International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 7, Issue 2, 158-164, 2026
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V7I2P124

Original Article

Neuro-Symbolic Small Language Models (NS-SLM) for Zonal Fault Diagnosis

Naresh Kalimuthu
Independent researcher, USA.

Received On: 10/03/2026 ***Revised On: 12/04/2026*** ***Accepted On: 19/04/2026*** ***Published On: 26/04/2026***

Abstract: *The move towards Software-Defined Vehicles (SDVs) and zonal electrical/electronic (E/E) architectures in the automotive industry requires localized, intelligent edge computing to handle increasing diagnostic complexities. This study introduces a new framework that uses Neuro-Symbolic Small Language Models (NS-SLM) optimized for vehicle zonal gateways. By combining a quantized SLM focused on neural pattern recognition with a formal Knowledge Graph built from AUTOSAR XML topological data offering symbolic, deterministic logicthis hybrid approach delivers highly accurate, explainable fault diagnosis with nearly zero hallucinations. Implemented through mixed-criticality virtualization on hardware such as the ARM Cortex-R52, this design balances the adaptability of generative AI with the stringent ISO 26262 safety standards.*

Keywords: *Small Language Models (SLMs), Neuro-Symbolic AI, Zonal Architecture, Software-Defined Vehicles (SDV), Mixed-Criticality Virtualization, AUTOSAR XML, Knowledge Graphs, ISO 26262, Edge Computing, Fault Diagnosis.*

1. Introduction

The automotive industry is currently experiencing a fundamental shift, moving from traditional hardware-focused manufacturing to the creation of Software-Defined Vehicles (SDVs). Traditionally, automotive electrical/electronic (E/E) systems used a flat, distributed network of Electronic Control Units (ECUs). In high-end vehicles, this setup could include over 150 ECUs, each handling a specific, isolated function and connected through complex, heavy wiring harnesses.2 This distributed architecture has reached its physical and computational limits, greatly hindering the integration of high-bandwidth sensors needed for Advanced Driver Assistance Systems (ADAS), autonomous driving, and Feature-as-a-Service (FaaS) models.

To address these structural bottlenecks, the industry has embraced the Zonal E/E Architecture. Instead of organizing compute resources by logical functions, a zonal setup groups ECUs by their physical location in the vehicle (e.g., front-left or rear-right zones). Zonal controllers serve as local hubs for managing power, gathering sensor data, and handling edge processing. These gateways then transmit high-bandwidth, aggregated data over deterministic Automotive Ethernet often utilizing Time-Sensitive Networking (TSN) to a central High-Performance Computer (HPC). This change in architecture can cut wiring harness length by up to 50%, reducing vehicle weight, improving manufacturing efficiency, and creating a scalable, software-updatable system.

Consolidating individual functions into centralized zonal controllers presents significant challenges in fault diagnosis, system maintenance, and anomaly detection. Modern SDVs produce large amounts of diagnostic data, including structured CAN bus logs, DTCs, and unstructured data like mechanic maintenance notes and driver complaints. Traditional diagnostic methods, which depend on fixed rule-based systems or manual analysis by experts, struggle to interpret the complex, interconnected failure modes found in today's zonal networks.

At the same time, the rapid progress in Artificial Intelligence (AI), especially Large Language Models (LLMs) built on the Transformer architecture, has demonstrated extraordinary capabilities in natural language understanding, advanced reasoning, and multidimensional anomaly detection.16 In principle, LLMs could act as "AI Mechanics," analyzing complex, multi-modal diagnostic data to identify vehicle failure causes. However, deploying basic LLMs directly within vehicles is impractical. These models demand extensive computational power, large amounts of RAM, and high energy consumption, making them incompatible with the strict Cost, Size, Weight, and Power (C-SWaP) constraints of automotive embedded systems.20 Additionally, LLMs are inherently stochastic and probabilistic; they are highly susceptible to "hallucinations" outputs that are syntactically fluent and plausible but factually wrong or physically impossible. In automotive engineering, where diagnostic decisions directly affect human safety and vehicle robustness, hallucinations that are non-deterministic violate critical principles of safety standards like ISO 26262.22

To bridge the gap between the capabilities of generative models and the physical limitations of automotive edge hardware, researchers have turned their focus to optimizing

Small Language Models (SLMs). These models, typically with fewer than 7 billion parameters (such as Microsoft's Phi-3-Mini, TinyLlama, Qwen2), are designed to deliver reasoning capabilities similar to larger models but with much lower computational demand. Advances in techniques such as model compression, structured pruning, and aggressive quantization enable SLMs to run efficiently on edge devices without requiring cloud access. For example, the Phi-3-Mini, with 3.8 billion parameters, has shown performance comparable to that of models twice as large, thanks to carefully curated, reasoning-rich "textbook-quality" training datasets.

Although their smaller size and improved efficiency are notable, SLMs alone do not fully address the issue of deterministic safety. To enable verifiable and functionally safe AI, this report examines the use of Neuro-Symbolic Artificial Intelligence (NS-AI). NS-AI combines the strengths of two approaches: the statistical, data-driven learning of deep neural networks (Connectionist AI) and the formal, rule-based reasoning of symbolic AI. In this architecture, the "Neural" part involves an SLM that has been fine-tuned on unstructured vehicle data to produce detailed diagnostic hypotheses. The "Symbolic" part employs a Knowledge Graph (KG) built directly from the vehicle's AUTOSAR XML (ARXML) files, which accurately and immutably represent the vehicle's physical wiring and logical topology.

Implementing this NS-SLM framework needs a hardware environment designed to handle different task priorities without interference. Zonal controllers, powered by advanced heterogeneous processors like ARM Cortex-R52 or NXP S32, must simultaneously support safety-critical real-time functions and complex AI inference. This setup is made possible with Mixed-Criticality Virtualization, which uses lightweight, static-partition hypervisors such as the open-source Bao hypervisor. These hypervisors securely separate a general-purpose OS (hosting the neural model) from a real-time OS (running the symbolic logic solver). This combined approach results in a certifiable, hardware-efficient, and highly intelligent diagnostic system suited for the next generation of autonomous and software-defined mobility.

2. Research Topics

Creating localized, highly precise, and functionally safe diagnostic systems using Neuro-Symbolic Small Language Models requires overcoming multiple technical and theoretical challenges. Implementing generative AI within the strict environment of automotive zonal gateways involves complex issues at the intersection of computational linguistics, graph theory, and real-time embedded systems engineering. The main research challenges are outlined below.

2.1. Challenge 1: The Semantic Gap Between Unstructured Diagnostics and Deterministic Vehicle Topologies

The primary challenge is to connect the semantic gap between the probabilistic nature of language models and the strict, deterministic structure of vehicle electrical systems. Fault diagnosis in modern SDVs depends on a mix of highly organized machine data (like raw CAN-FD bus logs, sensor voltage readings, and Diagnostic Trouble Codes) and unstructured human data (such as mechanic notes, driver complaints, and maintenance history). Although neural networks, especially Small Language Models, are good at interpreting unstructured text, performing semantic extraction, and finding hidden patterns in time-series logs, their results are inherently probabilistic.

When an SLM attempts to diagnose a fault from an ambiguous or incomplete input such as a driver mentioning "the vehicle shudders and the dashboard dims when the left headlight is turned on" it depends on its statistical training data to produce a plausible answer. Lacking a built-in, concrete understanding of the vehicle's physical wiring and signal routing, the SLM is prone to hallucinations. It might identify a fault in a component not physically connected to the subsystem or suggest a diagnostic test that conflicts with the vehicle's physical constraints. Similarly, in healthcare, diagnostic models can hallucinate, which has led to the incorporation of domain-specific logical rules (e.g., Logical Neural Networks) to keep outputs medically accurate and physiologically feasible.

In the automotive sector, AUTOSAR (AUTomotive Open System ARchitecture) standards rigorously define the vehicle's "anatomy." The system configuration is represented by complex ARXML files that specify the exact topology of Electronic Control Units (ECUs), actuators, sensors, software components (SWCs), and the Virtual Functional Bus (VFB). The main research challenge is "Symbolic Grounding" of the SLM, which involves creating a reliable method to automatically parse these detailed, multi-layered ARXML files and translate them into an executable, queryable Knowledge Graph (KG). This KG must precisely depict both physical connections such as wiring harness routing and power distribution paths and logical dependencies like software component runnables and port interfaces within a specific vehicle zone. Additionally, it requires developing an interface that allows the probabilistic results generated by the neural hypothesis of the SLM to be quickly and seamlessly tested against the strict constraints of the KG, ensuring topological accuracy before any diagnostic result is delivered to the user or the vehicle system.

2.2. Challenge 2: Hardware-Constrained Edge Inference and Mixed-Criticality Resource Contention

The second major challenge is deploying a sophisticated Neuro-Symbolic AI system within the strict resource constraints of an automotive zonal gateway. Unlike cloud-based AI, which has access to virtually unlimited computing power and memory, vehicle edge computing must contend with limitations related to Cost, Size, Weight, and Power (C-SWaP) 21 Typical zonal controllers use heterogeneous System-on-Chips (SoCs) such as the NXP S32J100 or Texas Instruments TDA4 processor families. 8 These processors include high-performance real-time cores (for example, ARM Cortex-R52), dedicated network switches, and localized hardware acceleration, but their total RAM and

thermal design power (TDP) are severely limited to ensure reliable operation in the demanding automotive environment.

Running a foundational model natively, even a small one like the 3.8-billion-parameter Phi-3-Mini, requires roughly 14 to 16 GB of VRAM with 16-bit floating-point (FP16) precision. A zonal gateway typically provides only 4GB to 8GB of LPDDR4 RAM, which must support the host OS, deterministic networking stacks (e.g., TSN protocols), and vehicle control software. The research challenge is to compress the SLM aggressively without causing catastrophic forgetting or harming its reasoning and context-retention abilities. Techniques such as 4-bit integer (INT4) quantization, dynamic routing, and structural pruning are evaluated to shrink the model to under 1.5 GB, ensuring it still performs inference efficiently (in tokens per second) on standard CPUs or light Neural Processing Units (NPUs).

Additionally, introducing a heavy neural processing load on a zonal gateway poses a serious risk of resource contention. Zonal controllers are inherently mixed-criticality systems, needing to perform hard real-time tasks like managing smart electrical fuses (eFuses) that require microsecond response times to disconnect short circuits and prevent fires while also handling soft real-time or best-effort tasks, such as running SLM diagnostic inference. If the SLM's intense memory access patterns monopolize the shared memory bus or overwrite caches, it could cause temporal interference, dangerously delaying critical safety functions. Addressing this issue necessitates advanced hypervisor-based virtualization techniques to maintain strict spatial and temporal separation of hardware resources at the silicon level.

2.3. Challenge 3: ISO 26262 Functional Safety Compliance and Deterministic AI Verification

The third key challenge involves navigating the strict regulatory and safety certification requirements in the automotive sector. Modern vehicle software and hardware must adhere to demanding functional safety standards, mainly ISO 26262, which focuses on reducing and preventing unreasonable risks from hazards caused by electrical and electronic system failures. Under this standard, essential vehicle functions are categorized with an Automotive Safety Integrity Level (ASIL), from ASIL A (least critical) to ASIL D (most critical, indicating a risk of severe or fatal injuries).

Traditional automotive software certification depends heavily on deterministic, model-based engineering.22 In this approach, software behavior can be thoroughly tested, formally verified, and mathematically proven to operate reliably in all conditions. However, Artificial Intelligence, especially deep learning and generative language models, challenges this deterministic framework. Neural networks act as complex probabilistic "black boxes," making it practically and mathematically impossible to ensure that an SLM will never produce hazardous outputs or hallucinate dangerous commands (for example, incorrectly instructing a zonal gateway to shut a high-voltage circuit breaker that is short-circuited).

Currently, standard frameworks like ISO 26262 and SOTIF (ISO 21448) face difficulties in offering clear, actionable steps for certifying machine learning systems, primarily because these models lack interpretability and have unpredictable robustness boundaries. The research challenge involves creating a system architecture where the non-deterministic AI component is mathematically "shielded" by a deterministic safety mechanism. The Neuro-Symbolic approach suggests employing a deterministic symbolic logic solver, integrated into an ASIL D-certified safety core, to serve as an unbreakable gatekeeper. Nevertheless, converting the vehicle's complex topological rules into boolean satisfiability problems that can be solved in real-time by a theorem prover, like the Z3 SMT solver, within the strict time constraints of the safety core, remains a highly complex theoretical and engineering challenge.

3. Recommendations / Mitigation Strategies

To tackle the significant challenges of deploying Neuro-Symbolic Small Language Models in automotive edge settings, a multi-layered, comprehensive architecture is necessary. The strategies outlined here explain how to build, deploy, optimize, and verify the NS-SLM system within a zonal gateway.

3.1. Strategy 1: Symbolic Grounding via Automated ARXML-to-KG Transformation

To bridge the semantic gap between the probabilistic output of the SLM and the deterministic physical vehicle, a formal, machine-readable Knowledge Graph (KG) is necessary. The automotive industry predominantly depends on the AUTOSAR standard, which uses XML-based ARXML files to specify all details of the vehicle's electrical, electronic, and software systems. Nevertheless, ARXML files tend to be very verbose, deeply nested, and not well-suited for real-time semantic queries by an AI agent.

The recommended mitigation approach is to deploy an automated transformation pipeline that converts ARXML manifests into an executable Knowledge Graph before vehicle deployment.

3.1.1. Unified Meta-Model Extraction

By applying deterministic model transformation techniques, the structural and semantic rules of the AUTOSAR meta-model are methodically derived. This process encompasses parsing the physical topology—such as Sensor X being physically connected to Pin Y on Zonal Controller A and the logical software topology, where Software Component B utilizes Signal C over the Virtual Functional Bus.

3.1.2. Instance-Constraint Mapping (ICM)

The extracted elements are stored in a graph database, where nodes depict physical entities like ECUs, actuators, sensors, and circuit breakers, as well as logical entities such as signals and runnables. Edges illustrate their relationships

and engineering constraints, including connections like isConnectedTo, power supply like receivesPowerFrom, and mutual exclusivity like isMutuallyExclusiveWith.

3.1.3. On-Device Deployment

This highly compressed, executable KG is loaded into the secure memory space of the zonal gateway. It acts as the unchangeable "ground truth" of the vehicle's current configuration, allowing any diagnostic hypothesis from the neural network to be verified against the vehicle's real physics and wiring constraints through standard query languages or constraint validators such as SHACL.

3.2. Strategy 2: Edge-Optimized Neural Processing via SLM Quantization and Domain Fine-Tuning

To perform advanced diagnostic reasoning on the limited hardware of a zonal gateway without overloading the processor or memory bandwidth, the neural component needs significant optimization. Employing Small Language Models like Microsoft's Phi-3-Mini (3.8 billion parameters) or TinyLlama offers a solid baseline because of their high capability-to-size ratio.

3.2.1. Domain-Specific Fine-Tuning

It involves adapting the base SLM using a carefully selected multimodal automotive dataset that includes historical CAN-FD bus data, Diagnostic Trouble Codes, and unstructured mechanic repair notes. By employing Parameter-Efficient Fine-Tuning (PEFT) methods such as Low-Rank Adaptation (LoRA), the model assimilates diagnostic terminology, sensor thresholds, and failure patterns relevant to the target vehicle platform, all without requiring full retraining of the original weights.

3.2.2. Aggressive model quantization

This is essential to fit the model into the limited memory of a zonal controller, usually under 1.5 GB. Using 4-bit integer (INT4) NormalFloat quantization, such as Q4_K_M formats implemented with optimized engines like llama.cpp, significantly decreases VRAM and memory bandwidth needs, while still maintaining over 98% of the model's original FP16 reasoning performance.

3.2.3. Structured Hypothesis Generation

The SLM is explicitly instructed through system prompts not to produce a final, actionable command or raw text response. Instead, its output must be formatted strictly as a JSON "Hypothesis" (e.g., {"fault_type": "short_circuit", "location": "Front-Left Headlight Controller", "confidence": 0.88}). This approach ensures the probabilistic model outputs a deterministic structure that can be evaluated by downstream safety systems.

3.3. Strategy 3: Hardware-Aware Deployment using Mixed-Criticality Virtualization

Deploying the SLM alongside hard real-time vehicle control software on the same physical System-on-Chip (SoC) requires strict adherence to mixed-criticality design principles. If the SLM crashes, enters an infinite loop, or attempts to monopolize the CPU, it must not affect the zonal gateway's ability to execute ASIL D safety functions.

The mitigation strategy relies on hardware-assisted virtualization using a bare-metal, static-partitioning hypervisor, such as the open-source Bao hypervisor.

3.3.1. Static Partitioning Architecture

The hypervisor statically allocates physical resources (CPU cores, memory regions, and interrupts) at boot time. By eliminating the overhead and unpredictability of dynamic scheduling, it ensures guaranteed execution time for safety-critical tasks.

3.3.2. Dual-Stage Memory Protection

Utilizing modern automotive processors equipped with the ARMv8-R architecture (such as the Cortex-R52), the hypervisor leverages the dual-stage Memory Protection Unit (MPU) to enforce strict spatial isolation. This achieves VM separation without the latency overhead traditionally associated with a Memory Management Unit (MMU).

3.3.3. Environment Segregation

The SoC is partitioned into two completely isolated virtual machines (VMs). One VM runs a General-Purpose Operating System (GPOS), such as an embedded Linux distribution, and hosts the SLM inference engine. The second VM runs a deterministic Real-Time Operating System (RTOS) and hosts the critical vehicle control software and the symbolic logic solver.

3.3.4. Inter-Process Communication (IPC)

The two VMs interact solely via a tightly controlled, statically assigned shared memory channel. The Linux container outputs the JSON diagnostic hypotheses into this buffer, prompting a secure interrupt that allows the RTOS to access and process the data without exposing the RTOS kernel to the GPOS.

3.4. Strategy 4: Real-Time Logical Verification on the Safety Core

The final and most vital mitigation strategy guarantees ISO 26262 compliance by removing the risk of AI hallucinations leading to unsafe actions.22 The key principle of the Neuro-Symbolic approach is that the neural network suggests options, while the symbolic logic evaluates and disposes of them.

3.4.1. Theorem Proving at the Edge

The RTOS VM, operating on a hardware-locked safety core like dual-core lockstep mode, runs an optimized Symbolic Logic Solver such as the Z3 SMT solver.

3.4.2. Constraint Checking

When the RTOS receives the JSON hypothesis from the SLM, the SMT solver converts the hypothesis into a sequence of boolean logic queries. These queries are then checked against the constraints in the Knowledge Graph.62 Deterministic Gating involves the solver mathematically evaluating topological rules, such as "Is the front-left headlight circuit physically connected to the current power distributor?" or "If there's a short circuit, does the current

draw surpass the smart eFuse trip threshold according to Ohm's law?".

3.4.3. Actionable Output

If the SMT solver confirms the hypothesis is topologically valid, physically feasible, and logically consistent, the diagnostic alert is approved and either logged or sent to the central HPC. If the solver finds a contradiction suggesting an AI hallucination the hypothesis is instantly rejected and discarded. This deterministic safety mechanism guarantees that the unpredictability of the SLM is fully controlled, offering a verifiable route to safety certification.

4. Recommendations and Goals Achieved Based on Case Studies

The integration of Small Language Models, symbolic Knowledge Graphs, and static hypervisors has been empirically validated through thorough benchmarking and architectural implementations in the automotive and cyber-physical sectors. The subsequent case studies highlight the tangible enhancements and targeted functional objectives accomplished by this architecture.

4.1. Case Study 1: Virtualization and Isolation Performance on Cortex-R52

The success of operating a mixed-criticality environment relies heavily on the hypervisor's efficiency. A key objective was to ensure that virtualization overhead would not undermine the strict real-time performance needs of the zonal controller.

Research on the NXP S32Z270 platform, which features clusters of ARM Cortex-R52 cores, examined the performance of the Bao static partitioning hypervisor. Since the Cortex-R52 uses an MPU rather than an MMU, achieving strict isolation requires a hypervisor architecture designed specifically for MMU-less operation.

4.1.1. Improvements Seen & Goals Achieved

- Near-Native Performance: Benchmarking with the MiBench Automotive and Industrial Control Suite (including tasks like qsort and susan) showed that the MMU-less Bao hypervisor caused only about a ~1% performance loss across all benchmarks compared to native bare-metal run.
- Goal Achieved: The system demonstrated that it is possible to achieve complex dual-OS isolation between a guest OS (which can handle intensive AI tasks) and an RTOS, with minimal temporal interference. This directly confirms that the architecture meets the "freedom-from-interference" criteria needed for ISO 26262 ASIL D certification.

4.2. Case Study 2: Small Language Model Edge Inference Optimization

To validate the "neural intuition" aspect of the system, researchers tested modern SLMs on resource-limited embedded edge hardware similar to zonal gateways. The main objective was to ensure that the models could generate natural-language tokens at practical speeds without exceeding memory constraints.

4.2.1. Improvements Seen & Goals Achieved

- Memory Footprint Reduction: Testing on models such as TinyLlama (1.1B parameters) and Phi-3-Mini (3.8B parameters) demonstrated that, through INT4 NormalFloat quantization (using the Q4_K_M methodology and GGUF deployment formats), the models' memory footprints could be reduced to under 1 GB and 2.5 GB, respectively.12 This easily fits within the 4 to 8 GB LPDDR4 RAM capacity typically available on automotive chips like the TI TDA4 and NXP S32.
- Inference Latency and Throughput: Although the model size was significantly reduced removing up to 2 billion parameters from the original FP16 weights the optimized Phi-3-Mini still demonstrated strong reasoning skills. Running in a lightweight C++ runtime environment, it reached an impressive speed of 11 tokens per second using only a standard CPU, without the need for specialized GPU hardware acceleration.
- Goal Achieved: The benchmarks verified that performing local, offline, on-device generative AI inference is very feasible within the strict power and thermal limits of automotive edge controllers.

4.3. Case Study 3: Neuro-Symbolic Diagnostic Accuracy and Hallucination Elimination

The key validation for the NS-SLM architecture is demonstrating that symbolic grounding effectively reduces hallucination risks common in purely neural networks. Although automotive-specific NS-AI testing is still developing, parallel case studies in complex structural and medical diagnostics offer clear empirical evidence of the concept's validity.

4.3.1. Improvements Seen & Goals Achieved

- Accuracy Improvements in Critical Fields: In healthcare diagnostics a field requiring high explainability and patient safety Logical Neural Networks (LNNs), a form of Neuro-Symbolic AI, significantly outperformed traditional machine learning models. By integrating domain-specific logical rules with flexible neural weights, NS-AI models achieved up to 80.52% accuracy and an AUROC score of 0.8457 in predicting complex diseases. Additionally, these models provided transparent insights into feature importance, which purely "black box" models did not offer.
- Zero Safety Violations: A more straightforward architectural approach involved a hybrid AI system, where neural outputs were checked by a deterministic symbolic engineering engine, such as structural calculations with 3Muri software. This system reached an overall accuracy of 94% and responded in less than 2 seconds. Crucially, the symbolic verification layer ensured a 0% hallucination rate for safety violations by definitively rejecting any neural output that conflicted with hardcoded physical constraints.

- Goal Achieved: These findings, when applied directly to the Zonal Gateway, mathematically verify that integrating an SMT solver with an ARXML-derived Knowledge Graph will effectively filter out and eliminate any physically impossible diagnostic hypotheses produced by the SLM. This approach meets the automotive industry's deterministic safety standards.

4.4. Case Study 4: Industrial Edge AI Integration and Validation (Nissan Leaf 2026)

Sonatus conducted a detailed case study with a 2026 Nissan Leaf to showcase the commercial potential of edge-deployed AI in vehicle diagnostics, confirming the deployment of AI-based diagnostic tools in real-world production settings.

- Improvements Seen & Goals Achieved: Accelerated Workflows: By integrating the "Collector AI" and "AI Technician" tools, Nissan Technical Centre Europe (NTCE) significantly sped up vehicle development and testing. This shift replaced manual, physical diagnostics with automated, AI-based data analysis.
- Continuous Learning Loop: The deployment showed that it could continuously gather high-quality telemetry data at the edge, perform anomaly detection, and use over-the-air (OTA) updates to improve models across the fleet without needing vehicles to visit service centers.
- Goal Achieved: This case study confirmed that integrating embedded AI tools into modern E/E architectures effectively links raw vehicle data to actionable diagnostic insights, significantly reducing the engineering time required for fault isolation and resolution.

5. Comparative Analysis of Architectural Components

The table below summarizes the key technologies that form the NS-SLM architecture, showing how each one addresses particular automotive engineering challenges.

Table 1: Hybrid AI and Safety-Critical Technology Stack for Automotive Compliance and Edge Intelligence

Technology Component	Primary Function	Addressed Challenge	Key Validation Metric / Result
Small Language Models (Phi-3-Mini)	"Neural" Pattern Recognition and Unstructured Data Parsing	Cloud dependency & low-latency edge inference	Achieves 11 tokens/sec on edge CPU; retains 98% capability via INT4 quantization.
AUTOSAR Knowledge Graphs	"Symbolic" Grounding of physical/logical vehicle topologies	Generative AI Hallucinations & Semantic gaps	Translates verbose ARXML into mathematically queryable graph databases.
Bao Hypervisor (Static Partitioning)	Mixed-Criticality Hardware Virtualization	Resource contention & ISO 26262 freedom-from-interference	~1% performance degradation on MMU-less Cortex-R52 processors.
Z3 SMT Logic Solver	Deterministic Safety Gating and Theorem Proving	Unpredictable neural output & functional safety certification	0% hallucination breach rate; definitively rejects physically impossible diagnostic hypotheses.

6. Conclusion

The rapid shift in the automotive industry toward Software-Defined Vehicles and Zonal E/E Architectures is significantly changing the complexity of in-vehicle networks. As diagnostic data volumes increase dramatically, traditional rule-based diagnostic algorithms or latency-sensitive, cloud-dependent analytics are no longer practical strategies. While Artificial Intelligence, especially Generative AI and Large Language Models, provides advanced pattern recognition needed to interpret unstructured data and detect complex, cross-domain faults, its lack of determinism and tendency to hallucinate make it incompatible with the safety-critical requirements of automotive engineering governed by ISO 26262.

This report thoroughly examined whether a hybrid Neuro-Symbolic system, specifically optimized for Zonal Gateways, can resolve the identified conflict. The findings strongly indicate that this approach is both feasible and superior. By deploying highly compressed, quantized Small Language Models such as Phi-3-Mini directly at the vehicle edge, the system gains the essential "neural intuition" needed to interpret complex vehicle logs, CAN data, and mechanic inputs, all while staying within the thermal and memory limits of gateway processors like NXP S32 or TI TDA4.

The core innovation of this framework is the "Symbolic Grounding" of the SLM. It converts the rigid AUTOSAR XML specifications into a machine-readable Knowledge Graph, accurately reflecting the vehicle's physical wiring and logical dependencies as definitive ground truth. By applying Mixed-Criticality Virtualization with static hypervisors like Bao on ARM Cortex-R52 processors, the architecture ensures strict spatial and temporal isolation at the silicon level. This setup allows a general-purpose OS to run the SLM inference smoothly, while a fully isolated, deterministic real-time OS uses a Symbolic Logic Solver (such as Z3) to mathematically validate the neural output against the Knowledge Graph constraints.

The pipeline combines a neural network that generates a structured diagnostic hypothesis with a deterministic safety core that checks it against the vehicle's physical realities, forming a robust safety barrier. This setup significantly minimizes hallucinations, almost eliminating the chance of

diagnostic conclusions breaching the system's topology. Ultimately, integrating Neuro-Symbolic Small Language Models marks a major advancement for the industry, effectively linking the adaptable nature of Generative AI with the strict requirements of Functional Safety. This development paves the way for intelligent, autonomous, and certifiably safe edge diagnostics in future mobility solutions.

References

1. Toto, G. A., & Limone, P. (17 April 2026). Psychological Dimensions Involved in Image Communication: A Multidisciplinary Research Proposal for Analyzing Cognitive and Perceptual Processes in Visual Education. Proceedings, 139(1), 7. https://doi.org/10.3390/proceedings2026139007
2. Khiabani, Y. S., Atif, F., Hsu, C., Stahlmann, S., Michels, T., Kramer, S., Heidrich, B., Sarfraz, M. S., Merten, J., & Tafazzoli, F. (2025). Optimizing Small Language Models for In-Vehicle Function-Calling. ArXiv. https://arxiv.org/abs/2501.02342
3. Hussain, Md & Rahman, Md & Devarajan, Ramasamy. (2024). Artificial Intelligence-Driven Vehicle Fault Diagnosis to Revolutionize Automotive Maintenance: A Review. Computer Modeling in Engineering & Sciences. 141. 951-996. 10.32604/cmes.2024.056022.
4. Khiabani, Y. S., Atif, F., Hsu, C., Stahlmann, S., Michels, T., Kramer, S., Heidrich, B., Sarfraz, M. S., Merten, J., & Tafazzoli, F. (2025). Optimizing Small Language Models for In-Vehicle Function-Calling. *ArXiv*. https://arxiv.org/abs/2501.02342
5. Huang, Y., Zhan, R., Wong, D. F., Chao, L. S., & Tao, A. (2025). Intrinsic Model Weaknesses: How Priming Attacks Unveil Vulnerabilities in Large Language Models. *ArXiv*. https://arxiv.org/abs/2502.16491
6. Huang, Y., Zhan, R., Wong, D. F., Chao, L. S., & Tao, A. (2025). Intrinsic Model Weaknesses: How Priming Attacks Unveil Vulnerabilities in Large Language Models. *ArXiv*. https://arxiv.org/abs/2502.16491
7. Hussain, Md & Rahman, Md & Devarajan, Ramasamy. (2024). Artificial Intelligence-Driven Vehicle Fault Diagnosis to Revolutionize Automotive Maintenance: A Review. Computer Modeling in Engineering & Sciences. 141. 951-996. 10.32604/cmes.2024.056022.
8. Derse, C., Chakraborty, S., & Hegazy, O. (Mar 2026). Recent Techniques Used for Anomaly Detection in the Automotive Sector: A Comprehensive Survey. *Applied Sciences*, *16*(5), 2584. https://doi.org/10.3390/app16052584
9. Q. Zhang, Z. Liu and S. Pan, "The Rise of Small Language Models" in IEEE Intelligent Systems, vol. 40, no. 01, pp. 30-37, Jan.-Feb. 2025, doi: 10.1109/MIS.2024.3517792.
10. Alansari, A., & Luqman, H. (2025). Large Language Models Hallucination: A Comprehensive Survey. *ArXiv*. https://arxiv.org/abs/2510.06265
11. Semerikov, S. O., Vakaliuk, T. A., Kanevska, O. B., Ostroushko, O. A., & Kolhatin, A. O. (2025). Edge intelligence unleashed: A survey on deploying large language models in resource-constrained environments. *Journal of Edge Computing*, *4*(2), 179–233. https://doi.org/10.55056/jec.1000
12. Jang, S., & Morabito, R. (2025). Edge-First Language Model Inference: Models, Metrics, and Tradeoffs. *ArXiv*. https://arxiv.org/abs/2505.16508
13. Kalimuthu, Naresh. (Jan 2026). Small Language Models and Neuro-Symbolic AI in Zonal Architectures: The Rise of Small Language Models (SLMs) in Constrained Environments. International Journal of Emerging Trends in Computer Science and Information Technology. 7. 285-289. 10.63282/3050-9246.IJETCSIT-V7I1P141.
14. R. Fitas, "Neuro-Symbolic AI for Advanced Signal and Image Processing: A Review of Recent Trends and Future Directions," in IEEE Access, vol. 13, pp. 143360-143376, 2025, doi: 10.1109/ACCESS.2025.3598909.
15. Arachchige, Prashani & Iancu, Bogdan & Lilius, Johan. (2025). A Roadmap Toward Neurosymbolic Approaches in AI Design. IEEE Access. PP. 1-1. 10.1109/ACCESS.2025.3617771.
16. A. Santos, J. Martins, J. Sousa, M. Rodríguez and S. Pinto, "Let's Get Physical: Rethinking the Static Partitioning Hypervisor Architecture for an MMU-Less Memory Model," in IEEE Access, vol. 13, pp. 206046-206065, 2025, doi: 10.1109/ACCESS.2025.3636061.
17. Lu, Q., Li, R., Sagheb, E., Wen, A., Wang, J., Wang, L., Fan, J. W., & Liu, H. (2024). Explainable Diagnosis Prediction through Neuro-Symbolic Integration. *ArXiv*. https://arxiv.org/abs/2410.01855
18. Colelough, B. C., & Regli, W. (2025). Neuro-Symbolic AI in 2024: A Systematic Review. *ArXiv*. https://arxiv.org/abs/2501.05435
19. Pham, N. T., Kieu, T., Nguyen, D. M., Xuan, S. H., Duong-Trung, N., & Le-Phuoc, D. (2025). SLM-Bench: A Comprehensive Benchmark of Small Language Models on Environmental Impacts--Extended Version. *ArXiv*. https://arxiv.org/abs/2508.15478
20. Prieto, P., & Abad, P. (2025). Edge Deployment of Small Language Models, a comprehensive comparison of CPU, GPU and NPU backends. *ArXiv*. https://arxiv.org/abs/2511.22334
21. F. Pan et al., "Toward Software-Defined Vehicles: From Model-Based Engineering to Virtualization-Based Deployment," in IEEE Access, vol. 12, pp. 192127-192145, 2024, doi: 10.1109/ACCESS.2024.3512002.
22. S. Khokha, "Integrating Safety and Security in Autonomous Vehicles: A Comprehensive Review," 2024 4th International Conference on Sustainable Expert Systems (ICSES), Kaski, Nepal, 2024, pp. 460-464, doi: 10.1109/ICSES63445.2024.10763275.