An Approach for Automotive ECU Diagnosis via Ethernet Snooping & Microcontroller Tracing

Zafer Attal
Chair of Integrated Systems
Technical University of Munich
Munich, Germany

Matthias Ernst
Infineon Technologies AG
Munich, Germany

Gasper Skvarc Bozic
Infineon Technologies AG
Munich, Germany

Ibai Irigoyen Ceberio
Infineon Technologies AG
Munich, Germany

Albrecht Mayer
Infineon Technologies AG
Munich, Germany

Thomas Wild
Chair of Integrated Systems
Technical University of Munich
Munich, Germany

Andreas Herkersdorf
Chair of Integrated Systems
Technical University of Munich
Munich, Germany

***Abstract*—The increasing software complexity in modern vehicles necessitates diagnostic capabilities beyond traditional systems. This paper presents a Diagnosis Unit (DU) that supports runtime detection and analysis of anomalies by correlating irregularities in Ethernet communication with ECU-internal processing behavior. The DU captures execution traces upon detecting anomalous communication and performs localized analysis to assist in uncovering potential root causes. Implemented on a ZCU102 platform and interfaced with Aurix ECUs, the prototype effectively detects both communication and processing anomalies with minimal impact on in-vehicle network bandwidth, supporting scalable, adaptive, and non-intrusive in-vehicle diagnostics.**

Keywords- Automotive, Diagnostics, Health Monitoring, Anomaly Detection, Trace Analysis

I. Introduction

Modern vehicles are evolving into software-defined systems, built on complex architectures with hundred of interconnected Electronic Control Units (ECUs) [1]. As vehicle functionality—from driver assistance to autonomous operation—relies heavily on software, the associated computational demands introduce significant challenges for software reliability and, consequently, for fault diagnosis. Traditional On-Board Diagnostics (OBD) systems [2], though effective for hardware faults, are not designed to detect transient, software-induced anomalies in real-world operation.

Recent diagnostic methods have begun addressing these limitations, yet they often fall short in correlating anomalies between network-level symptoms and ECU-internal behaviors. This gap is critical, as many communication irregularities may reflect deeper malfunctions within individual ECUs or their subsystem interactions.

We present a Diagnosis Unit (DU) that supports correlation-based fault analysis by monitoring in-vehicle Ethernet communication and retrieving execution traces from the responsible ECU. Integrated in a non-intrusive manner at a gateway or central service node (Fig. 1), the DU performs targeted, online trace-based analysis with minimal impact on system behavior, identifying irregularities as they manifest on the IVN.

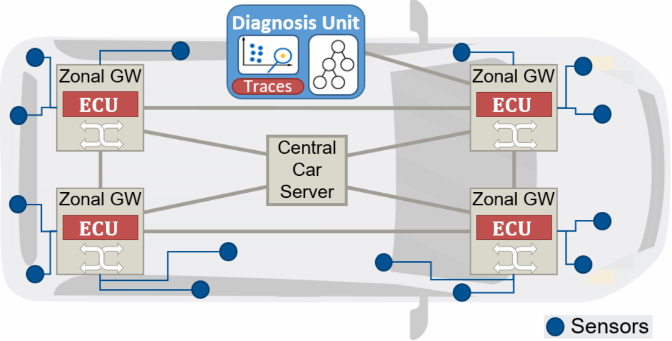


Figure 1. Diagnosis Unit deployment in a IVN.

Implemented on a ZCU102 platform and interfaced with Aurix ECUs, our prototype demonstrates the feasibility of detecting and correlating communication and ECU-internal processing anomalies.

The rest of this paper covers related work (II), the DU concept (III), system architecture and demonstration setup (IV), detection results and system performance (V), and conclusions with future directions (VI).

II. Related Work

Traditional diagnostic systems such as On-Board Diagnostics (OBD-II) are effective for hardware-level faults, but they are not designed to address dynamic, software-induced anomalies in modern vehicles [2]. As software complexity in in-vehicle systems increases, researchers have explored complementary diagnostic approaches.

Cloud-centric systems enhance diagnostic coverage by offloading data to backend processors for deeper analysis [3]. However, this approach incurs high bandwidth costs and cannot

This work was funded by the German Federal Ministry of Education and Research (BMBF) under grant number 16ME0457 (EMDRIVE).

Manuscript received April 30, 2025; revised July 23, 2025; accepted July 25, 2025. Published September 2, 2025.
Issue category: Special Issue on DSD/SEAA 2025 on Works in Progress (WiP) Session, Salerno, Italy, Sept. 2025
Paper category: Short
DOI: doi.org/10.64552/wipiec.v11i2.96

provide timely responses within the vehicle. Similarly, automated trace preprocessing frameworks, as shown in [4], offer rich analysis capabilities but are typically designed for offline use during system validation and lack feasibility for deployment in online scenarios within operational vehicles.

Vehicle Health Monitoring Systems (VHMSs) incorporate predictive diagnostics through sensor analytics and machine learning [5][6]. However, they often focus on individual subsystems and struggle to correlate behavior across architectural domains. Similarly, advanced anomaly detection models [7][8] emphasize pattern recognition in communication or control flows but fall short in identifying causal relationships between network anomalies and ECU processing behavior.

In contrast, the proposed Diagnosis Unit (DU) operates locally and autonomously within the vehicle. It monitors Ethernet communication for anomalies and retrieves ECU execution traces to analyze them for potential correlations. Rather than replacing existing diagnostics, the DU complements them by delivering runtime insights that support the identification of potential root causes.

III. Design and Operating Principles

A. Diagnosis Unit Subsystems

The DU comprises three tightly integrated components:

- **Gateway Snooping:** Passively monitors mirrored Ethernet traffic to detect timing or behavioral anomalies without disrupting normal operation.
- **Trace Control System:** Upon detecting an anomaly, the DU identifies the affected ECU and initiates trace recording via its Tool Access Socket (TAS) server. This requires ECUs to support hardware tracing and tooling for remote trace configuration and retrieval.
- **Trace Analyzer:** Retrieved traces are analyzed during runtime to identify irregularities such as delayed functions, excessive execution time, or control-flow deviations potentially linked to the observed communication anomaly.

This modular structure supports localized, event-driven diagnostics without requiring continuous cloud connectivity. The current prototype operates autonomously, with all core logic implemented on a ZCU102 platform.

B. Timing Requirements for Effective Trace Capture

Effective diagnosis depends on capturing the relevant processing history in the trace buffer corresponding to the observed anomaly. This observable history depends on the size of the trace buffer and on the tracing granularity—i.e., how many trace events are recorded per time unit. To ensure the trace includes the necessary context, the Recording Window (T_{RW}) must satisfy:

$$T_{RW} \geq \Delta t_{AProp} + \Delta t_{DU} \quad (1)$$

Here, Δt_{AProp} represents the internal propagation delay before a processing anomaly manifests on the network. Δt_{DU} includes anomaly detection, identification of the source ECU, and the time to stop tracing and initiate trace retrieval. Given the limited size of the trace buffer and the risk of overwriting older entries, this constraint ensures the capture of causally relevant events. Additionally, Δt_{TT} denotes the time required to transfer the trace from the ECU to the DU, though it does not impact the critical timing path for trace preservation. Fig. 2 illustrates the timing relationship between these components.

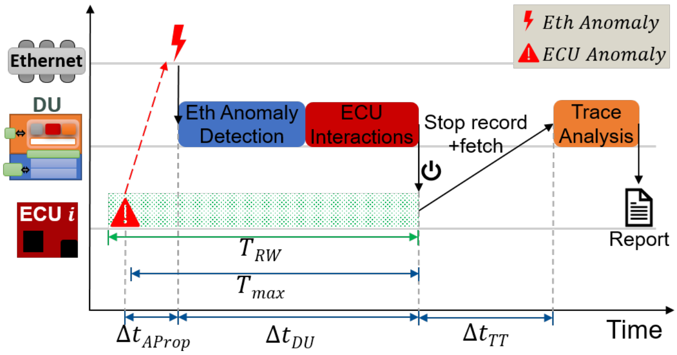


Figure 2. Timing coordination between anomaly detection and trace recording.

C. Diagnostic Advantages

The DU enables localized correlation between communication symptoms and processing anomalies during vehicle operation, eliminating the need for continuous cloud uploads or offline trace post-processing. It operates non-intrusively—without ECU software instrumentation—and integrated via gateway snooping and standard debug interfaces, assuming an existing ECU tracing subsystem. These features support future extensions such as cloud-assisted reconfiguration and adaptive anomaly classification for scalable, fleet-wide diagnosis.

IV. System Architecture & Demonstration Setup

A. Diagnosis Unit Implementation

The Diagnosis Unit (DU) is prototyped on a Xilinx ZCU102 board, which integrates a Zynq UltraScale+ MPSoC featuring programmable logic and a quad-core ARM Cortex-A53 processing system [9]. This heterogeneous architecture enables a clear separation between time-critical data-plane functions and flexible control-plane logic. The programmable logic (PL) hosts a custom hardware module for Ethernet traffic monitoring, anomaly detection, and timestamping with cycle-level precision. The processing system (PS) runs embedded Linux and hosts the DU Manager, which coordinates the Tool Access Socket (TAS) server [10], manages trace configurations and retrieval, and performs local analysis of ECU traces.

As shown in Fig. 3, the DU connects its monitoring port to a mirroring port on the in-vehicle Ethernet switch, ensuring non-intrusive monitoring of communication traffic. Upon detecting a communication anomaly, it identifies the affected ECU and configures trace capture via a Tool Access Socket (TAS) server. Retrieved traces are analyzed locally on the DU without relying on external computation resources during runtime. After analysis, the DU generates a compacted report summarizing the detected communication and processing anomalies.

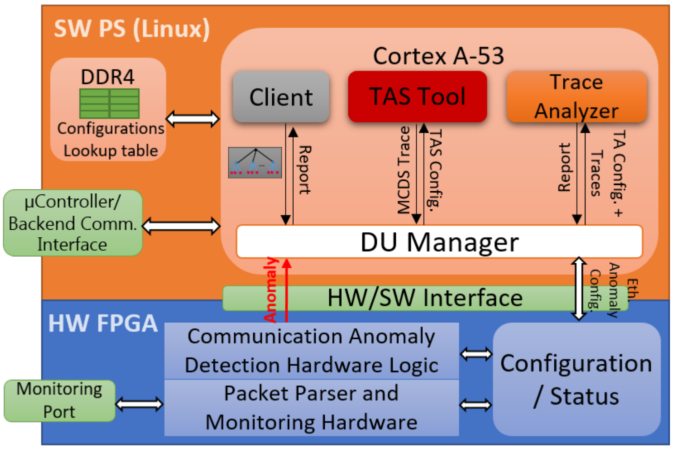


Figure 3. Modular Architecture of the Diagnosis Unit Prototype.

B. Demonstration Setup

To validate the DU's diagnostic capabilities, we developed a demonstration platform that combines real automotive hardware with simulation. The setup features three Infineon Aurix TC397 microcontroller boards, each equipped with a Multi-Core Debug Solution (MCDS) for trace recording [11]. These boards simulate different ECUs making up a distributed lane-keeping assistant application. The ECUs are connected to the CARLA simulator, which supplies real-time vehicle sensor inputs from a dynamic driving environment and receives steering commands.

The DU monitors Ethernet traffic for *communication anomalies*, including timing irregularities, missing messages, and burst structure deviations. When such anomalies are detected, the DU triggers trace collection to analyze *processing-level anomalies* such as delayed functions, task overruns, or atypical execution sequences. Fig. 4 shows the demonstration setup with integrated hardware components.

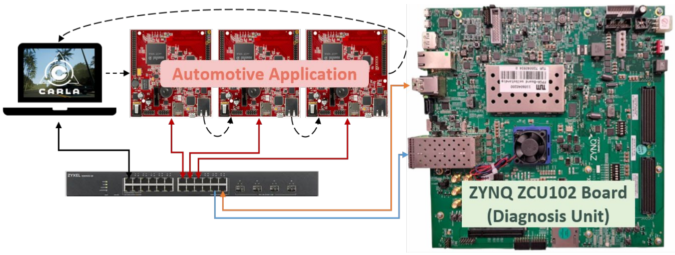


Figure 4. Demonstration Setup with Aurix ECUs and ZCU102-based Diagnosis Unit.

C. Anomaly Detection Criteria

To evaluate cross-domain detection capabilities, both communication and processing anomalies were deliberately introduced. Communication-level anomalies included timing deviations, altered periodicity, and packet drop patterns, as formally defined in Table I. Each anomaly type was detected based on predefined inequality-based thresholds for timing or packet count deviations.

TABLE I. COMM. ANOMALY TYPES MONITORED IN PROTOTYPE

Anomaly	Detection Rule
Timing Deviation Between Bursts	$T_B < TP_B - \Delta T_B$ *or* $TP_B + \Delta T_B < T_B$ [1]
Timing Deviation Between Packets	$T_P < TP_P - \Delta T_P$ *or* $TP_P + \Delta T_P < T_P$ [2]
Packet Count Deviation in Bursts	$P_{rec} < P_{exp} - \Delta P$ *or* $P_{exp} + \Delta P < P_{rec}$ [3]

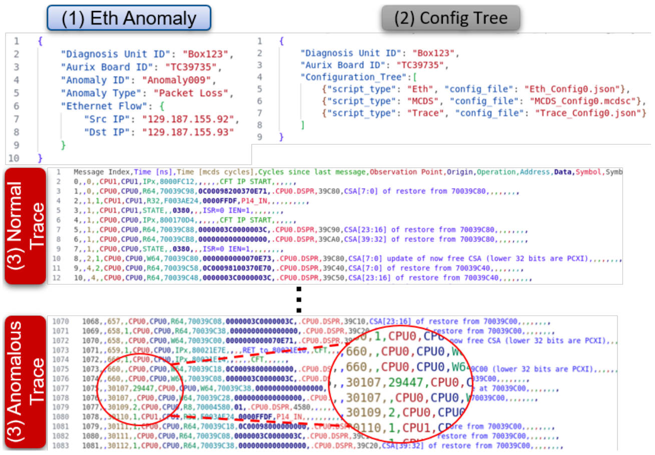
1. TP_B : Expected inter-burst interval, T_B : Observed inter-burst interval, ΔT_B : Burst corridor width threshold.
2. TP_P : Expected inter-packet interval, T_P : Observed inter-packet interval, ΔT_P : Packet corridor width threshold.
3. P_{exp} : Expected number of packets per burst, P_{rec} : Observed number of packets, ΔP: Burst size toleranc.

V. Results and Observations

A. Demonstration of Cross-Domain Anomaly Localization

Upon detecting a communication anomaly, the DU identified the source ECU and triggered trace retrieval via the TAS server. These traces enabled the analysis of related processing anomalies, such as prolonged execution delays, misordered instruction/function sequences, and irregular task load distribution. By linking anomalies in the communication domain with internal ECU behaviors, the DU demonstrated correlated, runtime insights across system domains.

Fig. 5 illustrates a trace excerpt highlighting instruction-level delays identified after a detected communication anomaly, confirming a processing deviation within the implicated ECU. Fig. 6 shows the physical prototype setup used for demonstration, featuring the ZCU102-based DU and connected Aurix ECUs with the Carla simulator as an environment for automotive application.



```
{
    "Diagnosis Unit ID": "Box123",
    "Aurix Board ID": "TC39735",
    "Anomaly ID": "Anomaly009",
    "Anomaly Type": "Packet Loss",
    "Ethernet Flow": {
        "Src IP": "129.187.155.92",
        "Dst IP": "129.187.155.93"
    }
}
```

```
{
    "Diagnosis Unit ID": "Box123",
    "Aurix Board ID": "TC39735",
    "Configuration_Tree":[
        {"script_type": "Eth", "config_file": "Eth_Config0.json"},
        {"script_type": "MCDS", "config_file": "MCDS_Config0.mcdsc"},
        {"script_type": "Trace", "config_file": "Trace_Config0.json"}
    ]
}
```

```
Message Index,Time [ns],Time [mcds cycles],Cycles since last message,Observation Point,Origin,Operation,Address,Data,Symbol,Symb
0,,0,,CPU1,CPU1,IPx,8000FC12,,,,,,CFT IP START,,,,,,
1,,0,,CPU0,CPU0,R64,70039C98,0C00098200370E71,.CPU0.DSPR,39C80,CSA[7:0] of restore from 70039C80,,,,,,,,
2,,1,1,CPU1,CPU1,R32,F003AE24,0000FFDF,P14_IN,,,,,,,,,,
3,,1,,CPU1,CPU1,STATE,,0380,,,ISR=0 IEN=1,,,,,,,,
4,,1,,CPU0,CPU0,IPx,800170D4,,,,,,CFT IP START,,,,,,
5,,1,,CPU0,CPU0,R64,70039C88,0000003C0000003C,.CPU0.DSPR,39C90,CSA[23:16] of restore from 70039C80,,,,,,,,
6,,1,,CPU0,CPU0,R64,70039CB8,0000000000000000,.CPU0.DSPR,39CA0,CSA[39:32] of restore from 70039C80,,,,,,,,
7,,1,,CPU0,CPU0,STATE,,0380,,,ISR=0 IEN=1,,,,,,,,
8,,2,1,CPU0,CPU0,W64,70039C80,0000000000070E73,.CPU0.DSPR,39C80,CSA[7:0] update of now free CSA (lower 32 bits are PCXI),,,,,,,,
9,,4,2,CPU0,CPU0,R64,70039C58,0C00098100370E70,.CPU0.DSPR,39C40,CSA[7:0] of restore from 70039C40,,,,,,,,
10,,4,,CPU0,CPU0,R64,70039C48,0000003C0000003C,.CPU0.DSPR,39C50,CSA[23:16] of restore from 70039C40,,,,,,,,
```

```
1068,,657,,CPU0,CPU0,R64,70039C08,0000003C0000003C,.CPU0.DSPR,39C10,CSA[23:16] of restore from 70039C00,,,,,,,,
1069,,658,1,CPU0,CPU0,R64,70039C38,0000000000000000,.CPU0.DSPR,39C2                       restore from 70039C00,,,,,,,,
1070,,658,,CPU0,CPU0,W64,70039C00,0000000000070E71,.CPU0.DSPR,3                            now free CSA (lower 32 bits are PCXI),,,
1071,,659,1,CPU0,CPU0,IPx,80021E7E,,,,RET to 80021E10,,CFT,,,
1072,,660,1,CPU0,CPU0,IPx,80021E14,,,,,,CFT,,,,,,
1073,,660,,CPU0,CPU0,W64,70039C18,0C00098000000000,.CPU0.DS                                 9C00 (lower 32 bits are PCXI),,,,,,
1074,,660,,CPU0,CPU0,W64,70039C08,0000003C0000003C,.CPU0.D                                   39C00,,,,,,,,
1075,,30107,29447,CPU0,CPU0,W64,70039C38,0000000000000000,                                  at 70039C00,,,,,,,,
1076,,30107,,CPU0,CPU0,W64,70039C28,0000000000000000,.CPU0.                                  70039C00,,,,,,,,
1077,,30109,2,CPU0,CPU0,R8,70004580,01,.CPU0.DSPR,4580,,,,,
1078,,30110,1,CPU1,CPU1,R32,F003AE24,0000FFDF,P14_IN,,,,,,,,,,
1079,,30111,1,CPU0,CPU0,R64,70039C18,0C00098000000000,.CPU0.DSPR                              re from 70039C00,,,,,,,,
1080,,30111,,CPU0,CPU0,R64,70039C08,0000003C0000003C,.CPU0.DSPR,39C                            restore from 70039C00,,,,,,,,
1081,,30112,1,CPU0,CPU0,R64,70039C38,0000000000000000,.CPU0.DSPR,39C20,CSA[39:32] of restore from 70039C00,,,,,,,,
```

Figure 5. Trace Analysis and Configuration Tree Output from the DU Prototype.

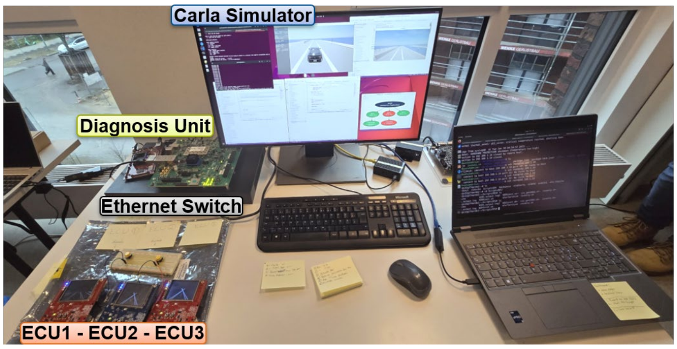


Figure 6. Physical demonstration of the Diagnosis Unit.

B. Detection Performance and Timing Behavior

To evaluate the responsiveness of the Diagnosis Unit, we measured its reaction time—defined here as the interval between the appearance of a communication anomaly on the Ethernet interface and the moment trace recording is halted. This time is critical to ensure that the trace buffer still retains the relevant processing history preceding the anomaly. While the propagation time from processing anomaly to their network manifestation is application-dependent, our measurement focuses on the DU's ability to respond quickly once a network-level deviation is observed.

Across multiple runs and under the configured trace buffer size, the DU required between 100 and 500 milliseconds to respond—depending on the recording granularity and the number of ECUs being traced concurrently. These values align with the buffer timing constraints outlined in Section III-B, ensuring adequate preservation of pre-anomaly trace context.

Trace retrieval occurred at approximately 6 MBps via the debug interface per ECU, indicating low bandwidth requirements on the IVN—an important factor in maintaining system non-intrusiveness. The subsequent local analysis of the retrieved trace typically completed within 300 milliseconds on average per anomaly case, depending on the trace length and granularity of recorded events. These results suggest that the DU is capable of real-time diagnosis while introducing low processing or communication overhead, supporting scalable in-vehicle deployment.

C. System Constraints and Prototype Limitations

The current prototype implementation presents practical constraints affecting diagnostic coverage and flexibility. A primary limitation is the 2~MB trace buffer per ECU, which restricts the retained processing history—particularly under fine-grained trace configurations where verbose logging can saturate the buffer. This bounds the diagnostic window and necessitates precise coordination between anomaly detection and trace retrieval.

While the DU performs local analysis independently of backend connectivity, its current evaluation logic is limited to predefined rule-based models. Planned backend integration—for dynamic rule updates and multi-vehicle correlation—was not included in the evaluated prototype. Similarly, advanced diagnostic methods, such as statistical learning or adaptive behavioral profiling, remain future work.

VI. Conclusion and Outlook

This work presents a cross-domain diagnostic approach—embodied in our Diagnosis Unit (DU)—that enables runtime anomaly detection in automotive systems by correlating communication anomalies with internal ECU processing behavior. Implemented on a ZCU102 platform and validated through a distributed lane-keeping assistant setup, the DU demonstrated its ability to localize anomalies efficiently and with low bandwidth overhead.

The DU is integrated via a mirrored switch port and debug interfaces, enabling non-intrusive deployment without requiring software modifications. Its modular design supports local, on-demand trace analysis, enhancing in-vehicle observability while minimizing reliance on backend infrastructure.

For fleet-scale deployment, distributed DUs autonomously detect and correlate anomalous events, forwarding concise reports to the backend. This low-bandwidth setup reduces network load, preserves privacy, and enables scalable diagnostics. Future versions will support cloud connectivity for remote control of diagnosis policies, result aggregation, and dynamic detection model updates.

Several enhancements are envisioned to improve the DU's precision and adaptability: (i) Semantic-level anomaly detection, such as recognizing out-of-range signals (e.g., sensor or actuator values); (ii) Learning-based classification to handle evolving or sporadic faults; (iii) Secure trace handling and integration with IVN security mechanisms to support encrypted communication monitoring.

Finally, for cost-effective deployment in production vehicles, we propose embedding DU functionalities directly into gateway Network Interface Controllers (NICs). Together, these improvements aim to deliver a scalable, resilient, and secure diagnostic infrastructure suited for the growing complexity of modern automotive systems.

Acknowledgment

This work was funded by the German Federal Ministry of Education and Research (BMBF) under grant number 16ME0457 (EMDRIVE). The responsibility for the content of this publication lies with the authors.

References

[1] SAE J3016 automated-driving graphic. Last Modified: 2020-05-15T14:03:15-04:00.

[2] Deepa Saibannavar, Mallikarjun M. Math, and Umakant P. Kulkarni. A survey on on-board diagnostic in vehicles. 2020.

[3] Malintha Amarasinghe, Sasikala Kottegoda, Asiri Liyana Arachchi, Shashika Ranga Muramudalige, Herath Mudiyanselage Nelanga Dilum Bandara, and Afkham Azeez. Cloud-based driver monitoring and vehicle diagnostic with obd2 telematics. 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer), pages 243–249, 2015.

[4] Artur Mrowca, Thomas Pramsohler, Sebastian Steinhorst, and Uwe Baumgarten. Automated interpretation and reduction of in-vehicle network traces at a large scale. 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), pages 1–6, 2018.

[5] Md. Arafatur Rahman, Md. Abdur Rahim, Md. Mustafizur Rahman, Nour Moustafa, Imran Razzak, Tanvir Ahmad, and Mohammad N. Patwary. A secure and intelligent framework for vehicle health monitoring exploiting big-data analytics. IEEE Transactions on Intelligent Transportation Systems, 23:19727–19742, 2022.

[6] Uferah Shafi, Asad Ali Safi, Ahmad Raza Shahid, Sheikh Ziauddin, and Muhammad Qaiser Saleem. Vehicle remote health monitoring and prognostic maintenance system. Journal of Advanced Transportation, 2018:1–10, 2018.

[7] ¨Ovg¨u ¨Ozdemir, M. Tu˘gberk ˙Is¸yapar, Pınar Karag¨oz, Klaus Werner Schmidt, Demet Demir, and N. Alpay Karag¨oz. A survey of anomaly detection in in-vehicle networks. arXiv preprint arXiv:2409.07505, 2024.

[8] Paolo Dini, Sergio Saponara, Carlo Rosadini, Walter Nesci, and Stefano Chiarelli. Anomaly and intrusion detection algorithms for can-bus networking security in automotive applications. Automotive SPIN Italia Workshop, 2023.

[9] AMD. Zcu102 evaluation kit board user guide, 2024. Accessed: 2024-08-08.

[10] A new generation automotive tool access architecture for remote in-field diagnosis. SAE Technical Paper Series, 2023.

[11] Lauterbach GmbH. Multi-core debug solution user's guide, 2024. Accessed: 2024-08-08. G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.