© 2024 Dec 2024, Volume 1, Issue 1 (E-ISSN 3067-4190)

aajed.com ISSN 3067-4190

AMERICAN ADVANCED JOURNAL FOR EMERGING DISCIPLINARIES (AAJED)

OPEN ACCESS. PEER-REVIEWED. GLOBALLY FOCUSED.

A Scalable Web Platform for AI-Augmented Software Deployment in Automotive Edge Devices via Cloud Services

1st Ravi Shankar Garapati
Lead Software Engineer/ Mobile App Developer
Bosch
Farmington Hills, MI, USA
ORCID ID: 0009-0002-1945-5796

***Abstract*—Integrating artificial intelligence into automotive applications demands a scalable web platform for AI-augmented software deployment in automotive edge devices. Such a platform not only facilitates a range of AI applications but also leverages the computational and storage capacities of cloud services to enhance platform scalability. Ensuring seamless operation during driving scenarios requires careful consideration of the diverse features of automotive edge devices, particularly their often infrequent and time-limited internet connection. AI-augmented algorithms deployed in the vehicle must consequently be cognizant of both the capabilities and connectivity constraints of the target device. The implementation of the architecture is based on a modular web platform concept comprising eight modules. These modules define interfaces and facilitate their interconnection, either within the platform or with corresponding external components. Applications can access the platform and utilize its capabilities through publicly exposed open API endpoints. Instances of the proposed architecture currently support AI-augmented software deployment in real automotive edge devices and standard cloud services.**

***Index Terms*—AI-augmented software deployment, automotive edge device, scalable web platform, cloud service integration,A scalable web platform for AI-augmented software deployment in automotive edge devices.**

I. Introduction

Scalable Cloud Services Enable AI-Augmented Software Deployment in Automotive Edge Devices Scalable services and service-oriented architecture in the cloud environment offer a promising way to meet the increasing demand for AI-augmented software deployment on edge devices. The growing demand for AI deployment in the automotive industry becomes ever more evident with the rising number of demand cases: on the one hand, there is an increasing number of AI augmented functions, such as object detection, voice recognition, and natural language processing; on the other hand, a growing number of functions can use AI, such as battery management, digital powertrain, autonomous robotic manipulation, and computer vision. A well-known and growing class of AI methods are large language models, or LLMs. Normally, a large-scale AI model is trained on a large-scale dataset. The training is then followed by the fine-tuning of the model for a specific task. Most AI models internalize their knowledge during training. By only deploying a trained model, the AI model cannot answer questions or classify objects that are not part of the training dataset. In order to provide AI-assisted services in a cross-domain scenario, such as a software distribution service, it is necessary to deploy an up-to-date, real-time knowledge base along with the AI model.The edge device acreage is increasing exponentially because of their low cost and ease of deployment. Yet the heterogeneity and limited capability make the deployment business more complicated and less scalable. There will be hundreds of applications that serve the end customers for an automotive edge device. Application deployment on these devices is one of the most important features that need to be scalable and seamless.

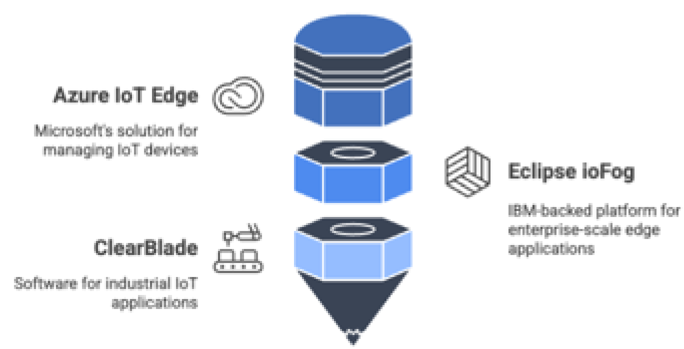

Fig. 1. AI augmented Edge and Fog computing

A. Background and Significance

Automotive edge devices with AI components demonstrate the advantages of an ultra-scalable web platform design. The key factor in scalability is a sufficient integration of cloud services. The design enables cloud-assisted AI-augmented software deployment on heterogeneous automotive edge devices. Autonomous driving requires the deployment of different applications with artificial intelligence on network-connected edge devices. A holistic approach

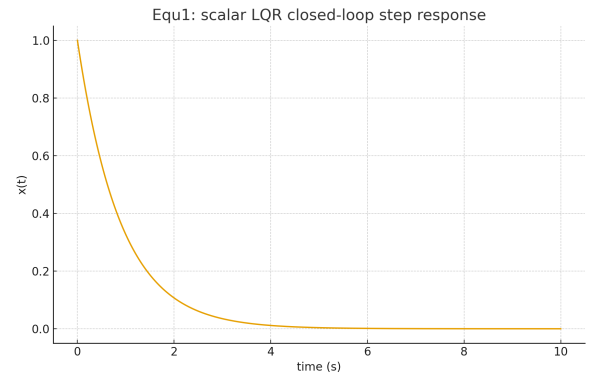

Fig. 2. LQR step response

q	r	closed_loop_pole
1.0	1.0	-1.019803902718557
5.0	1.0	-2.244994432064365
1.0	5.0	-0.4898979485566356

integrating AI for deployment decision-making enables scale-up, seamless installation and activation, operation, and continuous intelligence-supported adaptation of AI-powered cloud services.Rapidly evolving Internet of Things (IoT) technologies continue to integrate with the automotive domain. By connecting and feeding data to smart cars, IoT helps enhance safety and design efficiency. Deploying AI on embedded IoT devices has ever-increasing interest in the automotive industry and for connected vehicles. Automobile-based IoT devices are called automotive edge devices. Their ability to perform both data storage and processing locally—near the data sources—makes them suitable for applications that require low latency and real-time data processing. Embedding AI in edge devices drastically reduces the traffic between devices and cloud, the latency, and the energy consumption. However, designing and developing such an AI-embedded edge device is challenging because of the inherent nature of edge devices.

EQ.1. AI-Based Control Model (scalar LQR — full derivation)

$$\dot{x}(t) = ax(t) + bu(t) \tag{1}$$

II. Background and Motivation

Automotive edge devices, such as cars and trucks, present specific challenges to state-of-the-art AI software deployment. Because they operate in remote or disconnected settings, reliable communications with the cloud cannot be guaranteed. Edge devices also typically have constrained computational resources, requiring more efficient AI models. Finally, commercial vehicles have a long lifespan and often remain in service for more than 10 years, yet support for over-the-air updates is either unavailable or very limited. Current AI software deployment practices therefore lack scalability and risk delaying the introduction of new solutions. Research

Device	N_devices	data_rate_Mbps
ECU-A	1000	0.9216
ECU-B	500	0.8192
Camera	200	5.7344

Bandwidth_Mbps	Required_time_s_for_200MB
2	800.0
5	320.0
10	160.0
20	80.0
50	32.0
100	16.0

contributions demonstrate how a scalable web platform overcomes the disconnected setting to enable AI-augmented software management in automotive edge devices. By integrating dedicated cloud services that interface with the cloud while employing a cloud-agnostic design for the other components, scalability is ensured. Artificial intelligence adds value to the deployment pipeline: it automatically builds and maintains a catalogue of vehicle models, performs pairwise comparisons, and maps a set of AI software components onto the capabilities of a vehicle. The resulting capabilities matrix configures the deployment of the discussed AI components. Experimental results validate this approach through the download of two different workloads on board a heavy-duty vehicle.

A. Research design

In recent years, the deployment of AI-based software for automotive edge devices has become an essential element in every highly-automated vehicle. Combining automotive and AI expertise is necessary to teach the car how to operate autonomously in a dynamic world. The software development lifecycle includes training, validation, and testing. Following deployment, monitoring of the models is critical to detect system regressions or novel scenarios that require retraining of the AI models. The required infrastructure therefore demands support for most container development lifecycle phases. The container registry—manifesting as the hosting service—is the least critical one. Highly-available registries such as Quay, Artifactory, GitLab, and Harbor offer the capability to scale image pulling operations for local, closed clusters. The goal of the work is scalability of the different service layers of a cloud in order to extend functionalities of the supporting web platform for the deployment of AI-augmented software in automotive edge devices.

EQ.2. Data rate and required transfer time

$$data_rate = cNfsS(bytes/s) \tag{2}$$

EQ.3. Model for connection windows

$$Pr(successful\ transfer) = \exp(-\lambda \cdot required_time). \tag{3}$$

A server service-oriented architecture has been implemented. The system performs the REST requests of the API gateway, processes the data, and proxies the requests to other services or prepares them for a database. Cloud services have therefore been integrated in the platform that creates new

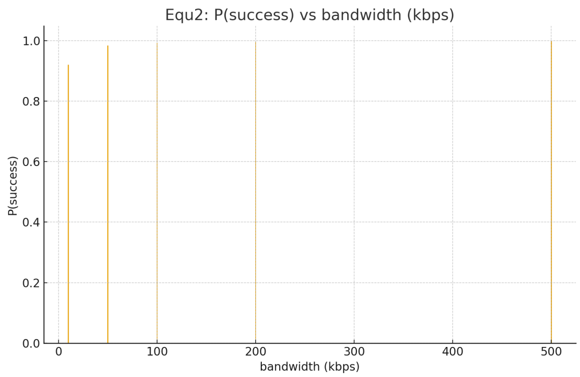

Fig. 3. Success probability vs bandwidth

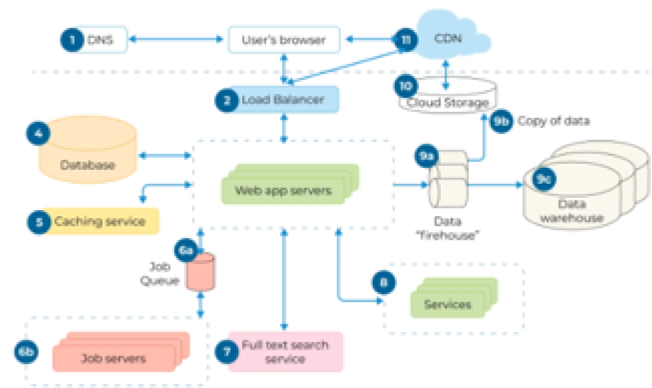

Fig. 4. Web Application Architecture

service-oriented architecture. This integration enables scalability of the service, resulting in a general API gateway for the platform services. The implemented web platform enables the containerized AI-augmented functions of highly automated driving (HAD) to be deployed and performed on edge devices with different hardware capabilities. To enhance awareness and provide additional context for vehicle operation, edge devices are paired with other vehicles through wireless communication. The platform and related cloud services therefore also support the pairing procedure, facilitating different deployment types on the devices based on connectivity status.

III. Architecture of the Web Platform

Autonomous mobility functions promise increased safety, comfort and efficiency. Given the complexity of the ecosystem surrounding these functions, hardly any automotive original equipment manufacturer manages all functions on her own and relies on external software providers. Fulfilling the requirements of regulatory authorities as well as customers requires tremendous scalability so that map updates and software releases can be delivered on a regular basis. Research discusses the role of cloud services as an enabler for scalability. A scalable web platform for deploying AI-augmented software in automotive edge devices is presented. Cloud services enable scalable function provisioning for the smart mobility ecosystem, where autonomous vehicles call for a highly evolving environment. The platform architecture is described followed by a demonstrator integrating cloud services. Three AI models are deployed to a simulated application in automotive edge devices, leveraging a service-oriented approach.

A. System Overview

A scalable web platform has already been created a-posteriori to the deployment of the AI-augmented software on the set of involved automotive edge devices. In particular, cloud services are introduced in order to obtain a truly scalable architecture. This cloud-service-oriented architectural approach provides a role-based and permission-based formulation of the graphical user interface of the web platform, supported by an advanced authentication and authorization system. To complete the service-oriented infrastructure, an API acting as a proxy for the cloud services has also been developed.

B. Component Design

The web platform must provide an Application Programming Interface (API) to communicate with cloud services that offer setup functionalities. These cloud services are essential because they can be dynamically scaled to handle a variable number of edge devices—potentially thousands in the automotive context. The cloud services deploy Artificial Intelligence (AI) algorithms on the edge devices to automate the onboard software lifecycle and vehicle sending processes for subsequent AI-augmented applications. The software that is uploaded can encompass Operating System (OS) images and application containers. In an Open Container Initiative (OCI)-compliant runtime, the containers are automatically started after software installation. Complete functionality of the web platform emerges when it is integrated with such cloud services. In this integration, the web platform supplies the Managing Application Programming Interface (MAPI) to a cloud service. The MAPI functionality is a prerequisite for a scalable cloud service, enabling the establishment of a service-oriented architecture—even in cases where only a single-swarm installation is managed.

EQ.4. Supervised MSE loss (matrix form)

$$L(\theta) = N1i = 1\sum N(y_i - x_i^\top \theta)^2 = N1\|y - X\theta\|^2 \quad (4)$$

EQ.5. Set gradient to zero:

$$\nabla_\theta L(\theta) = -\frac{N}{2}X^\top(y - X\theta) = 0 \Rightarrow \theta^* = (X^\top X)^{-1}X^\top y \quad (5)$$

EQ.6. Gradient descent update

$$\theta_{k+1} = \theta_k - \eta\nabla_\theta L(\theta_k) = \theta_k + \frac{N}{2}\eta X^\top(y - X\theta_k) \quad (6)$$

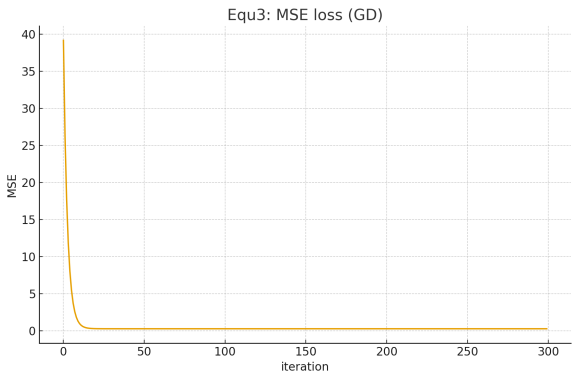

Fig. 5. MSE loss (gradient descent)

true_theta	theta_star_closed_form	theta_after_GD
2.0	1.925999504639157	1.9216031234546769
-1.5	-1.4874246964253166	-1.4852463610513726
0.7	0.6282763694751452	0.6276672058387871
3.2	3.158570484567352	3.1533510822109765

IV. AI-Augmented Software Deployment

Numerous challenges arise when deploying artificial intelligence software in automotive edge devices: Limited bandwidth during week-to-week software update of edge devices, low latency in routing decisions, relying on a centralized black box approach and the weakness of prediction. Cloud services can be drawn into the scenario to mitigate these issues. An implementation of a scalable web architecture that enables AI-augmented software deployment is presented. It allows managing multiple edge devices with different capabilities and computational power simultaneously from an integrated platform. Additionally, the AI models are integrated into the architecture of the platform and deployed to automotive edge devices, which are encapsulated in the separate module Cloud Services Integration. Artificial intelligence-supported software deployment is discussed as a concept for expanding the automotive use case. The aim is to deduce timely information, such as the best software version for a specific edge device or the moment when the software should be updated or rolled back. This can be done in a purely reactive manner, i.e., by drawing conclusions based on existing statistics, or proactively by forecasting the future. Conditions required for implementing these capabilities and concrete implications for the platform are identified. A scalable web platform is a high-level system architecture that contains the modular components for enabling AI-augmented software deployment. Each component is designed independently to support scalability for an increasing number of edge devices in operation.

A. AI Models Overview

Three different AI models will be deployed within the DemoCar vehicles in order to demonstrate the scalability of the web platform. The first model is a YOLOv5 model trained to detect cars and trucks on the road ahead. It is sent to the car with low priority but, when important service is required, it can be triggered to start and the detection results obtained can be consulted through the web using a specific API. The second model is a Yolov7 model specially trained to detect traffic signboards. It is sent to the car with high priority, as it is related to ADAS capabilities and the safety of the vehicle and its occupants. The detection results obtained can be consulted through the web using a specific API. The third model is a YAMNet audio classification model that categorizes the ambient vehicle noise in the 5 Democar vehicles. It is sent to the vehicles with high priority at specific times of the day. The characteristics of the AI models and the capabilities of the DemoCar edge devices determine their deployment priority.

B. Deployment Strategies

augmented software deployment for automotive edge devices follows several dimensions. While automatic tests provide an initial qualification of software components for functional and nonfunctional properties, certification complements these results with a safety argument. In some cases, the intention of AI augmentation is to relieve the Software Architect from having to assign software components to specific edges. Instead, the AI can take into account the platform model that contains the hardware configuration of each Edge Device and constraints from the Software Architect. The result of such optimization might then be a formalized deployment for each Edge Device. A related requirement is to consider the connectivity of each Edge Device. Since the connectivity is not necessarily bi-directional, the cloud cannot instruct the Edge Device on how to configure itself. Instead, the Edge Device needs to request the intended deployment using a polling approach. The initial intention of the AI augmentation was to support the Software Architect, especially for fleet-wide reconfiguration of Edge Devices.

EQ.7. Closed-Loop Cloud-Web-Robot Control (discrete-time with communication delay)

$$xk + 1 = Axk + Buk - d, uk = -Kxk \tag{7}$$

V. Cloud Services Integration

Cloud services fulfill numerous needs in connection with smart applications at the edge of society—mobility, health, and other areas. The increasing number of queries, ranging from health risks to navigation, demands scalable platforms that can provide fast and reliable answers. Cloud providers have recognized these needs and created platforms enabling developers to construct such services: Google offers Google App Engine and Google ML Kit; Amazon provides Amazon SageMaker; Microsoft operates Azure and AzureContainers Services; and IBM delivers IBM Cloud and Watson. These platforms facilitate the development and distribution of services across a broad spectrum of requirements. The three fundamental pillars employed to deploy the scalable web

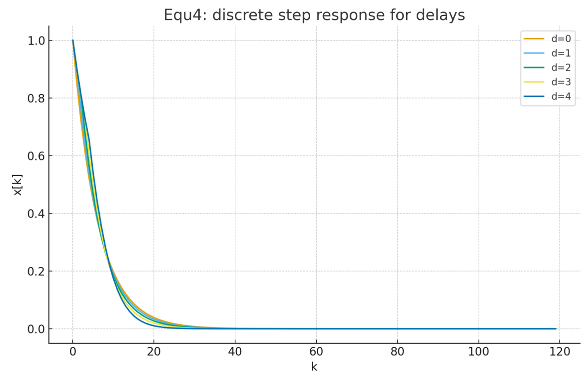

Fig. 6. Step responses for delays

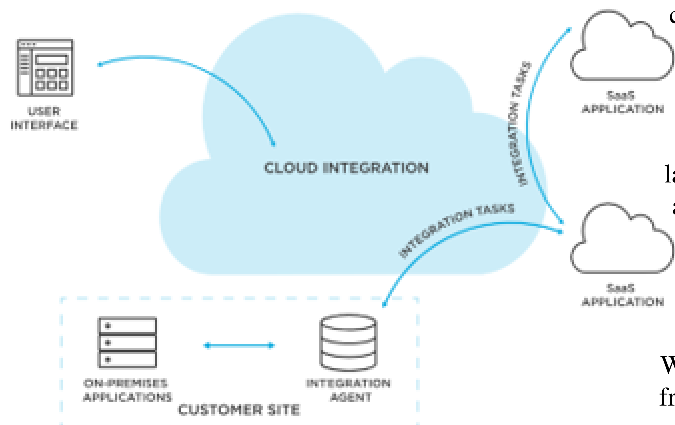

Fig. 7. Cloud Integration

platform for AI software deployment and execution on automotive edge devices are: Node.js with the Express.js framework to implement an IoT platform with service-oriented architecture and dedicated APIs; Watson AI services designed for specific tasks such as speech recognition and facial expression analysis; and IBM Cloud Container Services with Kubernetes orchestration to provide containerized services offering extreme scalability, the ability to process user requests in different geographical regions, and serving as backup to Node-RED IoT applications.

A. Service-Oriented Architecture

The overall architecture of the web platform follows a service-oriented approach, enabling highly modular and scalable deployment of the web platform over a set of container instances. Container orchestration can be scaled horizontally to account for high request volume or alternatively scaled down for a corresponding reduction in the traffic. As a result, the web platform can provide services for AI deployment in the automotive edge domain in a highly available, cost-effective, and scalable manner. The business logic of the web platform is also divided into multiple internal REST-based application programming interfaces (APIs) that later call cloud service providers services. As a result, various cloud service providers can offer own cloud services with which these internal REST-based APIs can communicate to provide the service in question. That means that the back-end of the Web platform enables service requests targeted at cloud service providers. These, in turn, should handle the request accordingly and provide the necessary services throughout the required cloud service provider's cloud infrastructure.

B. API Management

API Management aimed at ensuring scalable services and control of deployed software on electrified vehicles and related edge devices requires a Service-Oriented Architecture (SOA) with RESTful APIs. These APIs enable communication between stakeholders, control applications, cloud services, and vehicles with supporting backends. A robust platform for automated generation of these cloud services allows easy integration of AI models for augmenting software deployment on Edge devices. In a large-scale deployment scenario, RESTful APIs support the automation of model deployment and validation activities. The platform is deployed as a dockerized Python-based API-as-a-Service on a cloud infrastructure that exposes REST and WebSocket APIs supporting AI-augmented software deployment on automotive Edge devices. The WebSocket APIs support telemetry streaming data messages from the Edge device, while the REST APIs perform staged operations toward software deployment and related activities. The API deposits and communicates with the internal services of the platform, geared toward telemetry ingestion, filtering, classification, and Composer services for automated software provisioning on the Edge device.

EQ.8. Anomaly Detection & Health Monitoring (Mahalanobis distance)

$$D2(x) = (x -)^{\top}\Sigma^{-1}(x -) \tag{8}$$

VI. Automotive Edge Devices

The deployment of AI-augmented software on automotive edge devices is challenging due to computational resource constraints and intermittent low-bandwidth connectivity. To mitigate these limitations, computationally heavy AI tasks are offloaded to the cloud, utilizing its virtually unlimited storage, bandwidth, and processing resources while keeping the individual device burden low. In addition, artificial intelligence can be harnessed to support the complex software deployment process on automotive edge devices. This is accomplished by a scalable cloud service offered as a web platform that facilitates the deployment of AI-augmented, cloud-based, automated and highly customizable software on automotive edge devices. A particular category within the automotive edge device domain

x1	x2	D2	is_anomaly
-0.3235227463236941	-0.6299343219142147	0.2064632543923958	0
-0.5884002734258498	1.4539274467637542	2.240387135933475	0
0.9656558721992294	1.6135329347235206	2.08893749449181	0

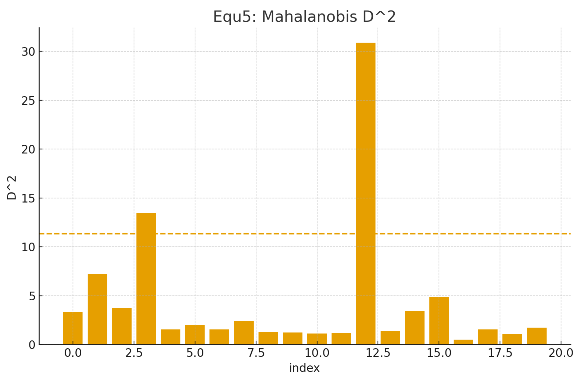

Fig. 8. Mahalanobis

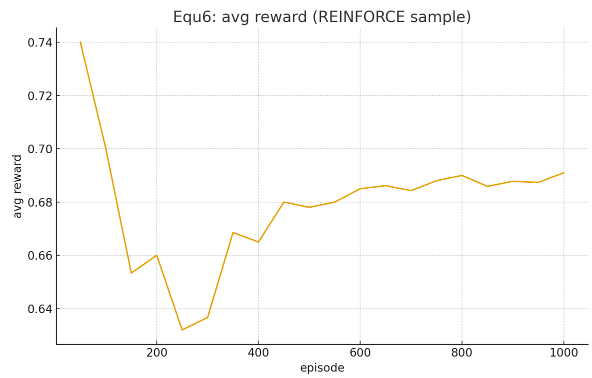

Fig. 9. Average reward (REINFORCE)

consists of specialized computing machines: drives, robots, and unmanned vehicles that are responsible for navigating the real world. For these systems to operate independently, they need to be equipped with intelligent functions realized as invisible executables running in the background. It would be immensely beneficial to have a computing system at one's command, capable of understanding the environment through images and videos, and capable of independently moving or assisting in movement. The natural question arises: Can a computing system be devised to fulfill such criteria? Can the autonomous functionality of an edge device be deployed using an auto-deploy feature?

A. Device Capabilities

Automotive edge devices encompass a wide spectrum, from classic embedded systems with adequate storage, processing power, and connectivity to control vehicle functions to more capable control units used in infotainment and Advanced Driver-Assistance Systems (ADAS). The capabilities of these control units continue to improve with each generation, with superior hardware enabling reduced latency for critical applications. Recent generations integrate powerful Graphics Processing Units (GPUs) alongside the main processing units, facilitating the deployment of AI-powered models on the smart devices themselves. The availability of connectivity with sufficient bandwidth and dependable Quality of Service (QoS) has grown in tandem with advancements in control units. Faster 5G networks and the emergence of dedicated communication systems such as C-V2X ensure comprehensive coverage, although certain geographical regions experience limitations. Connectivity requirements vary depending on the vehicle architecture: classic centralized architectures connect vehicles to the cloud via a central gateway, whereas distributed architectures, employing multiple control units with their own cellular modems, rely on the Control Unit itself for quality of service in interactions with cloud services.

B. Connectivity Challenges

The continual expansion of cloud technologies and the rise in automotive data production have positioned the car as the predominant sensor of the automotive edge, receiving increasing attention in the automotive edge domain. Consequently, application software capable of implementing Artificial Intelligence (AI) algorithms that capitalize on automotive data for service provision will reach the automotive edge. Automotive AI applications are often characterized by significant hardware requirements, which for many tasks represent a prohibitive condition, calling the scalability of the platform into question. The proposed web platform seeks to address these connectivity challenges by providing a comprehensive environment that integrates the abundant computational, storage, connectivity, and service-oriented capabilities of the cloud with the AI functions implemented at the automotive edge. The services offered by the cloud form the backbone through which scalability is attained, supporting the deployment of AI-augmented application software within automotive edge devices in a scalable manner.

EQ.9. Policy / softmax parameterizat (bandionit example)

$$pj(a) = ej0 + ej1eja \tag{9}$$

EQ.10. Policy gradient (REINFORCE, episodic single step):

$$\nabla_\theta J(\theta) \approx \mathbb{E}[\nabla_\theta \log \pi_\theta(a) G] \tag{10}$$

VII. Conclusion

The presented web platform fulfills all initial formulated requirements and is capable of supporting a growing and

arm	theta	final_prob	true_mean
0	-1.3916116274572865	0.05836630805250445	0.1
1	0.6707881660864718	0.45904057276986815	0.3
2	0.720823461370815	0.4825931191776274	0.5

vanishing number of vehicles in a scalable manner. The central idea is to leverage well-proven cloud services from Hyperscalers, which naturally provide scalability and load balancing, encapsulated within a suitably designed service-oriented architecture and accessed through standardized API calls. This approach is generally transferable to any other application domain where software must be pushed to a fluctuating number of edge devices. In addition to scalability, the platform also implements an AI solution. On the edge device, the software deployed with the highest priority runs on a dedicated hardware accelerator optimized for AI-prominent operations. The platform can be configured so that one of the other production OWCS runs an AI model in the background and, based on the outcome, triggers the deployment of more software functions, depending on the current situation. Overall, the solution provides scalable execution of software, either constrained by proprietary requirements (such as safety/security) or enabled solely by AI innovations.

A. Future Trends

Cloud services have repeatedly proven their scalability, once again enabling an autonomous and AI-augmented software deployment to edge devices, even in the demanding automotive domain. Within the next decade, new networks such as 5G are expected to be the main enabler for edge computing, along with the convergence of hardware, software, and AI in a self-adaptive manner. Therefore, future architectures must consider the benefits of cloud computing and the requirements of edge applications, which are shifting to the cloud. A service-oriented architecture (SOA) provides flexibility by splitting processes into smaller services, single-responsible services, that can be independently developed, deployed, scaled, and updated. Such architectures show the potential to satisfy the requirements of distributed edge services in any domain and provide scalable approaches for next-generation edge computing-driven services in the automotive domain.

Service-Oriented Architectures enable flexible management of applications and software supporting full software lifecycle management for members in the automotive value chain. Service-oriented systems offer acceleration for advanced software lifecycle management, including software testing, analysis, and business process generation. Their flexible design supports scaling for requirements in the increasingly interconnected automotive domain. Componentized and modular approaches pave the way for AI development and full software lifecycle management, thereby accelerating the fulfillment of business-oriented use cases.

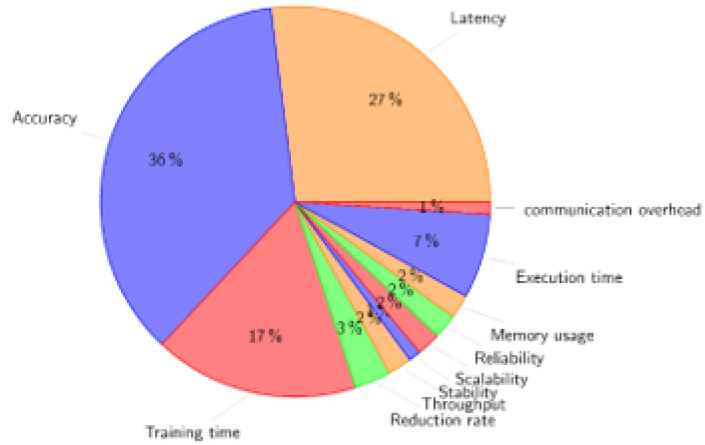

Fig. 10. Confluence of Artificial Intelligence and Edge Computing

References

[1] Halder, S., Ghosal, A., & Conti, M. (2020). Secure over-the-air software updates in connected vehicles: A survey. Computer Networks, 178, 107343. https://doi.org/10.1016/j.comnet.2020.107343.

[2] Sneha Singireddy. (2024). The Integration of AI and Machine Learning in Transforming Underwriting and Risk Assessment Across Personal and Commercial Insurance Lines . Journal of Computational Analysis and Applications (JoCAAA), 33(08), 3966–3991. Retrieved from https://eudoxuspress.com/index.php/pub/article/view/2732

[3] Maller, L. M., Suskovics, P., & Bokor, L. (2023). Edge computing in-the-loop simulation framework for automotive use-cases evaluation. Wireless Networks, 29(8), 3717–3735. https://doi.org/10.1007/s11276-023-03432-3.

[4] Mahesh Recharla, Karthik Chava, Chaitran Chakilam, & Sambasiva Rao Suura. (2024). Postpartum Depression: Molecular Insights and AI-Augmented Screening Techniques for Early Intervention. International Journal of Medical Toxicology and Legal Medicine, 27(5), 935–957. https://doi.org/10.47059/ijmtlm/V27I5/118

[5] Seisa, A. S. (2023). A Kubernetes-based edge architecture for controlling the trajectory of a resource-constrained aerial robot by enabling model predictive control. arXiv Preprint arXiv:2311.05610.

[6] Rani, F. (2024). Industrial Edge MLOps: Overview and challenges. In Edge MLOps and Industrial AI. Elsevier.

[7] Bhattacharjee, A., Mahmood, H., Lu, S., Ammar, N., Ganlath, A., & Shi, W. (2023). Edge-assisted over-the-air software updates. Pre-print.

[8] Lu, S., & Shi, W. (2023). Vehicle computing: Vision and challenges. Journal of Information and Intelligence.

[9] Shoker, A., Alves, F., & Esteves-Veríssimo, P. (2023). ScalOTA: Scalable secure over-the-air software updates for vehicles. arXiv Preprint arXiv:2302.09421

[10] Baqar, M., Khanda, R., & Naqvi, S. (2025). Self-Healing Software Systems: Lessons from Nature, Powered by AI. arXiv. https://arxiv.org/abs/2504.20093

[11] Inala, R., & Somu, B. (2024). Agentic AI in Retail Banking: Redefining Customer Service and Financial Decision-Making. Journal of Artificial Intelligence and Big Data Disciplines, 1(1).

[12] Amjad, A., Sthapit, S., & Syed, T. Q. (2025). An agentic system with reinforcement-learned subsystem improvements for parsing form-like documents. arXiv. https://arxiv.org/abs/2505.13504

[13] Mahesh Recharla, Karthik Chava, Chaitran Chakilam, & Sambasiva Rao Suura. (2024). Postpartum Depression: Molecular Insights and AI-Augmented Screening Techniques for Early Intervention. International Journal of Medical Toxicology and Legal Medicine, 27(5), 935–957. https://doi.org/10.47059/ijmtlm/V27I5/118

[14] Vakali, A., & Dimitriadis, I. (2025). FAIRTOPIA: Envisioning Multi-Agent Guardianship for Disrupting Unfair AI Pipelines. arXiv. https://arxiv.org/abs/2506.09107
[15] Sneha Singireddy. (2024). The Integration of AI and Machine Learning in Transforming Underwriting and Risk Assessment Across Personal and Commercial Insurance Lines . Journal of Computational Analysis and Applications (JoCAAA), 33(08), 3966–3991. Retrieved from https://eudoxuspress.com/index.php/pub/article/view/2732
[16] Inala, R., & Somu, B. (2024). Agentic AI in Retail Banking: Redefining Customer Service and Financial Decision-Making. Journal of Artificial Intelligence and Big Data Disciplines, 1(1).
[17] Halder, S., Ghosal, A., & Conti, M. (2020). Secure over-the-air software updates in connected vehicles: A survey. Computer Networks, 178, 107343.
[18] Raviteja Meda. (2024). Agentic AI in Multi-Tiered Paint Supply Chains: A Case Study on Efficiency and Responsiveness . Journal of Computational Analysis and Applications (JoCAAA), 33(08), 3994–4015. Retrieved from https://eudoxuspress.com/index.php/pub/article/view/2734
[19] Baqar, M., Khanda, R., & Naqvi, S. (2025). Self-Healing Software Systems: Lessons from Nature, Powered by AI. arXiv. https://arxiv.org/abs/2504.20093
[20] Koppolu, H. K. R., & Sheelam, G. K. (2024). Machine Learning-Driven Optimization in 6G Telecommunications: The Role of Intelligent Wireless and Semiconductor Innovation. Global Research Development (GRD) ISSN: 2455-5703, 9(12).
[21] Amjad, A., Sthapit, S., & Syed, T. Q. (2025). An agentic system with reinforcement-learned subsystem improvements for parsing form-like documents. arXiv.
[22] Kummari, D. N., & Burugulla, J. K. R. (2023). Decision Support Systems for Government Auditing: The Role of AI in Ensuring Transparency and Compliance. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 493-532.