehicles for real-time action [11], and, crucially, integrates Counterfactual XAI to provide a clear, auditable "why" and "how to change" path for every predicted latency state (High to Low/Medium, Low to High/Medium, Medium to Low/High). This comprehensive approach, including XAI, cooperative broadcast, and embedded decision rules, demonstrates superior functionality by tightly coupling prediction with actionable vehicle coordination and alerts [22,23].

2.2. Research gap

In summary, prior research in 5G-enabled vehicular networks has predominantly concentrated on either predictive modeling—focusing on traffic forecasting, latency estimation, or mobility prediction—or on system-level optimization aimed at improving MEC placement, resource allocation, or network throughput. However, few studies integrate these two domains into a unified, actionable framework. Existing approaches often stop at accurate prediction or efficient infrastructure configuration without transforming such intelligence into vehicle-level, prescriptive decision-making.

The present work addresses this critical gap by introducing an Agentic AI framework that operates directly at the vehicle edge. It not only predicts latency states but also translates these predictions into immediate, cooperative control actions through explainable and rule-based logic. This agentic integration bridges the divide between forecasting and action, enabling real-time, energy-efficient coordination among vehicles in 5G-MEC environments.

3. Proposed methodology

An intelligent latency-aware framework has been developed to enable real-time vehicle coordination using 5G-MEC (Multi-Access Edge Computing). The framework employs an AI model to predict network latency and a decision-making system to take appropriate actions. This methodology is broken down into several key steps, starting with dataset preparation and culminating in an agentic AI dashboard for real-time vehicle management as shown in Fig. 1.

3.1. Dataset preparation and feature engineering

The basis of the system is a dataset [16] that has been made up of 1000 records that are made up of 11 features. Among these, 10 of them are independent features that are gathered by different sensors and system logs and the other is the dependent feature, which is Latency_Category. Speed kms, Processing time ms, Energy consumed J, Latitude and Longitude are the numerical independent features. The categorical characteristics, as defined in Table 1, have been coded to be used in the

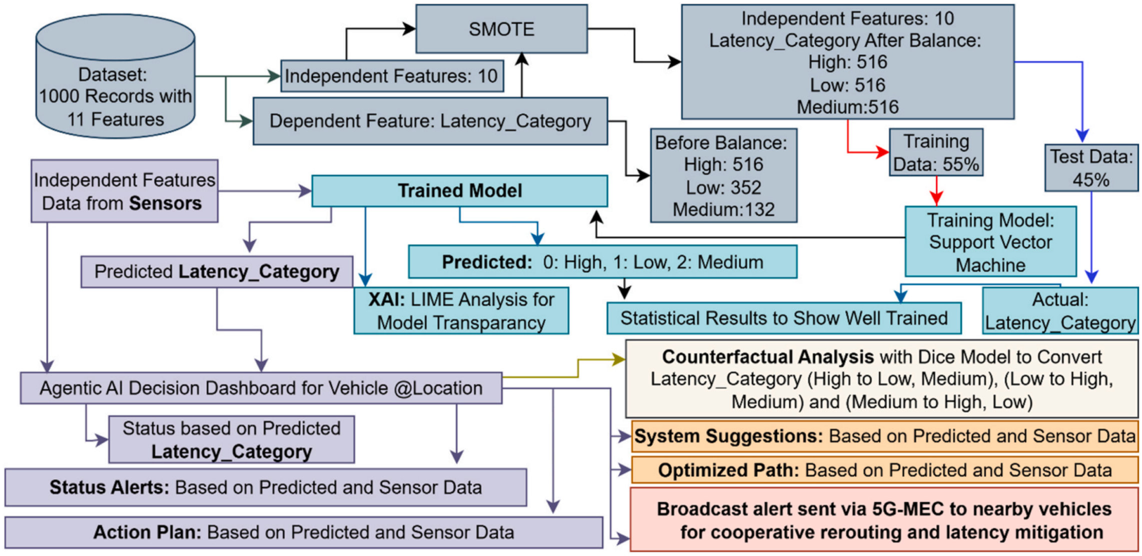

Fig. 1. An Intelligent Latency-Aware Framework for Real-Time Vehicle Coordination using 5G-MEC.

Table 1
Description of categorical features and encoded mappings.

Feature Name	Category Code	Category Label	Description
Direction	0	E	Vehicle heading East
	1	N	Vehicle heading North
	2	NE	Vehicle heading North-East
	3	NW	Vehicle heading North-West
	4	S	Vehicle heading South
	5	SE	Vehicle heading South-East
	6	SW	Vehicle heading South-West
	7	W	Vehicle heading West
Event_Type	0	Accident	Road accident detected
	1	Normal	No event or anomaly
	2	Traffic	Congestion or slowdown detected
	3	Weather	Environmental/weather impact
Urgency_Level	0	High	Requires immediate system action
	1	Low	Non-critical event
	2	Medium	Moderate urgency requiring monitoring
MEC_Server_ID	0	MEC1	Primary edge node (closest vehicle link)
	1	MEC2	Secondary MEC node
	2	MEC3	Mid-load edge node
	3	MEC4	Remote node with moderate delay
	4	MEC5	Backup or overflow MEC unit
Processed_By	0	Cloud	Centralized computation (higher delay)
	1	Fog	Edge computation (low latency)
Latency_Category	0	High	Critical latency — triggers rerouting & alert broadcast
	1	Low	Stable latency — maintain current path
	2	Medium	Moderate latency — hybrid MEC + cloud balancing

machine learning model. These are Direction, Event type, level of urgency, MEC server ID and Processed by. Latency category is also a categorical target variable that has three different classes: 0: High, 1: Low, and 2: Medium.

First of all, the dataset had a considerable imbalance in the distribution of Latency_Category. As illustrated in the bar chart below (Fig. 2) the High latency class (code 0) has 516 records with only 352 records in the Low (code 1) and Medium (code 2) classes. This imbalance may bias

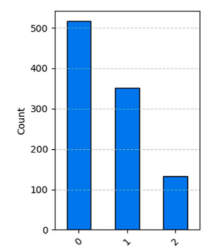

Fig. 2. Class distribution before balancing of Latency_Category.

an artificial intelligence model making it perform poorly on minor classes.

To solve this, the use of the BorderlineSMOTE (Synthetic Minority Over-sampling Technique) algorithm was used. The SMOTE algorithm is based on the principle of generating artificial instances of the minority classes using the same neighbors. The dataset was balanced by over-sampling the Low and Medium latency classes such that the number of records in each of the three classes is 516. The equal distribution, as shown in Fig. 3 debates that the model is fairly trained across all categories of latency.

Upon balancing, the data was divided into training and test set to determine the performance of the model on unknown data. A training to testing ratio of 55 to 45 was applied. Although a training set of 55 percent is generally good enough, the 45 percent test set is not negligible as it relates a solid quantity of data to be certain of having a sound and trustful assessment of the generalization capacity of the model. When a bigger test set is used, the variance of the performance estimation will be less, and the confidence that the measures it reports are truly

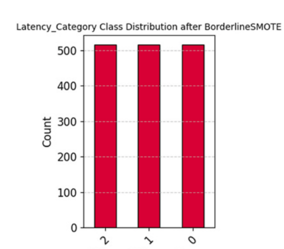

Fig. 3. Class distribution after balancing of Latency_Category.

representative of the real-world behavior of the model increases.

A tiny sample of 10 records of the resulting training and test sets is presented in Tables 2 and 3, respectively, exhibiting the model training and evaluation data format.

To prepare the dataset for machine learning, several preprocessing steps were applied. Categorical features (Direction, Event_Type, Urgency_Level, MEC_Server_ID, and Processed_By) were label-encoded to convert them into numerical representations suitable for the SVM model. Numerical features (Speed_kmh, Processing_Time_ms, Energy_Consumed_J, Latitude, and Longitude) were used in their raw form without normalization or standardization, as the model achieved high predictive accuracy (97%) without these steps. Moreover, using the actual numerical values is important for the downstream counterfactual analysis, as it allows the Agentic AI module to generate meaningful and actionable recommendations—e.g., how much speed to adjust, processing time thresholds, or energy consumption changes—to transition between latency categories. Therefore, standard scaling was not necessary and would in fact reduce the interpretability of the results in a real-world prescriptive setting. Class imbalance in the target variable Latency_Category was addressed using Borderline-SMOTE, generating synthetic samples for the minority classes to achieve equal representation across all three latency categories. These preprocessing steps ensured that the dataset was well-prepared for training and evaluation while maintaining the integrity, interpretability, and actionable utility of the features for the proposed Agentic AI framework.

3.2. Support Vector Machine model for training

Theclassification task was performed using a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel (implemented via sklearn.svm.SVC(probability = True, random_state = 42)). This choice is problem-driven, as the 5G-V2N dataset exhibits complex, non-linear relationships between sensor-derived features and latency categories, including environmental, vehicular, and network-state variables. The RBF kernel enables the SVM to capture these non-linear interactions efficiently, creating flexible decision boundaries that can accurately separate High, Medium, and Low latency states in high-dimensional feature space. Furthermore, SVMs are computationally lightweight and well-suited for on-board, low-latency inference, which aligns with the deployment requirements of the Agentic AI module. The probability = True parameter allows the model to output probability estimates for each class, supporting downstream prescriptive and counterfactual decision-making. Using random_state = 42 ensures reproducibility across experiments.

3.3. SVC model performance

Performance of the model is strictly measured with the help of a set of statistical measurements, such as Precision, Recall, Accuracy, and F1-Score. These classification measures give an overall perspective of the model in the detection of each latency category. Moreover, the regression indicators like MSE, R-squared and Mean Absolute Error (MAE) and Explained Variance Score (EVS) will also be used to review the numerical properties of the model prediction.

3.4. XAI-analysis

An XAI analysis is conducted in the form of LIME (Local Interpretable Model-agnostic Explanations) to gain the trust of users and make the model transparent. The idea of LIME is to build a simple interpretable model that estimates the behavior of the complex SVM model on a single prediction. This will enable system operators and users to know the reason why the model made a particular prediction on a particular test data point. LIME allows the identification of possible biases and makes the decisions of the model logical and consistent with the domain knowledge by giving local explanations.

Table 2
Sample of ten records from training set.

Speed_kmh	Direction	Event_Type	Urgency_Level	MEC_Server_ID	Processed_By	Processing_Time_ms	Energy_Consumed_J	Latitude	Longitude	Latency_Category
67.12	6	0	0	4	0	62.87	0.37	12.98	77.59408	0
103.7375	2	2	1	3	1	37.3975	0.286954	12.95561	77.61474	2
30.32408	1	1	1	3	1	33.59741	0.42463	12.98204	77.62381	2
20.39537	4	1	1	2	1	38.82616	0.617427	12.9827	77.61628	2
63.15	5	1	1	2	1	7.08	0.52	12.95739	77.59162	1
25.95	1	2	2	4	1	27.36	0.19	12.98279	77.61195	1
103.59	0	3	1	1	1	36.68	0.72	12.95538	77.58661	2
96.43	3	3	1	1	0	103.36	0.34	12.96835	77.61126	0
43.39	4	2	2	0	0	70.12	0.29	12.97334	77.59381	0
117.7904	2	1	1	2	1	27.40941	0.548734	12.97408	77.58924	1

Table 3
Sample of ten records from test set.

Speed_kmh	Direction	Event_Type	Urgency_Level	MEC_Server_ID	Processed_By	Processing_Time_ms	Energy_Consumed_J	Latitude	Longitude	Latency_Category
57.83	3	0	0	0	1	30.84	0.46	12.98781	77.60687	2
18.14	1	0	0	0	1	30.86	0.66	12.97364	77.58881	2
100.777	6	1	1	3	1	26.85482	0.244024	12.96569	77.58925	1
108.3761	0	3	1	0	1	37.40821	0.608953	12.95298	77.62498	2
39.75	7	0	0	4	1	27.11	0.65	12.99791	77.60294	1
108.48	6	2	2	2	1	31.37	0.11	12.96101	77.62282	2
25.46102	2	3	1	1	1	12.06209	0.260532	12.99592	77.62196	1
114.83	2	0	0	0	1	30.78	0.45	12.95718	77.59043	2
58.90271	0	1	1	3	1	36.57158	0.672477	12.95084	77.61935	2
53.36	7	1	1	0	1	34.45	0.41	12.95862	77.60715	2

3.5. Counterfactual analysis with DICE model for conversion of Latency_Category

The DiCE (Diverse Counterfactual Explanations) model of counterfactual analysis is an important element to give actionable information. It answers the question of what-if by creating other scenarios (counterfactuals) that would have altered the prediction of the model. For example.

- Low to High and Medium: DiCE can demonstrate which feature changes (e.g. an abrupt change in Speedkmh or ProcessingTimems) would cause the LatencyCategory to change to High or Medium. This assists in the pre-emption of risks and taking of precautionary measures.
- High to Low and Medium: In its turn, DiCE may also give advice on what steps (e.g., rerouting to a new MECServerID or minimizing ProcessingTimems) could turn a High latency scenario into a Low or a Medium one in order to be able to manage the risks.
- Medium to Low or High: The analysis would be beneficial in optimizing the allocation of the resources, by determining factors that would either reduce latency (Medium to Low) or prevent its aggravation (Medium to High).

3.6. Proposed agentic AI Decision Dashboard for Vehicles' coordination using 5G-MEC

The Agentic AI Decision Dashboard is designed as an autonomous, rule-driven in-vehicle decision module that integrates the trained SVM latency classifier with real-time sensor data to generate dynamic, prescriptive recommendations. The term agentic here emphasizes that the system acts as a bounded autonomous actor: it observes, decides, and acts on real-time conditions without human intervention, while coordinating with nearby vehicles through 5G-MEC. Although the module uses a lightweight SVM model for classification and predefined decision rules, it operates as an agent at the vehicle level, converting predictive insights into real-time cooperative actions such as rerouting alerts, adaptive offloading, and prioritized task execution.

The core of the system is the Agentic AI Decision Dashboard, which integrates the trained model with real-time sensor data to provide dynamic, actionable recommendations. The dashboard takes real-time data from various sensors (Table 4), feeds it into the trained SVM model to predict Latency_Category, and then uses a set of predefined proposed decision rules to generate a comprehensive system response.

Table 5 establishes a direct link between a vehicle's speed and the potential for network latency. At low speeds (10.04–46.69 km/h), the system assumes stable conditions and a Low latency category, so it maintains normal routing without broadcasting any alerts. At medium speeds (46.70–83.34 km/h), a Medium latency is expected, prompting the system to dynamically optimize MEC offloading to prevent delays. Finally, at high speeds (83.35–119.99 km/h), the system anticipates High latency due to the rapid change in location and potential handover issues, so it recommends a temporary speed reduction and broadcasts a rerouting alert via 5G-MEC to other vehicles.

This condition in Table 6 is concerned with time interval in which a task is serviced at the edge server. The Low processing time (5.01-43.33 ms) is a sign of Low latency scenario and the system proceeds with the present computation node. The processing time (43.34-81.65 ms) indicates a Medium value of latency, which would mean that the system would monitor the computation queue and offload to a Fog node. High latency (High processing time) 81.66-119.96 ms is an indicator of a high processing time that causes offloading to the nearest server on the MEC to minimize delay.

Table 7 is an association between energy consumption and network load and system efficiency. Low latency is linked with Low energy consumption (0.10-0.33 J) since tasks are probably efficiently being executed, and the system keeps the Cloud link. The 0.34-0.56 J of energy

Table 4
Used sensors to receive time data for independent features.

Feature	Typical Sensor or Data Source	Description/Function
Speed_kmh	OBD-II vehicle sensor, Wheel speed sensor, or GPS receiver	Measures instantaneous vehicle speed. Modern vehicles report this through the On-Board Diagnostics (OBD) port or vehicle CAN bus. GPS-based speed may also be used for redundancy.
Direction	🧭 Digital compass, GPS heading, or Inertial Measurement Unit (IMU)	Gives vehicle's movement direction (e.g., N, NE, E …). Useful for mobility prediction and link stability estimation between vehicle and edge node.
Event_Type	V2X event logs, ITS message detection, or camera/sensor fusion system	Derived from vehicular messages such as “accident detected”, “traffic congestion”, “weather alert”, or “normal condition”. Typically comes from the V2V/V2I communication stack or roadside units (RSUs).
Urgency_Level	Event classification logic/rule-based engine	Not directly measured by sensors; computed from contextual parameters — e.g., accident = high urgency, traffic = medium, weather = low.
MEC_Server_ID	☁ Edge infrastructure metadata	Assigned by the network orchestrator or edge controller when a vehicle connects to a specific MEC node (MEC1–MEC5). Indicates which edge server processes the task.
Processed_By	☁ Orchestration layer logs/API	Indicates whether computation was offloaded to the Fog node (nearby RSU or base station) or Cloud (centralized server). Collected from software-defined network logs.
Processing_Time_ms	⏱ MEC performance monitor	Measured at the MEC node or fog device itself — the time it takes to process a request (e.g., image classification, data aggregation). Available from edge resource manager APIs.
Energy_Consumed_J	🔋 On-board power sensor/ ECU or software estimator	Calculated from vehicle electronics (e.g., power draw of transmitter, CPU). Can be estimated using current × voltage × time from OBD sensors or fog node logs.
Latitude/Longitude	GPS receiver	Global positioning data used to determine vehicle location, server proximity, and potential network path delay.

Table 5
Decision rules by Speed_kmh (10.04 – 119.99) for agentic AI module.

Range Zone	Speed (km/h)	Latency_Category	System Decision/Action
Low	10.04 – 46.69	Low	Maintain normal routing; no broadcast needed
Medium	46.70 – 83.34	Medium	○ Slight delay detected — optimize MEC offload dynamically
High	83.35 – 119.99	High	● Reduce speed temporarily; broadcast reroute alert via 5G-MEC

Table 6
Decision rules by Processing_Time_ms (5.01 – 119.96) for agentic AI module.

Range Zone	ProcTime (ms)	Latency_Category	System Decision/Action
Low	5.01 – 43.33	Low	Stable processing — continue current computation node
Medium	43.34 – 81.65	Medium	⚠ Monitor computation queue; prepare offload to Fog
High	81.66 – 119.96	High	🚨 High delay — immediately offload to nearest MEC server

Table 7
Decision rules by Energy_Consumed_J (0.10 – 0.80) for agentic AI module.

Range Zone	Energy (J)	Latency_Category	System Decision/Action
Low	0.10 – 0.33	Low	Energy optimal — maintain Cloud link
Medium	0.34 – 0.56	Medium	⚡ Hybrid Cloud + Fog execution to balance energy
High	0.57 – 0.80	High	🔋 Overuse detected — throttle tasks, reduce transmission load

use indicates a Medium latency state, where the energy and the computational requirements are balanced by using a hybrid Cloud + Fog execution. High latency (0.57-0.80 J) is associated with High energy consumption, it means that the system is under stress, and it attempts to slow down tasks, decrease transmission load in order to save energy and achieve better performance.

Direction-based guidelines are applied in order to deduce the possibilities of network congestion depending on the usual traffic trends as depicted in Table 8. A Low latency path is attributed to travelling in the East, North-East, or North direction and hence the system enables the vehicle to proceed with its route. There is a Medium latency associated with the West and North-West directions, potentially causing congestion and thus a reroute check is triggered. Lastly, the directions of the South, South-East and South-West are associated with High latency and send a directive to turn the vehicle to the North-East direction of lower load through a MEC signal.

Table 9 maps real events of the world to latency and system actions. Normal event type means that the system is low latency in which only measurements are recorded. Traffic or Weather event has a Medium latency, which results in an adaptive throttling response or rerouting. An Accident event is the most critical one, which is associated with High latency, when a broadcast alert is activated immediately by 5G-MEC to cooperative rerouting of nearby vehicles.

Table 10 shows that this set of rules will match the urgency of the event with the reaction of the system. Low urgency event is addressed by routine data exchange, which is associated with Low latency. Medium urgency event demands prioritization of tasks with minimal load which is a reaction to the Medium latency. The High latency may be solved by an MEC offload and broadcast as High urgency event, like an accident.

Table 11 correlates the concrete MEC server that a vehicle is connected to and the anticipated latency. Low latency is related to connections to there are MEC1 and MEC2, which are presumably low load servers. The associations with MEC3 and MEC4 are associated with Medium latency that triggers monitoring and redistribution of the load

Table 8
Decision rules by direction (E, NE, N, NW, W, SW, S, SE) for agentic AI module.

Direction Zone	Latency_Category	System Decision/Action
East/North-East/ North	Low	Path clear — continue
West/North-West	Medium	Possible congestion — reroute check
South/South-East/ South-West	High	🧭 Redirect toward low-load NE direction via MEC signal

Table 9
Decision rules by Event_Type (accident, traffic, weather, normal) for agentic AI module.

Event Type	Latency_Category	System Decision/Action
Normal	Low	Stable operation — log metrics only
Traffic	Medium	🚗 Apply adaptive throttling and rerouting
Weather	Medium	🌧 Adjust driving pattern and compute offload dynamically
Accident	High	🚨 Broadcast alert via 5G-MEC for cooperative rerouting

Table 10
Decision rules by Urgency_Level (low, medium, high) for agentic AI module.

Urgency	Latency_Category	System Decision/Action
Low	Low	Routine data exchange only
Medium	Medium	Prioritize tasks with minimal load
High	High	Immediate MEC offload and reroute broadcast required

Table 11
Decision rules by MEC_Server_ID (MEC1 – MEC5) for agentic AI module.

Server Load Range	Latency_Category	System Decision/Action
MEC1–MEC2	Low	Allocate jobs normally
MEC3–MEC4	Medium	Monitor and redistribute load
MEC5	High	⚙ Offload to alternate MEC with lower congestion

by the system. High latency is connected to MEC5 which might be a backup unit or a overloaded unit and the system will offload the job to another MEC with less congestion.

Table 12 calculates system actions depending on location of computation. Latency is usually Low when the tasks are processed by Fog node, and local processing is preserved. The Medium latency of Offloading to the Cloud results into a balanced distribution of the load between the Cloud and Fog nodes. A Hybrid scheme in which the latency is High will cause rerouting of MEC to a low-latency area to overcome the delay.

Table 13 uses geographical location to predict network conditions. In Zone A (12.95–12.96, 77.58–77.59), the system assumes Low latency with a stable signal. In Zone B (12.96–12.98, 77.59–77.61), slight congestion is expected, correlating with Medium latency and recommending an alternate MEC. Zone C (12.98–12.99, 77.61–77.63) is identified as an overloaded region with High latency, prompting the system to broadcast a reroute to a less congested area.

The final and the critical rule, employs the forecasted latency group of the trained machine learning model to operate the final system reaction as depicted in Table 14. In the case of the model predicting Low latency, the response of the system is straightforward no broadcast and continued normal routing. An adaptive offload and pre-warning is triggered to the MEC by a Medium latency prediction. A High latency prediction is the most urgent step as the system automatically sends an alert to the surrounding vehicles using 5G-MEC, allowing cooperative rerouting and general latency reduction. This rule serves as the ultimate decision maker; it combines all the past sensor information and model knowledge into one unified action.

Table 12
Decision rules by Processed_By (cloud/Fog) for agentic AI module.

Processing Mode	Latency_Category	System Decision/Action
Fog	Low	Maintain local processing
Cloud	Medium	Balance load across Cloud and Fog nodes
Both (Hybrid)	High	Trigger MEC rerouting to low-latency region

Table 13
Decision rules by latitude/longitude (12.95–12.99/77.58–77.63) for agentic AI module.

Position Zone	Latency_Category	System Decision/Action
Zone A (12.95–12.96, 77.58–77.59)	Low	Standard route, stable signal
Zone B (12.96–12.98, 77.59–77.61)	Medium	Slight congestion, recommend alternate MEC
Zone C (12.98–12.99, 77.61–77.63)	High	📡 Overloaded region — broadcast reroute to NE

Table 14
Decision rules by predicted feature: Latency_Category for agentic AI module.

Predicted Label	Description	System Response
Low	Nominal network condition	No broadcast; continue standard routing
Medium	Moderate latency; transitional phase	Adaptive offload + pre-warning to MEC
High	Network overload or critical event	📡 Broadcast alert via 5G-MEC to nearby vehicles for cooperative rerouting and latency mitigation

4. Results and discussion

The implemented latency aware intelligent framework based on the Support Vector Machine (SVM) classifier was strictly tested. The model was trained using the processed and balanced training set (specified above in the preceding section) and tested on a 45% hidden test set. The results presented below confirm that the SVM model performs highly and is reliable when it comes to real-time prediction of the LatencyCategory. The visualization of the first validation of the prediction ability of the model is represented in Fig. 4. This is the plot of the actual (True) and the predicted LatencyCategory of the first 100 samples of the test set.

As illustrated in Fig. 4, the estimated (blue x marks) values seem to go hand in hand with the real (red circle markers) values in the first 100 samples. The lines of prediction are almost perfectly fidel to the true values. Lack of large deviations in the two series give a solid visual evidence that the trained SVM model has reached an impressive level of accuracy thus giving it the ability to classify the latency category of unseen data well and generalize. The first mapping is a strong indication that the model is highly trained and can be used with much degree of reliability in real-time. Fig. 5 Confusion Matrix details the quantitative performance of the model on the test set in terms of classification. The count of correct and incorrect prediction of each class is displayed in the matrix (on the left).

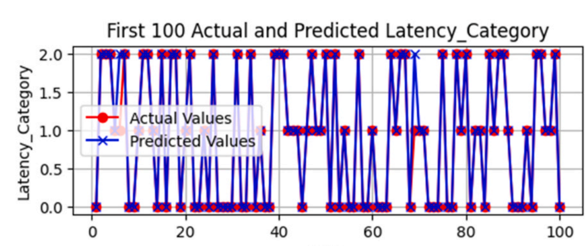

Fig. 4. First 100 Actual and Predicted Latency_Category using SVM.

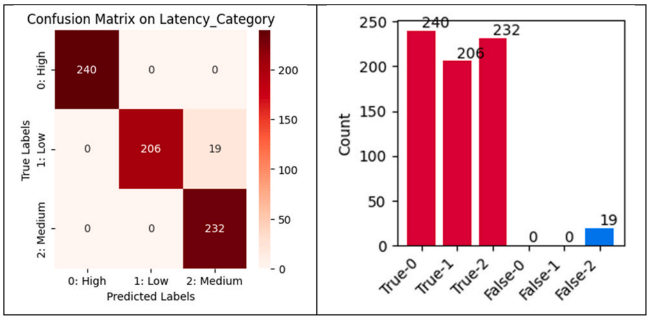

Fig. 5. Heat Map with True and False Prediction of Latency_Category using SVM Prediction.

– 0: High: There were 240 records that were correctly classified as High (True Positives).
– 1: Low: 206 records were judged as Low (True Positives).
– 2: Medium: The number of records that were rightly classified as Medium (True Positives) was 232.

Importantly the heat map indicates that there is no misclassification between the High (0) and the Low (1) class since there are zero misclassifications. Their mistakes amount to 19 cases of misclassification of 19 samples that were actually Low (1) but were predicted to be Medium (2) by the model. This is affirmed by the bar chart attached (on the right) where there is a False-2 count of 19. The few off-diagonal terms in the confusion matrix will testify the superb discriminative power of the model among the three classes of latency, and the discriminative model is highly accurate in differentiating the most critical High latency class.

4.1. Performance of SVM using statistical results

Statistical measures are essential variables to the precise quantification of the performance of a classification model. As it can be seen in Table 15 and Fig. 6, the proposed SVM model has exceptionally good performance.

The most important point is the overall Accuracy of 0.97 that is, the model rightly classified 97 percent of the test samples.

- The High (0) category had a perfect score: Precision (1.00), Recall (1.00) and F1-Score (1.00). This is crucial to a critical system as it would mean that the model is able to detect all single instances of High latency without false alarms (Precision) or false undetected real High latency events (Recall).
- The Low (1) class has the highest Precision (1.00) but lower Recall (0.92) which indicates that it has correctly classified all samples that it has predicted to be Low, but it has missed out on $8Percent of the true Low events (those 19 cases it has classified as being Medium).
- The Medium (2) class has an ideal Recall (1.00) and slightly worse Precision (0.92), that is, all true Medium events have been correctly predicted, however, 8-percent of the samples labelled as Medium were actually Low.

The Weighted Average F1-Score of 0.97 shows that the performance in all the three classes is strong and balanced.

Table 15
Proposed model performance for Latency_Category prediction.

Class	Precision	Recall	F1-Score
0 (High)	1.00	1.00	1.00
1 (Low)	1.00	0.92	0.96
2 (Medium)	0.92	1.00	0.96
Weighted Avg	0.97	0.97	0.97
Accuracy	0.97		

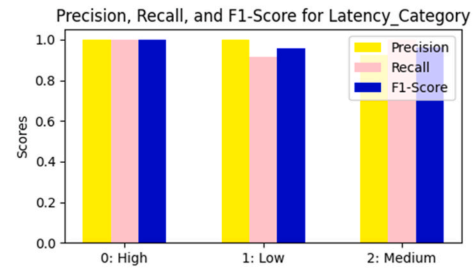

Fig. 6. Precision, recall, and F1-Score for predicted Latency_Category.

The results in Table 15 are also supported graphically by Fig. 6. The bar graph indicates clearly that the scores are close to or equal to the ideal score of 1.00 of all the metrics in all the classes. Bars of the High (0) type are the highest, which graphically proves the complete classification performance of the most important state of latency. The minor recalls decrease of Low class and Precision of the Medium class is also displayed in visual form, indicating the minor inter-class confusion experienced in the confusion matrix. By and large, the visualization indicates that the model is quite the best in terms of its performance in the multi-class prediction task.

Impact of Prediction Errors on Downstream Actions (False Alarms and Rerouting Quality). In real-world deployment, latency-state prediction may occasionally introduce errors relative to ground truth sensor/network conditions. In our evaluation, the observed misclassifications are limited to confusion between the Low and Medium states (19 samples), while no confusion occurs between the High and Low states in the confusion matrix. This behavior is important because the most safety-sensitive downstream actions (e.g., emergency broadcast alerts and cooperative rerouting) are associated with the High latency state, which is detected with perfect precision and recall in Table 15. Consequently, within the evaluated system, critical false alarms (Low→High) and missed critical events (High→Low) are not observed. Moreover, the Agentic AI module does not rely solely on the predicted Latency_Category; instead, it combines the model output (Table 14) with multiple sensor-driven rule checks (Tables 5–13), which provides redundancy and reduces the likelihood that a single prediction error propagates into an inappropriate action. When an occasional Low↔Medium confusion occurs, the resulting downstream response remains conservative (monitoring, adaptive offload, or load balancing) rather than disruptive emergency actions. Finally, the Medium state is treated as a transitional buffer by design, which helps absorb uncertainty and prevents abrupt oscillations in prescriptive actions under realistic noise.

4.2. Curves based evaluation of model performance

Curves-based evaluation gives a broad evaluation of the capability of the model to discriminate between classes especially the various probability thresholds. Such plots are critical in getting a profound insight into model robustness beyond those measures that are single-point such as accuracy. The Receiver Operating Characteristic (ROC) curve is a curve that is used to depict the True Positive Rate (Recall) compared to the False Positive Rate at different threshold settings. It is a powerful indication of the power of a model to discriminate classes. Fig. 7 illustrates the ROC curve of all the three classes. The curve passes right at the upper-left point of the plot, which has an Area Under the Curve (AUC) of 1.00. The value of AUC of 1.00 means that the model is perfectly separable; it is able to differentiate between the positive and negative classes across all three categories with 100% certainty in all classification thresholds. The outcome is very suggestive of a model that fits in the task exceptionally well.

The Precision-Recall (PR) curve is particularly valuable for multi-class problems or imbalanced datasets, as it focuses on the

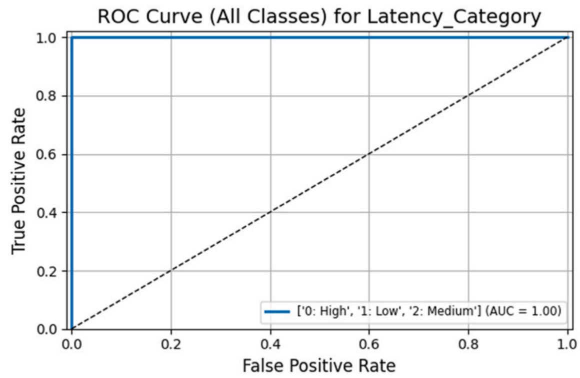

Fig. 7. Roc curve for Latency_Category.

performance of the positive class. It plots Precision against Recall. Fig. 8 displays the PR curve, which runs perfectly along the top-right corner of the plot, resulting in an Average Precision (AP) of 1.00. An AP of 1.00 confirms that the model maintains a perfect balance between Precision and Recall across all thresholds. This complements the perfect AUC score and reinforces the finding that the model is performing flawlessly in distinguishing all three latency categories.

The Cumulative Gain curve (or gain chart) assesses the efficiency of a model in identifying the target class compared to a random guess. It shows the proportion of true positives found when considering a given proportion of the samples sorted by the model's predicted probability. Fig. 9 shows the cumulative gain curves for all three classes (0, 1, and 2). The curves for all classes rise steeply and reach the maximum cumulative gain (1.0) by sampling only a small proportion of the total samples (around 30-40%). This is significantly superior to the Random Model (dashed grey line). For example, by analyzing the top 40% of the samples (those most confidently predicted as a class), the model has already identified nearly 100% of the instances belonging to that class. This steep rise confirms the model's high effectiveness in accurately ranking samples based on their predicted probability.

Lift curve is a result of the cumulative gain curve used to measure the degree of improvement of the model over a random selection. Both the Cumulative Gain (blue line) and the Lift Curve (green line) are shown in Fig. 10. The Cumulative Gain line also has its highest point of 1.0 long before the samples come to an end, once again indicating the high efficiency of the model. The Lift curve indicates that the lift value is about 2.0. The lift of 2.0 indicates that the model is just as good as a random guess of identifying the target class in the highest-scoring samples. The lift is far above the baseline of 1.0 on a vast majority of the samples indicating the consistently high predictive capability of the model on random chance.

Radar Charts (or spider charts) can be used to give a graphical overview of several performance indicators in one, multi-axis graph. A radar chart is illustrated in Fig. 11, each of the three classes has a radar

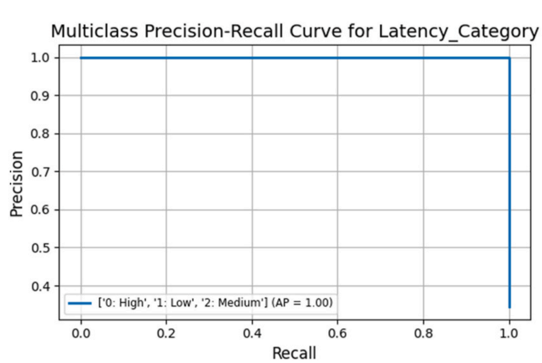

Fig. 8. Precision-recall curve for Latency_Category.

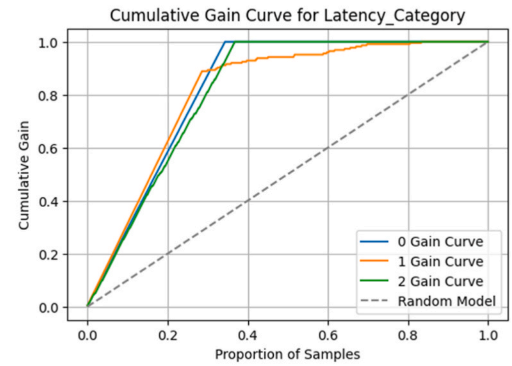

Fig. 9. Cumulative gain curve for Latency_Category.

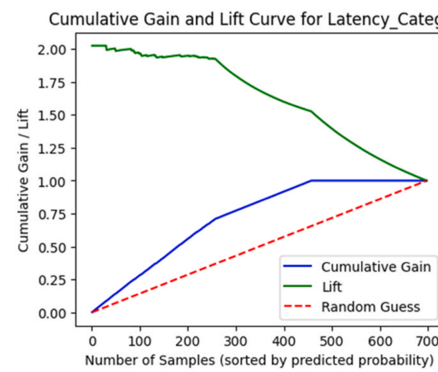

Fig. 10. Cumulative gain and lift curve for Latency_Category.

chart, Precision, Recall, F1-Score, ROC-AUC, and Accuracy are plotted on individual axes. In all three classes, the shaded region occupies almost the entire chart almost to the outer ring (which would be to have a score of 1.0). This graphic depiction is the brightest affirmation of the overall excellence of the model. The perfect limit is represented by the polygon of Class 0 (High) that proves its flawless scores on all metrics. In Classes 1 (Low) and 2 (Medium), the polygons are almost perfect which confirms the high-0.90 range scores of Table 15. The radar chart briefly summarises the solid, balanced, and almost flawless performance of the model in all the important statistical and curve based evaluation measures.

4.3. LIME based XAI analysis on proposed framework

In such critical systems like 5G-MEC vehicular networks, model performance, though mandatory, is not enough in itself. Accountability and transparency are critical towards the acceptance of the user and accountability to the system. In a bid to counter the black-box nature of the complex machine learning classifiers such as Support Vector Machines (SVM) we combine LIME (Local Interpretable Model-agnostic Explanations).

LIME is an explainable AI (XAI) method that is notably important since it provides local faithful explanations by approximating the black-box model behavior about a certain data point. This step shows the input features that have the greatest impact in influencing a certain prediction. This transparent performance, which is white-box, plays a critical role in:

Debugging and Validation: It is permitting domain experts to pass a test that the logic of the model is consistent with the principles of physical and networks, not with coercional relationships.

User Trust: Allowing end-users (e.g., system operators or vehicle control systems) to know the reason a particular latency prediction was achieved (High, Low, or Medium), and represents a confidence that the

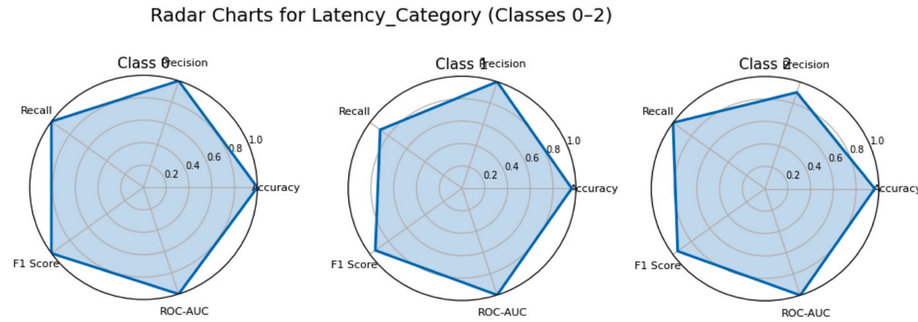

Fig. 11. Radar charts for Latency_Category.

resulting Agentic AI action plan will be trusted.

Regulatory Compliance: Despite making auditable track of critical decisions that influence the safety and quality of services.

As an illustration of this transparency, we have selected three records, all in the test data with the three categories of latency, one of which had the actual class the same as the predicted class, and created the LIME explanations of these records.

4.4. LIME analysis on predicted Latency_Category (high: 0)

Fig. 12 displays the LIME explanation for Record-100, where both the Actual and Predicted Latency_Category are High (0). The goal of this analysis is to identify the features contributing to the critical "High" latency classification. The key features driving the High latency prediction are primarily those associated with network congestion and resource strain. The most significant factor contributing to the positive prediction (factors pushing the prediction toward High: 0) is a high Processing_Time_ms, specifically condition 34.60 < Processing_Time_ms, which contributes a substantial numerical weight of +0.36. This clearly indicates that a high processing load at the edge is the strongest predictor of high latency for this specific record. Furthermore, Energy_Consumed_J greater than 0.45 (with an actual value of 0.58) adds a positive contribution of +0.07, suggesting higher energy expenditure linked to intensive communication or computation. Finally, the specific MEC_Server_ID (Value: 2.00) contributes +0.04, indicating that connection to a specific, potentially congested, MEC server ID is increasing the risk. Conversely, Speed_kmh being less than 37.80 km/h (with an actual value of 65.57 km/h) acts as the primary counter-force, adding a negative weight of −0.03. While the vehicle's speed is high, its negative contribution suggests that other factors, like processing time, are more dominant than speed alone in this specific instance. The strong, concentrated positive weight from Processing_Time_ms (0.36) clearly explains the model's correct and confident prediction of High Latency (0).

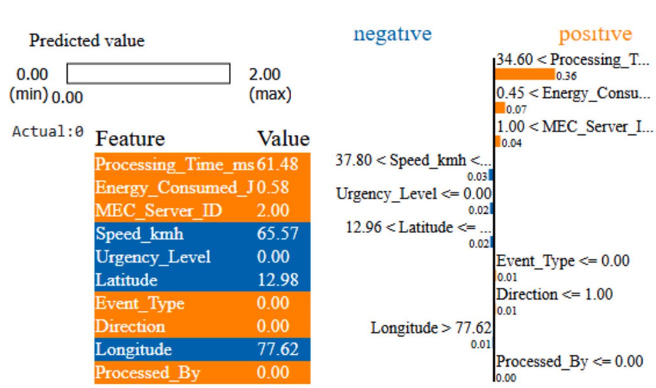

Fig. 12. Lime explanation for predicted Latency_Category (high: 0) on Record-100.

4.5. LIME analysis on predicted Latency_Category (low: 1)

Fig. 13 illustrates the LIME explanation for Record-400, where the Actual and Predicted Latency_Category are correctly identified as Low (1). This analysis confirms the model's reasoning for classifying a state as safe and stable. The most dominant factor contributing positively (factors pushing the prediction toward Low: 1) is a low value for Processing_Time_ms being less than a certain threshold, with an actual value of 11.38 ms. This efficient processing condition contributes a significant +0.02 to the Low latency prediction, acting as a direct indicator of a low-latency environment. A low MEC_Server_ID (Value: 0.00), likely representing a low-load or preferred server, also contributes positively +0.02. Additionally, Longitude being within a specific low-congestion range adds a minor positive contribution of +0.01. Conversely, the largest negative contributor (factors pushing the prediction away from Low: 1) is Speed_kmh being in the mid-range (Value: 83.00 km/h), with a magnitude of −0.05. High Energy_Consumed_J (Value: 0.78 J) also contributes negatively −0.03, as high energy use often implies complex, high-bandwidth operations that could potentially increase latency. Despite these negative contributions from speed and energy consumption, the combined, influential impact of efficient Processing_Time_ms and favorable MEC_Server_ID conditions confirms the model's robust classification of this record as Low Latency (1).

4.6. LIME analysis on predicted Latency_Category (medium: 2)

Fig. 14 presents the LIME analysis for Record-10, where the model accurately predicts a Medium Latency (2) state. This explanation is crucial for understanding the features that create an indeterminate or transitional network state. The feature that provides the most significant positive influence (factors pushing the prediction toward Medium: 2) is Processing_Time_ms having a value in the intermediate range of 22.16 < Processing_Time_ms < 31.80. This condition provides an overwhelming contribution of +1.04, clearly establishing processing time in the non-critical but non-optimal range as the primary indicator of the Medium state. A high Speed_kmh greater than 93.35 km/h (Value: 94.80 km/h)

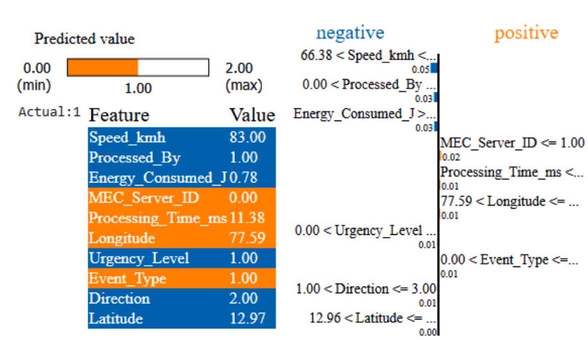

Fig. 13. Lime explanation for predicted Latency_Category (low: 1) on Record-400.

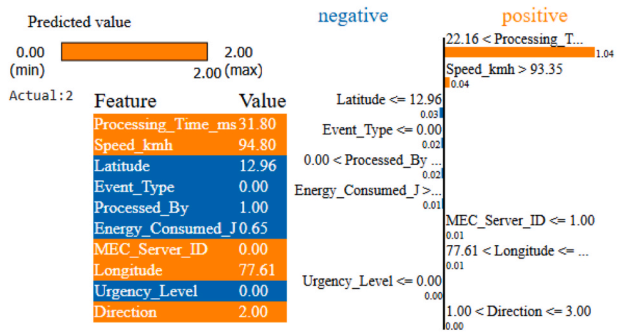

Fig. 14. Lime explanation for predicted Latency_Category (medium: 2) on Record-10.

also contributes positively +0.04. High speed, while not immediately critical, places the environment into a state of flux, increasing the probability of medium latency. The primary negative factor (factors pushing the prediction away from Medium: 2) is Latitude being less than 12.96 (Value: 12.96), contributing −0.03. This suggests that the vehicle's geographical position is still somewhat stable, which works against the Medium prediction. Other factors, such as Event_Type (Value: 0.00) and Processed_By (Value: 1.00), offer minor negative contributions, indicating the absence of certain critical events or processing bottlenecks. The overwhelmingly dominant positive influence from the intermediate Processing_Time_ms is the clear driver for the Medium Latency (2) prediction, correctly identifying a state that requires system attention but is not yet critical. The LIME analysis across all three classes successfully translates the black-box SVM decision into transparent, feature-level contributions, establishing the model's reliability and building user trust.

4.7. Results of agentic AI coordination using 5G-MEC

The main task of the trained machine learning model is used as the predictive core of the Agentic AI Coordination Module of the 5G-MEC vehicular network structure. This module works under the closed-loop concept: real-time data provided by on-board vehicle sensors, according to the independent features of the suggested methodology (Table 4)), is directly fed into the trained model. The model is predictive of the Latency_Category (High, Low or Medium). According to this forecast and the sensor information about the vehicle, the Agentic AI will provide the vehicle with critical and instantaneous output parts of the dashboard, which consists of: a status alert, action plan, system recommendations, and an optimized route. A collection of established rules of operation controls these outputs (Tables 5–14). Moreover, the Agentic AI takes advantage of the fact that 5G-MEC has ultra-low latency and uses it to send proactive alerts to nearby vehicles, enabling them to rerouted cooperatively and reduce network latency. Counterfactual Analysis has made this decision-making process more transparent and has offered actionable intelligence that enables how the independent features, which are read by the sensors, may be altered to change between the latency categories. In particular, this analysis illustrates the way to reduce the risk (High to Low/Medium), provide the protection of system care (Low to High/Medium), or address transitional states (Medium to Low/High). The subsequent passages describe the adopted simulation on Python platform and display the dashboard output with the broadcast alert system of each of the predicted latency categories.

4.7.1. Agentic AI dashboard and broadcast alert for high latency

The simulation for Record-100 results in the prediction of Latency_Category High (0), a critical state demanding immediate intervention. The Agentic AI Decision Dashboard for Vehicle @ (12.97515, 77.61882), as depicted in Fig. 15, reflects this risk. The status alert identifies a "High latency risk detected," corroborated by sensor readings showing a high ProcTime of 61.48 ms (Mid) and a high Energy consumption of 0.58 J (High), indicating severe resource congestion, potentially due to cloud overload. The immediate action plan formulated by the Agentic AI is a multi-step mitigation strategy: "Reduce speed temporarily to stabilize link," "Offload computation to nearby MEC server," and "Initiate rerouting through alternate low-load area." The system suggestions further recommend strategic, long-term actions such as "Train adaptive latency correction model" and "Request network priority for high-urgency vehicles." Concurrently, to ensure collective safety and network stability, the Agentic AI triggers a Broadcast alert sent via 5G-MEC to nearby vehicles (Fig. 16), explicitly stating a "High latency risk detected" and broadcasting a reroute message (Event:

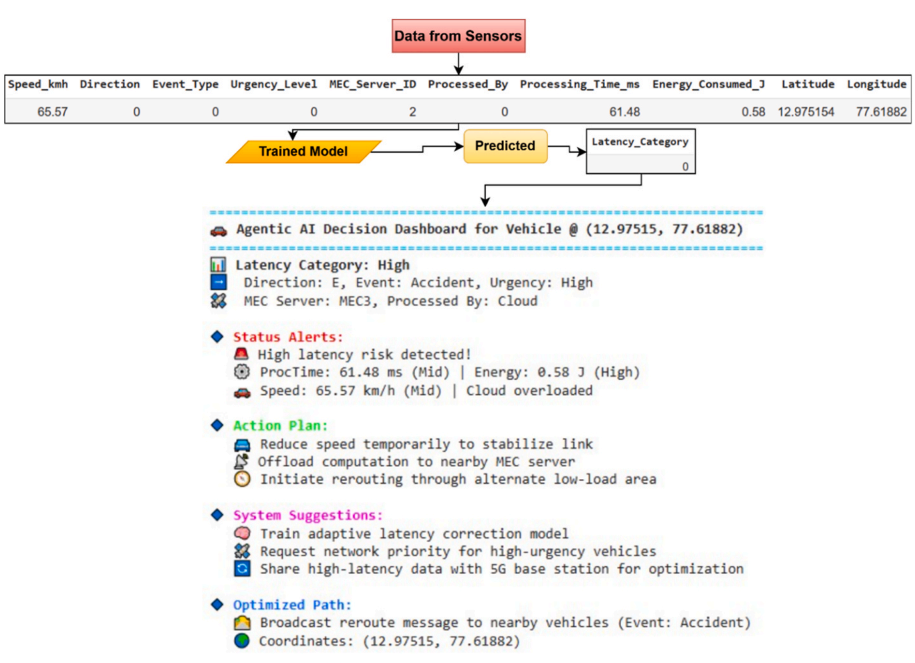

Fig. 15. Agentic AI decision dashboard for by predicting Latency_Category high for Record-100.

```
==================================================================
📡 Broadcast alert sent via 5G-MEC to nearby vehicles
   for cooperative rerouting and latency mitigation.
==================================================================

{'Status Alerts': ['🚨 High latency risk detected!',
  '⚙ ProcTime: 61.48 ms (Mid) | Energy: 0.58 J (High)',
  '🚗 Speed: 65.57 km/h (Mid) | Cloud overloaded'],
 'Action Plan': ['🚙 Reduce speed temporarily to stabilize link',
  '📡 Offload computation to nearby MEC server',
  '🕓 Initiate rerouting through alternate low-load area'],
 'System Suggestions': ['🧠 Train adaptive latency correction model',
  '🛰 Request network priority for high-urgency vehicles',
  '🔄 Share high-latency data with 5G base station for optimization'],
 'Optimized Path': ['📢 Broadcast reroute message to nearby vehicles (Event: Accident)',
  '🌍 Coordinates: (12.97515, 77.61882)']}
```

Fig. 16. Broadcast alert sent via 5G-MEC to nearby vehicles by Predicting Latency_Category High for Record-100.

Accident) along with the coordinates.

The Agentic AI further employs Counterfactual Analysis to determine the minimum necessary modifications to sensor-read features required to alter the prediction from High to a less critical state (Low or Medium), thereby providing executable guidance for risk removal.

Table 16 details the transition from the Actual High latency state (where Processing_Time_ms is 61.48 ms to a desirable Low latency state. The analysis shows that reducing the Processing_Time_ms is the dominant feature for risk mitigation. In Counterfactual #4, by only changing Processed_By from 0 to 1 (shifting the computation method) and dramatically reducing Processing_Time_ms from 61.48 ms to a mere 7.58 ms, the model successfully predicts the desired Low latency. This confirms the direct causal link between processing load and network criticality.

Table 17 illustrates the less drastic, intermediate step of reducing the risk from High to Medium latency. Here, the required adjustment to Processing_Time_ms is less severe, dropping from 61.48 ms in the actual record to approximately 40 ms in the counterfactual examples. For instance, Counterfactual #4 achieves a Medium prediction by reducing Processing_Time_ms to 39.26 ms and slightly modifying Latitude. This analysis confirms that the Medium category acts as a transitional buffer, requiring only a partial, but significant, reduction in the primary bottleneck feature.

4.7.2. Agentic AI dashboard and broadcast alert for low latency

For Record-500, the prediction is a desirable Latency_Category Low (1). Fig. 17 shows the dashboard confirming “Stable low latency detected (Low)," with highly efficient processing time of 7.07 ms (Low). The action plan here is conservative, aimed at maintaining the current operational status: “Maintain speed and network configuration” and “Continue energy-efficient mode.” System suggestions focus on optimization and sharing: “Continue localized edge computation” and “Share optimized performance data with nearby vehicles.” The corresponding broadcast alert shown in Fig. 8 simply communicates the Stable low latency detected (Low) status, informing nearby agents of the current un-congested conditions. Fig. 18 shows that the predicted latency category low for record-500.

The counterfactual analysis for the Low state is critical for “care” assessment, predicting the potential for risk if operational features deteriorate.

Table 18 demonstrates how the system can be pushed from a Low state (Actual Processing_Time_ms is 6.66 ms to the critical High state. Counterfactual #1 shows that simply increasing Processing_Time_ms from 6.66 ms to a high value of 89.26 ms is sufficient to trigger the High classification, while keeping all other features constant. Counterfactual #2 shows a combination of a moderately high Speed_kmh 50.97 km/h and a high Processing_Time_ms 73.96 ms also forces the High category. This highlights that Processing_Time_ms is the primary sensitivity factor that, when neglected, rapidly escalates the risk level.

Table 19 identifies the feature adjustments needed to reach the Medium transitional state. This requires a moderate increase in Processing_Time_ms to the intermediate range. For example, in Counterfactual #2, increasing Processing_Time_ms to 44.36 ms is enough to push the prediction to Medium. This analysis provides the system administrators with a clear threshold to monitor, indicating the operational point at which a stable state transitions into a state of potential congestion.

4.7.3. Agentic AI dashboard and broadcast alert for medium latency

Record-10 simulation has a predicted Latency_Category Medium (2) which is a medium congestions and potential risk state. The alert presented in Fig. 19 is a “Moderate latency observed - possible congestion” with the backup of Processing time of 31.20 ms (Mid). The action plan of the Agentic AI is devoted to the load balancing and adaptation: “Reduce transmission frequency slightly,” “Switch hybrid (Fog + Cloud) mode load balancing,” and “Check congestion along existent route.' Some of the system recommendations are, Retrain model using prevailing environmental information and Adjust power allocation and MEC routing dynamically. The optimal direction proposes an Alternate path direction: NW. The broadcast alert to the local vehicles is verified through Fig. 20 which informs about the condition of moderate latency observed.

The counterfactual analysis of the Medium state is necessary to identify the risk mitigation (Medium to Low) as well as require the conditions of a critical failure (Medium to High). Table 20 indicates the transformation of the actual Medium state with ProcessingTimems of 31.80 ms to the critical High state. Counterfactual #3 goes to show that the high level of ProcessingTimems of 111.08 ms is a direct route to the High latency. Such analysis gives the threshold of the worst-case scenario, which makes the system aware of the processing load value that causes critical service failure.

Table 21 shows what has to be done to reduce the moderate risk and solutions on returning to the stable Low latency state. A Medium latency is changed into Low in Counterfactual #2 by reducing markedly the ProcessingTimems of 31.80 ms down to 14.36 ms. This solidifies the idea that it is the most effective and most important control over the processing load of a system that can be used to control latency, and system performance so that the Agentic AI can place priority in resource allocation on this counterfactual understanding.

4.8. Crossvalidation for Proposed Model

To evaluate the robustness and generalizability of the proposed lightweight SVM model, a rigorous 5-fold cross-validation was conducted on the training set. This process involves partitioning the training data into five equal subsets, where the model is iteratively trained on four folds and validated on the remaining fold. This approach ensures that the performance metrics are not biased by a single favorable data split and provides a reliable estimate of how the model will perform on

Table 16
Modifying features for a Latency_Category high to low for Record-100.

Record_Type	Speed_kmh	Direction	Event_Type	Urgency_Level	MEC_Server_ID	Processed_By	Processing_Time_ms	Energy_Consumed_J	Latitude	Longitude	Latency_Category
Actual	65.57	0	0	0	2	0	61.48	0.58	12.975154	77.618820	High
Counterfactual #1	65.57	0	3	0	2	0	21.30	0.58	12.975154	77.618821	Low
Counterfactual #2	65.57	0	0	0	2	0	21.30	0.58	12.975154	77.618821	Low
Counterfactual #3	65.57	0	0	0	2	0	26.44	0.58	12.975154	77.618821	Low
Counterfactual #4	65.57	0	0	0	2	1	7.58	0.58	12.975154	77.618821	Low

Table 17
Modifying features for a Latency_Category high to medium for Record-100.

Record_Type	Speed_kmh	Direction	Event_Type	Urgency_Level	MEC_Server_ID	Processed_By	Processing_Time_ms	Energy_Consumed_J	Latitude	Longitude	Latency_Category
Actual	65.57	0	0	0	2	0	61.48	0.58	12.975154	77.618820	High
Counterfactual-1	65.57	0	2	0	2	0	46.37	0.58	12.975154	77.618821	Medium
Counterfactual-2	65.57	0	0	0	2	0	42.59	0.58	12.975154	77.618821	Medium
Counterfactual-3	65.57	0	0	0	2	0	42.47	0.58	12.975154	77.618821	Medium
Counterfactual-4	65.57	0	0	0	2	0	39.26	0.58	12.968921	77.618821	Medium

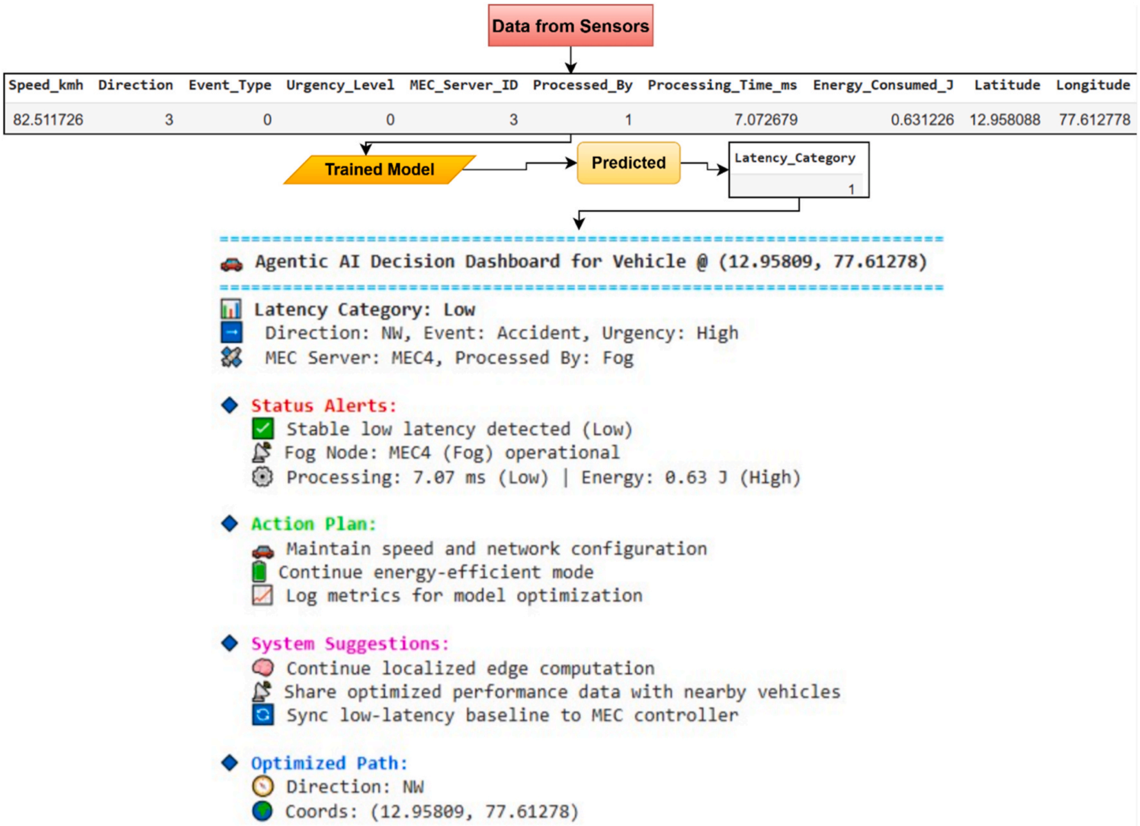

Fig. 17. Agentic AI decision dashboard for by predicting Latency_Category low for Record-500.

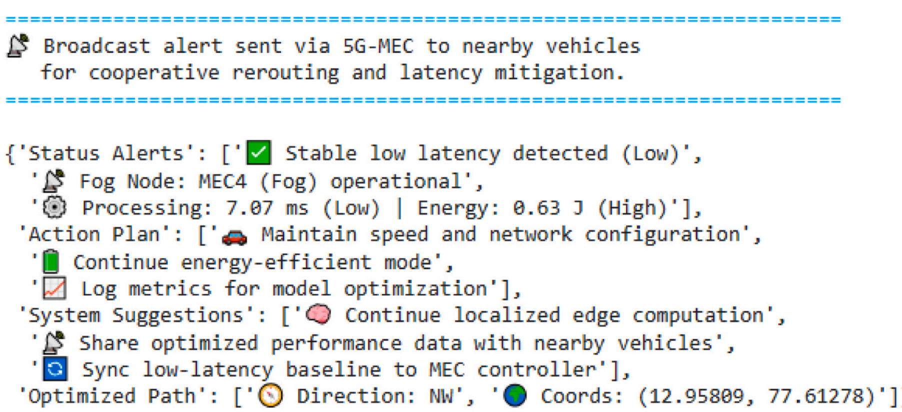

Fig. 18. Broadcast alert sent via 5G-MEC to nearby vehicles by Predicting Latency_Category Low for Record-500.

unseen data. The results, as illustrated in Fig. 21, demonstrate exceptional stability across all folds. The model achieved a mean Accuracy of 0.9636 ± 0.0201, indicating high consistency with very low variance. Furthermore, the Macro F1-score and Weighted F1-score were both calculated at 0.9634, confirming that the model maintains a high balance between precision and recall across all latency categories, even when accounting for potential class imbalances within the training subsets.

The selection of a 55% training and 45% test data split was determined to be optimal for this specific 5G-MEC vehicular dataset. While traditional machine learning often utilizes a larger training ratio (e.g., 80/20), the proposed 55/45 split was chosen to provide a more rigorous challenge for the lightweight SVM model and to ensure an extensive evaluation set for the Agentic AI module's prescriptive rules. By allocating a substantial 45% of the data to the test set, we provide a larger variety of real-time sensor scenarios for the Agentic AI to process, thereby better validating its ability to generate accurate Status Alerts, Action Plans, and Counterfactual Analysis across a wide range of network conditions. This balanced split ensures that the model is sufficiently trained to capture the complex spatiotemporal patterns of the 5G-V2N data while simultaneously proving that the system remains highly accurate and reliable when faced with a large, diverse set of unseen operational data.

4.9. Mapping classification performance to system-level KPIs and discussion on generalization

While this work primarily evaluates the predictive component using standard classification metrics (Accuracy, Precision, Recall, F1-score, ROC-AUC), it is important to clarify how these metrics translate into

Table 18
Modifying features for a Latency_Category low to high for Record-500.

Record_Type	Speed_kmh	Direction	Event_Type	Urgency_Level	MEC_Server_ID	Processed_By	Processing_Time_ms	Energy_Consumed_J	Latitude	Longitude	Latency_Category
Actual	29.857101	2	1	1	0	1	6.657786	0.552367	12.965424	77.602493	Low
Counterfactual-1	29.857101	2	1	1	0	1	89.26	0.552367	12.965424	77.602492	High
Counterfactual-2	50.970000	2	1	1	0	1	73.96	0.552367	12.965424	77.602492	High
Counterfactual-3	29.857101	2	1	1	0	1	59.01	0.460000	12.965424	77.602492	High
Counterfactual-4	29.857101	2	1	1	0	1	93.57	0.552367	12.965424	77.602492	High

Table 19
Modifying features for a Latency_Category low to medium for Record-500.

Record_Type	Speed_kmh	Direction	Event_Type	Urgency_Level	MEC_Server_ID	Processed_By	Processing_Time_ms	Energy_Consumed_J	Latitude	Longitude	Latency_Category
Actual	29.857101	2	1	1	0	1	6.657786	0.552367	12.965424	77.602493	Low
Counterfactual-1	29.857101	1	1	1	0	1	43.72	0.552367	12.965424	77.602492	Medium
Counterfactual-2	29.857101	2	1	1	0	1	44.36	0.552367	12.965424	77.602492	Medium
Counterfactual-3	15.960000	2	1	1	0	1	32.26	0.552367	12.965424	77.602492	Medium
Counterfactual-4	29.857101	2	1	1	0	1	51.25	0.552367	12.965424	77.602492	Medium

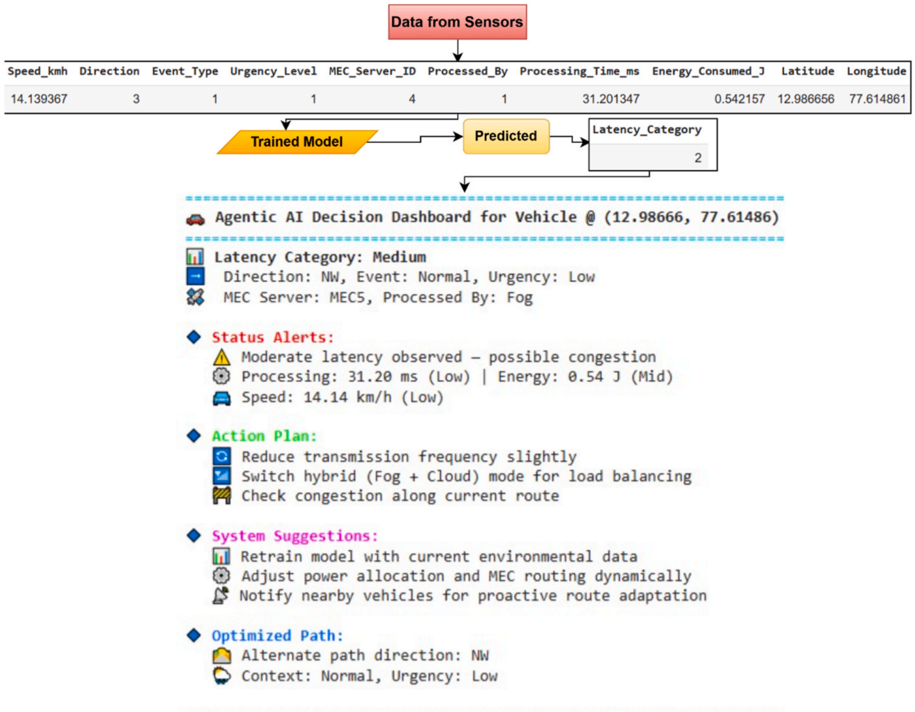

Fig. 19. Agentic AI decision dashboard for by predicting Latency_Category medium for Record-10.

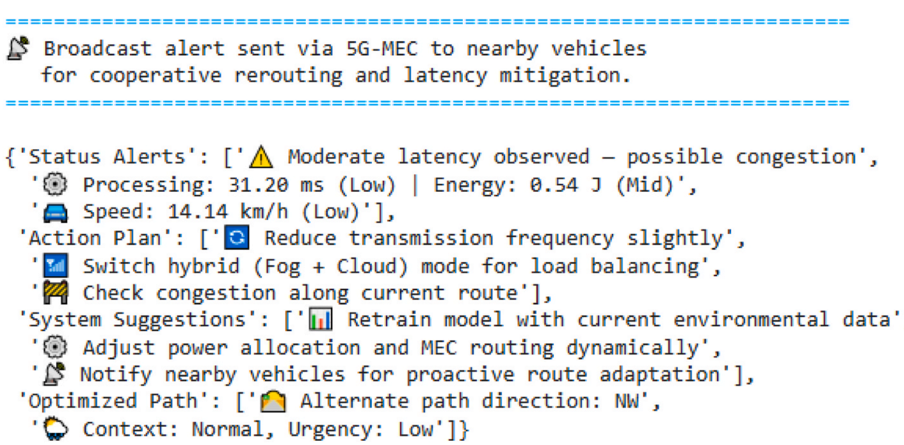

Fig. 20. Broadcast alert sent via 5G-MEC to nearby vehicles by Predicting Latency_Category Medium for Record-10.

meaningful system-level performance for the proposed Agentic AI framework. In the proposed architecture, prediction quality directly governs the quality of downstream actions such as rerouting, offloading decisions, and broadcast alerts. For instance, the perfect Precision and Recall achieved for the High latency class (Table 15) imply that critical events triggering emergency broadcasts and cooperative rerouting are neither falsely generated nor missed within the evaluated system. This directly supports the reliability of safety-critical actions and prevents unnecessary congestion amplification due to false alarms.

Similarly, the limited confusion observed only between Low and Medium states (19 samples) leads exclusively to conservative downstream effects, such as temporary monitoring or adaptive offload, rather than disruptive rerouting. In system terms, this means that occasional misclassification does not degrade route stability, increase congestion, or introduce unsafe actions, but instead results in precautionary behavior aligned with the design of the Medium state as a buffer. Therefore, the classification metrics reported are not abstract indicators but directly reflect operational quality of rerouting reliability, delay mitigation behavior, and congestion-aware coordination.

Regarding generalization, we explicitly acknowledge that the dataset used is limited in size (1000 samples) and that class balancing was performed using Borderline-SMOTE. To mitigate this concern, we added a full 5-fold cross-validation analysis (Section 4.8), which demonstrates consistent performance across folds (mean Accuracy = 0.9636 ± 0.0201, Macro and Weighted F1-score = 0.9634). This stability indicates that the model does not rely on a favorable single partition and maintains robust behavior under different data splits. Furthermore, the 45% held-out test set already represents a large proportion of unseen data relative to typical machine learning studies, strengthening confidence in generalization within the available dataset.

Nevertheless, we recognize that real-world deployment would benefit from future evaluation on larger-scale and externally collected datasets, and we explicitly frame the proposed system as a validated prescriptive framework whose architectural principles, explainability mechanisms, and agentic coordination logic are transferable beyond the specific dataset used in this study.

Table 20
Modifying features for a Latency_Category medium to high for Record-10.

Record_Type	Speed_kmh	Direction	Event_Type	Urgency_Level	MEC_Server_ID	Processed_By	Processing_Time_ms	Energy_Consumed_J	Latitude	Longitude	Latency_Category
Actual	94.799759	2	0	0	0	1	31.802691	0.652935	12.958105	77.612404	Medium
Counterfactual-1	94.799757	2	0	0	0	1	69.72	0.652935	12.958105	77.612405	High
Counterfactual-2	16.600000	2	0	0	0	1	100.09	0.652935	12.958105	77.612405	High
Counterfactual-3	94.799757	2	0	0	0	1	111.08	0.652935	12.958105	77.612405	High
Counterfactual-4	94.799757	2	2	0	0	1	89.09	0.652935	12.958105	77.612405	High

Table 21
Modifying features for a Latency_Category medium to low for Record-10.

Record_Type	Speed_kmh	Direction	Event_Type	Urgency_Level	MEC_Server_ID	Processed_By	Processing_Time_ms	Energy_Consumed_J	Latitude	Longitude	Latency_Category
Actual	94.799759	2	0	0	0	1	31.802691	0.652935	12.958105	77.612404	2
Counterfactual-1	94.799757	2	3	0	0	1	14.36	0.652935	12.958105	77.612405	1
Counterfactual-2	94.799757	2	0	0	0	1	14.36	0.652935	12.958105	77.612405	1
Counterfactual-3	94.799757	2	0	0	4	1	7.35	0.652935	12.958105	77.612405	1
Counterfactual-4	94.799757	2	0	0	0	0	15.01	0.652935	12.958105	77.612405	1

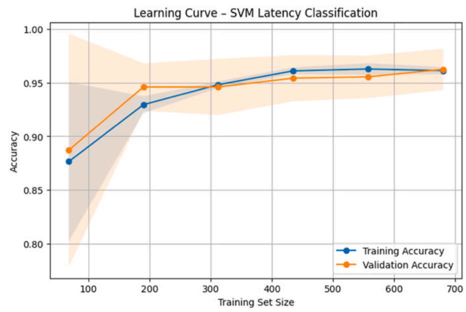

Fig. 21. Cross validation for proposed model.

4.10. Efficiency claims and computational scope

The efficiency claim in this work refers to architectural and algorithmic simplicity rather than to experimentally measured runtime or system-level latency gains. Specifically, the proposed framework is characterized as lightweight because it relies on (i) a compact SVM model operating on only 10 input features, (ii) deterministic rule-based decision logic (Tables 5–14) that introduces negligible computational overhead, and (iii) the absence of deep or iterative feature extraction pipelines such as CNNs, GNNs, or LSTMs commonly used in related vehicular intelligence studies. This design choice directly supports feasibility for on-board inference in vehicles and aligns with low-complexity deployment constraints.

However, we explicitly acknowledge that this study does not report measured inference time (e.g., milliseconds per prediction), MEC processing overhead, or end-to-end latency reduction under a deployed network or simulator. The presented results therefore demonstrate predictive performance, explainability, and prescriptive behavior of the Agentic AI framework, rather than empirical system-level timing benchmarks.

Table 22
Comparison of proposed work with similar studies.

Existing Work	Used model in existing work	Objective of Existing Work	Key limitation(s) of Existing Work	Advantages of the Proposed Work
Forecasting & Improving Latency in 5G V2X (MELECON 2024) [15]	Network/latency forecasting models (ML-based)	Forecast latency in 5G V2X to improve autonomous driving safety	Focuses on forecasting; limited integration into a prescriptive/cooperative control loop.	The proposed work converts predicted latency into rule-based outputs (alerts, reroutes) for immediate cooperative mitigation via 5G-MEC
Rethinking the mobile edge for vehicular services (2024) [17]	MEC deployment/system design (analytic + simulation)	Propose realistic MEC placement/config for vehicular services to reduce latency	System-level placement focus; does not demonstrate closed-loop vehicle-level prescriptive decisioning.	The proposed work embeds lightweight prediction and agentic decision module on vehicles for real-time action
VANET Edge Computing Model to Minimize Latency using 5G (Journal of Grid Computing 2024) [20]	Edge/MEC task offloading algorithms	Minimize latency and delay in smart-city VANET applications	Emphasizes optimization/offloading strategies; limited XAI/counterfactual or cooperative broadcast integration.	The proposed work integrates counterfactual analysis and broadcasts rerouting alerts to neighboring vehicles
GCN-DML/Deep learning model for efficient traffic forecasting (2025) [6]	GCN + LSTM in a distributed ML (edge) setup	High-accuracy traffic forecasting on NGSIM, edge-friendly distributed training	Strong predictive performance but mainly prediction-oriented; limited demonstration of prescriptive actions/in-vehicle agent.	The proposed work provides prescriptive rule sets and broadcast functionality per vehicle to respond to predicted latency
Dynamic Graph Conv. Networks with Temporal Rep. Learning (DGCN-TRL) (2025) [10]	Dynamic GCN + temporal representation	Capture dynamic temporal dependencies in traffic for better forecasting	Complexity and model size can limit edge deployment; still prediction-focused.	The proposed work uses lightweight SVM model and rules for minimal computational overhead
LSTM-based Graph Attention for vehicle trajectory (2024) [13]	GAT/LSTM hybrid	Encode vehicle interactions and predict trajectories	High model complexity; centralized training & limited edge evaluation.	The proposed work demonstrates agent-side model deployment and decision rules integrated into vehicle dashboards
ML-driven latency optimization for MEC (survey/2025) [22]	ML-based offloading & scheduling	Optimize latency and offloading decisions in MEC-enabled networks	Survey indicates many works still lack end-to-end vehicle-level prescriptive control and cooperative V2V broadcast.	The proposed work closes loop by coupling prediction with actionable vehicle coordination and alerts
Simulating & validating Vehicular Cloud/MEC-enabled 5G apps (2024) [23]	Simulation platform for VCC + MEC applications	Provide a testbed to design and test vehicular + edge applications	Tooling focus; few studies use real-time agentic prescriptive modules in-simulation.	The proposed work implements agentic AI module that performs real-time prescriptive decisions and broadcast alerts
Urban Traffic Forecasting w/integrated travel time (2024) [16]	GCN + LSTM + conformal prediction	Improve travel-time-aware traffic forecasting and uncertainty estimates	Strong on uncertainty/prediction quality, but doesn't provide vehicle-level prescriptive actions.	The proposed work issues prescriptive alerts and coordinated rerouting messages based on predictions
Spatiotemporal Synchronous Graph (2025) [11]	Advanced spatiotemporal GCN layers	Capture dynamic spatiotemporal traffic correlations for forecasting	Sophisticated layers increase inference cost; edge deployment and prescriptive use not demonstrated.	The proposed work uses a lean predictor + decision rules that permit immediate dashboard and broadcast outputs
LSTGAN: Long-term Spatio-Temporal Graph Attention Network (2025) [14]	Graph attention + long-term modeling	Long-term traffic pattern modeling and forecasting	Focused on long-term history and accuracy; not focused on immediate per-vehicle actionable control.	The proposed work focuses on short-horizon latency classification and counterfactual transitions between latency states
Adaptive-spatiotemporal GCN (Nature Sci Reports 2025) [24]	Adaptive spatiotemporal GCN + attention	Better capture spatiotemporal correlations for traffic flow prediction	High model complexity and centralized training requirements.	The proposed work is designed for lightweight inference on vehicles with embedded decision rules and broadcast paths
MEC-enabled resource allocation in Internet of Vehicles (2024) [21]	Joint task sequencing & resource allocation algorithms	Balance resource allocation and sequencing to reduce latency	Algorithmic focus; lacks integrated vehicle-level predictive decisioning and cooperative broadcast logic.	The proposed work combines predictive classification, rule-based decision, and cooperative rerouting broadcast via 5G-MEC
Continual learning framework for real-time traffic (2025) [25]	Continual learning + GCN for real-time prediction	Maintain model performance over time and nonstationary traffic	Addresses model drift but doesn't convert predictions into prescriptive cooperative control.	The proposed work includes counterfactual analysis plus dashboard + broadcast for prescriptive action under varying latency states

A full runtime profiling, including model inference latency, communication delay through 5G-MEC, and quantitative impact on end-to-end vehicular coordination, would require either a real-world testbed or a large-scale vehicular network simulator integrated with MEC infrastructure. Such evaluation is beyond the scope of the current dataset-driven study and is identified as an important direction for future work.

4.11. Comparison of proposed work with similar studies

The comparative analysis highlights that most existing studies between 2024 and 2025 in Table 22 primarily emphasize prediction accuracy, traffic forecasting, or resource optimization within 5G-MEC and vehicular networks. Techniques such as GCN, LSTM, and GAT are commonly employed for spatiotemporal feature extraction and latency or trajectory prediction, yet these models largely remain forecast-oriented. They do not translate predictive outcomes into real-time, prescriptive actions or cooperative decision mechanisms at the vehicle level. Furthermore, several approaches suffer from high computational complexity, centralized training, and a lack of counterfactual or explainable AI (XAI) integration.

In contrast, the proposed work moves beyond prediction by integrating a lightweight SVM model within an Agentic AI framework capable of generating rule-based prescriptive outputs such as alerts, rerouting decisions, and cooperative broadcasts through 5G-MEC. It uniquely employs counterfactual analysis to interpret and adjust sensor features for latency mitigation, ensuring real-time vehicle coordination, explainability, and low computational overhead suitable for deployment on edge devices.

5. Conclusion

5.1. Recapitulation

This study was able to create and test a new Agentic AI Coordination Framework that can fill this gap that exists between passive latency prediction and real-time, prescriptive action in 5G-MEC vehicular networks. This framework, in contrast to most of the existing literature that is confined to high-accuracy prediction or generic system optimization, illustrates an end-to-end closed-loop solution, which converts instantaneous latency threats to coordinated and cooperative mitigations. The preliminary stage of the work created an efficient predictive core using the lightweight Support Vector Machine (SVM) model and a newly developed 5G-V2N communication dataset, and obtained the best results at classifying the Latency_Category (High, Low, or Medium). This low-cost method is particularly intended to get around the computational limits and centralized training demands of detailed Graph Neural Networks (GCNs) and other deep learning models found in the literature. In particular, the SVM model had a high overall classification accuracy of 97% and weighted precision of 97% and weighted F1-score of 97% which provides a very strong basis on which the further Agentic AI decisions will be made. Importantly, the model was shown to perform perfectly (F1-score of 1.00) on the critical High Latency (Class 0) prediction, which guarantees the highest degree of reliability of the state that is most required in terms of safety.

The estimated latency classification was then the stimulus of the Agentic AI Module that is the heart of the innovation as it offers prescriptive, vehicle-level decisioning that did not exist in the former literature. The use of simulation results proved that the module had the potential to generate instantaneous driver alerts, as shown in all three latency categories, resulting in a real-time dashboard release that includes a Status Alert, Action Plan, System Suggestions and an Optimized Path, indicating the driver what to do now to reduce the risk. Moreover, the module allows collaborative rerouting with the help of ultra-low latency of the 5G-MEC and broadcasting a custom alert message to neighboring vehicles to support the distributed control of traffic to avoid congestion on a regional level. To achieve transparency, the framework incorporates the LIME (Local Interpretable Model-agnostic Explanations) methodology that allows human-interpreting justification of each prediction, to promote user trust which is a requirement of the safety-critical application. The system most notably, offers Actionable Counterfactual Analysis generating concrete situations which are descriptions of how a vehicle can change its present high-risk latency condition (High to Low/Medium) or how other state transitions (Low to High/Medium, Medium to Low/High) can be facilitated by modifications of certain sensor-read independent features, providing a straight forward path to the elimination of risk or care.

5.2. Future studies

This end-to-end architecture explicitly resolves the major weaknesses of the present research space by addressing the constraints of prediction-only frameworks by introducing a rule-based prescriptive layer, mitigating the shortcomings of centralized planning and complicated GCN/LSTM framework by deploying a lean predictor and decision-engine at the vehicle edge, and addressing the lack of interpretability by optimization-driven works by linking every decision to XAI and Counterfactual Analysis. The effective combination of lightweight classification, rule-based prescriptive action, and 5G-MEC cooperative broadcasting offers a strong solution to the next generation of vehicular networks with the potential to achieve significant safety and efficiency increases compared to the existing systems based on a single approach or the combination of prediction and reaction as a static system to the environment. Future studies will focus on the extension to capture the dynamic resource slicing and task offloading mechanisms based on the forecasted category of latency that will entail incorporation of adaptive multi-agent reinforcement learning methods to further refine the prescriptive rules and optimize resource utilization across various MEC servers in a decentralized fashion.

CRediT authorship contribution statement

Sheikh Muhammad Saqib: Methodology, Funding acquisition, Conceptualization. **Tehseen Mazhar:** Writing – review & editing, Writing – original draft, Visualization, Validation. **Muhammad Usman Tariq:** Resources, Investigation, Formal analysis, Data curation. **Tariq Shahzad:** Resources, Investigation, Data curation. **Asem Ibrahim AlAlwan:** Writing – original draft, Validation, Resources. **Habib Hamam:** Methodology, Funding acquisition, Formal analysis, Conceptualization.

Funding

No.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] K. Ali, et al., Raven: vision-based connected vehicle safety platform using infrastructure sensing, 5G, and MEC, IEEE Trans. Veh. Technol. 73 (11) (2024) 17290–17304, https://doi.org/10.1109/TVT.2024.3430815.

[2] P.K. Singh, S.K. Nandi, S. Nandi, A tutorial survey on vehicular communication state of the art, and future research directions, Veh. Commun. 18 (2019), https://doi.org/10.1016/j.vehcom.2019.100164.

[3] A. Almutairi, A.F. Al Asmari, F. Alanazi, T. Alqubaysi, A. Armghan, Deep learning based predictive models for real time accident prevention in autonomous vehicle networks, Sci. Rep. 15 (1) (2025), https://doi.org/10.1038/s41598-025-04867-8.
[4] R. Shrestha, S.Y. Nam, R. Bajracharya, S. Kim, Evolution of v2x communication and integration of blockchain for security enhancements, Electron 9 (9) (2020) 1–33, https://doi.org/10.3390/electronics9091338.
[5] S. Lokesh, R. Selvameena, T. Babu, A.S. George, Intelligent deep learning architectures for real-time traffic analysis and decision support in IoT-Enabled autonomous vehicles, AI's role Enhanc. Automot. Saf. (2025) 143–158, https://doi.org/10.4018/979-8-3373-0442-7.ch010.
[6] S. Khan, et al., Deep learning model for efficient traffic forecasting in intelligent transportation systems, Neural Comput. Appl. 37 (20) (2025) 14673–14686, https://doi.org/10.1007/s00521-024-10537-z.
[7] S.S. Chauhan, J. Singh, G.S. Gupta, M.K. Pathak, Deep learning for traffic prediction in VANET, Stud. Comput. Intell. 1207 (2025) 139–175, https://doi.org/10.1007/978-981-96-5190-0_7.
[8] M.U. Tariq, et al., Edge-enabled smart agriculture framework: integrating IoT, lightweight deep learning, and agentic AI for context-aware farming, Results Eng. (2025) 107342, https://doi.org/10.1016/j.rineng.2025.107342.
[9] R. Mahaveerakannan, et al., Information centric networking based cooperative caching framework for 5G communication systems, Comput. Mater. Continua (CMC) 78 (2024), https://doi.org/10.32604/cmc.2024.051611.
[10] A. Zhang, Dynamic graph convolutional networks with temporal representation learning for traffic flow prediction, Sci. Rep. 15 (1) (2025), https://doi.org/10.1038/s41598-025-01696-7.
[11] J. Gan, Spatiotemporal interactive dynamic synchronous graph convolution network for traffic flow forecasting, J. Intell. Transp. Syst. (2025), https://doi.org/10.1080/15472450.2025.2519262.
[12] G. Balram, et al., MEDALS: a sustainable AI framework for energy-efficient routing in 5G vehicular networks, Sustainable Computing: Informatics and Systems 48 (2025) 101210, https://doi.org/10.1016/j.suscom.2025.101210. ISSN 2210-5379, 2025.
[13] J. Wang, K. Liu, H. Li, LSTM-based graph attention network for vehicle trajectory prediction, Comput. Netw. 248 (2024), https://doi.org/10.1016/j.comnet.2024.110477.
[14] B. Remmouche, D. Boukraa, A. Zakharova, T. Bouwmans, M. Taffar, Long-term spatio-temporal graph attention network for traffic forecasting, Expert Syst. Appl. 288 (2025), https://doi.org/10.1016/j.eswa.2025.128244.
[15] R. Barbosa, P. Georgieva, S. Sargento, P. Rito, M. Araujo, A. Goes, Forecasting and improving latency in 5G V2X networks for autonomous driving scenarios. 2024 IEEE 22nd Mediterr, Electrotech. Conf. MELECON 2024, 2024, pp. 838–843, https://doi.org/10.1109/MELECON56669.2024.10608556.
[16] M. Patil, Q. Ahmed, S. Midlam-Mohler, Urban traffic forecasting with integrated travel time and data availability in a conformal graph neural network framework. IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC, 2024, pp. 2482–2487, https://doi.org/10.1109/ITSC58415.2024.10920016.
[17] P. Parastar, G. Caso, J.A.O. Iglesias, A. Lutu, O. Alay, Rethinking the mobile edge for vehicular services, Comput. Netw. 253 (2024), https://doi.org/10.1016/j.comnet.2024.110687.
[18] N. Alzaben, et al., DEEPCO-RIS: joint BD-RIS and hybrid NOMA/OMA optimization for energy-efficient vehicular networks, Sustainable Computing: Informatics and Systems 47 (2025) 101145, https://doi.org/10.1016/j.suscom.2025.101145. ISSN 2210-5379, 2025.
[19] L. Han, et al., Energy-efficient cloud-edge collaborative model integrating digital twins and machine learning for scalable and adaptive distributed networks, Sustainable Computing: Informatics and Systems 47 (2025) 101157, https://doi.org/10.1016/j.suscom.2025.101157.
[20] M. Wang, et al., Smart City transportation: a VANET edge computing model to minimize latency and delay utilizing 5G network, J. Grid Comput. 22 (1) (2024), https://doi.org/10.1007/s10723-024-09747-5.
[21] Y. Xiao, J. Zhao, Q. Zhang, Y. Huang, H. Quan, L. Fan, MEC-enabled resource allocation in internet of vehicles, Phys. Commun. 65 (2024), https://doi.org/10.1016/j.phycom.2024.102368.
[22] A. Armat, D. Thankachan, ML-driven latency optimization for mobile edge computing in fiber-wireless access networks, MethodsX 15 (2025), https://doi.org/10.1016/j.mex.2025.103594.
[23] P.B. Angelo Feraudo, Alessandro Calvio, Simulating and validating vehicular cloud computing applications in MEC-enabled 5G environments, J. Simul. Eng. 4 (2025).
[24] H. Xiao, B. Zou, J. Xiao, Graph convolution networks based on adaptive spatiotemporal attention for traffic flow forecasting, Sci. Rep. 15 (1) (2025), https://doi.org/10.1038/s41598-025-88706-w.
[25] M.L. Francies, A.T. Khalil, H.M. Amer, M.M. Ata, A framework for continual learning in real-time traffic forecasting utilizing spatial–temporal graph convolutional recurrent networks, Complex Intell. Syst. 11 (10) (2025), https://doi.org/10.1007/s40747-025-02049-7.