
www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 17(2), 2026, 62-75. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2026.v17i2.1263

Incremental Development of a Multilayer Neural Network for CAN bus Attack Detection

Ponciano J. Escamilla-Ambrosio, Juan C. Venegas-Segura, Magdalena Saldaña-Pérez, Abraham Rodríguez-Mota
Instituto Politécnico Nacional, Centro de Investigación en Computación. Av. Juan de Dios Bátiz, esq. Miguel Othón de Mendizábal, C.P. 07738
pescamilla@cic.ipn.mx, prim_vsjc@hotmail.com, amagdasaldana@cic.ipn.mx, arodrigm@cic.ipn.mx

Abstract. The growing integration of electronic systems in vehicles has transformed the automotive industry, enabling improved performance, reduced costs, and enhanced functionality. However, the increased complexity of vehicular networks and the lack of built-in cybersecurity mechanisms in communication protocols such as the Controller Area Network (CAN) bus have left modern vehicles vulnerable to cyberattacks. This paper presents a comprehensive investigation on intrusion detection in the CAN bus using multilayer neural networks (MLPs). A process is presented to find a neural network capable of detecting three types of attacks. This consisted of a series of experiments to find the minimal structure of a neural network that can detect malicious traffic on the CAN bus, allowing the identification of attack-free data and data with three types of attacks, denial-of-service (DoS), impersonation, and fuzzy attacks. Additionally, this work contextualizes CAN security issues, discusses related contributions, and presents future research opportunities.
Keywords: Cybersecurity, Intrusion Detection System (IDS), Multilayer Neural Network (MLP), CAN bus, Electronic Control Unit (ECU).

Article Info
Received December 2, 2025
Accepted Jan 11, 2026

1 Introduction

Technological progress has driven significant advancements across multiple industries, enabling products to offer increasingly complex and efficient functionalities. In the automotive sector, these developments have enhanced numerous vehicle processes that demand high precision and speed. Among the most notable benefits are the reduction in wiring complexity, overall vehicle weight, and manufacturing costs.

A pivotal innovation in this transformation has been the integration of Electronic Control Units (ECUs) [1], which are embedded computers responsible for monitoring and managing various subsystems within a vehicle. The emergence of the Controller Area Network (CAN) bus [2] further enabled seamless and efficient communication between ECUs, ensuring the constant exchange of data required for optimal system performance. Today, the CAN bus serves as the standard communication protocol in most modern vehicles.

The automotive industry has evolved into a highly digitized sector. Vehicles now integrate 50–80 ECUs, each interconnected through the CAN bus. This network allows for high-speed data exchange, lightweight wiring, and reduced costs. Despite its advantages, the CAN protocol was designed without authentication or encryption, making it inherently insecure. Cybersecurity researchers have demonstrated the feasibility of attacks, such as Denial of Service (DoS) and message spoofing, which can jeopardize passenger safety [1]. Hence, cybersecurity mechanisms must complement vehicular communication.

In this context, the CAN bus has become a key attack surface for adversaries seeking to compromise vehicular systems [3]. This research addresses that challenge by proposing an intrusion detection approach based on multilayer neural networks, aiming to identify and classify different types of malicious activity within the CAN bus.

Specifically, the proposed model is designed to detect and differentiate between three common types of attacks, DoS, impersonation attacks, and fuzzy attacks, as well as to identify normal attack-free traffic. Unlike deep neural networks that often require extensive tuning and computational resources, the proposed approach explores a structured, gradually enhanced, multilayer neural network architecture suitable for deployment in automotive environments.

The remainder of this paper is structured as follows. Section 2 presents a review of related work. Sections 3 and 4 discuss the technical considerations and methodologies employed in this study. Section 5 describes the experimental process used to determine the optimal network architecture, including the number of hidden layers, neurons per layer, choice of optimizer, and input data formatting. Finally, Section 6 concludes the paper with final remarks and potential directions for future work.

2 Related Works

There is an extensive body of research focused on developing security systems for protecting in-vehicle networks, with particular emphasis on the CAN bus. One widely used characteristic for intrusion detection is the transmission frequency of CAN messages from individual ECUs, which tends to deviate significantly during attacks. This behavior has been exploited in several intrusion detection methods, such as those based on statistical profiling [4].

Intrusion Detection Systems (IDS) for vehicular networks have used a wide range of techniques, including:

- Statistical approaches, which monitor message frequency and timing to detect anomalies in CAN traffic [4].
- Machine learning algorithms, such as Support Vector Machines (SVM), Random Forests, and unsupervised methods like k-means clustering, to identify abnormal communication patterns [5], [6].
- Deep learning models, including Convolutional Neural Networks (CNNs), Deep Neural Networks (DNNs), and Generative Adversarial Networks (GANs), which have demonstrated detection accuracies above 98–99%, but often at the cost of high computational requirements, limiting their feasibility for deployment in embedded automotive systems [7], [8].

For instance, Song et al. [6] proposed a CNN-based IDS that achieved 99% detection accuracy, while Kang et al. [5] utilized a deep neural network with 98% accuracy. Song et al. [7] later proposed a GAN-based approach, also reaching 98% accuracy. More recently, Huang et al. [8] presented a CNN-hybrid intrusion detection system implemented on FPGA hardware, achieving over 99% accuracy and low latency, showing promise for real-time applications in constrained automotive environments.

Unlike these prior studies, this research focuses on determining the simplest possible neural network architecture that still delivers high detection performance. An exhaustive experimentation process is conducted to find:

1 the number of hidden layers,
2 the number of neurons per layer,
3 the choice of optimizer,
4 and the data structure in the input layer (including sliding window configuration and frame sizes).

This approach aims to achieve a balance between detection accuracy and computational efficiency, making the proposed IDS viable for real-world embedded automotive platforms.

3 Vehicular Networks

The development of computing technologies has marked a transformative era in the automotive industry. With the introduction of computers into vehicles, many subsystems that were previously mechanical or analog have been replaced by embedded controllers. These Electronic Control Units (ECUs) manage diverse vehicle functions, ranging from engine control and braking to infotainment and safety, thus enabling higher performance, efficiency, and adaptability.

Modern vehicles typically integrate 50–100 ECUs, depending on the model and manufacturer, interconnected via in-vehicle communication networks. These control units are interconnected through in-vehicle communication protocols, among which the CAN bus remains the most widely adopted due to its cost-effectiveness and deterministic performance in real-time environments. Research surveys report that many intelligent vehicles employ approximately 70 ECUs, each exchanging message traffic over one or more CAN buses [9].

The CAN bus was developed by Robert Bosch GmbH in the early 1980s and formally introduced in 1986 at the SAE Congress. Its design goal was to reduce wiring complexity, improve reliability, and ensure deterministic message delivery without the need for a centralized master controller [10]. The CAN protocol employs a bus topology for the transmission of messages and supports communication rates up to 1 Mbps in classical implementations, with CAN FD (Flexible Data-Rate) later introduced to extend throughput and payload capacity. Each ECU listens to all messages but processes only those relevant to its identifier, which simplifies communication but exposes critical vulnerabilities [11].

From a cybersecurity perspective, the CAN bus protocol was specified without authentication, encryption, or integrity verification mechanisms. All circulating messages are implicitly trusted, which makes CAN inherently insecure. Consequently, adversaries can exploit this flaw by injecting, replaying, or spoofing frames that are indistinguishable from legitimate ones. This has been demonstrated in multiple studies where Denial of Service (DoS), impersonation, and fuzzing attacks disrupted or even hijacked vehicular systems [12].

Recent surveys emphasize that although CAN remains dominant in the automotive industry, its vulnerabilities necessitate supplementary defense mechanisms. Security frameworks incorporating cryptographic authentication, anomaly detection, and Intrusion Detection Systems (IDS) have been proposed as critical measures to safeguard vehicular networks against evolving cyber threats [9], [13].

4 Neural Networks

A neural network is a computational model composed of interconnected processing units (called neurons) that try to emulate certain aspects of biological neural systems to solve complex tasks. These units (neurons) process inputs through weighted connections and activation functions, enabling the network to learn non-linear relationships in data [14], [15].

Neural networks are structured in layers: an input layer, one or more hidden layers, and an output layer. Critical design elements include the selection of the optimizer (e.g., SGD, Adam, RMSProp), the architecture (number of hidden layers, neurons per layer), the data representation and preprocessing, activation functions (ReLU, sigmoid, etc.), loss functions, regularization methods (dropout, weight decay), and training parameters (learning rate, batch size, number of epochs) [15].

In recent years, neural networks (especially multilayer perceptrons, convolutional neural networks, and recurrent neural networks) have been employed in cybersecurity contexts to detect anomalies, intrusions, or attacks in different systems and networks [16].

This work aims to address the cybersecurity challenges in modern vehicles. As a countermeasure within the vehicle's communication systems, this study proposes the development of a classifier based on a multilayer neural network capable of identifying various types of cyber-attacks on the CAN bus, such as message spoofing, DoS, and fuzzy (random payload) attacks. The classifier is designed to be lightweight, efficient, and suitable for real-time or near real-time deployment in embedded automotive systems.

5 Development of the Neural Network Structure for CAN Attack Detection

In the machine learning literature, there is no single protocol or standard that indicates a fixed neural network architecture tailored for all specific tasks. Determining the optimal structure of a neural network, such as the number of hidden layers, the number of neurons per layer, choice of optimizer, activation functions, and other hyperparameters, remains highly problem-dependent. Researchers often must balance expressive power, computational efficiency, and risk of overfitting [17], [18].

The experience and domain knowledge of the designer are frequently instrumental in proposing an initial architecture. However, this is often complemented by systematic experimentation or hyperparameter optimization techniques. For example, studies show that methods like particle swarm optimization (PSO), genetic algorithms, or more recently, neural architecture search (NAS) and meta-heuristic approaches can substantially aid in identifying efficient architectures [19], [20].

In the context of this work, a sequence of experiments was conducted to discover a neural network structure capable of detecting and classifying attack-related data over a vehicle's CAN bus. Two overarching experimental configurations were evaluated:

1. Multiclass Classification: Training the neural network to distinguish between three attack types, DoS, impersonation, fuzzing, and normal traffic.

2. Binary Classification: Distinguishing simply between normal and malicious traffic (aggregating attack types into one category).

Additionally, the effect of different input data configurations on detection performance was investigated. Key configurations included:

- Using data frames of various sizes (i.e., grouping a fixed number of CAN messages per sample).
- Applying sliding windows over sequences of messages to incorporate temporal context.

Figure 1 illustrates the overall process followed in this study to engineer and refine the neural network architecture. The methodological steps were:

- Start with a baseline architecture: one input layer, one hidden layer, and one output layer, with a certain number of neurons, trained with a particular optimizer and loss (cost) function.
- Evaluate several cost functions (such as categorical cross-entropy, mean squared error, focal loss) and various optimization algorithms (SGD, Adam, RMSProp) to find what yields better detection accuracy and classification metrics.
- Incrementally add hidden layers and modify neuron counts in each layer, observing the trade-offs in performance (accuracy, precision/recall) versus complexity (training time, memory usage).
- Continue refining until additional complexity fails to yield statistically significant improvement in detection metrics.

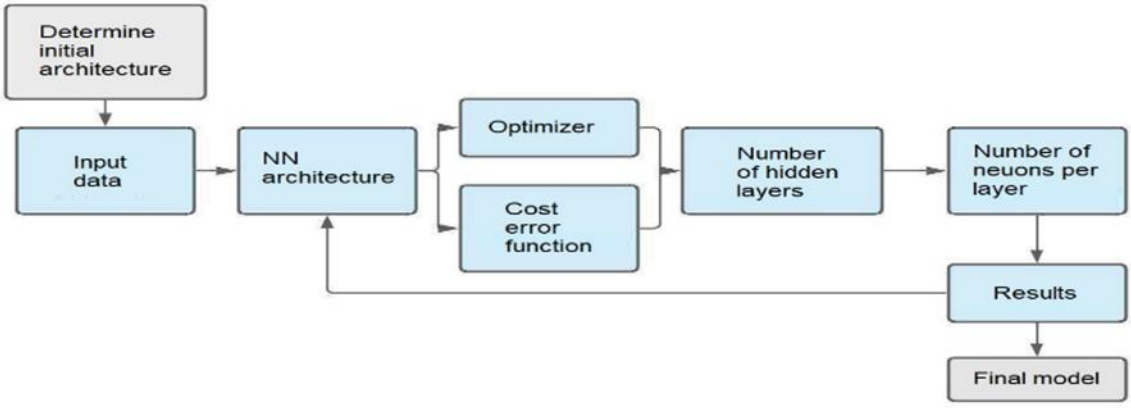

Fig. 1. Diagram of the process followed for the development of the neural network.

Through this process, the simplest architecture that satisfies the performance criteria (i.e., good detection rate across attack types, acceptable false positive rate, and feasible computational cost) was selected. Emphasis was placed on achieving effectiveness with minimal complexity to enable implementation in real embedded automotive systems.

5.1 Data Set

The dataset employed in this study was collected by the Hacking and Countermeasure Research Laboratory (HCRL) for research purposes [21]. The data consists of Controller Area Network (CAN) messages recorded during real driving sessions of approximately 40 minutes per trial. The dataset includes both normal traffic and traffic injected with malicious patterns. Three types of attacks were considered, as they are among the most widely studied in the literature on automotive cybersecurity:

1. Denial of Service (DoS) – This kind of attack occurs when the CAN bus is flooded with high-priority frames, preventing legitimate messages from losing priority access.
2. Fuzzy attack – This attack is seen when adversaries inject random data values (including identifier ID and data field) into the CAN bus. The objective is to create unpredictable and disruptive traffic.
3. Impersonation attack – The attack occurs when an attacker steals the identity of a legitimate ECU using its identifier ID. The attacker sent falsified information to the CAN bus that will be accepted and processed because the system thinks the information is authentic.

Table 1 summarizes the number of CAN messages in the dataset, separated by attack type and normal operation.

Table 1. Number of messages in the data sets

Data set	Number of messages
Free of attacks	2,369,868
DoS attack	656,579
Fuzzy attack	591,990
Impersonation attack	995,472

The structure of a CAN message, illustrated in Figure 2, is composed of several fields necessary for ECUs to interpret and execute control actions. The key fields are:

- Start of Frame (SOF): Indicates the beginning of a new message.
- Identifier (ID): Provides arbitration priority and identifies the message content. This field is critical, since ECUs filter and process only those messages relevant to their operation.
- Remote Transmission Request (RTR): Differentiates between data frames and remote request frames.
- Data Field: Contains the actual payload (up to 8 bytes in classical CAN, and up to 64 bytes in CAN FD).
- Control, CRC, and Acknowledge fields: Ensure message integrity and proper synchronization between transmitting and receiving ECUs [22].

In this work, the ID field and the Data Field were extracted and used as features for training the neural network. These two fields provide essential information for distinguishing between normal and malicious traffic, since most CAN attacks exploit manipulated identifiers or payload data.

Fig. 2. Structure of a CAN message.

For training purposes, the dataset was divided into two subsets: 80% for training and 20% for testing, following common practice in machine learning to ensure unbiased evaluation [23]. To provide temporal context, consecutive CAN messages were grouped into frames or sliding windows, which formed the input vectors of the neural network.

The initial experiments used frames of 9 consecutive CAN messages, as shown in Figure 3. This approach allowed the model to learn temporal dependencies and recognize patterns that span multiple messages, improving detection accuracy for fuzzing and impersonation attacks where malicious payloads may not be evident in single messages. Further experiments investigated alternative frame sizes and sliding window mechanisms to assess their impact on classification performance.

ID1 DF1 DF2 DF3 DF4 DF5 DF6 DF7 DF8
ID2 DF1 DF2 DF3 DF4 DF5 DF6 DF7 DF8
ID3 DF1 DF2 DF3 DF4 DF5 DF6 DF7 DF8
ID4 DF1 DF2 DF3 DF4 DF5 DF6 DF7 DF8
ID5 DF1 DF2 DF3 DF4 DF5 DF6 DF7 DF8
ID6 DF1 DF2 DF3 DF4 DF5 DF6 DF7 DF8
ID7 DF1 DF2 DF3 DF4 DF5 DF6 DF7 DF8
ID8 DF1 DF2 DF3 DF4 DF5 DF6 DF7 DF8
ID9 DF1 DF2 DF3 DF4 DF5 DF6 DF7 DF8

Fig. 3. Structure of the frames presented to the neural network.

5.2 Neural Network Validation Tests

One of the main objectives of these experiments was to determine the architectural elements that best fit a neural network designed for classification and detection of cyberattacks on the CAN bus. The tests aimed to identify the most suitable configuration of

hidden layers, number of neurons per layer, choice of optimizers, and error (loss) functions to achieve the best detection accuracy while maintaining computational efficiency.

The first experiments were carried out using a baseline neural network architecture composed of:

- An input layer, of 81 neurons, corresponding to the features extracted from CAN message frames.
- A single hidden layer, with a variable number of neurons (n).
- An output layer with 4 neurons, representing the four classification categories (normal, DoS, fuzzy, impersonation).

This initial setup was used to analyze the performance of different optimizers (Adam, Adagrad, and Stochastic Gradient Descent (SGD)) and error functions (binary cross entropy and sparse categorical cross entropy). The results indicated that the Adam optimizer combined with the sparse categorical cross entropy loss function provided the most stable convergence and the highest classification accuracy. Table 2 summarizes these results.

Table 2. Results obtained during tests for a neural network with a single hidden layer

Optimizer	No. of neurons (hidden layer)	Accuracy (Training)	Accuracy (Validation)
SGD	40	0.6515	0.6507
Adam	40	0.9099	0.9019
Adagrad	40	0.7091	0.7109

To determine the optimal number of neurons in the first hidden layer, experiments were conducted by starting with a single neuron and gradually increasing up to 50 neurons. As shown in Figure 4 (error vs. no. of neurons) and Figure 5 (accuracy vs. no. of neurons), convergence in both error reduction and accuracy improvement was observed at approximately 40 neurons. Consequently, the first hidden layer was fixed at 40 neurons for subsequent tests.

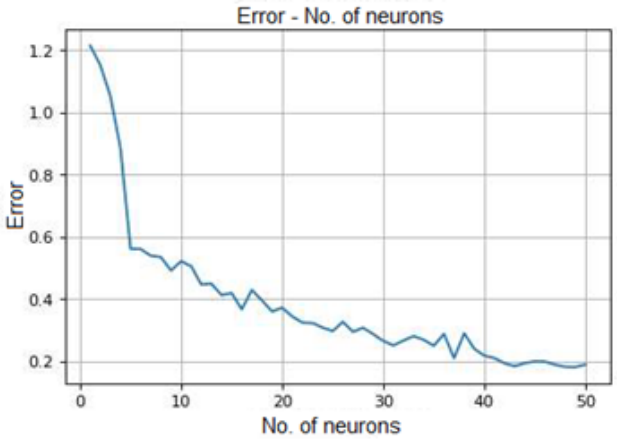

Fig. 4. Error graph-number of neurons in the hidden layer of the proposed neural network.

Next, the architecture was extended to include a second hidden layer. Following a similar incremental process, the number of neurons in this layer was varied between 1 and 50. The results, shown in Figures 6 and 7, indicated that performance converged with approximately 30 neurons in the second hidden layer. Table 3 presents the results, where the two-hidden-layer network achieved an accuracy of 92%, compared to 90% for the single-hidden-layer network.

Further experiments investigated deeper architectures with three, four, and five hidden layers. The same incremental methodology was applied to determine the number of neurons per layer, starting from one neuron and increasing up to 50. The results are shown in Table 4, which revealed:

- Accuracy improved consistently from one to three hidden layers.
- Performance plateaued with four hidden layers.
- Accuracy decreased slightly with five hidden layers, suggesting diminishing returns and possible overfitting.

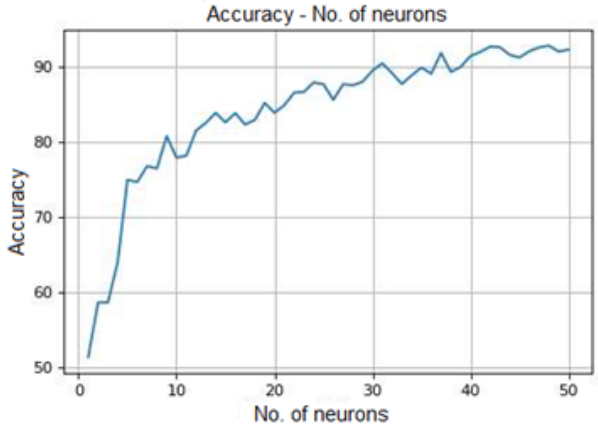

Fig. 5. Accuracy-number of neurons in the hidden layer of the proposed neural network.

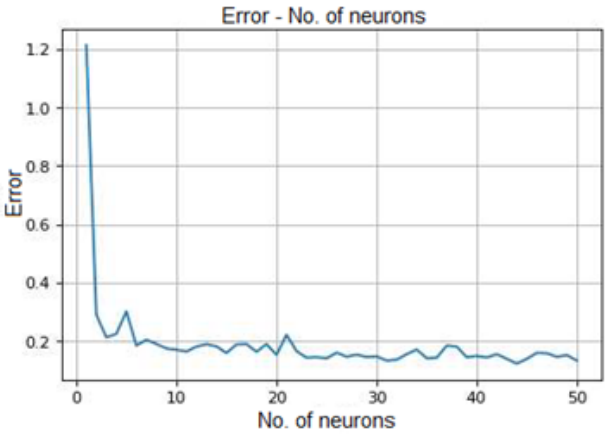

Fig. 6. Error-number of neurons plot for the neural network with two hidden layers.

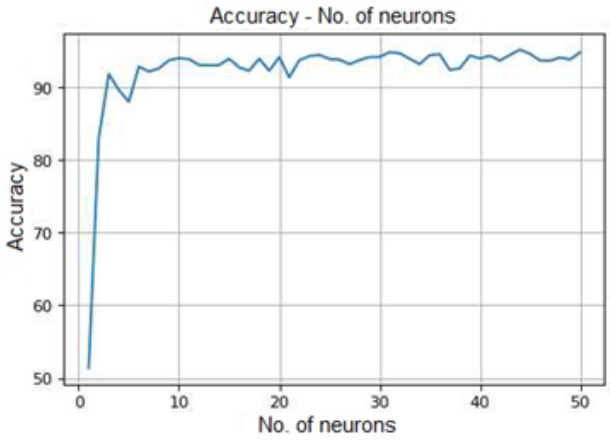

Fig. 7. Accuracy-number of neurons plot for the neural network with two hidden layers.

Based on these observations, it was concluded that three hidden layers represent the optimal trade-off between accuracy and complexity for this problem. Architectures with more layers can yield marginal improvements but also introduce higher computational costs and greater risk of overfitting, which is undesirable for real-time, resource-constrained environments such as in-vehicle ECUs.

Table 3. Results obtained during tests for a neural network with two hidden layers

Optimizer	No. of neurons (1^{st}, 2^{nd} hidden layers)	Accuracy (Training)	Accuracy (Validation)
SGD	40,30	0.6036	0.6507
Adam	40,30	0.9257	0.9247
Adagrad	40,30	0.7165	0.7109

Table 4. Results obtained with different numbers of hidden layers

Hidden layers	No. of neurons per layer	Accuracy (Training)	Accuracy (Validation)
1	40	0.9099	0.9019
2	40,30	0.9257	0.9247
3	**40,30,28**	**0.9514**	**0.9379**
4	40,30,28,26	0.9532	0.9404
5	40,20,28,26,24	0.9505	0.9315

In summary, the experiments demonstrated that:

- Adam optimizer and sparse categorical cross entropy loss are well suited for CAN attack detection.
- The optimal architecture for this work consists of three hidden layers with approximately 40, 30, and a smaller number of neurons in the third layer.
- Increasing depth beyond three hidden layers introduces complexity without significant performance gains.

This systematic exploration of architectural choices ensured that the neural network was as simple as possible but sufficiently complex to capture the patterns of malicious CAN traffic.

5.3 Testing with different frame sizes for the input layer

The experiments described in this section were designed to determine the optimal frame size for feeding data into the input layer of the neural network for effective classification of CAN bus traffic. The goal was to identify a frame size that balances sufficient contextual information with generalization capability, thereby ensuring accurate detection of both normal and malicious data patterns.

Each frame consisted of a sequence of CAN messages arranged into an array of dimension N × 9, where N represents the number of consecutive CAN messages grouped together and 9 corresponds to the features extracted per message (e.g., CAN ID and data field bytes). The tests began with a minimal frame of size 1 × 9 (one message), progressively increasing to frames containing 50 × 9 messages. This incremental approach enabled evaluation of how temporal context impacts the neural network's performance in attack detection.

Figures 8 and 9 illustrate the results of these experiments. Figure 8 presents the evolution of the error across different frame sizes, while Figure 9 shows the corresponding accuracy values. The results reveal several important trends:

- Very small frames (e.g., 1 × 9 or 3 × 9) provided insufficient information for reliable classification. As a result, accuracy values were low, and the models exhibited difficulty in distinguishing between normal and attack traffic.
- Performance improved significantly as the frame size increased. Larger frames provided the neural network with more temporal context, improving its ability to detect anomalies that may not be apparent in a single message.
- However, frames exceeding the size of 20 × 9 began to show reduced generalization capability. Although accuracy sometimes improved within the training set, overfitting became apparent, and performance deteriorated on unseen test data.

Based on these findings, three frame sizes were identified as optimal trade-offs:

- Minimum viable size: 9 × 9 — sufficient to provide meaningful context without excessive complexity.
- Preferred size: 16 × 9 — offering consistently high accuracy and robust generalization across attack types.
- Upper bound: 20 × 9 — beyond this point, the risk of overfitting outweighs the marginal performance gains.

Table 5 presents the results obtained using the three-hidden-layer neural network across different frame sizes. For comparison, results obtained with larger frames are also included, showing diminishing returns in accuracy when exceeding the recommended maximum.

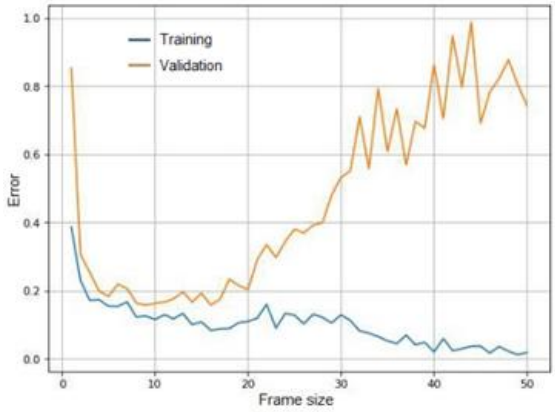

Fig. 8 Error-frame size plot for the neural network.

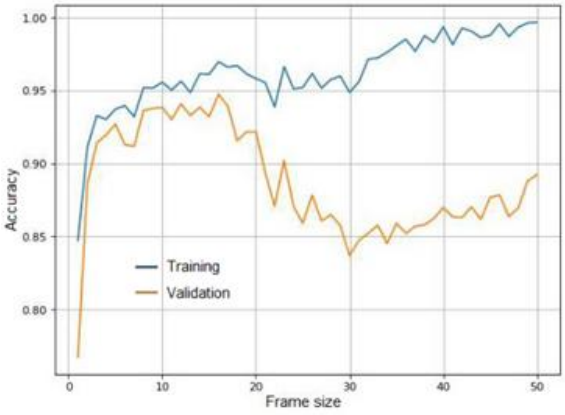

Fig. 9. Accuracy-frame size graph for the neural network.

Table 5. Results obtained for the neural network considering different data frame sizes

Data frame size	No. of neurons (1st, 2nd, 3rd hidden layer)	Accuracy (Training)	Accuracy (Validation)
9x9	40,30,28	0.9517	0.9377
16x9	**40,30,28**	**0.9684**	**0.9581**
20x9	40,30,28	0.9605	0.9487
26x9	40,30,28	0.9509	0.8983

For the neural network architectures considered in this work, a frame size of 16 × 9 is recommended. This configuration provided the best balance between accuracy, computational efficiency, and generalization, making it well suited for real-time detection of CAN bus attacks in embedded automotive environments.

5.4 Testing with bi-class neural networks

In this stage of the study, a set of experiments was conducted using binary classification neural networks. Unlike the multiclass configuration explored earlier, these networks were designed to classify CAN bus traffic into only two categories: attack-free data

and malicious data of a specific type. The rationale behind this approach was to evaluate whether simplifying the classification task would lead to improved detection performance for individual attack types.

All tests employed frames of size 16 × 9, as previously identified as the optimal configuration for balancing accuracy and generalization. Four different binary classification models were trained, each targeting a specific detection scenario:

1. Normal vs. DoS traffic – distinguishing between benign CAN messages and those containing denial-of-service (DoS) patterns.
2. Normal vs. Fuzzy traffic – detecting random identifier and payload injections characteristic of fuzzy attacks.
3. Normal vs. Impersonation traffic – identifying instances where an attacker injects forged frames mimicking legitimate ECU identifiers.
4. Normal vs. Attack traffic (all types) – classifying traffic into attack-free versus any malicious pattern (DoS, fuzzy, or impersonation combined).

The outcomes of these tests are summarized in Table 6. Across all scenarios, the binary classifiers exhibited improved accuracy compared to the multiclass model, confirming that reducing the number of output classes simplifies the decision boundaries and reduces classification ambiguity. Summarizing these results:

- The Normal vs. DoS classifier achieved particularly high accuracy, as DoS attacks generate distinctive traffic patterns (e.g., overwhelming high-priority frames) that are relatively easier to detect.
- The Normal vs. Fuzzy classifier also performed well, though accuracy values were slightly lower due to the randomness and unpredictability of fuzzy attack payloads.
- The Normal vs. Impersonation classifier achieved strong performance, reflecting the model's ability to detect subtle manipulations of CAN identifiers.
- The Normal vs. Attack (combined) classifier yielded high overall detection accuracy, making it a promising option for practical deployment in real-time intrusion detection systems where identifying the presence of any attack is sufficient.

Table 6. Results obtained for the neural network considering different data frame sizes

Neural network	No. of neurons (1^{st}, 2^{nd}, 3^{rd} hidden layer)	Accuracy (Training)	Accuracy (Validation)
1	40,30,28	0.9999	0.9995
2	40,30,28	0.9997	0.9942
3	40,30,28	0.9735	0.9400
4	40,30,28	0.9792	0.9488

These results highlight the trade-off between binary and multiclass classification strategies in intrusion detection. While multiclass models provide richer information by identifying the exact type of attack, binary models deliver higher accuracy and robustness by focusing on simpler decision boundaries. For embedded vehicular systems with limited computational resources, a hybrid approach could be considered: using binary classifiers for real-time monitoring and multiclass models offline for forensic analysis and attack characterization.

5.4 Testing with sliding windows

Finally, a set of experiments was conducted in which the input data to the neural network were processed using a sliding window format. In this approach, the frame presented to the input layer is continuously updated: as each new CAN message arrives, it is appended to the frame, while the oldest message is simultaneously discarded. This rolling update ensures that the frame always contains the most recent sequence of messages.

This methodology contrasts with the earlier fixed-frame experiments, where it was necessary to accumulate a complete set of 9 or 16 messages before constructing an input frame. With sliding windows, however, a new frame is formed with every incoming

CAN message. This characteristic is particularly advantageous for real-time intrusion detection, as it eliminates latency introduced by waiting for a full frame to be assembled. Instead, the neural network continuously receives updated frames, allowing faster and more responsive attack detection.

The experimental results demonstrate that using sliding windows improves the classification performance of the neural network. Specifically:

- Accuracy gains: Accuracy values either matched or exceeded those obtained in earlier fixed-frame tests.
- Timeliness: Since new frames are available for every message received, the system is better suited for real-time environments, where delays in detecting malicious activity can have safety-critical consequences.
- Stability: Sliding windows help capture short-term temporal dependencies, which are valuable for detecting attacks such as impersonation or fuzzy injections that may only manifest over a short sequence of messages.

Figure 10 illustrates the sliding window mechanism, showing how overlapping frames are formed from consecutive CAN messages. The detailed results of these tests are presented in Table 7, which confirms that the sliding window approach provides consistently strong or improved accuracy compared to fixed-frame experiments.

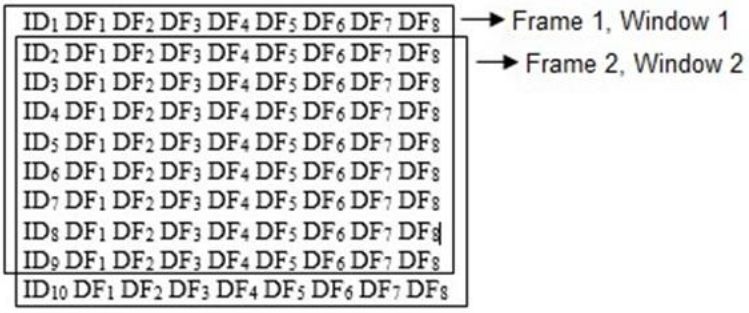

Fig. 10. Accuracy-frame size graph for the neural network.

Table 7. Results obtained for the neural network considering sliding windows

Neural network	No. of neurons (1^{st}, 2^{nd}, 3^{rd} hidden layer)	Accuracy (Training)	Accuracy (Validation)
4 classes	40,30,28	0.9856	0.9766
3 classes	40,30,28	0.9999	0.9994
DoS – Free	40,30,28	0.9999	0.9999
Fuzzy – Free	40,30,28	0.9998	0.9995
Imper – Free	40,30,28	0.9713	0.9632
Attack - Free	40,30,28	0.9792	0.9488

In addition, we include Table 8 with information related to the following metrics: Recall, Specificity, Precision, FPR, FNR, and F1-Score.

Table 8. Results obtained for the neural network considering sliding windows

Neural network	Recall	Specificity	Presicion	FPR	FNR	F1-Score
4 classes	0.9712	0.9879	0.9888	0.012	0.0287	0.9799
3 classes	0.9998	0.9997	0.9999	0.0012	0.0010	0.9998
DoS – Free	0.9999	0.9999	0.9999	0.0001	0.0001	0.9999
Fuzzy – Free	0.9998	0.9991	0.9997	0.0008	0.0001	0.9997
Imper – Free	0.9781	0.9621	0.9843	0.0378	0.0218	0.9812

Precision [25] is the proportion of correctly predicted positive cases out of all cases predicted as positive. Recall [25] is the proportion of correctly predicted positive cases out of all actual positive cases. The F1 [25] score is a harmonic mean of precision and recall. It is useful for unbalanced datasets, as it seeks a balance between these two metrics. FPR is the ratio of false positives to the sum of false positives and true negatives. It measures the proportion of actual negative cases that are incorrectly classified as positive by the model. FNR is the ratio of false negatives to the sum of false negatives and true positives. It measures the proportion of actual positive cases that are incorrectly classified as negative by the model. A high FNR or FPR means that the model is making a lot of errors.

In summary, adopting a sliding window input structure enhances both the accuracy and the practical applicability of the neural network model. By enabling near-instantaneous frame construction, this method provides a pathway for real-time intrusion detection in embedded automotive systems, where minimizing detection latency is essential for ensuring passenger safety and system reliability.

6 Conclusion and Future Work

In this work, a machine learning–based classifier has been proposed to detect abnormal behavior in the data flow of a vehicle's Controller Area Network (CAN) bus. The classifier employs a multilayer perceptron (MLP) neural network, specifically designed to identify deviations from normal traffic patterns and to classify different types of cyberattacks targeting in-vehicle communication systems.

The proposed approach is capable of distinguishing between:

1. Normal traffic (attack-free data),
2. Denial of Service (DoS) attacks,
3. Fuzzy attacks, and
4. Impersonation attacks.

The experiments confirmed that the classifier achieves robust performance in detecting and classifying these categories. However, some limitations were observed. For binary-class neural networks, the model occasionally misclassified normal traffic as anomalous when using frames of size 16×9 with sliding windows. Furthermore, when larger frames (e.g., 20×9) were employed, the classifier exhibited difficulties in generalizing to unseen cases, highlighting the risk of overfitting when temporal context is excessively extended.

Through systematic experimentation, the following configuration was identified as optimal for this task:

- Optimizer: Adam, which provided faster convergence and stable learning dynamics.
- Loss function: Sparse categorical cross entropy, enabling effective classification across multiple categories.
- Hidden layers: Three layers with 40, 30, and 28 neurons, respectively.

This architecture, in conjunction with sliding window input formatting, achieved the highest accuracy among all tested configurations. Figures presented earlier in this study (e.g., evolution of error and accuracy with neuron counts, and sliding window results) support the conclusion that this architecture represents the best trade-off between complexity, accuracy, and computational feasibility.

The results demonstrate that high detection accuracy can be achieved without resorting to more computationally demanding architectures, such as convolutional or deep residual networks. This makes the proposed classifier particularly suitable for deployment in embedded automotive systems, where processing and memory resources are constrained.

Table 9 compares the accuracy values obtained in this work with those reported in related studies. While deep learning approaches such as CNNs and GANs have reported slightly higher accuracies in some cases [5], [6], the proposed MLP achieves comparable results with significantly lower computational cost, thereby enhancing its potential for real-world, real-time intrusion detection applications in vehicles.

Table 9. Results obtained for the neural network considering sliding windows

Reference	Accuracy	Precision	Structure
[24] DCNN	99%	99%	3 filters (several layers)
[6]	98%	98%	11 hidden layers
[5]	99%	98%	2 NN (convolutional, deep)
This proposal	99%	99%	3 hidden layers

Overall, this work confirms that an appropriately optimized MLP, combined with carefully designed input representations such as sliding windows, is a viable and efficient solution for CAN bus intrusion detection. By achieving strong performance with a relatively simple architecture, this study contributes to bridging the gap between theoretical research and practical implementation in vehicular cybersecurity.

6.1 Future Work

Although this study demonstrates that optimized multilayer neural network can effectively detect attacks on the CAN bus, several avenues remain open for future research:

- Real-time deployment and validation. Future efforts should focus on implementing the proposed classifier in embedded automotive hardware to validate its performance under strict memory and processing constraints. Real-time experiments on physical testbeds or in-the-loop simulations would provide insights into latency, scalability, and practical feasibility.
- Online learning and adaptability. The current model is trained offline with pre-collected data. In real-world environments, however, attack patterns may evolve over time. Incorporating online learning mechanisms or incremental training strategies could allow the classifier to adapt dynamically to new or unforeseen attack types.
- Integration with hybrid detection systems. While neural networks provide strong detection performance, combining them with statistical anomaly detection, rule-based systems, or cryptographic measures could enhance robustness. Hybrid intrusion detection systems (IDS) may reduce false positives while improving resilience against sophisticated attacks.
- Evaluation with extended datasets. Additional experiments should be conducted using datasets that include other types of CAN attacks (e.g., replay, masquerade, and more advanced fuzzing techniques). Testing with multi-vehicle and multi-vendor datasets would ensure greater generalization across diverse automotive platforms.
- Explainability and interpretability. As neural networks are often treated as “black-box” models, exploring methods to improve the explainability of predictions would be valuable. Understanding why the model classifies a frame as malicious could help automotive engineers in diagnosing anomalies and strengthening system trust.
- Migration toward the next generation of automotive networks. With the adoption of CAN FD and Automotive Ethernet in modern vehicles, future research should investigate the transferability of the proposed method to these protocols. Evaluating the scalability of the approach across different in-vehicle networks will be critical for long-term applicability.
- Implementation in an embedded system that includes a Raspberry Pi for real cases of studio. This advice will be connected to the OBD port of the automobile to register and analyze traffic data in the vehicle network.
- Development and implementation of an application in which the user will be warned through a message or with an android/IOS application.

In summary, future work should not only aim at improving detection accuracy but also at ensuring that intrusion detection systems are adaptive, interpretable, and feasible for real-world deployment in the evolving landscape of vehicular cybersecurity.

Acknowledgments.
The authors would like to thank SECIHTI, and IPN for their support under project SIP-20251080.

References

1. Kaiser, M. (2015). *Electronic control unit (ECU).* In K. Reif (Ed.), *Gasoline engine management: Systems and components* (pp. 254–259). Springer Vieweg. https://doi.org/10.1007/978-3-658-03964-6_16

2. Ran, L., Wu, J., Wang, H., & Li, G. (2010, October). *Design method of CAN BUS network communication structure for electric vehicle.* In *2010 International Forum on Strategic Technology (IFOST)* (pp. 326–329). IEEE. https://doi.org/10.1109/IFOST.2010.5668017
3. Bozdal, M., Samie, M., Aslam, S., & Jennions, I. (2020). *Evaluation of CAN bus security challenges. Sensors, 20*(8), 2364. https://doi.org/10.3390/s20082364
4. Khan, J., Lim, D.-W., & Kim, Y.-S. (2023). *Intrusion detection system CAN-bus in-vehicle networks based on the statistical characteristics of attacks. Sensors, 23*(7), 3554. https://doi.org/10.3390/s23073554
5. Kang, M.-J., & Kang, J.-W. (2016). *Intrusion detection system using deep neural network for in-vehicle network security. PLOS ONE, 11*(6), e0155781. https://doi.org/10.1371/journal.pone.0155781
6. Song, H. M., Woo, J., & Kim, H. K. (2020). *In-vehicle network intrusion detection using deep convolutional neural network. Vehicular Communications, 21*, 100198. https://doi.org/10.1016/j.vehcom.2019.100198
7. Song, Y., Hyun, S., & Cheong, Y.-G. (2021). *Analysis of autoencoders for network intrusion detection. Sensors, 21*(13), 4294. https://doi.org/10.3390/s21134294
8. Rangsikunpum, A., Amiri, S., & Ost, L. (2024, March). *An FPGA-based intrusion detection system using binarised neural network for CAN bus systems.* In *2024 IEEE International Conference on Industrial Technology (ICIT)* (pp. 1–6). IEEE. https://doi.org/10.1109/icit58233.2024.10540960
9. Avatefipour, O., & Malik, H. (2018). *State-of-the-art survey on in-vehicle network communication (CAN-bus) security and vulnerabilities. arXiv.* https://arxiv.org/abs/1802.01725
10. CAN in Automation (CiA). (n.d.). *History of CAN technology.* Recovery of https://www.can-cia.org/can-knowledge/history-of-can-technology
11. Lotto, A., Marchiori, F., Brighente, A., & Conti, M. (2024). *A survey and comparative analysis of security properties of CAN authentication protocols. IEEE Communications Surveys & Tutorials, 27*(4), 2470–2504. https://doi.org/10.1109/COMST.2024.3486367
12. Rai, R., Grover, J., Sharma, P., & Pareek, A. (2025). *Securing the CAN bus using deep learning for intrusion detection in vehicles. Scientific Reports, 15*(1), 13820. https://doi.org/10.1038/s41598-025-98433-x
13. Rathore, R. S., Hewage, C., Kaiwartya, O., & Lloret, J. (2022). *In-vehicle communication cyber security: Challenges and solutions. Sensors, 22*(17), 6679. https://doi.org/10.3390/s22176679
14. Taherdoost, H. (2023). *Deep learning and neural networks: Decision-making implications. Symmetry, 15*(9), 1723. https://doi.org/10.3390/sym15091723
15. Sarker, I. H. (2021). *Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. SN Computer Science, 2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1
16. Pawlicki, M., Kozik, R., & Choraś, M. (2022). *A survey on neural networks for (cyber-) security and (cyber-) security of neural networks. Neurocomputing, 500*, 1075–1087. https://doi.org/10.1016/j.neucom.2022.06.002
17. Hunter, D., Yu, H., Pukish III, M. S., Kolbusz, J., & Wilamowski, B. M. (2012). *Selection of proper neural network sizes and architectures—A comparative study. IEEE Transactions on Industrial Informatics, 8*(2), 228–240. https://doi.org/10.1109/TII.2012.21879
18. Teslyuk, V., Kazarian, A., Kryvinska, N., & Tsmots, I. (2021). *Optimal artificial neural network type selection method for usage in smart house systems. Sensors, 21*(1), 47. https://doi.org/10.3390/s21010047
19. Jamous, R., ALRahhal, H., & El-Darieby, M. (2021). *Neural network architecture selection using particle swarm optimization technique. Applied Artificial Intelligence, 35*(15), 1219–1236. https://doi.org/10.1080/08839514.2021.1972251
20. Yang, H., & Effatparvar, M. (2025). *A deep learning based intrusion detection system for CAN vehicle based on combination of triple attention mechanism and GGO algorithm. Scientific Reports, 15*(1), 19462. https://doi.org/10.1038/s41598-025-04720-y
21. Hacking and Countermeasure Research Lab (HCRL). (n.d.). *Datasets.* https://ocslab.hksecurity.net/Datasets
22. Robert Bosch GmbH. (1991). *CAN specification: Version 2.0.*
23. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* The MIT Press.
24. Miller, C., & Valasek, C. (2014). *A survey of remote automotive attack surfaces* [Technical report]. IOActive.
25. Naidu, G., Zuva, T., & Sibanda, E. M. (2023). *A review of evaluation metrics in machine learning algorithms.* In R. Silhavy & P. Silhavy (Eds.), *Artificial intelligence application in networks and systems (CSOC 2023)* (pp. 15–25). Springer. https://doi.org/10.1007/978-3-031-35314-7_2