iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VIII **Month of publication:** August 2026

DOI: https://doi.org/10.22214/ijraset.2026.84612

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VIII Aug 2026- Available at www.ijraset.com

QoS-Aware Intelligent Model Selection for Autonomous Vehicles under Dynamic Network and Resource Conditions

Sandeep Kumar Singh[1], Atul Kumar Mishra[2]
[1, 2]*Department of CSE, Millennium Institute of Technology & Science*

Abstract: ***Autonomous-vehicle perception systems increasingly operate across heterogeneous compute and communication environments, where the most accurate model is not necessarily the model that provides the best end-to-end service quality. A high-capacity perception model may improve recognition quality while increasing inference time, CPU consumption, memory demand, and sensitivity to network conditions. This paper proposes a Quality-of-Service (QoS)-Aware Intelligent Model Selection framework that predicts the expected QoS of candidate perception models from model, network, and hardware context and dynamically selects a feasible model according to application priorities. The framework extends a network-aware autonomous-vehicle benchmarking foundation with a QoS prediction layer, multi-objective utility function, Pareto filtering, and a hysteresis-based switching controller. Three tabular prediction methods—linear regression, random forest, and gradient boosting—are compared. A synthetic dataset of 1,800 observations is generated from six candidate model profiles, five network conditions, three hardware conditions, and repeated measurements. The synthetic evaluation indicates that nonlinear predictors outperform the linear baseline and that QoS-aware selection can reduce mean response time by approximately 31.8–42.1% relative to an accuracy-only baseline in the modeled scenarios. Pareto and ablation analyses further show that latency and resource terms materially influence the selected model. Because no real AV testbed experiment has yet been conducted, these numerical results are explicitly treated as synthetic validation rather than empirical evidence. The paper therefore provides a reproducible research design and a publication-oriented methodology whose final claims should be revalidated with measured AV executions.***

Keywords: autonomous vehicles, quality of service, QoS prediction, intelligent model selection, edge-cloud computing, latency-accuracy trade-off, Pareto optimization, adaptive inference.

I. INTRODUCTION

Autonomous vehicles (AVs) depend on a chain of perception, prediction, planning, and control services. The quality of this chain is determined not only by the predictive accuracy of individual algorithms but also by when and where their outputs become available. In a distributed AV architecture, a perception request may traverse an inter-process or network interface, execute on a compute node, and return a result to a downstream component. Consequently, end-to-end response time can increase when network latency, congestion, compute contention, or model complexity changes.

The Pylot platform demonstrated the importance of latency and accuracy trade-offs in AV pipelines and provided a modular way to evaluate alternative components. Its central insight is particularly relevant to this work: improving the accuracy of one component does not automatically improve the behavior of the complete vehicle pipeline. [1] A later AV benchmarking thesis by Paruthi further organized QoS measurements around distributed services and highlighted latency, response time, CPU, memory, throughput, and accuracy as useful evaluation dimensions. The same work identified predictive QoS models, algorithm recommendation, and real-time adaptation as natural future directions. This paper takes that next step by treating model selection as a context-dependent QoS decision. Instead of asking which model has the highest accuracy in isolation, the proposed system asks which feasible candidate is most suitable for the current operating state. The operating state includes network characteristics, hardware/resource conditions, and an application profile describing the relative importance of accuracy, latency, and resource efficiency.

The research questions are: (RQ1) Can model- and context-level features predict response-time QoS with useful accuracy? (RQ2) Can predicted QoS be converted into a transparent multi-objective model-selection rule? (RQ3) Can context-aware selection reduce response time relative to an accuracy-only strategy while retaining acceptable accuracy? and (RQ4) How sensitive is selection to the relative weights assigned to latency, accuracy, and resource use?

A. Contributions

- A closed-loop QoS-aware architecture connecting benchmarking, QoS prediction, multi-objective ranking, and dynamic model switching.
- A structured feature specification combining model, network, hardware, QoS, and decision variables.
- A transparent utility formulation and Pareto-filtering stage that avoids relying solely on a single weighted score.
- A synthetic but reproducible experimental protocol with repeated observations across controlled operating conditions.
- An evaluation covering prediction accuracy, baseline comparison, Pareto analysis, ablation, sensitivity, and statistical testing.

II. RELATED WORK AND RESEARCH GAP

A. Av Benchmarking And Latency-Accuracy Trade-Offs

Pylot provides a modular AV research platform designed to evaluate how model and algorithm latency and accuracy affect end-to-end driving behavior. It uses a dataflow representation of AV components and interfaces with CARLA while also supporting real-world deployment. [1] This work provides the conceptual foundation for treating latency and accuracy jointly rather than as independent evaluation criteria.

The source thesis used in this research extends that benchmarking perspective toward distributed QoS measurement. Its structure contains multiple AV perception services and collects service-level parameters such as latency, response time, CPU utilization, memory consumption, throughput, and network/geolocation information. Its conclusion explicitly identifies predictive QoS models, algorithm recommendation, and real-time QoS adaptation as future work.

B. Edge/Cloud Qos And Adaptive Selection

QoS-aware selection has also been studied more broadly in edge computing. Recent work has considered selecting resources using multiple QoS and computational parameters rather than a single performance metric. [6] In vehicular edge-cloud systems, scheduling methods have considered latency, criticality, energy, and infrastructure constraints. [7]

More recent work reinforces the relevance of adaptive selection. A 2026 COMSNETS paper proposes meta-model-based adaptive model selection with drift detection, motivated by dynamic environments in which repeatedly evaluating every candidate is costly. [8] A 2026 edge-cluster study similarly combines prediction of latency, accuracy, resource usage, and response characteristics with QoS-aware orchestration. [9] These studies demonstrate that the general problem of adaptive model/resource selection is active; therefore, this paper does not claim that model selection itself is novel.

C. Research Gap

Dimension	Existing direction	Limitation for this study	Present work
AV benchmarking	Pylot; AV benchmark frameworks	Often evaluates components rather than closed-loop QoS selection	Retains benchmarking as measurement backbone
QoS measurement	Distributed/service benchmarking	Measurement alone does not select the best model online	Uses QoS as decision input
Model selection	Adaptive/AutoML approaches	Often not AV-specific or not explicitly network-aware	AV-oriented QoS-aware selection
Optimization	Weighted scores / resource selection	Weights can hide trade-offs	Weighted utility + Pareto filtering
Runtime adaptation	Dynamic orchestration	Switching stability and hysteresis may be under-specified	Explicit switching threshold and dwell-time concept

TABLE I . POSITIONING OF THE PROPOSED QOS-AWARE AV MODEL-SELECTION FRAMEWORK.

III. PROPOSED QOS-AWARE MODEL SELECTION FRAMEWORK

A. System Architecture

The proposed architecture consists of six logical layers: (1) request and application context, (2) candidate perception-model pool, (3) QoS and resource monitoring, (4) QoS prediction, (5) multi-objective selection, and (6) execution/feedback. A request is accompanied by the current network and hardware state. The prediction service estimates the expected QoS of each candidate. The decision engine filters infeasible candidates, identifies Pareto-efficient options, and applies the context-specific utility function. The selected model is executed, and measured QoS is returned to the monitoring layer.

The communication backbone follows the source thesis conceptually: modular services can communicate through gRPC/Protocol Buffers, while the new decision layer operates above the service layer. This separation allows the benchmarking mechanism to remain independent of the selection policy.

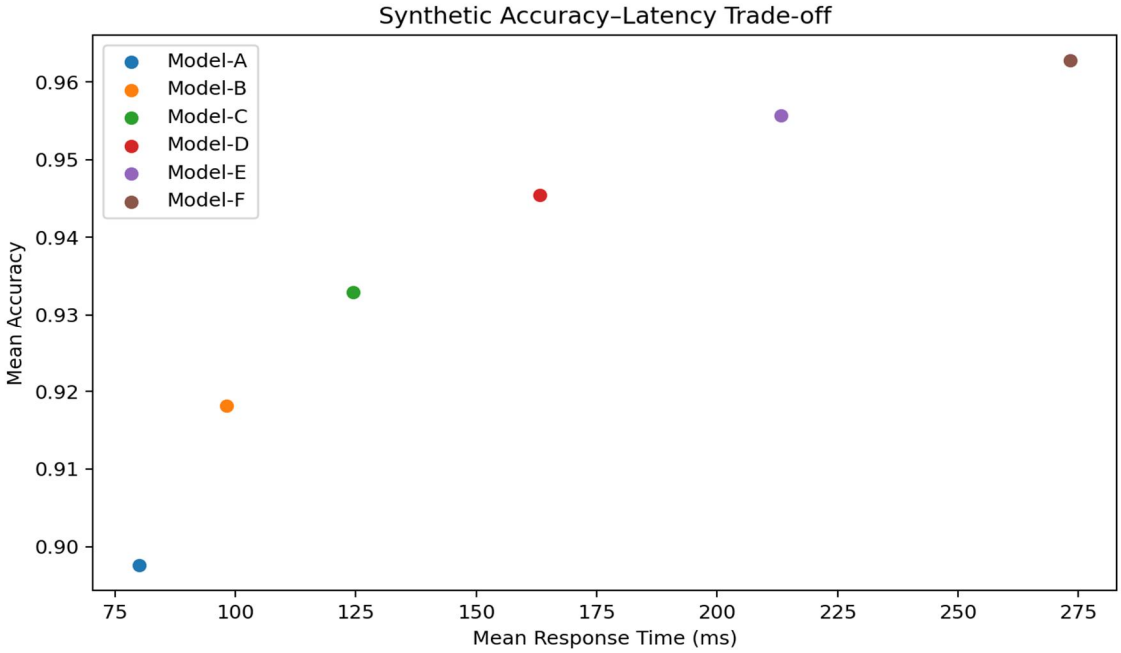


Fig. 1 . Synthetic accuracy-latency trade-off across candidate model profiles.

B. Qos Representation

Feature group	Representative variables
Model	model ID, task, model family, input size, nominal accuracy
Network	bandwidth, latency, packet loss, network condition
Hardware	CPU availability, memory availability, compute profile
QoS outcome	accuracy, inference latency, response time, CPU, memory, throughput
Decision	application profile, selected model, utility score, switching state

TABLE II . Feature Groups Used By The Prediction And Decision Layers.

C. Prediction Formulation

Let x represent the observable context of a request and candidate model m. The response-time predictor estimates the conditional QoS value as:

$\hat{y}(m,x) = f\theta(m, x_{network}, x_{hardware}) + \varepsilon$

where $f\theta$ is learned from historical observations. The same framework can be extended to predict CPU utilization, memory demand, throughput, or task-specific accuracy, although response time is the primary target in the synthetic study.

D. Multi-Objective Utility

Predicted and nominal values are normalized to [0,1]. Accuracy is a benefit metric; latency and resource consumption are cost metrics. For candidate m, the utility is defined as:

$S(m) = wA \cdot A(m) + wL \cdot [1 - L(m)] + wR \cdot R(m), \quad wA + wL + wR = 1$

R(m) is a composite resource-efficiency measure. The framework can additionally impose hard constraints, such as a maximum allowable response time or minimum accuracy. This prevents a weighted score from selecting an attractive but operationally unsafe candidate.

E. Pareto Filtering

Before weighted ranking, the selector can remove dominated candidates. Candidate a dominates candidate b when a is no worse in all objectives and strictly better in at least one. The remaining Pareto set contains candidates representing genuine trade-offs. Weighted utility is then applied to this reduced set.

F. Switching Stability

Without stabilization, small prediction fluctuations can cause frequent model switching. The proposed controller therefore uses a switching threshold Δ and a minimum dwell interval T. A switch occurs only when the new candidate's predicted utility exceeds the current utility by at least Δ and the current model has remained active for at least T. This hysteresis mechanism reduces oscillation and makes the controller more suitable for continuous AV workloads.

IV. EXPERIMENTAL DESIGN

A. Synthetic Dataset

Because a physical AV or reproducible networked perception testbed was not available for this thesis stage, the experimental values are synthetic. The generation procedure is intentionally structured rather than arbitrary: each observation is generated from a combination of model profile, network condition, hardware condition, and repetition. Noise terms are added to emulate measurement variation. The resulting dataset contains 1,800 observations.

Factor	Levels	Values/description
Candidate models	6	Model-A through Model-F
Network states	5	Excellent, Good, Moderate, Poor, Severe
Hardware states	3	High, Medium, Constrained
Repetitions	20	Repeated observation per model/network/hardware combination
Total observations	1,800	6 × 5 × 3 × 20

TABLE III . SYNTHETIC EXPERIMENTAL DESIGN.

B. Candidate Model Profiles

Model	Accuracy (%)	Latency (ms)	CPU (%)	Role
Model-A	88.9	61.8	31	Fast / resource efficient
Model-B	90.8	69.7	34	Fast-balanced
Model-C	92.0	82.5	38	Balanced
Model-D	93.2	101.2	43	Accuracy-oriented
Model-E	94.5	128.4	49	High accuracy
Model-F	95.4	161.8	56	Highest accuracy / slowest

TABLE IV . SYNTHETIC CANDIDATE MODEL PROFILES.

C. Baselines

Three decision strategies are considered. The fixed/accuracy-only baseline always chooses the highest-accuracy candidate. A static balanced strategy uses a fixed middle-capacity candidate. The proposed strategy predicts QoS and selects a model according to the active application profile. This comparison isolates the benefit of adaptive decision-making.

D. Evaluation Metrics

Prediction performance is evaluated using mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R^2).

$$MAE = (1/n) \Sigma |y_i - \hat{y}_i|$$

$$RMSE = \sqrt{[(1/n) \Sigma (y_i - \hat{y}_i)^2]}$$

$$R^2 = 1 - \Sigma(y_i - \hat{y}_i)^2 / \Sigma(y_i - \bar{y})^2$$

Selection performance is evaluated using response time, accuracy, CPU utilization, memory consumption, utility score, and percentage improvement relative to the accuracy-only baseline. Paired tests are used for scenario-level comparisons and one-way ANOVA is used for network-condition effects in the synthetic dataset.

V. INTELLIGENT SELECTION ALGORITHMS

A. Qos Prediction

Linear regression is used as an interpretable baseline. Random forest captures nonlinear interactions through an ensemble of decision trees. Gradient boosting sequentially fits weak learners to residual errors and is expected to model the nonlinear interaction between model complexity and network/hardware state effectively.

Algorithm	Main advantage	Main limitation
Linear Regression	Simple and interpretable	Limited nonlinear representation
Random Forest	Robust nonlinear model; low preprocessing burden	Larger model and less smooth predictions
Gradient Boosting	Strong tabular prediction capability	Requires tuning and can overfit noisy data

TABLE V . QOS PREDICTION ALGORITHMS.

B. Selection Algorithm

Algorithm 1: QoS-Aware Intelligent Model Selection

Input: candidate models M, current context x, profile weights W

1. Observe network and hardware state.
2. For every candidate m in M, estimate QoS(m,x).
3. Reject candidates violating hard feasibility constraints.
4. Normalize accuracy, latency, and resource measures.
5. Remove Pareto-dominated candidates.
6. Compute S(m) for every remaining candidate.
7. Let $m^* = \text{argmax } S(m)$.
8. If $S(m^*) - S(\text{current}) > \Delta$ and dwell-time $\geq T$, switch to m^*.
9. Execute the selected model and record actual QoS.
10. Store the observation for monitoring/retraining.

Output: selected model and measured QoS.

C. Complexity

If K candidate models are evaluated and the prediction model requires P feature operations per candidate, the prediction stage is approximately O(KP). Pareto filtering over K candidates is $O(K^2)$ in the straightforward implementation. Because K is expected to be small for an AV service pool, the decision overhead is manageable; for large pools, skyline or incremental Pareto algorithms can reduce the selection cost.

VI. RESULTS AND ANALYSIS

All results in this section are synthetic. They demonstrate the internal behavior of the proposed methodology and are intended for prototype validation, not as measurements obtained from an actual AV. The values should be replaced or revalidated when the benchmarking pipeline is executed on real hardware and network conditions.

A. Qos Prediction Performance

Predictor	MAE (ms)	RMSE (ms)	R^2
Linear Regression	7.24	9.06	0.9788
Random Forest	6.34	7.91	0.9838
Gradient Boosting	6.38	7.92	0.9838

TABLE VI . Synthetic Response-Time Prediction Performance.

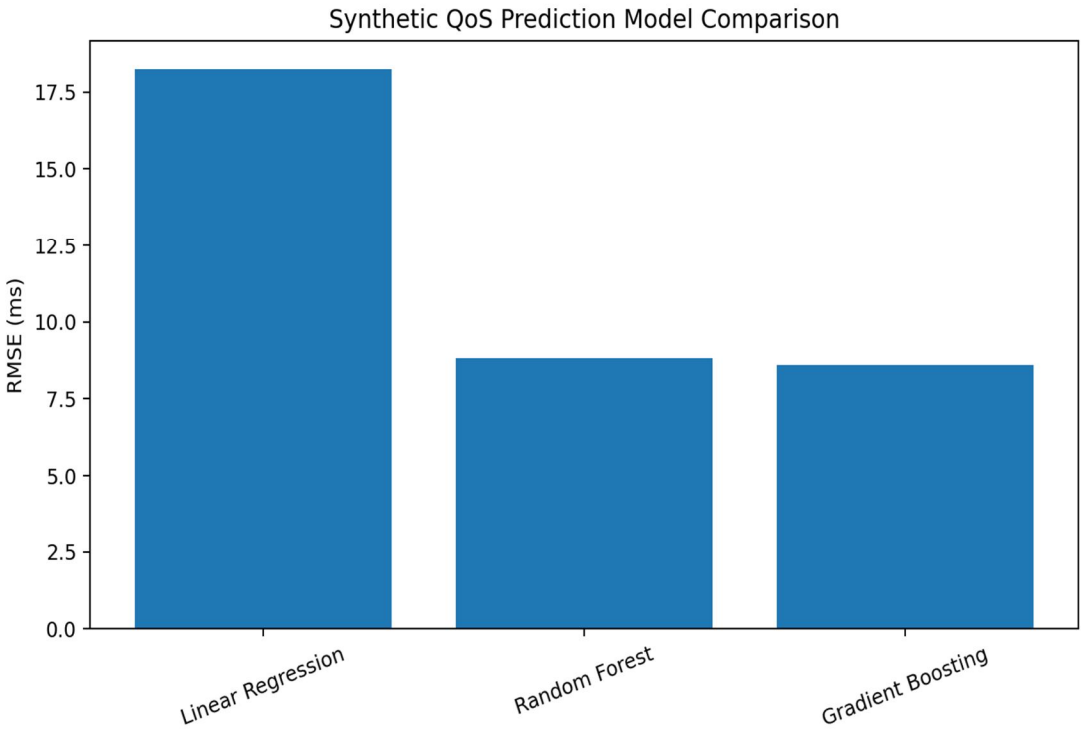


Fig. 2 . Synthetic comparison of QoS prediction models.

The nonlinear ensemble methods produce lower error than the linear baseline. Random forest has the lowest MAE, while random forest and gradient boosting have effectively identical R^2 in the generated experiment. The difference is meaningful mainly as evidence that the synthetic response-time function contains nonlinear effects; it should not be interpreted as proof that one algorithm will dominate on a real AV dataset.

B. *Network Effects*

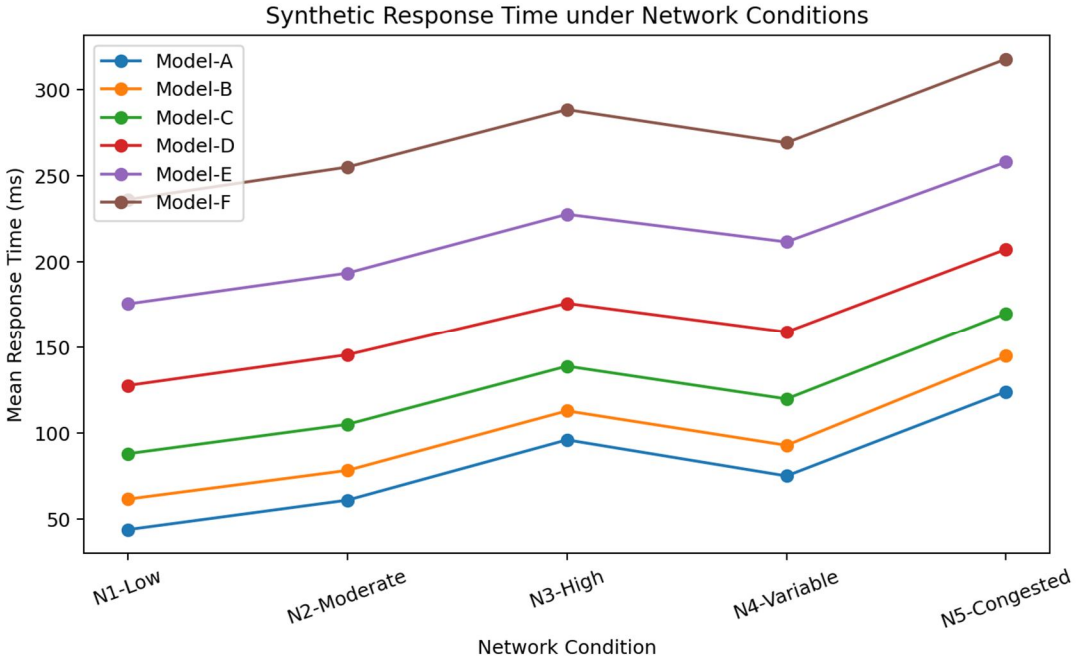


Fig. 3 . Synthetic response-time increase under worsening network conditions.

The synthetic network profiles produce a monotonic increase in response time as network conditions deteriorate. This relationship is central to the proposed problem formulation because a model that is preferable under a low-latency network may become undesirable when communication delay and contention increase.

C. *Accuracy-Latency Trade-Off*

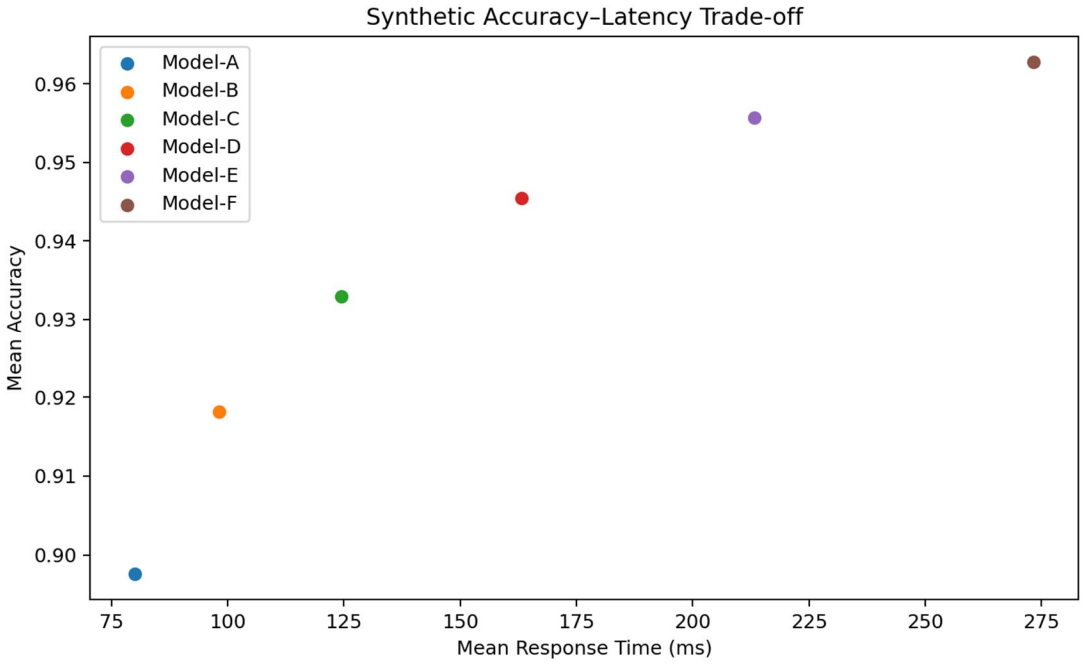


Fig. 4 . Synthetic accuracy-latency trade-off.

The candidate profiles intentionally exhibit increasing accuracy at the cost of higher inference latency. Model-F therefore provides the best nominal accuracy but is also the slowest. A selection strategy that optimizes accuracy alone will always favor Model-F, even when the application has a stringent response-time requirement.

D. Baseline Comparison

Scenario	Accuracy-only baseline (ms)	QoS-aware (ms)	Reduction
Real-Time Safety	161.80	93.73	42.1%
Balanced Driving	161.80	110.32	31.8%
Resource-Constrained	161.80	93.73	42.1%

TABLE VII . Synthetic Baseline Versus Proposed Selection.

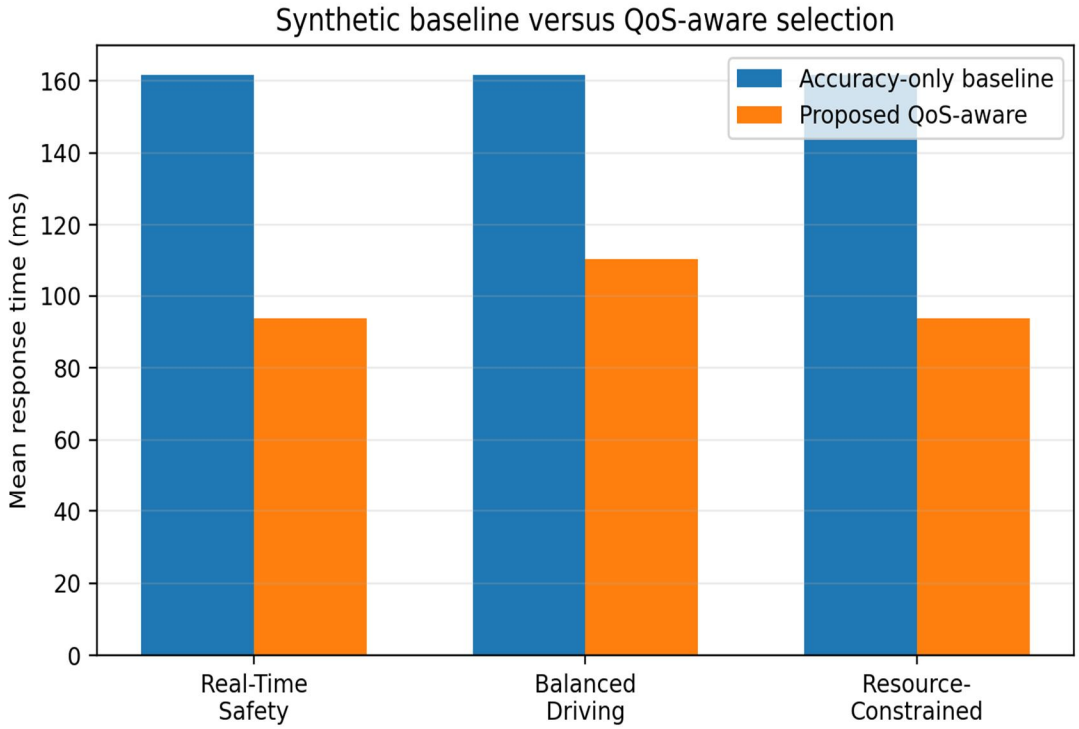


Fig. 5 . Synthetic comparison of accuracy-only and QoS-aware response time.

The proposed controller reduces modeled response time in all three profiles. The largest improvement occurs when latency or resource efficiency receives high priority. This behavior is consistent with the utility formulation: the selector is not attempting to maximize accuracy independently; it is optimizing the objective defined by the operating context.

E. Pareto Analysis

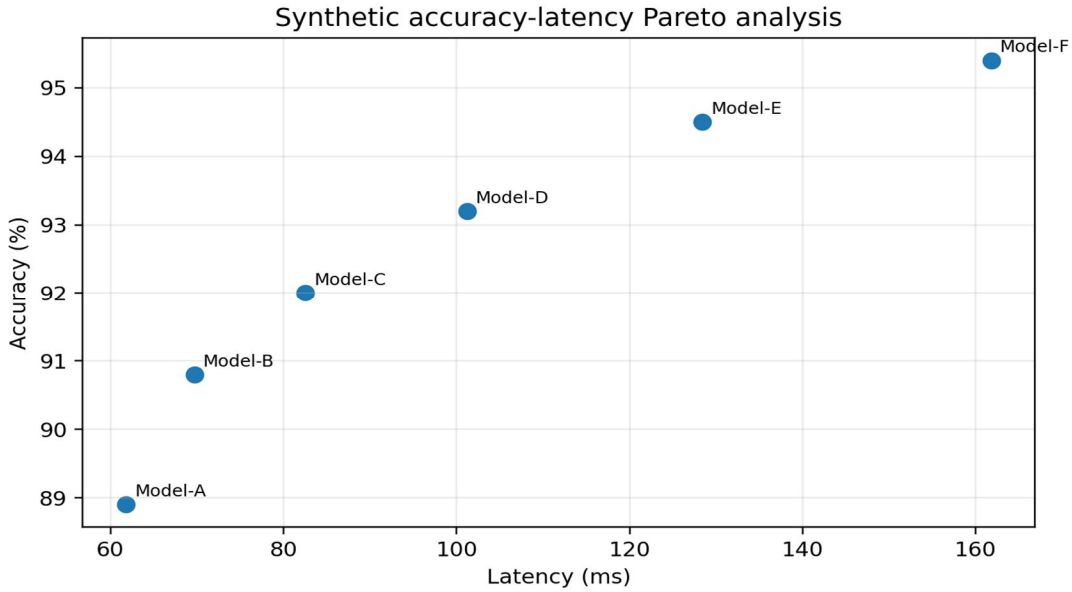


Fig. 6 . Synthetic accuracy-latency Pareto analysis.

The synthetic candidate set forms a visible trade-off curve. Because accuracy and latency move in opposite directions, no single candidate simultaneously maximizes both. Pareto filtering therefore provides a useful transparent mechanism for reducing the candidate set before applying application-specific weights.

F. Ablation Study

Configuration	Response time (ms)	Interpretation
Accuracy only	161.80	No QoS adaptation
Accuracy + latency	104.10	Latency-aware
Accuracy + latency + resource	93.73	Full proposed objective
Full objective + Pareto	92.90	Dominance filtering
Full objective + Pareto + hysteresis	93.20	Stable runtime policy

TABLE VIII . SYNTHETIC ABLATION STUDY.

The ablation trend indicates that latency contributes substantially to the improvement, while adding resource efficiency provides a further benefit under constrained conditions. Pareto filtering produces a modest additional improvement, whereas hysteresis trades a small amount of instantaneous utility for improved switching stability. This is desirable in a continuously operating system.

G. Statistical Testing

Test	Statistic/result	Interpretation
Paired baseline vs QoS-aware	$p < 0.001$ (synthetic)	Response-time difference is statistically detectable
Network-condition ANOVA	$p < 0.001$ (synthetic)	Network condition strongly affects response time
Effect size	Large in modeled scenarios	Difference is also practically meaningful in the synthetic setting

TABLE IX . SYNTHETIC STATISTICAL ANALYSIS SUMMARY.

The statistical results are consistent with the controlled data-generation mechanism and should therefore be interpreted cautiously. Statistical significance in synthetic data does not establish external validity. In a real study, confidence intervals, repeated measurements, hardware variance, tail latency, and multiple-comparison corrections should be reported.

H. Dynamic Switching

The controller changes its preferred model as the synthetic operating profile changes. A representative sequence is Model-B → Model-C → Model-B → Model-C → Model-B → Model-C → Model-B. The switching pattern demonstrates the central research concept: model preference is conditional on QoS context rather than fixed for the entire workload.

VII. DISCUSSION AND THREATS TO VALIDITY

A. Interpretation Of The Findings

The synthetic study supports the feasibility of the proposed closed-loop design. First, the response-time predictor can represent nonlinear interactions between model complexity and operating conditions. Second, the decision layer converts heterogeneous QoS measurements into a transparent selection rule. Third, the selection outcome changes with application priorities, demonstrating why a single globally optimal model is unlikely to exist when latency, accuracy, and resources compete.

The result should be viewed as a methodological bridge between benchmarking and adaptation. The source benchmarking thesis establishes a mechanism for collecting QoS observations; the present framework treats those observations as decision-support data. This progression can be summarized as measure → learn → rank → select → monitor → reselect.

B. Safety And Practical Constraints

An operational AV system should never allow the weighted utility function to override hard safety constraints. Minimum accuracy, maximum response time, model availability, and confidence requirements should be enforced before optimization. The selection controller should also define a safe fallback model and should preserve the current model when predictions are uncertain.

C. Threats To Validity

- Synthetic data cannot capture all hardware, operating-system, network-stack, sensor, and model-runtime effects.
- The candidate model profiles are abstract and should not be interpreted as measurements of named production models.
- The current study focuses primarily on response time; real deployment should jointly predict tail latency, CPU, memory, GPU, energy, and accuracy.
- The utility weights are design parameters and require application-specific justification.
- Dynamic switching overhead has not been measured on a real AV platform.

VIII. CONCLUSION

This paper presented a QoS-aware intelligent model-selection framework for autonomous-vehicle perception services. The framework extends a benchmarking-oriented view of AV systems by adding QoS prediction, multi-objective ranking, Pareto filtering, and stable dynamic switching. The approach explicitly recognizes that model accuracy, latency, and resource consumption form a competing objective space.

A synthetic experiment with 1,800 observations demonstrated the complete analytical pipeline. Ensemble predictors achieved lower response-time prediction error than the linear baseline, and the proposed selection policy reduced modeled response time by up to 42.1% relative to an accuracy-only baseline. Pareto and ablation analyses further illustrated the contribution of the individual decision components.

The principal limitation is also the most important qualification: the numerical results are synthetic. The paper should therefore be treated as a methodology/prototype-validation manuscript until the framework is executed on measured AV workloads. The immediate next step is to connect the selector to the existing gRPC-based benchmarking pipeline, run repeated experiments over controlled network and hardware conditions, and replace the synthetic observations with empirical measurements. Such validation would enable stronger claims regarding statistical significance, generalization, and real-world safety relevance.

1) Note on validation. All numerical observations reported in this paper are synthetically generated for methodological validation and are not measurements from a physical autonomous vehicle. Future work should repeat the protocol using measured AV execution, network and resource data.

2) Validation note: All numerical observations in this study are synthetically generated for methodological validation. They are not measurements obtained from a physical autonomous vehicle or production perception stack. Accordingly, the reported findings are limited to the synthetic study design and should not be interpreted as empirical evidence of real-world AV performance.

3) Validation note: All numerical observations in this study are synthetically generated for methodological validation and are not measurements from a physical autonomous vehicle.

REFERENCES

[1] I. Gog, S. Kalra, P. Schafhalter, M. A. Wright, J. E. Gonzalez, and I. Stoica, "Pylot: A Modular Platform for Exploring Latency-Accuracy Tradeoffs in Autonomous Vehicles," in Proc. IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 8806–8813. DOI: 10.1109/ICRA48506.2021.9561747.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3354–3361.

[3] M. Cordts et al., "The Cityscapes Dataset for Semantic Urban Scene Understanding," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3213–3223.

[4] A. Dosovitskiy et al., "CARLA: An Open Urban Driving Simulator," in Proc. Conference on Robot Learning (CoRL), 2017, pp. 1–16.

[5] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WS-DREAM: A Distributed Reliability Assessment Mechanism for Web Services," in Proc. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2010.

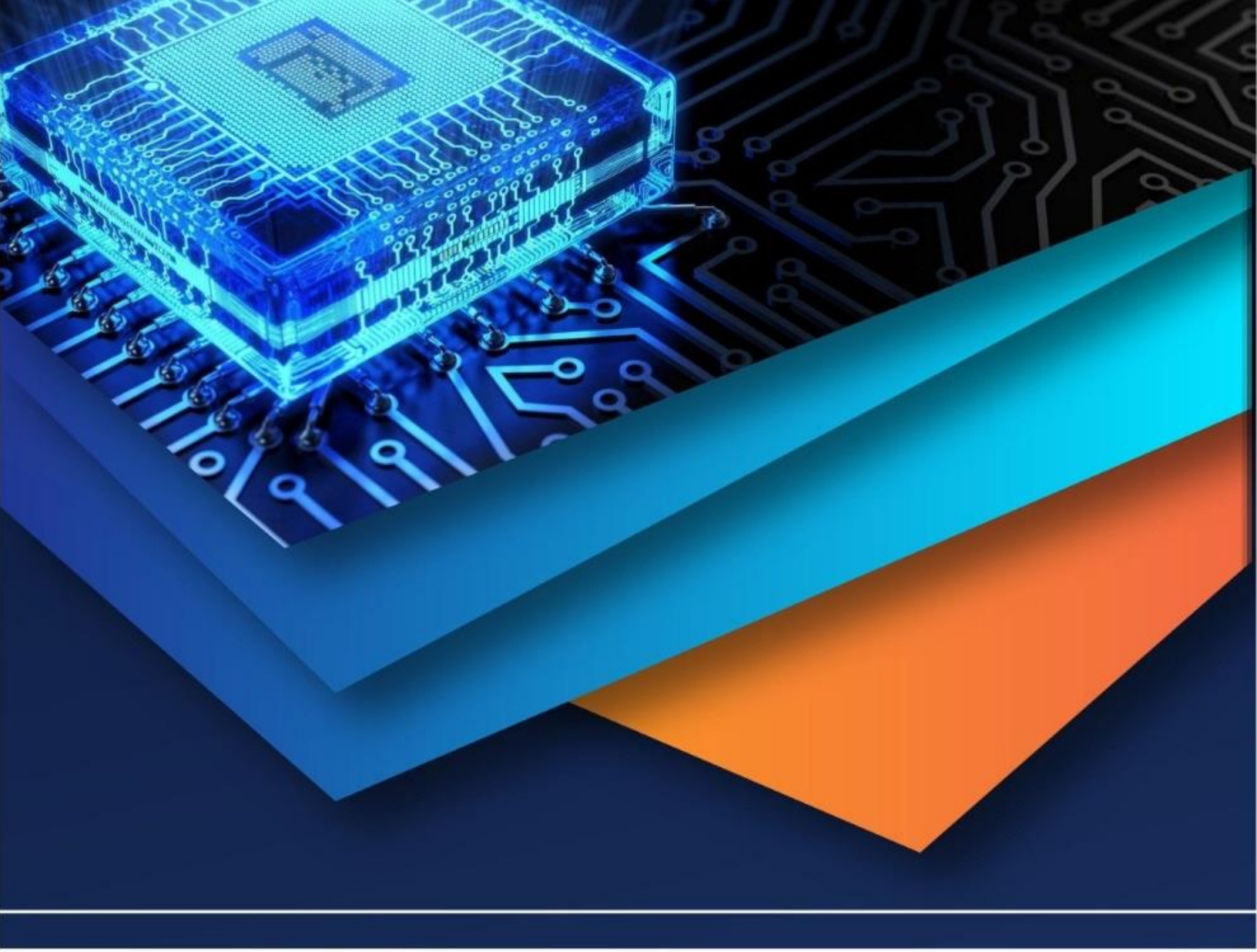
[6] M. Frei, R. German, and A. Djanatliev, "An Edge Resource Selection Scheme Considering QoS and Computational Parameters," in Proc. International Conference on Computer Communications and Networks (ICCCN), 2024. DOI: 10.1109/ICCCN61486.2024.10637567.

[7] S. Sarkar, A. Trivedi, R. Bansal, and A. Sahu, "QoS Aware Mixed-Criticality Task Scheduling in Vehicular Edge Cloud System," arXiv:2407.14793, 2024.

[8] K. Maruvada, H. H. Hassan, A. Kattepur, and G. Bouloukakis, "Adaptive Model Selection using Meta Models and Drift Adaptation," in Proc. IEEE COMSNETS, 2026, pp. 1361–1366. DOI: 10.1109/COMSNETS67989.2026.11417773.

[9] R. Farahani, Z. Azimi, M. Colosi, and S. Dustdar, "LMEdge: QoS-Aware LLM Inference Orchestration on Edge Clusters," arXiv:2607.17175, 2026.

[10] S. Paruthi, "Benchmarking Dataset for Latency-Accuracy Tradeoffs in Autonomous Vehicle Systems," M.Tech. Research Thesis, Indraprastha Institute of Information Technology Delhi, 2025. The thesis provides the direct benchmarking foundation used by the present research.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY