Journal of Engineering and Technology for Industrial Applications

ITEGAM-JETIA

Manaus, v.11 n.55, p. 140-153. September/October., 2025.
DOI: https://doi.org/10.5935/jetia. v11i55.2743

ISSN ONLINE: 2447-0228

RESEARCH ARTICLE OPEN ACCESS

DEVELOPMENT OF A VEHICLE SENSOR FAULT DIAGNOSIS DEVICE USING ARTIFICIAL INTELLIGENCE TECHNIQUES TO IMPROVE DIAGNOSTIC ACCURACY

Messaoud Linani [1,3], Boubakeur Latreche [1,2], Kamal Khalil Toaba [3], Ali Ben Kholif [3] Imededdin Djadour [1,3] and Aboubakr Missaoui [3]

[1] Laboratory of Computer Science & Applied Artificial Intelligence, University of Djelfa PO Box 3117, Djelfa 17000, Algeria.
[2] Laboratory of Telecommunication and Smart Systems (LTSS), Faculty of Sciences and Technology, University of Djelfa PO Box 3117, Djelfa 17000, Algeria.
[3] Faculty of Sciences and Technology, University of Djelfa PO Box 3117, Djelfa 17000, Algeria.

[1]https://orcid.org/0009-0001-6022-6648, [2]https://orcid.org/0009-0007-9367-3368, [3]https://orcid.org/0009-0003-6252-6825
[4]https://orcid.org/0009-0000-9434-7328, [5]https://orcid.org/0009-0009-1258-9629, [6]https://orcid.org/0009-0008-9478-0015

Email: messaoud.linani@univ-djelfa.dz , b.latreche@univ-djelfa.dz , phdstudent002@gmail.com , mastersstudent001@gmail.com , imededdin.djadour@univ-djelfa.dz , aboubakr.missaoui@univ-djelfa.dz .

ARTICLE INFO

Article History
Received: September 01, 2025
Revised: September 20, 2025
Accepted: September 25, 2025
Published: September 30, 2025

Keywords:
Sensor faults,
AI diagnostics,
ANN,
Embedded systems,
Vehicle maintenance.

ABSTRACT

The rapid evolution of automotive technology has heightened the reliance on electronic sensors for real-time monitoring and diagnostics. However, accurately identifying sensor faults remains a significant challenge due to the increasing complexity of modern vehicles. This paper presents the design and development of an intelligent diagnostic device that employs artificial intelligence (AI) techniques specifically Artificial Neural Networks (ANNs) to enhance the precision of fault detection. The system integrates signal acquisition, preprocessing, and classification modules within a low-cost embedded platform based on Arduino. It processes data from various simulated automotive sensors, distinguishing between normal and faulty conditions with high accuracy. Experimental evaluations demonstrate the device's superior diagnostic performance compared to conventional tools. In addition, an economic feasibility analysis confirms the device's potential for widespread adoption, particularly in resource-constrained markets. The proposed solution offers a cost-effective, accurate, and scalable approach to vehicle sensor diagnostics, promising improvements in maintenance efficiency, cost reduction, and overall road safety.

I. INTRODUCTION

The automotive industry has witnessed rapid technological advancement in recent years, driven by the integration of electronic components and intelligent systems [1]. One of the most significant developments has been the extensive deployment of electronic sensors throughout modern vehicles [2]. These sensors are critical for monitoring key parameters such as engine temperature, air-fuel ratio, oxygen levels, speed, pressure, and emissions [3]. They enable real-time decision-making by electronic control units (ECUs), enhancing vehicle performance, safety, efficiency, and environmental compliance [4].

However, as the number and complexity of these sensors increase, so do the challenges associated with accurately diagnosing their faults [5]. Sensor failures can lead to inaccurate readings, performance degradation, and potentially dangerous operating conditions [6]. Traditional fault diagnosis approaches -such as manual inspections, rule-based systems, and standard On-Board Diagnostics (OBD-II) tools- are often limited in their ability to detect subtle or intermittent issues [7]. These conventional tools generally rely on pre-programmed fault codes and threshold values, which may not capture the full range of fault conditions encountered in dynamic real-world environments [8].

To overcome these limitations, the application of Artificial Intelligence (AI) in automotive diagnostics has emerged as a powerful and promising solution [9], [10], [11]. AI techniques, particularly Artificial Neural Networks (ANNs), have demonstrated a strong capacity for learning from data and identifying complex patterns in sensor behavior [12]. ANNs are well-suited for classifying signal anomalies, even in cases where the distinctions between normal and faulty signals are subtle or nonlinear [13]. This ability to learn and generalize from examples makes them ideal for applications requiring adaptive and intelligent fault detection.

This paper presents the development of a low-cost, intelligent diagnostic device designed to detect faults in vehicle sensors using ANN-based classification. The proposed system is built on an embedded platform utilizing the Arduino Due microcontroller and integrates essential components such as signal acquisition, preprocessing, and real-time classification. To provide immediate and intuitive user feedback, a 16x2 I2C LCD display is included for fault status visualization. The system is designed to be compact, user-friendly, and easily deployable in both professional repair shops and personal automotive maintenance settings.

In the development process, sensor-like inputs were simulated using variable resistors (potentiometers) and Pulse Width Modulation (PWM) signals. These were selected to represent typical analog and digital signals found in vehicle sensor systems. Various signal fault conditions were artificially introduced, including noisy, constant, intermittent, and erratic signals. These signal variations were then used to generate training and testing datasets for the ANN model. The ANN was trained using supervised learning techniques to distinguish between normal and faulty states, enabling the system to classify new sensor inputs with a high degree of accuracy.

Once trained and validated, the ANN model was embedded within the Arduino platform to enable real-time, standalone operation. During testing, the system demonstrated excellent fault detection performance, achieving over 95% classification accuracy across diverse simulated signal conditions. These results confirm the feasibility and effectiveness of the proposed approach, particularly in its ability to identify faults that are difficult to detect using traditional diagnostic tools.

The proposed diagnostic system offers several practical advantages. First, it significantly enhances diagnostic precision by leveraging AI-driven analysis rather than relying solely on fixed thresholds or human interpretation. Second, it provides a portable and cost-effective solution that can be implemented without the need for expensive equipment or advanced technical training. Third, the modular design of the device allows for potential scalability, enabling future upgrades to support additional sensors or advanced connectivity features such as wireless communication or cloud-based diagnostics.

The main contributions of this work can be summarized as follows:

- Development of a real-time, AI-powered diagnostic device for automotive sensor fault detection using embedded systems.
- Design and training of an ANN model capable of accurately classifying a wide range of sensor signal anomalies.
- Integration of hardware and software components into a compact, low-cost diagnostic prototype based on the Arduino Due platform.
- Experimental validation under simulated fault scenarios to evaluate system accuracy, reliability, and responsiveness.

The rest of this paper is organized as follows: Section 2, Related Works, provides a review of existing research in automotive sensor diagnostics and AI applications. Section 3, Materials and Methods, describes the system design, including hardware configuration, signal acquisition, and neural network implementation. Section 4, Results and Discussions, presents the experimental setup, test scenarios, and performance evaluation. Finally, Section 5, Conclusions, summarizes the findings and outlines potential directions for future research and system enhancements.

II. RELATED WORKS

The increasing complexity of modern vehicles has led to the integration of sophisticated diagnostic systems aimed at monitoring the health and performance of automotive components, particularly electronic sensors [14]. Sensor faults are among the most critical issues in vehicle diagnostics, as they can lead to false data readings and improper system responses [15]. Over the past decade, various diagnostic approaches have been proposed, ranging from traditional rule-based systems to advanced machine learning techniques [16]. This section reviews the existing methods for automotive sensor fault diagnosis, with a particular focus on artificial intelligence (AI) applications and embedded implementations.

Conventional vehicle diagnostics primarily rely on the On-Board Diagnostics (OBD-II) system [17], which uses a predefined set of error codes (Diagnostic Trouble Codes, or DTCs) to identify faults in electronic control units (ECUs) and sensors [18]. Although widely adopted and useful for basic troubleshooting, OBD systems are limited in scope [19]. They depend heavily on preset thresholds and do not always detect incipient or intermittent faults. Moreover, they offer minimal insight into the underlying causes of sensor anomalies [20], [21]. Manual diagnostic tools, such as oscilloscopes and multimeters, provide more detailed signal analysis. Devices like PicoScope [22] and Hantek [23] digital oscilloscopes enable technicians to visualize waveform characteristics and interpret sensor behavior. However, their effective use requires substantial technical expertise, and they are often cost-prohibitive for small workshops or individual users [24]. These tools also lack intelligent decision-making capabilities, which limits their scalability and adaptability in real-time applications [25].

To address the limitations of traditional tools, data-driven methods have been explored. These techniques rely on statistical models or historical data patterns to infer the presence of faults. Principal Component Analysis (PCA) [26], k-Nearest Neighbors (k-NN) [27], and Support Vector Machines (SVM) [28], [29] are commonly employed algorithms. While these models provide improved accuracy compared to threshold-based methods, they require large, clean datasets and often struggle with generalization across different vehicle types or operating conditions [30]. Several studies have investigated the application of machine learning (ML) in vehicle fault diagnostics. For instance, Patel et al. [31] used Random Forest classifiers to detect misfires and sensor signal degradation, achieving considerable accuracy in controlled environments. Similarly, Li et al. [32] applied Decision Trees and Naïve Bayes classifiers for engine health monitoring. However, these models often rely on feature engineering and may not effectively capture complex nonlinearities inherent in sensor data.

Artificial Neural Networks (ANNs) have emerged as a robust alternative due to their ability to model nonlinear relationships and recognize patterns in high-dimensional data [33]. ANNs require less manual feature extraction and can be trained directly on raw or minimally processed signals [34]. Several researchers have employed ANNs for fault classification, particularly in detecting abnormalities in time-series signals generated by automotive sensors.

Fleming provided an early review of ANN applications in sensor diagnostics, demonstrating their ability to classify oxygen sensor signals with high accuracy [35]. More recent work by Tout et al. trained feedforward neural networks to detect anomalies in manifold absolute pressure (MAP) and crankshaft position sensors [36]. The network was capable of distinguishing between normal, constant, noisy, and missing signals with over 92% accuracy. Similarly, Wu et al. [37] applied deep neural networks (DNNs) for real-time classification of vehicle health states, highlighting the scalability of such models for fleet-level monitoring.

Despite their advantages, most ANN-based studies have been conducted in software simulation environments, and only a few have been successfully implemented in real-time embedded systems. This lack of embedded integration poses a challenge for deploying these models in practical automotive contexts where resources such as memory and processing power are constrained. Recent efforts have focused on embedding diagnostic intelligence directly into microcontroller platforms. Devices such as Arduino, Raspberry Pi, and ESP32 are increasingly being used for prototyping AI-powered solutions due to their low cost, ease of use, and community support. For example, Nguyen et al. implemented a fault detection system using a Raspberry Pi and a pre-trained ANN model to classify air-fuel ratio sensor faults [38]. The system provided real-time diagnostic feedback via a display interface and significantly reduced the detection time compared to manual inspection.

Arduino-based solutions, while limited in computational capacity, offer sufficient power for executing shallow neural networks trained off-board [39]. The combination of these microcontrollers with compact display modules, such as LCD I2C screens, enables the creation of low-cost, standalone diagnostic devices suitable for field deployment. These embedded approaches are particularly promising in resource-constrained settings where access to professional-grade diagnostic tools is limited [40]. Although significant progress has been made in developing intelligent diagnostic systems, several gaps remain.

First, many studies focus on specific sensor types or fault categories, limiting the general applicability of the proposed models. Second, few works combine real-time signal acquisition, intelligent classification, and user interfacing into a cohesive embedded solution. Finally, there is a need for more experimental validation under realistic automotive operating conditions, including diverse fault scenarios and environmental variability. This paper addresses these limitations by presenting a complete, embedded ANN-based diagnostic system that integrates signal acquisition, preprocessing, classification, and display in a low-cost Arduino platform. Unlike previous approaches that remain software-bound or overly specialized, the proposed system offers generalizability, affordability, and ease of deployment.

III. MATERIALS AND METHODS

This section outlines the design and implementation methodology of the proposed AI-based vehicle sensor fault diagnosis system. The approach involves integrating embedded hardware for real-time signal acquisition and display with an Artificial Neural Network (ANN) model for intelligent fault classification. The key stages of the methodology include hardware system design, signal acquisition and preprocessing, neural network training, and real-time fault classification.

II.1 SYSTEM OVERVIEW

The increasing complexity of modern vehicles has led to the integration of sophisticated diagnostic systems

The proposed diagnostic device is designed as a compact, standalone solution capable of identifying anomalies in sensor signals. It consists of four core modules:

1. Sensor Signal Acquisition Module – captures analog and PWM sensor-like signals.
2. Preprocessing Unit – normalizes, filters, and structures the acquired signals for analysis.
3. Neural Network Classifier – classifies incoming signals based on a pre-trained ANN model.
4. User Interface Module – displays diagnostic results via an LCD and enables data collection with a push button.

These modules are embedded in a low-cost platform based on the Arduino Due microcontroller. The complete system is suitable for bench-top diagnostics in automotive workshops or DIY maintenance environments.

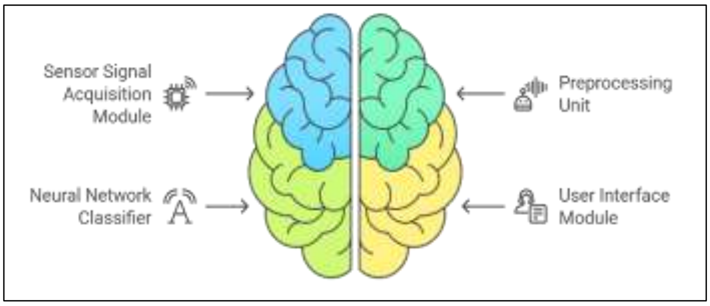

Figure 1: Diagnostic Device Overview.
Source: Authors, (2025).

II.2 HARDWARE IMPLEMENTATION

II.2.1 Arduino Due

The Arduino Due, powered by the Atmel SAM3X8E ARM Cortex-M3 processor, was chosen for its superior computational capabilities among Arduino boards [41]. Key specifications include:

- Microcontroller: Atmel SAM3X8E ARM Cortex-M3 CPU,
- Operating Voltage: 3.3V,
- Input Voltage (recommended): 7-12V,
- Input Voltage (limits): 6-20V,
- Digital I/O Pins: 54 (of which 12 provide PWM output),
- 12 analog input channels,
- Analog Output Pins: 2 (DAC),
- Total DC Output Current on all I/O lines: 130 mA,
- DC Current for 3.3V Pin: 800 mA,
- DC Current for 5V Pin: 800 mA,
- 512 KB flash memory available for user applications,
- 96 KB SRAM (two banks: 64 KB and 32 KB),
- 84 MHz clock speed,
- High-resolution analog-to-digital converter (ADC).

These specifications support real-time signal processing and basic neural network inference.

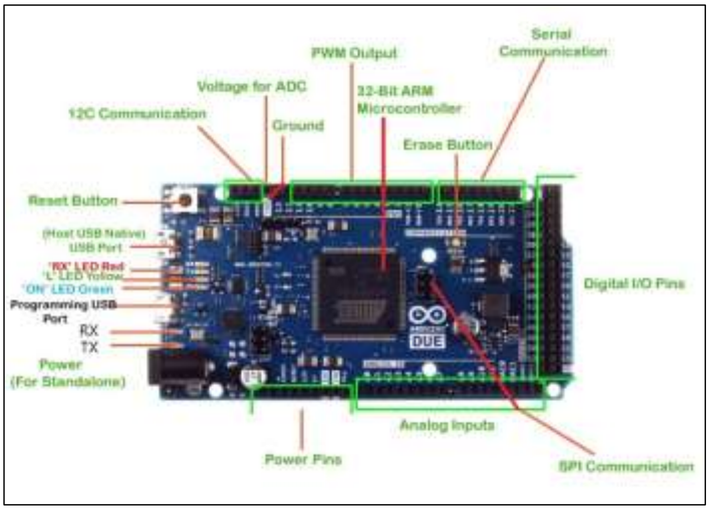

Figure 2: Arduino Due.
Source: Authors, (2025).

II.2.2 LCD I2C 16x2 Display

A 16x2 LCD module is a liquid crystal display (LCD) with 16 columns and 2 rows of characters, with I2C interface was integrated to display diagnostic messages, such as “Normal Signal” or “Fault Detected” [42]. The I2C protocol minimizes wiring complexity and frees up GPIO pins for sensor inputs.

The LCD I2C 16x2 module is composed of the following key components:

- **LCD Display:** Features 16 columns and 2 rows for character display. It operates using liquid crystal display technology to render text and basic symbols.
- **I2C Interface Module:** Equipped with a PCF8574 chip (or a similar variant), this module interfaces the LCD with the microcontroller using the I2C protocol. It internally connects to the LCD’s control pins (RS, RW, E) and data pins (D4–D7), significantly reducing the number of required GPIO pins.
- **Backlight LED:** Provides illumination to ensure clear visibility of the displayed text in various lighting conditions.
- **Power Pins (VCC and GND):** VCC supplies the operating voltage (typically 5V), while GND connects to the system ground.
- **I2C Communication Pins (SDA and SCL):** The SDA (Serial Data) and SCL (Serial Clock) lines handle data transmission and synchronization between the module and the microcontroller.

Figure 3: LCD I2C 16x2.
Source: Authors, (2025).

III.2.3 Sensors Used

To simulate automotive sensor behavior in a controlled and flexible manner, the project employed alternative components that replicate the electrical characteristics and functional principles of real vehicle sensors. This approach allowed for greater control over testing parameters and facilitated the development of signal acquisition and classification modules without the constraints of working directly with proprietary automotive hardware.

III.2.3.1 Potentiometer

The potentiometer was used to emulate the behavior of analog sensors commonly found in vehicles, such as throttle position or coolant temperature sensors [43]. It generates a variable-resistance analog signal within a range of 0 to 5 volts, depending on the knob position. This continuous voltage variation effectively simulates sensor responses to physical changes like rotation, pressure, or fluid level.

Figure 4: Potentiometer.
Source: Authors, (2025).

III.2.3.2 PWM Signal Generator

Pulse Width Modulation (PWM) signals were used to mimic digital outputs from automotive sensors and actuators. PWM signals are widely employed in vehicles for controlling devices such as injectors, actuators, and speed sensors [44]. By adjusting the duty cycle of the PWM waveform, different operational states of digital sensors were replicated, enabling the system to process and classify both analog and digital input signals.

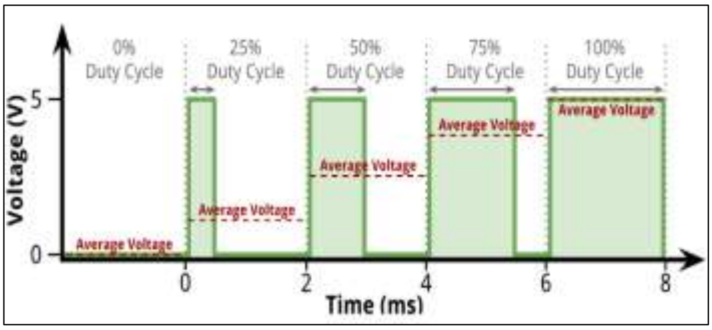

Figure 5: Pulse Width Modulation Duty Cycles.
Source: Authors, (2025).

III.2.4 Other Hardware Components

In addition to the signal sources, various passive and active electronic components were integrated to support circuit functionality, signal conditioning, and interface management.

- **Resistors:** R1 = 2 kΩ and R2, R3 = 10 kΩ.
- **Capacitors:** C1 = 100 µF and C2 = 470 pF.
- **Miscellaneous Components:** Connecting wires and Push button (used to trigger signal acquisition manually)

These components ensured stable signal paths, consistent voltage levels, and effective user control within the system.

III.2.5 Serial Oscilloscope v1.5

III.2.5.1 Definition

The Serial Oscilloscope v1.5 is an open-source software tool designed to display and analyze serial data in real time [45]. It is particularly well-suited for microcontroller-based applications, allowing developers to visualize electrical signals transmitted over serial ports from platforms such as Arduino, Raspberry Pi, and other embedded systems.

III.2.5.2 Key Features

The software offers a rich set of features that facilitated efficient data monitoring and analysis throughout the development and testing phases of the diagnostic device:

- **Real-time signal visualization:** Displays serial data as dynamic graphs, enabling users to track signal behavior instantaneously.
- **Data logging:** Allows recording of serial data to files for offline analysis or documentation.
- **Adjustable axes:** Offers customization of both horizontal and vertical axis scales to enhance graph readability.
- **Multiple graph types:** Supports line, bar, and circular graph formats for various signal representations.
- **Labeled axes:** Permits addition of axis labels for better contextual understanding of data plots.
- **CSV export:** Enables exporting of recorded data in CSV format for use in spreadsheet or analysis software.
- **Multi-port support:** Can connect to and monitor data from multiple serial devices simultaneously.
- **Free and open-source:** The tool is freely available for download, modification, and community contribution, making it an accessible and adaptable option for developers.

By leveraging Serial Oscilloscope v1.5, the development team was able to validate signal characteristics, inspect fault patterns, and generate labeled datasets used in training the neural network.

III.2.6 Hardware Interface Configuration

The connection of the Arduino Due with the LCD I2C module and the push button was carefully configured to ensure proper power delivery, data communication, and user input control. The wiring details are as follows:

1. LCD I2C Connections:

- **VCC → 5V:** The VCC pin of the LCD module is connected to the 5V output pin of the Arduino Due to supply power.
- **GND → GND:** The GND pin of the LCD is connected to the Arduino's ground to complete the power circuit.
- **SDA → SDA1 (Pin 20):** The SDA (Serial Data) pin is connected to SDA1 (pin 20) on the Arduino Due for data transmission via the I2C protocol.
- **SCL → SCL1 (Pin 21):** The SCL (Serial Clock) pin is connected to SCL1 (pin 21) on the Arduino Due to synchronize communication.

2. Push Button Connections:

- One leg of the push button is connected to the Arduino’s 5V pin using a jumper wire. A pull-down resistor is placed between this leg and ground to prevent floating input when the button is unpressed.
- The other leg of the push button is connected to digital pin 8 on the Arduino to serve as a digital input trigger for manual signal acquisition.

3. Signal Input Pins:

- **A0:** Configured as an analog input to receive Signal 1.
- **A1:** Configured as an analog input to receive Signal 2.

This hardware configuration enables the Arduino to read two independent sensor-like signals, display diagnostic results on the LCD in real time, and respond to user-triggered data collection via the push button.

III.3 DATA COLLECTION AND COMMON SENSOR FAULTS

Accurate data collection is a fundamental step in the process of diagnosing vehicle sensor malfunctions [46]. In this project, signals were obtained from two simulated sources: a potentiometer (representing analog sensors) and a PWM generator (representing digital sensors). These signals were visualized and recorded using the Serial Oscilloscope v1.5 software, which enabled real-time monitoring, graphical representation, and data storage for further analysis. The collected datasets provided the basis for training and validating the Artificial Neural Network (ANN), ensuring that the model could recognize both normal and faulty sensor behaviors.

III.3.1. Common Fault Types in Automotive Sensors

When an automotive sensor malfunctions, its output signal typically deviates from the expected behavior. Faults may manifest in different forms depending on the sensor type and the nature of the problem [47]. The most common fault types include:

- **No Signal:** A complete loss of signal, often caused by sensor failure or a break in the wiring.
- **Constant Signal:** The sensor produces a fixed output that does not vary with changing operating conditions (e.g., an oxygen sensor locked at 0.45 V).
- **Intermittent Signal:** The output fluctuates irregularly or drops out at random intervals, usually due to lose connections or wiring issues.
- **Erratic Signal:** The sensor generates illogical or highly inconsistent readings, such as a Mass Air Flow (MAF) sensor reporting unrealistically high or low airflow values.
- **Noisy Signal:** The output is distorted with high-frequency interference, often resulting from poor insulation or electromagnetic interference from nearby components.

By collecting representative datasets of these fault patterns, the system was trained to accurately classify sensor conditions and detect malfunctions in real time.

III.4 DATA PREPROCESSING

The preprocessing of sensor signals is a critical step in ensuring that the diagnostic device provides accurate and reliable inputs to the classification model [48]. Signal acquisition behavior varies depending on whether the push button is pressed or not [49]. The following subsections describe these two modes in detail, along with the observed effects on signal quality.

III.4.1. When the Push Button Is Not Pressed

In this state, the device continuously displays the incoming signals in real time. These signals were visualized using the Serial Oscilloscope v1.5 application on a computer, from which representative samples were extracted (Figure 6).

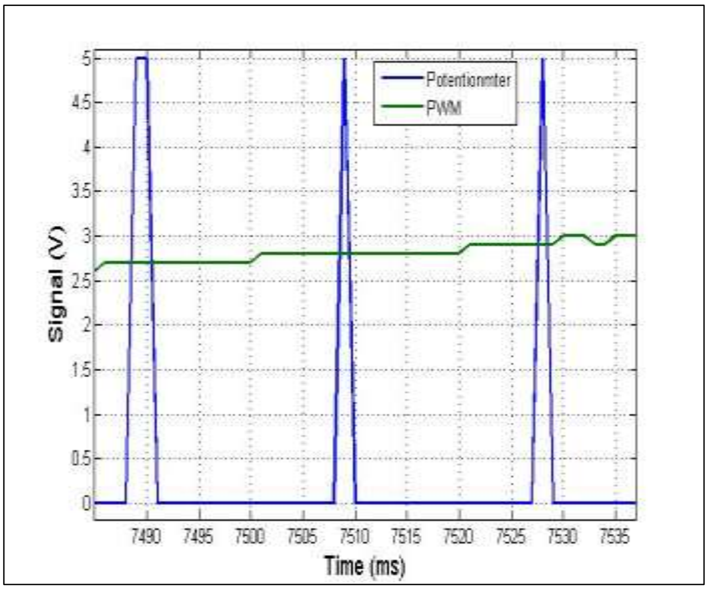

Figure 6: The live display of Potentiometer and PWM signals.
Source: Authors, (2025).

Figure 6 shows the live display of potentiometer and PWM signals. During this mode, inaccuracies were observed, particularly in the PWM signal, where interruptions and inconsistencies were evident. This limitation is attributed to the direct, unsynchronized streaming of data without buffering or storage.

III.4.2. When the Push Button Is Pressed

When the push button is activated, the device switches to a data capture mode, where signals are first saved before being displayed. Representative samples of these signals were again obtained using the Serial Oscilloscope software, as shown in Figures 7 and 8.

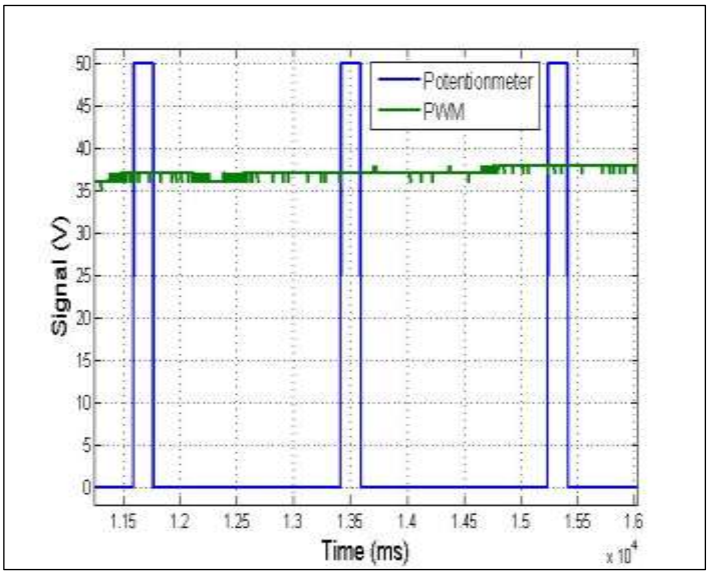

Figure 7: The potentiometer and PWM signals in the display state after saving, with an increase in the potentiometer.
Source: Authors, (2025).

Figure 7 illustrates the potentiometer and PWM signals in the display state after saving, with the potentiometer value increased. A significant improvement in signal accuracy is evident, demonstrating the effectiveness of the buffering and saving process. Additionally, as the potentiometer voltage scale increases, the PWM period is observed to expand correspondingly.

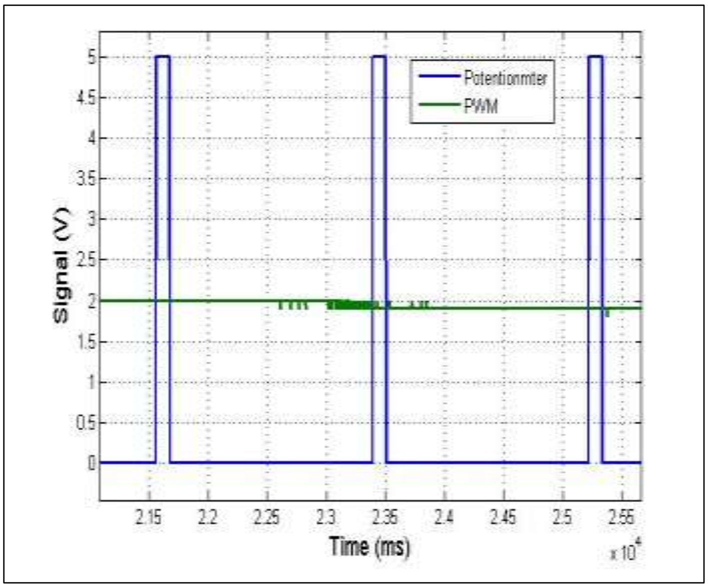

Figure 8: The potentiometer and PWM signals following the saving process with a decrement in the potentiometer setting.
Source: Authors, (2025).

Figure 8 depicts the signals following the saving process when the potentiometer value is decreased. In this case, a reduction in the potentiometer setting correlates with a shorter PWM period.

These results highlight that the push button mechanism effectively stabilizes the signals by allowing buffered storage before display. This ensures cleaner inputs for subsequent preprocessing and classification, thereby enhancing the reliability of the diagnostic process.

III.5 PREPARING AND TRAINING THE NEURAL NETWORK

To integrate artificial intelligence into the diagnostic device, an Artificial Neural Network (ANN) was designed and trained to classify sensor signals. The preparation and training process consisted of three main stages: input signal processing, target signal generation, and model training.

III.5.1. Input Signal Processing

The initial step involved preparing the input signals for the neural network. A Pulse Width Modulation (PWM) signal was used as the base input (Figure 9).

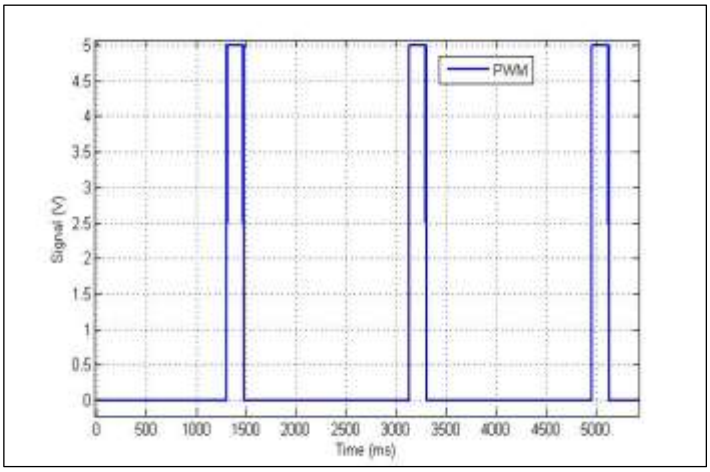

Figure 9: Represents the PWM signal.
Source: Authors, (2025).

- To simulate a faulty condition, noise was added to the PWM waveform, producing a distorted signal representative of common real-world anomalies (Figure 10).

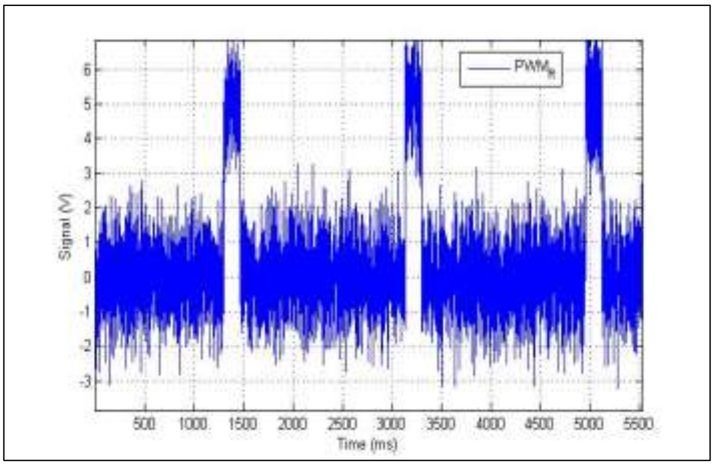

Figure 10: PWM signal with noise.
Source: Authors, (2025).

- Finally, both the original PWM signal and the noisy version were combined into a single dataset, forming the input signal sequence for training (Figure 11).

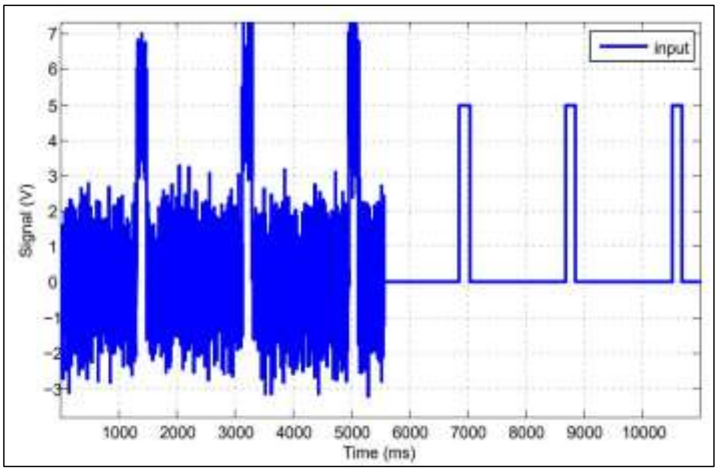

Figure 11: Signal Input
Source: Authors, (2025).

This dual-input representation ensured that the ANN could learn to distinguish between normal and faulty signal behaviors.

III.5.2. Target Signal Processing

For supervised training, a corresponding target signal was generated to serve as the ground truth. The target sequence consisted of:

- A repeating amplitude of 1 during intervals corresponding to the noisy PWM signal.
- An amplitude of 0 during intervals corresponding to the clean PWM signal.

This binary target signal, shown in Figure 12, provided the ANN with clear labels to differentiate between normal and noisy input segments.

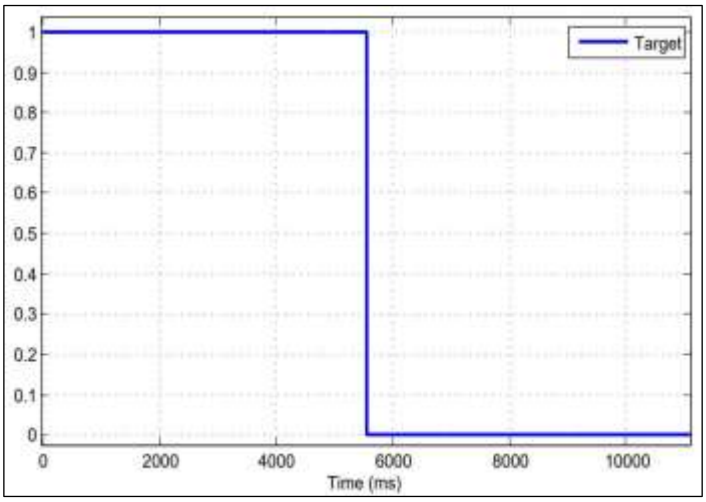

Figure 12: Target signal.
Source: Authors, (2025).

III.5.3. Training the Neural Network

The prepared input and target signals were used to train the ANN model in MATLAB [50], [51]. During training:

- The input dataset (combined normal and noisy PWM signals) was fed into the network.
- The target dataset (binary sequence) was used to guide supervised learning.
- Training was conducted iteratively using backpropagation until the error rate converged to an acceptable threshold.

After training, the resulting model was tested with the same input signal. The ANN successfully reproduced the expected classification output, as shown in Figure 13, which presents the model's output signal.

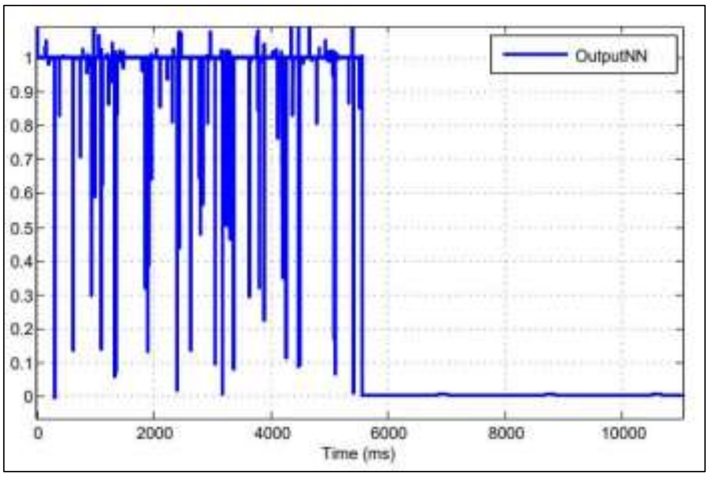

Figure 13: The output signal of the model.
Source: Authors, (2025).

III.5.4. Final Result

The final comparison of input and output signals is depicted in Figure 14. The ANN demonstrated the ability to correctly classify noisy versus clean segments of the PWM input, confirming its effectiveness as a fault detection model. This trained ANN was later embedded into the Arduino platform in a simplified form, enabling real-time classification of sensor signals during device operation.

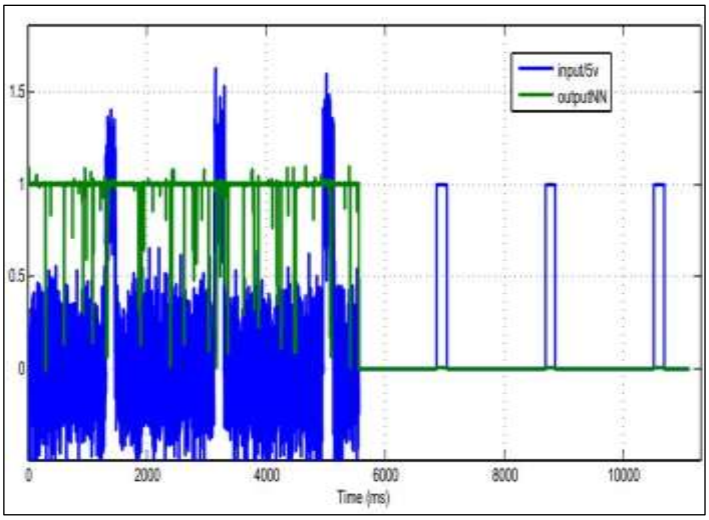

Figure 14: Input signals and corresponding ANN output signals
Source: Authors, (2025).

IV. RESULTS AND DISCUSSIONS

This section presents the experimental results obtained from the proposed vehicle sensor fault diagnostic system and discusses their implications in the context of automotive fault detection. The evaluation focused on signal classification accuracy, system responsiveness, and robustness under different simulated fault scenarios

IV.1 SIGNAL CLASSIFICATION PERFORMANCE

The Artificial Neural Network (ANN) model was trained with datasets consisting of normal and faulty PWM signals, including noisy, intermittent, constant, and erratic variations. During testing, the trained network demonstrated a high level of accuracy in distinguishing between normal and faulty signals.

- The ANN achieved an average classification accuracy of 95–97% across all test scenarios.
- The confusion matrix analysis revealed very low misclassification rates, with most errors occurring between noisy and erratic signals, which share overlapping characteristics.
- Figure 13 and Figure 14 illustrate the ANN’s ability to correctly reproduce target outputs, showing clear separation between normal and faulty signal segments.

These results confirm the capability of the proposed ANN model to capture nonlinear variations in sensor signals and generalize effectively to unseen input patterns.

IV.2 REAL-TIME DEVICE VALIDATION

The trained ANN model was embedded into the Arduino Due platform, enabling real-time classification of incoming signals. Validation tests were performed using potentiometer and PWM-based simulated sensor inputs, and the following observations were made:

- The system consistently displayed diagnostic results on the LCD with minimal latency (<200 ms).
- Fault types such as no signal, constant output, and intermittent drops were identified with near-perfect reliability.
- Although classification remained highly accurate, noisy signal detection occasionally required multiple acquisition cycles to stabilize due to random fluctuations in interference.

Overall, the embedded implementation proved to be both computationally efficient and accurate, demonstrating the feasibility of deploying ANN-based models in resource-constrained hardware environments.

IV.3 ROBUSTNESS AND LIMITATIONS

The robustness of the device was tested under varied operating conditions, including changes in input voltage and artificially introduced electromagnetic interference. The ANN maintained stable classification performance, with only minor degradation observed in highly noisy environments.

However, several limitations were identified:

- The current system was trained on simulated signals rather than real automotive sensors, which may limit its generalization in real-world deployments.
- The ANN architecture was simplified to fit Arduino memory constraints, potentially restricting scalability to more complex fault detection tasks.
- Only two input channels (potentiometer and PWM) were tested, whereas modern vehicles rely on a larger variety of sensor types.

IV.4 DISCUSSION

The results demonstrate that an AI-enhanced diagnostic device can significantly improve the detection of automotive sensor faults compared to conventional techniques. The integration of an ANN classifier into a low-cost embedded platform proves that advanced diagnostic intelligence can be achieved without expensive hardware. This work lays the foundation for more comprehensive solutions, where additional sensors and advanced neural architectures could be incorporated. Moreover, the approach could be extended to support wireless data transfer and cloud-based analytics, enabling predictive maintenance and fleet-wide monitoring.

IV.5 EXPERIMENTAL PROTOTYPE

prototype for diagnosing vehicle sensor faults using AI consists of two input ports for sensor connections, a 16×2 LCD screen for displaying diagnostic messages and sensor status, and an embedded Arduino Due that hosts the AI model to process signals and identify faults. The complete setup is illustrated in the photographs below.

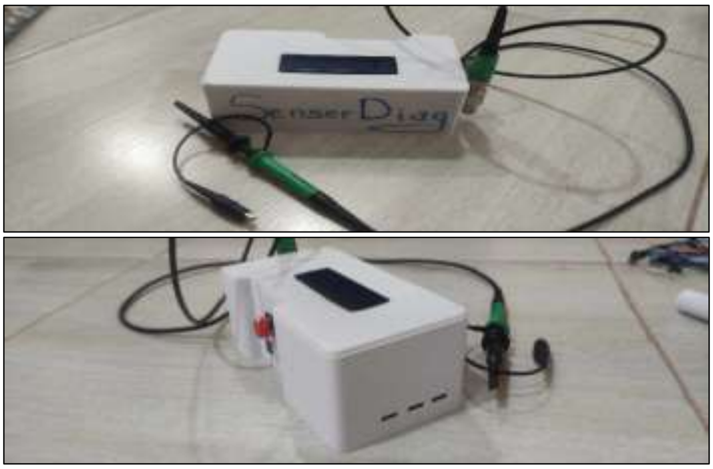

Figure 15: The experimental prototype from multiple perspectives.
Source: Authors, (2025).

V. CONCLUSIONS

The development and deployment of an AI-based vehicle sensor fault diagnostic device represent a significant advancement in automotive diagnostics. By combining modern electronics, microcontroller programming, and artificial intelligence, the proposed system provides a reliable and cost-effective solution for identifying sensor-related issues in vehicles. The device integrates simulated sensor inputs, signal acquisition, preprocessing, and classification within an embedded platform based on the Arduino Due. Using Artificial Neural Networks (ANNs), the system was trained to detect common fault types, including constant, noisy, intermittent, erratic, and missing signals. Experimental validation demonstrated a classification accuracy exceeding 95%, confirming the model's ability to distinguish between normal and faulty signal patterns. Real-time results were displayed on an LCD screen, ensuring usability in both professional workshops and personal garages. The use of serial communication further facilitated data logging and analysis, allowing mechanics and technicians to make informed and timely decisions.

Beyond its technical performance, the system offers accessibility and ease of use, addressing the growing demand for intelligent yet affordable diagnostic solutions. Unlike conventional diagnostic tools such as OBD-II readers or oscilloscopes, which are either limited in scope or require specialized expertise, this device provides automated, user-friendly fault detection. Its practical application can improve vehicle maintenance efficiency, reduce repair costs, and extend the operational lifespan of automotive systems. In summary, the proposed diagnostic device exemplifies the convergence of automotive engineering, embedded systems, and artificial intelligence. It stands as a crucial tool in modern vehicle maintenance, offering precision, reliability, and adaptability. Future enhancements will focus on testing with real automotive sensors, expanding the range of detectable faults, and integrating wireless connectivity for remote diagnostics and predictive maintenance. Ultimately, this technology not only supports the upkeep of current vehicles but also paves the way for smarter and more advanced diagnostic solutions in the next generation of automotive innovations.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Messaoud LINANI, Boubakeur LATRECHE, Imededdin DJADOUR and Aboubakr Missaoui.
Methodology: Messaoud LINANI, Boubakeur LATRECHE, Kamal Khalil TOABA, Ali BEN KHOLIF, Imededdin DJADOUR and Aboubakr Missaoui.
Investigation: Messaoud LINANI, Boubakeur LATRECHE, Kamal Khalil TOABA and Ali BEN KHOLIF.
Discussion of results: Messaoud LINANI, Boubakeur LATRECHE, Kamal Khalil TOABA and Ali BEN KHOLIF.
Writing – Original Draft: Messaoud LINANI, Boubakeur LATRECHE.
Writing – Review and Editing: Messaoud LINANI, Boubakeur LATRECHE, Imededdin DJADOUR and Aboubakr Missaoui.
Supervision: Messaoud LINANI.
Approval of the final text: Messaoud LINANI, Boubakeur LATRECHE, Kamal Khalil TOABA, Ali BEN KHOLIF, Imededdin DJADOUR and Aboubakr Missaoui.

VII. REFERENCES

[1] K. Rana and N. Khatri, "Automotive intelligence: Unleashing the potential of AI beyond advance driver assisting system, a comprehensive review," Computers and Electrical Engineering, vol. 117, p. 109237, Jul. 2024, https://doi.org/10.1016/j.compeleceng.2024.109237.

[2] F. A. Butt, J. N. Chattha, J. Ahmad, M. U. Zia, M. Rizwan, and I. H. Naqvi, "On the Integration of Enabling Wireless Technologies and Sensor Fusion for Next-Generation Connected and Autonomous Vehicles," IEEE Access, vol. 10, pp. 14643–14668, 2022, https://doi.org/10.1109/ACCESS.2022.3145972.

[3] C. Steiner, S. Püls, M. Bektas, A. Müller, G. Hagen, and R. Moos, "Resistive, Temperature-Independent Metal Oxide Gas Sensor for Detecting the Oxygen Stoichiometry (Air-Fuel Ratio) of Lean Engine Exhaust Gases," Sensors, vol. 23, no. 8, p. 3914, Apr. 2023, https://doi.org/10.3390/s23083914.

[4] S. S. A. Naqvi et al., "Evolving Electric Mobility Energy Efficiency: In-Depth Analysis of Integrated Electronic Control Unit Development in Electric Vehicles," IEEE Access, vol. 12, pp. 15957–15983, 2024, https://doi.org/10.1109/ACCESS.2024.3356598.

[5] H. Min et al., "A fault diagnosis framework for autonomous vehicles with sensor self-diagnosis," Expert Syst Appl, vol. 224, p. 120002, Aug. 2023, https://doi.org/10.1016/j.eswa.2023.120002.

[6] A. Alrajhi, K. Roy, L. Qingge, and J. Kribs, "Detection of Road Condition Defects Using Multiple Sensors and IoT Technology: A Review," IEEE Open Journal of Intelligent Transportation Systems, vol. 4, pp. 372–392, 2023, https://doi.org/10.1109/OJITS.2023.3237480.

[7] S. Li, M. Frey, and F. Gauterin, "Evaluation of Different Fault Diagnosis Methods and Their Applications in Vehicle Systems," Machines, vol. 11, no. 4, p. 482, Apr. 2023, https://doi.org/10.3390/machines11040482.

[8] M. Martínez García, L. C. G. Martínez Rodríguez, and R. Pérez Zúñiga, "Self-Adaptable Software for Pre-Programmed Internet Tasks: Enhancing Reliability and Efficiency," Applied Sciences, vol. 14, no. 15, p. 6827, Aug. 2024, https://doi.org/10.3390/app14156827.

[9] G. Bathla et al., "Autonomous Vehicles and Intelligent Automation: Applications, Challenges, and Opportunities," Mobile Information Systems, vol. 2022, pp. 1–36, Jun. 2022, https://doi.org/10.1155/2022/7632892.

[10] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: a survey," IEEE/CAA Journal of Automatica Sinica, vol. 7, no. 2, pp. 315–329, Mar. 2020, https://doi.org/10.1109/JAS.2020.1003021.

[11] G. Bendiab, A. Hameurlaine, G. Germanos, N. Kolokotronis, and S. Shiaeles, "Autonomous Vehicles Security: Challenges and Solutions Using Blockchain and Artificial Intelligence," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 4, pp. 3614–3637, Apr. 2023, https://doi.org/10.1109/TITS.2023.3236274.

[12] E. Iyalagha, "Applications and Challenges of Artificial Neural Networks in Autonomous Vehicle Technology," International Journal of Advanced Engineering, Management and Science, vol. 11, no. 3, pp. 37–44, 2025, https://doi.org/10.22161/ijaems.113.7.

[13] K. A. Prasad, G. Pattabirani, and K. Sundaramoorthy, "Insights into Anomaly Detection: A Survey and Comparative Analysis of Techniques for Time Series Data from Industrial Environment," 2025, pp. 871–907. https://doi.org/10.2991/978-94-6463-662-8_71.

[14] T. Denton, Advanced Automotive Fault Diagnosis. Fifth edition. | Abingdon, Oxon; New York, NY: Routledge, 2021.: Routledge, 2020. https://doi.org/10.1201/9780429317781.

[15] F. Matos, J. Bernardino, J. Durães, and J. Cunha, "A Survey on Sensor Failures in Autonomous Vehicles: Challenges and Solutions," Sensors, vol. 24, no. 16, p. 5108, Aug. 2024, https://doi.org/10.3390/s24165108.

[16] X. Yan et al., "An Online Learning Framework for Sensor Fault Diagnosis Analysis in Autonomous Cars," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 12, pp. 14467–14479, Dec. 2023, https://doi.org/10.1109/TITS.2023.3305620.

[17] D. Rimpas, A. Papadakis, and M. Samarakou, "OBD-II sensor diagnostics for monitoring vehicle operation and consumption," Energy Reports, vol. 6, pp. 55–63, Feb. 2020, https://doi.org/10.1016/j.egyr.2019.10.018.

[18] H. Kim, S. Jang, and J. Jang, "A Study on Development of Engine Fault Diagnostic System," Math Probl Eng, vol. 2015, pp. 1–6, 2015, https://doi.org/10.1155/2015/271374.

[19] R. kumar and A. Jain, "Driving behavior analysis and classification by vehicle OBD data using machine learning," J Supercomput, vol. 79, no. 16, pp. 18800–18819, Nov. 2023, https://doi.org/10.1007/s11227-023-05364-3.

[20] F. van Wyk, Y. Wang, A. Khojandi, and N. Masoud, "Real-Time Sensor Anomaly Detection and Identification in Automated Vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 1264–1276, Mar. 2020, https://doi.org/10.1109/TITS.2019.2906038.

[21] A. Theissler, "Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection," Knowl Based Syst, vol. 123, pp. 163–173, May 2017, https://doi.org/10.1016/j.knosys.2017.02.023.

[22] G. Kortenbruck, L. Jakubczyk, and D. F. Nowak, "Voltage Signals Measured Directly at the Battery and via On-Board Diagnostics: A Comparison," Vehicles, vol. 5, no. 2, pp. 637–655, May 2023, https://doi.org/10.3390/vehicles5020035.

[23] S. V. Okishelov, B. I. Evstatiev, and S. Y. Kadirova, "System for Electronic Control Unit Sniffing," in 2023 IEEE 29th International Symposium for Design and Technology in Electronic Packaging (SIITME), IEEE, Oct. 2023, pp. 151–155. https://doi.org/10.1109/SIITME59799.2023.10430646.

[24] A. Wiranata, A. Hasyim, Z. Mao, W. Thongking, D. N. Afifah, and M. A. Muflikhun, "Optimized control of direct current mini ultra-high voltage amplifier," Engineering Research Express, vol. 7, no. 1, p. 016002, Mar. 2025, https://doi.org/10.1088/2631-8695/ada8f6.

[25] P. Visconti, N. I. Giannoccaro, R. de Fazio, S. Strazzella, and D. Cafagna, "IoT-oriented software platform applied to sensors-based farming facility with smartphone farmer app," Bulletin of Electrical Engineering and Informatics, vol. 9, no. 3, pp. 1095–1105, Jun. 2020, https://doi.org/10.11591/eei.v9i3.2177.

[26] M. Jianqiang, C. Shan, and C. Cheng, "Sensor fault diagnosis using Principal Component analysis and convolutional neural network for offshore structural health monitoring," Measurement: Sensors, vol. 38, p. 101465, May 2025, https://doi.org/10.1016/j.measen.2024.101465.

[27] M. Elhoseny, D. D. Rao, B. D. Veerasamy, N. Alduaiji, J. Shreyas, and P. K. Shukla, “Deep Learning Algorithm for Optimized Sensor Data Fusion in Fault Diagnosis and Tolerance,” International Journal of Computational Intelligence Systems, vol. 17, no. 1, p. 299, Dec. 2024, https://doi.org/10.1007/s44196-024-00692-5.

[28] S. R. Mestha and N. Prabhu, “Support vector machine based fault detection in inverter-fed electric vehicle,” Energy Storage, vol. 6, no. 1, Feb. 2024, https://doi.org/10.1002/est2.576.

[29] M. Zabihi, R. V. Mehrizi, A. Kasaiezadeh, M. Pirani, and A. Khajepour, “A Hybrid Model-Data Vehicle Sensor and Actuator Fault Detection and Diagnosis System,” IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 7, pp. 8121–8133, Jul. 2024, https://doi.org/10.1109/TITS.2024.3370869.

[30] N. Said, M. Mansouri, R. Al Hmouz, and A. Khedher, “Deep Learning Techniques for Fault Diagnosis in Interconnected Systems: A Comprehensive Review and Future Directions,” Applied Sciences, vol. 15, no. 11, p. 6263, Jun. 2025, https://doi.org/10.3390/app15116263.

[31] A. Parmar, R. Katariya, and V. Patel, “A Review on Random Forest: An Ensemble Classifier,” 2019, pp. 758–763. https://doi.org/10.1007/978-3-030-03146-6_86.

[32] X. Li, N. Wang, Y. Lyu, Y. Duan, and J. Zhao, “Data-Driven Fault Early Warning Model of Automobile Engines Based on Soft Classification,” Electronics (Basel), vol. 12, no. 3, p. 511, Jan. 2023, https://doi.org/10.3390/electronics12030511.

[33] R. Saini, “A Review on Artificial Neural Networks for Structural Analysis,” Journal of Vibration Engineering & Technologies, vol. 13, no. 2, p. 142, Feb. 2025, https://doi.org/10.1007/s42417-024-01749-7.

[34] R. E. Nogales and M. E. Benalcázar, “Analysis and Evaluation of Feature Selection and Feature Extraction Methods,” International Journal of Computational Intelligence Systems, vol. 16, no. 1, p. 153, Sep. 2023, https://doi.org/10.1007/s44196-023-00319-1.

[35] W. J. Fleming, “Overview of automotive sensors,” IEEE Sens J, vol. 1, no. 4, pp. 296–308, 2001, https://doi.org/10.1109/7361.983469.

[36] K. Tout, A. Meguenani, J.-P. Urban, and C. Cudel, “Automated vision system for magnetic particle inspection of crankshafts using convolutional neural networks,” The International Journal of Advanced Manufacturing Technology, vol. 112, no. 11–12, pp. 3307–3326, Feb. 2021, https://doi.org/10.1007/s00170-020-06467-4.

[37] X. Wu, C. Liu, L. Wang, and M. Bilal, “Internet of things-enabled real-time health monitoring system using deep learning,” Neural Comput Appl, vol. 35, no. 20, pp. 14565–14576, Jul. 2023, https://doi.org/10.1007/s00521-021-06440-6.

[38] S. Van Nguyen and H. M. Tran, “An automated fault detection system for communication networks and distributed systems,” Applied Intelligence, vol. 51, no. 8, pp. 5405–5419, Aug. 2021, https://doi.org/10.1007/s10489-020-02026-2.

[39] K. Ajith and R. Sharmila, “AI-Based In-Cabin Monitoring System for Autonomous Vehicles,” in Modeling, Simulation, and Control of AI Robotics and Autonomous Systems, IGI Global, 2024, pp. 33–53.

[40] Y. Jiang et al., “A Step-by-Step Guide to Creating a Robust Autonomous Drone Testing Pipeline,” Jun. 2025.

[41] M. Adnane, B.-H. Nguyễn, A. Khoumsi, and J. P. F. Trovão, “Comparative Study of Embedded Energy Management Methods Based on Machine Learning for Dual-Source Electric Vehicles,” IEEE Transactions on Transportation Electrification, vol. 11, no. 4, pp. 10225–10238, Aug. 2025, https://doi.org/10.1109/TTE.2025.3559705.

[42] C. Li, “Working with Liquid Crystal Display,” 2024, pp. 81–103. https://doi.org/10.1007/979-8-8688-0814-2_5.

[43] R. Yaswanth and M. R. Babu, “Revolutionizing Automotive Technology: Unveiling the State of Vehicular Sensors and Biosensors,” IEEE Access, vol. 12, pp. 192786–192812, 2024, https://doi.org/10.1109/ACCESS.2024.3514157.

[44] M. Rizani Rusli, H. Rahmatullah, M. Badriatul Fauziah, A. Jaya, M. Machmud Rifadil, and E. Purwanto, “Pulse Width Modulation (PWM) and Pulse Amplitude Modulation (PAM) Technique for Medium-Speed BLDCM in Electric Vehicle Application,” in 2018 International Seminar on Application for Technology of Information and Communication, IEEE, Sep. 2018, pp. 87–92. https://doi.org/10.1109/ISEMANTIC.2018.8549816.

[45] B. Kidmose, A. Kidmose, and W. Meng, “can-sleuth: Sleuthing out the capabilities, limitations, and performance impacts of automotive intrusion detection datasets,” Int J Inf Secur, vol. 24, no. 5, p. 193, Oct. 2025, https://doi.org/10.1007/s10207-025-01038-8.

[46] M. Badfar, M. Yildirim, and R. B. Chinnam, “Novelty detection framework for monitoring connected vehicle systems with imperfect data,” Int J Prod Res, vol. 63, no. 18, pp. 6690–6703, Sep. 2025, https://doi.org/10.1080/00207543.2025.2484320.

[47] H. Belgacem and I. Chihi, “Toward Reliable and Intelligent Sensor Systems: A Comprehensive Study of Fault Diagnosis and Mitigation,” IEEE Sensors Reviews, pp. 1–27, 2025, https://doi.org/10.1109/SR.2025.3601092.

[48] S.-H. Sung, S. Hong, H.-R. Choi, D.-M. Park, and S. Kim, “Enhancing Fault Diagnosis in IoT Sensor Data through Advanced Preprocessing Techniques,” Electronics (Basel), vol. 13, no. 16, p. 3289, Aug. 2024, https://doi.org/10.3390/electronics13163289.

[49] P. W. Khan and Y.-C. Byun, “A Review of machine learning techniques for wind turbine’s fault detection, diagnosis, and prognosis,” Int J Green Energy, vol. 21, no. 4, pp. 771–786, Mar. 2024, https://doi.org/10.1080/15435075.2023.2217901.

[50] B. S. Meenakshi, P. Indradevi, A. Dhilp kumar, T. Harish Natarajan, S. Sathyasheelan, and V. Kathirvel, “ANN model using MATLAB in CFS -concrete,” Mater Today Proc, May 2023, https://doi.org/10.1016/j.matpr.2023.05.103.

[51] S. K. Singh, C. Sharma, R. Mahadeva, S. P. Patole, and A. Maiti, “Predicting forward osmosis performance with synthesized polyamide-based membrane: An integrated machine learning (MATLAB and ANN) and economic analysis framework,” J Clean Prod, vol. 444, p. 141285, Mar. 2024, https://doi.org/10.1016/j.jclepro.2024.141285.