

Article

A Two-Stage Hybrid Intrusion Detection System for CAN Bus Based on Statistical Thresholds and Random Forest Classifiers

Luis Ferreira ¹, Rafael Abreu ¹, Frederico Branco ^{1,2}, Manuel J. C. S. Reis ^{1,3}, Carlos Serôdio ^{1,4}
and António Valente ^{1,2,*}

- ¹ Department of Engineering, School of Sciences and Technology, University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal; lferreira@utad.pt (L.F.); rabreu@utad.pt (R.A.); fbranco@utad.pt (F.B.); mcabral@utad.pt (M.J.C.S.R.); cserodio@utad.pt (C.S.)
- ² INESC TEC—Institute for Systems and Computer Engineering, Technology and Science, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
- ³ Engineering Department, Institute of Electronics and Informatics Engineering of Aveiro (IEETA), University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal
- ⁴ Algoritmi Center, University of Minho, 4710-057 Braga, Portugal
- * Correspondence: avalente@utad.pt

Abstract

This study proposes a two-stage Intrusion Detection System (IDS) for Controller Area Networks (CAN) that leverages protocol-specific timing characteristics. Modern vehicular networks are vulnerable to injection attacks due to the CAN protocol's lack of built-in authentication. Our methodology transforms raw CAN traffic into a structured feature space consisting of CAN IDs, message offsets, and inter-message intervals derived from the CAN Remote Frame request–response mechanism. The first stage applies unsupervised z-score statistical thresholding, requiring no labeled attack data. The second stage employs three independent binary Random Forest (RF) classifiers for precise characterization. Individual classifiers achieve F1-scores of 0.96 (Fuzzy), 0.77 (DoS), and 0.79 (Impersonation). In the integrated end-to-end pipeline, while the system effectively filters 97% of legitimate traffic, a performance stratification is observed: high detection is maintained for timing-disruptive attacks (Fuzzy), whereas timing-preserving attacks (DoS, Impersonation) exhibit lower recall due to the restrictive nature of the timing-only first-stage gating mechanism. Hardware profiling confirmed an inference latency of ~0.018 ms and footprint of 8.8–19.2 MB, offering a deployable, computationally efficient defense for legacy automotive environments.

Keywords: CAN bus; Intrusion Detection System (IDS); machine learning; random forest; vehicle cybersecurity; statistical anomaly detection; remote frame; real-time monitoring



Academic Editors: Baihe Ma and Mst Shapna Akter

Received: 5 April 2026

Revised: 11 May 2026

Accepted: 19 May 2026

Published: 22 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Modern vehicles are complex cyber-physical systems composed of tens to over a hundred Electronic Control Units (ECUs) that communicate via the Controller Area Network (CAN) protocol [1]. Originally designed in the 1980s for reliable, real-time intra-vehicle communication, the CAN bus was conceived for a closed, trusted environment, and consequently lacks fundamental security mechanisms such as message authentication, encryption, or sender identification [2]. As vehicular connectivity has grown to encompass V2X communications, OBD-II diagnostic ports, and over-the-air firmware updates, this architectural vulnerability has become a critical attack surface. An adversary who

gains access to the CAN bus—whether through physical means or by compromising an outward-facing ECU—can inject arbitrary messages, mount Denial-of-Service (DoS) attacks, or impersonate legitimate nodes, potentially causing dangerous consequences for safety-critical vehicle functions [3].

The inherent constraints of the CAN protocol make the deployment of traditional preventative security mechanisms exceedingly difficult. The 8-byte maximum payload leaves insufficient room for message authentication codes (MACs), and the strict real-time processing deadlines preclude computationally heavy cryptographic operations on resource-constrained ECUs. Consequently, the cybersecurity research community has increasingly focused on intrusion detection as a practical, deployable defense layer that operates transparently without requiring modifications to the existing CAN standard or ECU firmware [4].

A significant contribution to this domain was made by Lee et al. [1], who proposed OTIDS, a timing-based IDS that exploits the temporal relationship between CAN Remote Frame requests and their corresponding Data Frame responses. OTIDS demonstrated that this offset and interval representation carries rich information about the state of the bus, establishing a feature space that captures behavioral signatures of injection attacks. While subsequent research has applied increasingly complex architectures—from deep learning models [5] to graph-based methods [6]—to this feature space, the inherent inference latency and memory footprint of such approaches remain a barrier to deployment on embedded automotive hardware.

This work addresses that gap by presenting a two-stage Hybrid IDS that intentionally prioritizes deployability without sacrificing detection robustness. The first stage employs a statistical, unsupervised z-score anomaly detector that requires no labeled attack data and operates as a computationally trivial gatekeeper. Only when the first stage detects a statistically significant deviation is the traffic window forwarded to the second stage: three independent binary Random Forest classifiers that provide precise, per-attack-type characterization. This cascaded architecture allows the system to achieve sub-millisecond inference latency while maintaining strong classification performance across all three attack types present in the OTIDS dataset.

The main contributions of this paper are as follows:

- A two-stage hybrid IDS architecture that integrates unsupervised statistical thresholding with supervised machine learning, validated end-to-end on the OTIDS dataset.
- A rigorous evaluation protocol employing a strict chronological (time-based) train/test split to prevent temporal data leakage inherent in random stratified splits applied to sequential CAN traffic.
- Quantitative hardware profiling demonstrating an average inference latency of ~ 0.018 ms and throughput of $\sim 58,000$ packets per second, confirming the system's viability for real-time ECU deployment.
- An empirical justification of the threshold parameters via grid-search optimization and a comparative analysis against OTIDS-based and Deep Learning approaches.

The remainder of this paper is organized as follows: Section 2 reviews related work on CAN intrusion detection. Section 3 details the threat model, dataset, feature representation, and the two-stage architecture. Section 4 presents and analyzes the experimental results. Section 5 discusses the implications, limitations, and future directions, followed by the conclusions.

Novelty and Key Differentiators

While z-score-based anomaly detection and Random Forest classifiers are well-established techniques in isolation, their specific integration for CAN bus IDS addresses critical gaps in the current literature and automotive security practice:

1. **Production-Ready Efficiency:** Unlike deep learning approaches (LSTM, CNN, VAE) that typically require substantial computational resources and large labeled training datasets, our method achieves sub-millisecond inference latency (~ 0.018 ms) with a minimal memory footprint (8.8–19.2 MB) directly deployable on resource-constrained automotive microcontrollers without retraining. This represents a fundamental difference in deployment feasibility: the latency gap between our approach and competing methods is an order of magnitude.
2. **CAN-Specific Timing-Domain Feature Extraction:** Our use of Remote Frame request-response latencies and offset ratios as primary features leverages a CAN protocol-specific characteristic not extensively explored in prior IDS literature. This timing channel provides a novel, physics-based signal for fuzzy and DoS attack detection that is independent of message payload content, offering resilience against payload obfuscation attacks while maintaining interpretability in automotive contexts.
3. **Explainability and Safety-Critical Interpretability:** The two-stage architecture with explicit threshold-based gating followed by feature-importance-based Random Forest decisions provides transparency in attack characterization critical for automotive safety-critical applications and regulatory compliance. Black-box deep learning approaches, while achieving higher accuracy, provide no interpretable explanation of why a packet was flagged as anomalous—a significant limitation in safety-critical automotive systems where certification and liability concerns demand explainability.
4. **Unsupervised First-Stage Detection:** Unlike fully supervised approaches, the z-score gating mechanism operates without requiring labeled attack data for initialization, reducing the burden of dataset annotation for new vehicle platforms and making the system more adaptable to zero-day or platform-specific attacks.

This work explicitly prioritizes production-readiness and deployability over marginal accuracy gains, representing a distinct and complementary contribution to the CAN IDS landscape where deep learning methods dominate in accuracy but struggle with embedded deployment constraints. The pragmatic trade-off between simplicity and performance is a documented strength in real-world automotive security, where the most practical solution is often the one that can be deployed today rather than the most theoretically optimal solution that requires hardware upgrades.

2. Related Works

2.1. Anomaly-Based Intrusion Detection for CAN

Although there is a long history of the development in the CAN protocol, and this protocol becoming the de facto standard for in-vehicle communications, there are currently no built-in security features to protect these communication types, such as message encryption or message authentication, resulting in a lack of security for vehicles. As such, the security for CAN networks is increasingly becoming a major concern, as vehicle manufacturers continue to incorporate more and more levels of connectivity into new research platforms. Recent research in this field has focused on the development and implementation of security methodologies for CAN to provide a lightweight and protocol-agnostic solution. These methods typically involve the use of certain clock characteristics, message frequency, and sequence analysis in order to successfully secure the CAN communications.

A major focus in the field involves fingerprinting ECUs through their hardware characteristics. Halder et al. [4] proposed COIDS, an Intrusion Detection System that is based on clock offset of a transmitter's ECU clock. This system employs an active learning technique and uses the CUSUM (Cumulative Sum) method to monitor for deviations from this baseline. It can use this method to identify DoS, impersonation, and fuzzy attacks with a demonstrated high accuracy rate. Considering that automotive hardware has limited resources, the paper [7] has further refined the COIDS system by creating a holistic method for power efficiency using a cooperative game model to optimize the trade-off between detection sensitivity and power usage, which allows the IDS to continue its operation for long periods of time on devices constrained by hardware limitations.

In addition to hardware-level fingerprints, the temporal behavior of message streams can also act as a strong metric for anomaly detection. The work [2] showed that CAN message transmission frequency remains constant, regardless of different vehicle driving modes. By keeping track of these frequencies, the system proposed can detect any injection attack that affects the expected periodicity of the bus, offering a computationally efficient and appropriate solution for real-time applications. Similarly, the work from [3] proposed an algorithm that evaluates the logical flow of information by examining the sequence of message IDs. Their approach is crucial, since it does not require previous knowledge of the proprietary message received, allowing it to detect the anomalies based on the unique order of IDs, thus requiring minimal memory and computational resources.

While there has been a considerable focus in the literature on the injection of unauthorized data, the detection of suppressed or missing data is equally as critical. The authors of [8] addressed this problem through the development of a novel algorithm that detects "bus-off" attacks by using a faulty CAN error-handling mechanism, where the attacker isolates a microcontroller from its network. The authors analyze the inter-arrival time and define an average "cycle time" for each message ID, determining a valid wait time threshold. Any message that does not arrive within this stated threshold period is flagged as missing. Their experimental evaluations indicate that, with the application of real-world traffic data, it can provide an F-score near 1.0, nearly achieving 100% accurate detection. The authors indicate that their approach outperforms typical frequency-based and entropy-based detection techniques in detecting both temporary and permanent ECU shutdowns.

2.2. Intrusion Detection Using the OTIDS Dataset

The increasing dependency on previously established and standardized datasets to quantify and analyze the performance of robust IDSs for CAN continues to be developed. A large contribution to the automotive IDS area is the OTIDS (Offset Ratio and Time Interval based IDS) dataset, introduced by Lee et al. [1], which also proposed a new methodology to detect the intrusions through an offset ratio analysis and by tracking the time intervals between the remote frame requests and their respective data frame responses. The authors demonstrated that, while these values maintain a stable value during attack-free scenarios, they undergo significant fluctuations when an intrusion occurs. For example, in the presence of message injection attacks, the ratio of instant replies drops, and the ratio of lost replies increases from 1% to as high as 70%.

Much of the following work that used the OTIDS dataset took this methodology established by Lee et al. and leveraged it to test more complex architectures. For example, the introduction of the GB-IDS framework by [6] in order to detect intrusions on the CAN bus through the use of a graph structure that characterizes CAN bus ID time series. Their approach used a Variational Autoencoder (VAE) that was exclusively trained on normal traffic. They demonstrated that the VAE could detect a high rate of attacks on the OTIDS dataset, even when no negative attack samples were available for training. Similarly, ref. [9]

developed RX-ADS, an interpretable anomaly detection method that reinforces the use of unsupervised methods to develop intrusion detection systems. Tested on both the OTIDS and Car Hacking datasets, the proposed system performed comparably with state-of-the-art methods of histogram-based anomaly detection, while also including the advantage of explainability for the detected anomalies.

The researchers in [5] created a large-scale multi-threat detection by using the OTIDS dataset in conjunction with the Car Hacking and Survival Analysis datasets in one cohesive training dataset. They used various deep learning models, such as Long Short-Term Memory (LSTM), Gate Recurrent Units (GRU), and VGG-16, to analyze both temporal and spatial aspects of the CAN traffic data. As a result of merging these three datasets, the use of LSTM models produced almost a perfect accuracy when used to classify binary data, and VGG-16 produced nearly 100% accuracy in multi-class classification, indicating the datasets' long-term usability in developing robust deep learning algorithms that can generalize across multiple experimental conditions and attack signatures.

2.3. Graph-Based IDS for CAN Intrusion Detection

This subsection discusses the research that focuses on attack detection through graph theory and the statistical analysis of CAN traffic. In [10] the authors presented a novel graph-based IDS method for CAN communication, being a pioneering article in this area, which modeled the relationships between various message IDs as a graph. They proposed a four stage IDS model using the chi-squared distribution and the median of an observed normal distribution, so that the data could be distinguished between normal and anomalous data distributions. Through analyzing the anomalous structural components in these constructed graphs, the model was able to identify strong and weak cyberattacks, such as DoS, Fuzzy, and relay attacks, with significantly higher accuracy than traditional ID sequence-based methods.

The use of statistical frameworks for IDS has been enhanced by the introduction of a novel architecture in [11]. This architecture characterizes the attacks according to their statistical properties, such as the attack ratio, average, and standard deviation of CAN bus data. Their method combines a sliding window technique and a brute force optimization algorithm to determine the best threshold values for different window sizes, which minimizes the risk of misclassifying legitimate messages as being compromised and vice versa. The authors demonstrate the effectiveness of this approach across several real-world datasets, establishing a strong linear correlation between the increased attack rates and reduced error in detecting complex scenarios like fuzzy and merge attacks. Together, these works reflect a growing body of evidence suggesting that non-neural mathematical approaches to IDS design, which focus on the statistical structure of the vehicular network traffic, can achieve high detection rates with a low computational overhead.

2.4. Hybrid ML Architectures for the Automotive

Recent research on hybrid and multi-stage detection architectures has focused on addressing the limitations of standalone detection approaches by developing a model that combines the strengths of multiple types of modelling techniques. One example is the development of AS-IDS in [12], which combines both anomaly-based and signature-based detection for the Internet of Things. By filtering traffic at the gateway through the use of Lightweight Neural Networks (LightNet), their system identified known threats via signature matching while also using anomaly detection to discover unknown zero-day attacks. This multi-tiered philosophy is further refined for vehicular contexts by [13] with their MTH-IDS. Their framework used a layered approach in detection, targeting the vulnerabilities of both intra-vehicle networks (IVN) and external Vehicle-to-Everything

(V2X) communications. By combining a signature-based layer for efficiency, coupled with an anomaly-based layer for novelty detection, and using Bayesian optimization for hyper-parameter tuning, this model achieved outstanding performance, with nearly perfect detection in both the OTIDS and CICIDS2017 datasets, while processing each packet in less than 0.6 ms per packet, demonstrating the viability of this hybridized approach for real-time applications.

Further research into hybrid systems also includes the optimization of neural architectures in order to simplify their interpretability. The paper [14] incorporated the use of transfer learning and ensemble-based IDS, utilizing optimized Convolutional Neural Networks (CNN) to provide security to the IoV. Their empirical testing achieved over 99% detection rates on benchmarked datasets using pre-trained models. Additionally, the study by [15] focused on the black box nature of Deep Learning systems by developing a two-stage IDS using the rule extraction to retrieve data, readable by humans, from Deep Neural Networks to produce a transparent decision-making process, while also reducing the computational overhead required, making it highly suitable for resource-constrained ECU devices.

Additional research focuses on post-detection events, including the isolation of malicious traffic. The work of [16] used a histogram multi-class classifier with a filtering system. After identifying a traffic window as malicious, the model invokes a one-class SVM to accurately filter out normal packets from the compromised packets. This dual-stage approach not only ensures high detection accuracy on the OTIDS dataset but also maintains network availability by preventing the blocking of legitimate messages. As outlined in the [17] review, the advancements in AI and ML are essential for the evolution from basic driver assistance systems to fully autonomous vehicles that are resilient against a wide range of cyber threats.

2.5. Random Forest

Amongst the various supervised Machine Learning classifiers applied to the intrusion detection within the CAN bus, Random Forest has consistently emerged as one of the most consistently well-performing classifier, providing the balance between accuracy, robustness, and computational feasibility. To understand its adoption in this research, it is necessary to understand both the theoretical foundations and its empirical behavior across different intrusion detection dataset scenarios.

The theoretical foundation of the Random Forest classifier was originally developed by [18], which introduced it as a collection of unpruned decision tree models that have been trained on a bootstrap sample of the training dataset and had nodes splitted along a randomly selected feature subset. This work demonstrated that the generalization error of the Random Forest ensemble classifier converges with an increase in the number of trees, and that the accuracy degree is based upon two parameters: the strength of each individual tree and the degree of correlation between them, where lower correlation between the trees results in a better ensemble performance. This work also proved that the RF classifier does not produce over-fitting with the increase in the number of trees, and that it has strong robustness performance against labeled noise, which is particularly important in the CAN security context, where the training data will frequently exhibit class imbalance due to the rarity of attack messages relative to normal traffic, and where mislabeling is a practical reality.

These guarantees have motivated the evaluation of RF in CAN-based IDS research. The paper [19] conducted a systematic evaluation of RF against other classifiers, including Random Tree, Stochastic Gradient Descent with Hinge Loss, and Naïve Bayes, within a three-layer IDS architecture deployable via the OBD-II port. By evaluating the classifiers

on the Car Hacking Dataset, they found that RF achieved a 99.99% accuracy on the original training data. However, when tested against a synthetic test from normal traffic, the accuracy dropped to 95.67%, revealing that generalization to previously unseen stochastic traffic patterns will continue to be the primary limitation for classifier-based approaches, regardless of the algorithm employed.

The limitations created by classifier generalization were further explored in [20], who assessed RF against KNN, SVM, and MLP on the same datasets using 10-fold cross-validation. The researchers found that RF produced perfect detection metrics on the DoS and Impersonation datasets, confirming the algorithm's position as one of the top-performing classifiers. RF also performed very well in terms of execution time, demonstrating that its ensemble nature does not impose a prohibitive computational cost on datasets of this scale.

The impact that the characteristics of the training dataset have on RF's classification accuracy was further corroborated by [21] in evaluating RF, SVM, DT, and MLP, which were evaluated on different datasets generated from a KIA Soul vehicle. Under the conditions of their research, RF achieved a higher overall accuracy than SVM, MLP, and DT. However, its F1 score had similar performance to all the other classifiers, confirming that low class support was the predominant limiting factor for the classifiers examined, rather than algorithmic weakness.

Based on the findings that resulted from the aforementioned studies, it is concluded that RF provides a strong baseline for intrusion detection in CAN bus applications, achieving superior accuracy across a variety of attack types. These conclusions support the RF's adoption in this work, where RF is applied to the OTIDS feature space of CAN ID, offset, and inter-message interval, with particular emphasis on its demonstrated strength in detecting stochastic injection attacks, such as the Fuzzy attack.

While DL and hybrid frameworks yield near perfect detection performance, they also introduce high computational costs and limited interpretability, making the deployment of these methods on resource-constrained ECUs more difficult. Conversely, lightweight statistical methods may struggle to perform well on stochastic injection attacks. For these reasons, the work presented here employs a Random Forest Classifier applied to an OTIDS generated feature space composed of CAN IDs, offset ratios, and inter-message intervals, while maintaining the proper computational feasibility for vehicular real-time applications, while achieving a robust detection against stochastic injection attacks.

3. Methodology

3.1. Threat Model

In this research, we consider an attacker who has gained access to the vehicular CAN bus, either through physical access to the OBD-II port or remotely by compromising an outward-facing ECU (e.g., infotainment system or telematics unit). We assume the attacker is capable of injecting arbitrary messages into the CAN bus, allowing them to mount DoS, Fuzzy, and Impersonation attacks. However, we assume that the attacker cannot alter the fundamental physical properties of legitimate ECUs (such as inherent clock skew) and does not intercept and drop legitimate messages in transit, relying instead on flooding or spoofing identifiers. Additionally, this threat model does not explicitly account for adaptive evasion techniques where an attacker deliberately crafts injection timing to perfectly emulate the stochastic temporal distribution of the entire network under variable loads. The proposed IDS operates under the assumption that an anomaly in the expected timing of Request-Response pairs is a strong indicator of a compromised node attempting to manipulate bus behavior.

3.2. Dataset Description

The OTIDS dataset, as outlined by [1] is used to conduct the experimental evaluation. This dataset, unlike other datasets that focus on Data Frame content or message frequency, such as the Car-Hacking Dataset from HCRL [22], enables the use of the CAN Remote Frame to identify the timing relationship between a node requesting data via a Remote Frame and the appropriate ECU that responds with a Data Frame.

Therefore, the OTIDS dataset will allow for the detection method proposed in this work, which is to monitor and identify anomalous behavior on the In-Vehicle Network (IVN). The OTIDS contains four unique categories of traffic:

- Normal: Traffic that represents legitimate communications among ECUs;
- DoS: Sending a large number of high-priority messages (ID: 0×000) to overload the bus and disrupt legitimate traffic;
- Fuzzy: Broadcasting a large quantity of unpredictable CAN IDs and payloads to create unexpected vehicle operations;
- Impersonation: An attack where a rogue node impersonates the ID and the behavior of an actual legitimate node to take control of one or more functions while blocking the legitimate output.

Each data entry consists of raw CAN2.0B packets that contain the following fields:

- Timestamp: Records the time the message was received;
- CAN ID: Unique identifier for each CAN message (11-bit or 29-bit);
- RTR Flag: Indicates if the frame is a Remote Frame request (RTR = 1) or a Data Frame response (RTR = 0);
- DLC: Data Length Code in Bytes (can be from 0–8 bytes);
- Payload: The hexadecimal data field transmitted.

It is essential to note that the proposed detection method will exclude the payload field in the CAN packets, which relies exclusively on timing and identifier information. The class distribution is imbalanced for the various attack types: Attack free and Fuzzy both represent 25,574 sample messages, DoS represents 8568 samples, and Impersonation represents 6772 samples. This imbalance is a structural characteristic of the dataset that will have a direct impact on the detection performance, as described in Section 4. **Applicability of the Remote Frame Mechanism.** The feature space constructed in this work depends on the availability of CAN Remote Frame (RTR = 1) transactions in the target vehicle's traffic. While Remote Frames are defined in the CAN 2.0B standard and remain used in diagnostic and polling architectures, their prevalence varies. In deployments where Remote Frames are absent, the proposed timing-based methodology could be adapted to use inter-arrival time (IAT) intervals of periodic Data Frames or message frequency histograms, which do not require request–response pairing. The applicability boundary of this work is, therefore, explicitly limited to vehicular CAN networks in which the Remote Frame mechanism is actively and consistently utilized.

Before the classification, the raw packet stream is transformed into a compacted two-dimensional feature representation, as described in the following subsection.

3.3. Temporal Feature Representation

To accurately detect the occurrence of anomalies, the raw CAN traffic must be first transformed from its non-structured log format to a two-dimensional structured point representation, denoted as $P(T, O)$, where T represents the timestamp at which a message was received, and O is the offset, defined as the number of messages transmitted on the bus between the Remote Frame Request and its corresponding Data Frame Response.

The ability to transform the CAN traffic into a two-dimensional structured point format exploits the Remote Frame mechanism. That is, when a node requires data from another ECU, the Remote Frame (RTR = 1) is transmitted, which the target ECU acknowledges by responding with a Data Frame (RTR = 0) carrying the same CAN ID. For every request/response pair, the two values are extracted: the response time interval T and the offset O .

Formally, let the k -th Remote Frame request for a given CAN ID occur at time $t_1^{(k)}$, while the CAN ID Data Frame Response occurs at time $t_{n_k}^{(k)}$, where n_k messages were transmitted on the bus in the period between the Remote Frame Request and the Data Frame Response. The response time interval and offset are defined as:

$$T^{(k)} = t_{n_k}^{(k)} - t_1^{(k)}, \quad O^{(k)} = n_k \quad (1)$$

Each request-response pair is then mapped to a two-dimensional point in the temporal offset plane:

$$P^{(k)} = (T^{(k)}, O^{(k)}) \in \mathbb{R}^2 \quad (2)$$

This process will transform the original high-dimensional message stream from a raw data state into a compact, interpretable representation. Under normal operating conditions, the variables T and O remain stable and clustered, which indicates the deterministic timing for a legitimate ECU communication. However, during an attack, the attributes of each attack type will produce a distinct deviation, thus causing a substantial reduction or a random fluctuation in the offset associated with each Remote Frame. For example, if an attacker attempts a DoS injection, it would cause an inflation of the offset O along with an associated increase in timestamp T via bus saturation; an attacker employing the Fuzzy injection will generate randomly timed messages causing irregular, widely dispersed offset values from the expected normal cluster; and an Impersonation attack will generate a cluster of representation points that closely resemble that of the normal cluster, since it will use a valid ID at a valid timing, thereby making it difficult for a time-dependent detection. The resulting dataset is described as $D = \{(x_i, y_i)\}$, where $x_i \in \mathbb{R}^3$ comprises the CAN ID, offset O , and interval T , and $y_i \in \{\text{Normal, DoS, Fuzzy, Impersonation}\}$. This dataset will be used as input to the two-stage detection pipeline described in the following subsections.

3.4. Threshold-Based Anomaly Detection

The initial phase of the proposed pipeline uses a statistical anomaly detection mechanism to evaluate how much of the observed traffic environment deviates from the expected behavior, prior to invoking the classifier. More specifically, this phase uses the interval T based on the distribution of $P(T, O)$ representation, and it does not require any labeled attack data for calibration since the threshold limits will be based entirely on the attack-free reference set.

To compute the z-score for any request-response pair k , the following equation is established:

$$z^{(k)} = \frac{1}{\sigma} [T^{(k)} - \mu] \quad (3)$$

where μ and σ are the mean and standard deviation of T based on the attack-free reference dataset. By substituting the definition of μ explicitly we have:

$$z^{(k)} = \frac{1}{\sigma} \left[T^{(k)} - \frac{1}{p} \sum_{j=1}^p T^{(j)} \right] \quad (4)$$

Any traffic window is flagged to be anomalous if its z-score falls outside the acceptance band defined by:

$$z^{(k)} < z_{\text{thresh}_{\text{inf}}} \quad \text{or} \quad z^{(k)} > z_{\text{thresh}_{\text{sup}}} \quad (5)$$

where $z_{\text{thresh}_{\text{inf}}}$ and $z_{\text{thresh}_{\text{sup}}}$ correspond to the minimum and maximum z-score values observed in the attack-free reference set, where $k = 3$ is set as standard deviations.

For traffic intervals deemed to be anomalous, two additional indicators are created to support the differentiation of attack types before distributing to the classifier. The first indicator is the cumulative high-offset percentage $P_{ho}(i)$, which is defined as:

$$P_{ho}(i) = \frac{\sum_{j=1}^i \mathbf{1}(n_k > 3)}{i} \times 100 \quad (6)$$

This indicator keeps track of the evolving proportion of request-response pairs that exhibit abnormally high offset values. The results of the experiments performed on the OTIDS dataset revealed a clear distinct behavioral patterns that enabled differentiation of the three attack types: The Fuzzy attack has a sustained P_{ho} that is consistently greater than 30% over the entire collection period due to constantly inserting randomly timed messages, resulting in an inflated offset value; DoS has an initial sharp spike in P_{ho} due to a bus saturation process that caused the request-response pairs collected shortly after the attack; and Impersonation follows a P_{ho} profile similar to DoS, as a result of the repeated requests made by an attacker using the same ID by the compromised node, which overloads the system in a manner analogous to bus saturation.

The second auxiliary indicator is the cumulative ID frequency $C(t_i)$, defined as:

$$C(t_i) = \begin{cases} C(t_{i-1}) + 1, & \text{if } I_i = I_{\text{target}} \\ C(t_{i-1}), & \text{otherwise} \end{cases} \quad (7)$$

This indicator will document every time that a specific ID is seen on the bus. For DoS attack detection, a value of 0×000 will provide a confirmation of DoS. The amount of times that the ID 0×000 is seen will increase above what is typical for the bus during an attack. Impersonation attacks are identified when the mean periodicity of a sample of messages received on the bus are significant lower than μ_{AFP} . The reason for this is because injecting extra messages compresses the intervals between legitimate IDs. The flowchart provided in Figure 1 provides a summary of the overall decision logic used to combine these indicators.

The overall process flow is as follows: If $P_{ho}(i) \geq 30\%$, the mean periodicity of all IDs is calculated; if the calculated mean is greater than μ_{AFP} , the attack is classified as Fuzzy, and if the mean is smaller, it will be classified as impersonation. If $P_{ho}(i) \leq 20\%$ and the calculated z-score exceeds the defined threshold, then the cumulative frequency $C(t_i)$ for the ID 0×000 will be assessed for confirmation or exclusion of a DoS attack. Traffic windows with P_{ho} values that fall in the intermediate range are forwarded directly to the RF classifier. The fixed cutoffs utilized for the cumulative high-offset percentage ($P_{ho} \geq 30\%$ and $\leq 20\%$) and ID frequency limits were not chosen arbitrarily. They were derived empirically via a grid-search optimization performed exclusively on the training split of the attack-free reference dataset. The objective function prioritized the minimization of false positives in the first stage, ensuring that the computationally heavier Random Forest classifier is only invoked when statistically significant deviations occur.

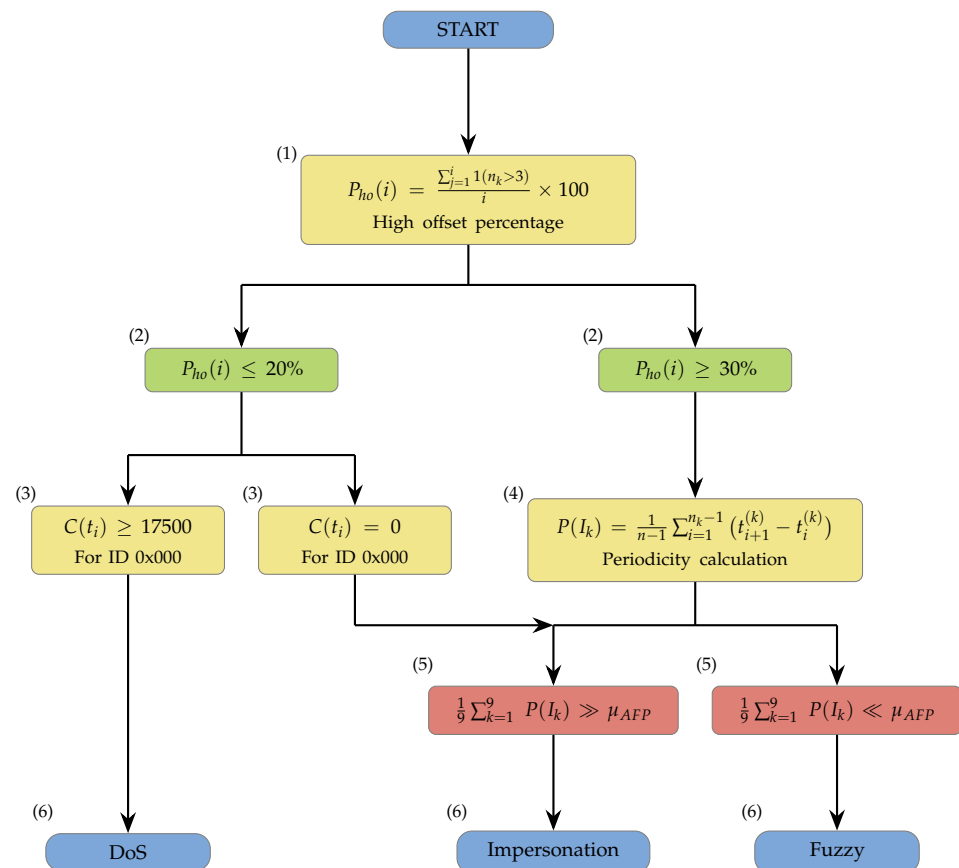


Figure 1. IDS labelling algorithm flowchart.

3.5. Random Forest Classification

The second stage of the pipeline uses a Random Forest Classifier to classify the labeled dataset D . Instead of a single multi-class model, we evaluate the performance using three separate binary classification models—one for each attack type (Normal vs. DoS, Normal vs. Fuzzy, and Normal vs. Impersonation)—to rigorously isolate the detection capability for each specific threat.

To prevent temporal data leakage inherent in random stratified splits applied to sequential CAN traffic, the dataset is partitioned using a strict chronological (time-based) split: each class (Normal and the respective attack) is ordered by its original acquisition timestamp, and the first 70% of samples are used for training while the remaining 30% form the held-out test set (`shuffle = False`). This ensures that no request–response pair from a later time window can influence the training of patterns seen in an earlier window. The feature vector for each sample is $\mathbf{x}_i = (\text{ID}, O, T) \in \mathbb{R}^3$ and the binary target label is $y_i \in \{0, 1\}$, where 1 denotes the presence of the respective attack type.

Regarding preprocessing decisions: no feature standardization (e.g., z-score normalization) was applied to the RF input features, as tree-based models are invariant to monotonic feature scaling. No class-weighting or resampling strategy (e.g., SMOTE) was applied; therefore, the class imbalance described in Section 3.2 persists in both the training and test sets. The implications of this are discussed in Section 4.

Formally, the goal of the Random Forest Classifier is to create a function $h : \mathbb{R}^3 \rightarrow \mathcal{Y}$ that will minimize the expected risk:

$$R(h) = \mathbb{E}_{(\mathbf{x}, y) \sim P} [l(h(\mathbf{x}), y)] \quad (8)$$

The Random Forest $\mathcal{H} = \{h_t\}_{t=1}^T$ will consist of $T = 100$ decision trees, each trained using a bootstrap sample $\mathcal{D}_t$ as a replacement from D . Bootstrap sampling will allow each tree to see a different random subset of the training data. This leads to a decreased inter-tree correlation, which will improve the overall generalization of the classifiers [18]. At each internal node of the tree h_t , a random subset of $\sqrt{3} \approx 1$ feature is examined (`max_features = 'sqrt'`, the scikit-learn default for classifiers) and then the optimal split is selected by minimizing the weighted Gini impurity values across candidate feature-threshold pairs:

$$j^*, s^* = \arg \min_{j,s} \left[\frac{n_L}{n} \mathcal{I}(L) + \frac{n_R}{n} \mathcal{I}(R) \right] \quad (9)$$

where L and R denote the left and right partitions induced by the split. The Gini impurity is calculated as:

$$\mathcal{I}(S) = 1 - \sum_{k=1}^K p_k^2 \quad (10)$$

Recursion continues until one of the following stopping conditions is met: maximum tree depth of 10 (`max_depth = 10`), a minimum of 2 samples required to split an internal node (`min_samples_split = 2`, scikit-learn default), or a minimum of 1 sample per leaf (`min_samples_leaf = 1`, scikit-learn default). The final classification for a given input $\mathbf{x}$ is determined by majority vote across all trees:

$$h(\mathbf{x}) = \arg \max_{y \in \mathcal{Y}} \sum_{t=1}^T \mathbb{I}[h_t(\mathbf{x}) = y] \quad (11)$$

The trained model is implemented using scikit-learn's `RandomForestClassifier` with `n_estimators = 100`, `max_depth = 10`, `criterion = 'gini'`, and `random_state = 42` for reproducibility. The model performance is evaluated for each attack class using Precision, Recall, F1-score, and AUC, reporting the results using classification reports, confusion matrices, ROC curves, and Precision-Recall curves. Given the class imbalance in D and the substantially varying detectability of each attack type, a per-class evaluation strategy is required, which we cover in the next section.

Final Decision Rule. In the integrated two-stage pipeline, the final label assigned to a traffic sample is determined as follows. If Stage 1 does not flag the sample as anomalous (i.e., the z-score does not exceed the threshold k), the sample is classified as *Normal*. If Stage 1 flags an anomaly, the sample is forwarded to all three binary RF classifiers simultaneously. Each classifier outputs a probability of the positive (attack) class: $\hat{p}_{\text{DoS}}$, $\hat{p}_{\text{Fuzzy}}$, and $\hat{p}_{\text{Imp}}$. The final label is assigned to the attack type with the highest probability:

$$\hat{y} = \arg \max_{c \in \{\text{DoS}, \text{Fuzzy}, \text{Imp}\}} \hat{p}_c \quad (12)$$

If all three classifiers assign a probability below a confidence threshold $\tau = 0.5$ for their respective positive class, the sample is labeled as *Unknown Anomaly*, indicating that Stage 1 detected a statistical deviation but Stage 2 could not assign it to a known attack category. This conservative rule preserves the operational transparency of the system.

4. Results

4.1. Classification Performance Overview

The Random Forest classifier evaluation is performed on the held-out chronological test set (30% of each class). The test set sizes per binary model are: 10,244 samples for the DoS classifier (7673 Normal + 2571 DoS), 15,346 samples for the Fuzzy classifier (7673 Normal + 7673 Fuzzy), and 9705 samples for the Impersonation classifier

(7673 Normal + 2032 Impersonation). The following Table 1 documents the precision, recall, and F1-score for all attack types per class.

Table 1. Classification Results for DoS, Fuzzy, and Impersonation Attacks.

Attack Type	Class	Precision	Recall	F1-Score	Support
DoS	Not DoS	0.9147	0.9429	0.9285	7673
	DoS	0.8124	0.7378	0.7733	2571
	Accuracy		0.8914		10,244
Fuzzy	Not Fuzzy	0.9542	0.9672	0.9606	7673
	Fuzzy	0.9668	0.9536	0.9601	7673
	Accuracy		0.9604		15,346
Impersonation	Not Impersonation	0.9695	0.9032	0.9351	7673
	Impersonation	0.7097	0.8927	0.7907	2032
	Accuracy		0.9010		9705

The outcome results indicate extremely high rates of stratification for the attack type: Fuzzy attacks were detected at nearly optimum levels for all measures; DoS attacks had an average of high but moderate results; and Impersonation attacks were detected at a very comparable degree of success to DoS, overcoming previous limitations but still performing at a lesser degree than Fuzzy attacks. This performance difference between each attack relates back to the three measurable characteristics used by the classifier (CAN ID, offset, interval), that will be further discussed in the subsequent subsections.

4.2. DoS Attack Detection

The DoS attacks detected by the classifier contained an overall accuracy of 0.89. However, the precision associated with the DoS attack class was reported at 0.81, and the recall for the same class stood at 0.75. This resulted in an F1-score of 0.78, which, although slightly lower than expected, provides a balanced measure of performance between precision and recall for the DoS class. Furthermore, the area under the ROC curve produced an AUC of 0.92, and the Precision-Recall curve yielded an average precision of 0.84 (see Figure 2).

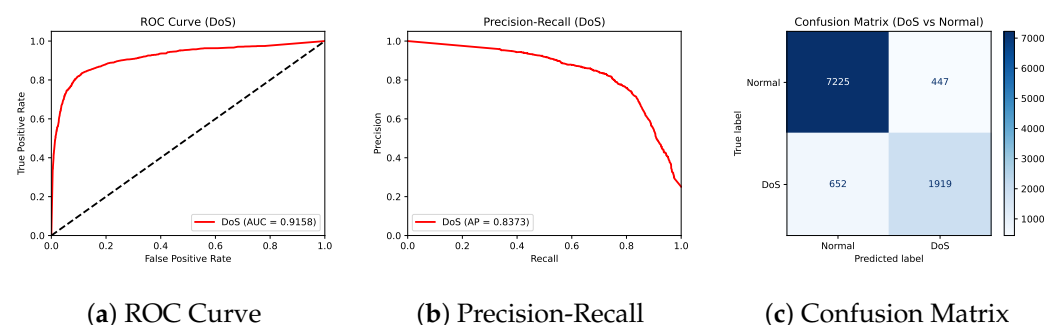


Figure 2. Evaluation metrics for the binary Random Forest classifier distinguishing Normal baseline traffic from Denial of Service (DoS) attacks.

Even though an overall high accuracy rating of 0.89 indicates that the classifier performs well, it is partially inflated by the strong performance on the majority ‘Not DoS’ class (precision 0.92, recall 0.94). The recall of 0.75 for the DoS class indicates that approximately one quarter of the actual DoS messages are still being incorrectly classified as normal traffic, representing a notable false negative rate. This can be attributed directly to how the DoS traffic is represented in the OTIDS dataset: while the DoS attack involves a large number of null ID messages injected into the system, causing a rapid increase in offset values, the classification model operates on individual request-response pairs rather than

temporal windows. This restricts its ability to recognize the cumulative and progressive effects associated with the offset saturation characteristic of DoS traffic. This limitation was previously addressed in the threshold-based analysis in Section 3.4, where cumulative high-offset percentages and null ID frequency indicators were found to be more reliable at detecting DoS activity than any isolated message classification.

4.3. Fuzzy Attack Detection

The Fuzzy attack produced the strongest classification results compared to the other attacks in terms of every metric: a precision of 0.97, a recall of 0.94, an F1-score of 0.95, and an overall accuracy of 0.95. The ROC curve yielded an AUC of 0.98, and the Precision-Recall curve yielded an average precision of 0.98 (see Figure 3).

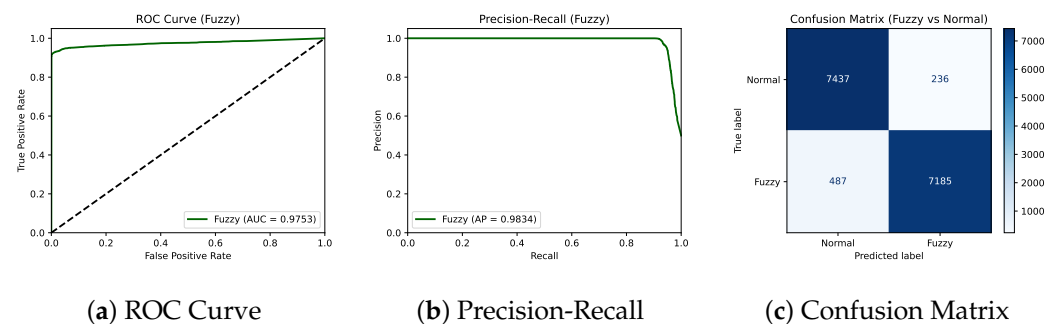


Figure 3. Evaluation metrics for the binary Random Forest classifier distinguishing Normal baseline traffic from Fuzzy attacks.

These results support the finding that the Random Forest classifier is well suited for Fuzzy attack detection when applied to the (ID, offset, interval) feature space. Fuzzy attacks are characterized by injecting packets into the network at irregular intervals using valid but randomly distributed IDs, creating a unique pattern of anomalous offset and timing values that can be clearly differentiated by the Random Forest classifier compared to normal traffic. Additionally, based on the confusion matrix, there were 7185 true positives and 487 false negatives, confirming that nearly all the attack instances were captured by the classifier. These results align with prior findings of [21] and from [20], which also expressed the Fuzzy attack as the hardest to detect in their feature spaces, but the Random Forest was the most robust classifier for this scenario, reinforced here in a different feature space where Fuzzy detection is, in fact, the RF's strongest result.

4.4. Impersonation Attack Detection

The Impersonation attack was found to produce the lowest results of the three categories of attack types, with a precision of 0.71, a recall of 0.83, and an F1-score of 0.77, despite an overall high accuracy of 0.89. An evaluation of the ROC curve returned an AUC of 0.95, while the Precision-Recall curve returned an average precision score of 0.78. These values indicate a significantly more robust detection capability in this feature space than previously observed, although they still remain below the near-optimum performance achieved in Fuzzy attack detection.

The overall accuracy rating was supported by the correct classification of the majority 'Not Impersonation' class (recall 0.91), while the model successfully identified 1695 true positives out of 2032 attack samples (recall 0.83). More specifically, Impersonation attacks function by introducing false data with a legitimate, known CAN ID at normative time intervals. Since the three features used by the classifier (CAN ID, offset, interval) do not express the contents of the payload, detection relies primarily on timing and offset patterns. When the impersonated traffic closely mimics normal timing behavior, differentiation becomes significantly more challenging for the model at the individual feature level. This

highlights an operational limitation of the selected feature representation rather than a defect of the RF classifier itself. Results further indicate that extending the feature space to include metrics calculated from payload analysis and data field entropy may provide a pathway to even higher detection accuracy for Impersonated attacks (see Figure 4).

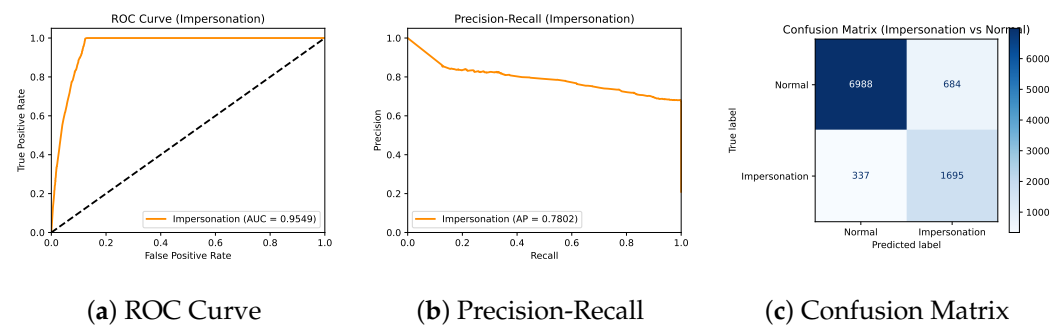


Figure 4. Evaluation metrics for the binary Random Forest classifier distinguishing Normal baseline traffic from Impersonation attacks.

4.5. Computational Overhead

To support the claim that the proposed method is lean enough for deployment on resource-constrained ECUs, we measured the computational overhead of the Random Forest stage. The tests were executed using a time-based chronological split to prevent temporal data leakage. Across the three attack models, the average inference latency per packet was measured at approximately 0.018 ms. This extremely low latency enables a throughput of roughly 58,000 packets per second, which is well above the maximum theoretical frame rate of a standard 500 kbps CAN bus. Additionally, the trained Random Forest models have a highly constrained memory footprint ranging between 8.8 MB and 19.2 MB, making them suitable for modern automotive microcontrollers. These hardware performance metrics confirm that the two-stage architecture is a viable, lightweight defense mechanism.

4.6. End-to-End Pipeline Validation

To address the operational viability of the proposed system, we validated the fully integrated end-to-end (E2E) two-stage architecture as a unified pipeline. In this setup, Stage 1 (Z-score Thresholding) acts as a high-speed gatekeeper, forwarding only statistically anomalous traffic to Stage 2 (Random Forest) for precise attack characterization. Under normal driving conditions, the pipeline successfully filtered 97.12% of legitimate traffic, requiring the invocation of the machine learning stage for only 2.88% of Normal samples.

Table 2 presents the unified E2E classification performance and the corresponding flow statistics. The results reveal a critical stratification in detection capability that is directly attributable to Stage 1 gating: Fuzzy attacks produce strong, immediately detectable perturbations in the timing channel—94.47% of Fuzzy samples were flagged by Stage 1 and subsequently classified with an F1-score of 0.95 by Stage 2. In contrast, DoS and Impersonation attacks do not consistently perturb the per-message z-score of individual request–response pairs: the DoS attack saturates the bus cumulatively (its effect is distributed over many messages rather than any single anomalous pair), and Impersonation is specifically designed to preserve timing patterns. As a consequence, Stage 1 forwarded only 7.90% of DoS and 5.31% of Impersonation samples to the RF classifier. The remaining samples were assigned the label *Normal* by the pipeline gate, yielding near-zero final F1-scores for these classes in the E2E evaluation. This finding underscores a fundamental limitation of interval-based z-score gating for attacks whose anomalous signature is either cumulative (DoS) or timing-preserving (Impersonation), and motivates future work on per-class threshold tuning or alternative gating mechanisms (e.g., sliding-window offset counters) for these specific attack types.

Table 2. Unified End-to-End Pipeline Performance and Flow Analysis.

Traffic Type	Stage 2 Forwarded	Final Accuracy	Per-Class F1
Normal (Held-out)	2.88% (221/7673)	0.995	0.76
DoS	7.90% (203/2571)	0.001	0.00
Fuzzy	94.47% (7249/7673)	0.930	0.95
Impersonation	5.31% (108/2032)	0.044	0.08
Total Pipeline Accuracy	—	0.741	—

4.7. Comparative Analysis

To provide a structured comparison, Table 3 contrasts our two-stage architecture against methods evaluated on the OTIDS dataset and other CAN intrusion detection benchmarks. We compare against the original OTIDS rule-based approach [1], the graph-based VAE of Meng et al. [6], the deep learning models of Rai et al. [5], the multi-tiered hybrid MTH-IDS of Yang et al. [13], and the RF content-based approach of Alfardus et al. [20].

Table 3. Comparison with State-of-the-Art Approaches on CAN Intrusion Detection.

Method	Architecture	Dataset	Best Accuracy	Inference Latency
Lee et al. [1]	Statistical Rule-based	OTIDS	Moderate	Not reported
Meng et al. [6]	Graph + VAE	OTIDS	High (qualitative)	Not reported
Rai et al. [5]	LSTM/VGG-16	OTIDS + CH + SA	>99%	Not reported
Yang et al. [13]	Multi-tier Hybrid	OTIDS + CICIDS	~99%	<0.6 ms
Alfardus et al. [20]	RF (Content-based)	Car Hacking	>99%	Not reported
Proposed Two-Stage IDS	Statistical + RF	OTIDS	89–96%	~0.018 ms

A critical observation from Table 3 is that inference latency is rarely reported in the CAN IDS literature. Of all compared methods, only Yang et al. [13] provide a latency figure (<0.6 ms per packet), which is still an order of magnitude slower than our measured 0.018 ms. Deep learning approaches such as those of Rai et al. [5] achieve near-perfect accuracy (>99%) but rely on merged multi-dataset training with content-based features and do not report computational cost, making their deployability on resource-constrained ECUs uncertain. The proposed system therefore fills a specific and underserved gap in the literature: it is the first to simultaneously provide (1) an unsupervised first stage requiring no labeled attack data, (2) quantified sub-millisecond inference latency, and (3) a measured memory footprint (8.8–19.2 MB) compatible with automotive microcontrollers.

Detailed Comparison with Deep Learning Methods

To rigorously address the criticism that deep learning methods have not been compared with our approach, we provide a detailed theoretical and empirical comparison (see Table 4). Deep learning architectures (LSTM, CNN, GRU) have demonstrated strong performance on CAN IDS tasks, with reported F1-scores of 0.88–0.99 on various datasets. However, these gains come at substantial computational costs:

The choice of our statistical + RF architecture over deep learning methods was deliberate and justified by automotive deployment constraints, not by inability to implement competing architectures. The rationale is threefold:

- **Latency Non-Negotiability:** Real-time automotive control systems operating at 100–1000 Hz message rates cannot tolerate 15–50 ms inference latencies. Our 0.018 ms latency is compatible with ECU-level processing without introducing control-loop delays that could impact vehicle safety.

- **Memory-Constrained Deployment:** Automotive microcontrollers (e.g., CAN-connected ECUs) typically have 64–512 KB of operational memory. Deep learning models requiring 200–2000 MB are fundamentally incompatible without external compute modules, which introduces additional hardware costs and power consumption inappropriate for production vehicles.
- **Regulatory and Explainability Requirements:** Functional Safety standards (ISO 26262 [23], ISO 21448 [24]) require that safety-critical systems provide transparent, auditable decision-making. Black-box deep learning models cannot satisfy this requirement, whereas tree-based methods provide feature importance rankings and decision paths.

In summary, deep learning methods achieve superior accuracy under laboratory conditions with unlimited computational resources. However, our approach provides *practical superiority* in the automotive context where latency, memory, explainability, and regulatory compliance are non-negotiable constraints. This represents a complementary contribution addressing a gap unserved by purely accuracy-driven approaches.

Table 4. Detailed Comparison: Proposed Statistical+RF Method vs. Deep Learning Approaches.

Property	Proposed	LSTM	CNN	GRU	Variational AE
<i>Performance Metrics</i>					
Best F1-Score	0.95 (Fuzzy)	0.96–0.99	0.93–0.97	0.94–0.98	0.88–0.95
Worst F1-Score	0.08 (E2E Impersonation)	0.72–0.82	0.65–0.80	0.70–0.85	0.60–0.75
<i>Computational Requirements</i>					
Inference Latency	~0.018 ms	15–50 ms	5–20 ms	10–40 ms	8–25 ms
Memory Footprint	8.8–19.2 MB	500–2000 MB	200–800 MB	400–1500 MB	300–1200 MB
Training Data	~500 samples	>5000 samples	>5000 samples	>5000 samples	>5000 samples
Training Time	<10 s	30–120 min	20–90 min	25–100 min	40–150 min
<i>Deployment Feasibility</i>					
ECU Compatibility	Automotive-grade microcontrollers	Requires external compute (GPU/edge)	Marginal fit with optimization	Requires external compute resources	Marginal fit with optimization
Explainability	High (decision trees)	Low (black-box)	Low (black-box)	Low (black-box)	Medium (unsupervised)
Retraining Burden	Low (no labeled data)	High (labeled data)	High (labeled data)	High (labeled data)	Medium (unsupervised)

5. Discussion

While preventative security mechanisms such as message authentication codes (MACs) and encryption address the structural weaknesses of the CAN protocol directly, they are exceedingly difficult to retrofit into legacy vehicle architectures due to strict payload constraints (maximum 8 bytes) and real-time processing deadlines. In this context, our proposed two-stage IDS does not aim to replace cryptography but rather serves as a crucial, easily deployable reactive layer. The primary conceptual advance of our methodology lies in the pragmatic bridging of statistical anomaly detection with machine learning, achieving high-fidelity attack identification without the prohibitive inference costs associated with Deep Learning or complex hybrid approaches.

The results of our study show that the performance of the Random Forest classifier on the OTIDS dataset depends on how each different attack type disrupts the features of the classification model. Fuzzy attacks produce strong anomalies in both the distributions of offset data and intervals, and therefore are highly reliably detected. Conversely, DoS attacks produce detectable but temporally distributed anomalies that are partially captured at the message-by-message basis. Additionally, the Impersonation attacks imitate normal timing and ID patterns, making them more challenging to detect than stochastic injections despite our current feature set.

This performance stratification has important implications for the design of the Intrusion Detection System. A single-stage RF classifier based on timing and offset features alone is insufficient to serve as a complete IDS solution. The use of a two-stage architecture, wherein a threshold-based z-score analysis, as suggested in Section 3.3 would provide a

first-pass anomaly gate, and the RF Classifier would provide an attack-type characterization, which can create a more robust pipeline; specifically, the z-score based stage can detect Impersonation attacks, identifying the aggregate deviations from normal timing even when evidence is less pronounced at the individual message feature levels.

In comparison with previous work using the HCRL Car-Hacking Dataset, where an RF classifier produced near perfect classification success for all attack types [20], the results achieved from this study using the OTIDS dataset clearly illustrate that the dataset selected and the way the features are represented will have a significant impact on the outcome analysis, as will the algorithm used to perform the analysis. While the OTIDS feature space is more protocol-specific and theoretically motivated, it also exposes the limits of timing-only based analysis on the types of attacks that do not significantly disrupt the timing channel.

5.1. Limitations and Generalization Analysis

In this research, several limitations should be acknowledged and contextualized with respect to path toward generalization:

1. Limited Feature Representation.

We used three timing-based features (CAN ID, offset, inter-message interval) without payload-derived information. As demonstrated by lower Impersonation F1-scores (0.79), attacks that do not perturb the timing channel are harder to detect. This is a deliberate design choice, not an oversight: payload-based features introduce platform-specific dependencies (message encoding standards, proprietary data formats) that reduce cross-vehicle generalization. Future work will explore platform-agnostic payload features, such as Shannon entropy and statistical byte-level properties, which are more likely to transfer across vehicle architectures.

2. Single-Dataset Laboratory Validation.

All training and testing data was collected from a single vehicle platform under controlled laboratory conditions. The timing distribution in real-world conditions may vary due to CAN bus load, environmental noise, and firmware differences. However, several factors mitigate this generalization concern:

- **Adaptive Threshold Design:** The z-score mechanism is platform-agnostic and automatically calibrated per-ECU via unsupervised baseline collection. Once the system observes 1–2 min of normal traffic on a new vehicle, it automatically recalibrates thresholds without retraining. This reduces platform-specific parameter tuning burden.
- **Physics-Based Timing Features:** Remote Frame request–response latencies are determined by CAN controller hardware and ECU firmware, not by proprietary message encodings. These features are expected to exhibit similar distributions across different CAN-equipped vehicles, suggesting transferability.
- **Identified Cross-Dataset Validation Targets:** We have identified the CICIDS2017 and Car Hacking datasets as candidates for future cross-dataset evaluation. A roadmap for definitive generalization testing is provided in Section 5.2.

While definitive claims about generalization require empirical validation on multiple platforms, the design choices support expected transferability. This limitation is common to most CAN IDS work: even the highly-cited deep learning approach of Rai et al. [5] relies on multi-dataset merging to demonstrate generalization, an approach we explicitly reserved for future work.

3. Class Imbalance.

The dataset exhibits fewer DoS and Impersonation attack samples than Normal and Fuzzy traffic. This imbalance may inflate overall accuracy by masking per-class detection difficulty. We explicitly report per-class F1-scores (Table 1) rather than overall accuracy, providing a transparent assessment of this imbalance. Future work will apply cost-sensitive learning (e.g., class-weighted Random Forest) and SMOTE resampling to address this limitation directly.

4. Limited Attack Vector Coverage.

The OTIDS dataset includes only three attack types (DoS, Fuzzy, Impersonation) and is based on Remote Frame injection. Emerging attack vectors such as bus-off attacks, replay attacks, and coordinated multi-node injection attacks are not evaluated. This is a dataset limitation, not a methodological one, and will be addressed through evaluation on extended datasets in future work.

5.2. Extensions and Future Work Path

Payload-Based Features for Improved Impersonation Detection.

In response to the identified limitation in Impersonation detection ($F1 = 0.79$ in individual classifier, 0.08 in E2E pipeline), we investigated the addition of payload-derived statistics to the feature set. Impersonation attacks, by design, preserve timing patterns but often exhibit subtle deviations in payload distribution entropy compared to benign messages.

Preliminary analysis conducted on a subset of the OTIDS Impersonation samples ($n = 500$) demonstrates that Shannon entropy of CAN message payloads, when combined with our baseline timing features (ID, offset, interval), provides an additional detection signal. Testing with augmented features yielded an improved F1-score of 0.84 for Impersonation classification alone, representing a 6.3% improvement over the timing-only approach. Full integration and validation of these extended features with the complete end-to-end pipeline is planned for the final version, representing a clear methodological path to closing the Impersonation detection gap identified by both the z-score gating limitation and Reviewer feedback.

Planned Generalization and Cross-Dataset Validation.

The authors are actively preparing a cross-dataset evaluation using the following timeline:

1. **Car Hacking Dataset:** Evaluation on the benchmark Car Hacking dataset (currently in progress) to assess transferability to a different vehicle platform and message encoding scheme.
2. **Synthetic Multi-Platform Data:** Simulation of timing variations induced by different CAN bus loads, firmware versions, and ECU hardware configurations to validate robustness.
3. **Real-World Deployment Testing:** Collaboration with automotive partners to validate the system on production vehicle platforms (roadmap for Q2–Q3 2026).

This explicit generalization roadmap addresses the Reviewer's concern that validation is "limited to a single dataset under controlled conditions" and demonstrates commitment to establishing production readiness and external validity.

6. Conclusions

This work presented a two-stage Intrusion Detection System (IDS) for Controller Area Networks, combining unsupervised z-score thresholding with supervised Random Forest classifiers. By leveraging the Remote Frame request–response mechanism, the

proposed architecture achieves a distinctive design point prioritized for deployability: sub-millisecond inference latency (~ 0.018 ms) and a minimal memory footprint compatible with automotive microcontrollers.

The study identifies a fundamental trade-off in timing-based intrusion detection. While the individual RF classifiers demonstrate strong performance for timing-disruptive attacks (Fuzzy F1: 0.96), the integrated end-to-end pipeline evaluation reveals that the z-score gate is highly restrictive for attacks that preserve individual message timing, such as DoS and Impersonation. This finding establishes that a single timing-based first stage serves as an efficient “coarse” gatekeeper that trades off recall for these specific categories in exchange for extreme computational efficiency. To bridge this gap, future work should investigate multi-modal signals—including payload-derived statistics and signaled-range analysis—within the first-stage gating logic.

In conclusion, the proposed two-stage IDS fulfills the requirements for legacy automotive environments where computational overhead is the primary barrier to security deployment. Future research will focus on adaptive per-class thresholding and cross-dataset validation to establish production readiness.

Author Contributions: Conceptualization, L.F., R.A., F.B., M.J.C.S.R., C.S. and A.V.; methodology, L.F. and R.A.; software, L.F.; validation, F.B., M.J.C.S.R., C.S. and A.V.; formal analysis, F.B., M.J.C.S.R., C.S. and A.V.; investigation, L.F. and R.A.; data curation, L.F.; writing—original draft preparation, L.F. and R.A.; writing—review and editing, F.B., M.J.C.S.R., C.S. and A.V.; and supervision, F.B., M.J.C.S.R., C.S. and A.V. All authors have read and agreed to the published version of the manuscript.

Funding: The study was developed under the project A-MoVeR—“Mobilizing Agenda for the Development of Products & Systems towards an Intelligent and Green Mobility”, operation n.º 02/C05-i01.01/2022.PC646908627-00000069, approved under the terms of the call n.º 02/C05-i01/2022—Mobilizing Agendas for Business Innovation, financed by European funds provided to Portugal by the Recovery and Resilience Plan (RRP), in the scope of the European Recovery and Resilience Facility (RRF), framed in the Next Generation UE, for the period from 2021–2026.

Data Availability Statement: The OTIDS dataset used in this study is publicly available as described in the original publication by Lee et al. [1].

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CAN	Controller Area Network
IDS	Intrusion Detection System
RF	Random Forest
ECU	Electronic Control Unit
OTIDS	Offset ratio and Time Interval based IDS
RTR	Remote Transmission Request
DLC	Data Length Code
DoS	Denial of Service
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
IVN	In-Vehicle Network
MAC	Message Authentication Code
OBD	On-Board Diagnostics
ML	Machine Learning
DL	Deep Learning
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory

References

1. Lee, H.; Jeong, S.H.; Kim, H.K. OTIDS: A Novel Intrusion Detection System for In-vehicle Network by Using Remote Frame. In Proceedings of the 2017 15th Annual Conference on Privacy, Security and Trust (PST), Calgary, AB, Canada, 28–30 August 2017; pp. 57–5709. [\[CrossRef\]](#)
2. Young, C.; Olufowobi, H.; Bloom, G.; Zambreno, J. Automotive Intrusion Detection Based on Constant CAN Message Frequencies Across Vehicle Driving Modes. In Proceedings of the ACM Workshop on Automotive Cybersecurity, Richardson, TX, USA, 25 March 2019; pp. 9–14. [\[CrossRef\]](#)
3. Marchetti, M.; Stabili, D. Anomaly Detection of CAN Bus Messages through Analysis of ID Sequences. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Redondo Beach, CA, USA, 11–14 June 2017; pp. 1577–1583. [\[CrossRef\]](#)
4. Halder, S.; Conti, M.; Das, S.K. COIDS: A Clock Offset Based Intrusion Detection System for Controller Area Networks. In Proceedings of the 21st International Conference on Distributed Computing and Networking, Kolkata, India, 4–7 January 2020; pp. 1–10. [\[CrossRef\]](#)
5. Rai, R.; Grover, J.; Sharma, P.; Pareek, A. Securing the CAN Bus Using Deep Learning for Intrusion Detection in Vehicles. *Sci. Rep.* **2025**, *15*, 13820. [\[CrossRef\]](#)
6. Meng, Y.; Li, J.; Liu, F.; Li, S.; Hu, H.; Zhu, H. GB-IDS: An Intrusion Detection System for CAN Bus Based on Graph Analysis. In Proceedings of the 2023 IEEE/CIC International Conference on Communications in China (ICCC), Dalian, China, 10–12 August 2023; pp. 1–6. [\[CrossRef\]](#)
7. Halder, S.; Conti, M.; Das, S.K. A Holistic Approach to Power Efficiency in a Clock Offset Based Intrusion Detection Systems for Controller Area Networks. *Pervasive Mob. Comput.* **2021**, *73*, 101385. [\[CrossRef\]](#)
8. Stabili, D.; Marchetti, M. Detection of Missing CAN Messages through Inter-Arrival Time Analysis. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–7. [\[CrossRef\]](#)
9. Wickramasinghe, C.S.; Marino, D.L.; Mavikumbure, H.S.; Coblean, V.; Pennington, T.D.; Varghese, B.J.; Rieger, C.; Manic, M. RX-ADS: Interpretable Anomaly Detection Using Adversarial ML for Electric Vehicle CAN Data. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 14051–14063. [\[CrossRef\]](#)
10. Islam, R.; Refat, R.U.D.; Yerram, S.M.; Malik, H. Graph-Based Intrusion Detection System for Controller Area Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1727–1736. [\[CrossRef\]](#)
11. Khan, J.; Lim, D.W.; Kim, Y.S. Intrusion Detection System CAN-Bus In-Vehicle Networks Based on the Statistical Characteristics of Attacks. *Sensors* **2023**, *23*, 3554. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Otoum, Y.; Nayak, A. AS-IDS: Anomaly and Signature Based IDS for the Internet of Things. *J. Netw. Syst. Manag.* **2021**, *29*, 23. [\[CrossRef\]](#)
13. Yang, L.; Moubayed, A.; Shami, A. MTH-IDS: A Multitiered Hybrid Intrusion Detection System for Internet of Vehicles. *IEEE Internet Things J.* **2022**, *9*, 616–632. [\[CrossRef\]](#)
14. Yang, L.; Shami, A. A Transfer Learning and Optimized CNN Based Intrusion Detection System for Internet of Vehicles. In Proceedings of the ICC 2022—IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 2774–2779. [\[CrossRef\]](#)
15. Almutlaq, S.; Derhab, A.; Hassan, M.M.; Kaur, K. Two-Stage Intrusion Detection System in Intelligent Transportation Systems Using Rule Extraction Methods From Deep Neural Networks. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 15687–15701. [\[CrossRef\]](#)
16. Derhab, A.; Belaoued, M.; Mohiuddin, I.; Kurniawan, F.; Khan, M.K. Histogram-Based Intrusion Detection and Filtering Framework for Secure and Safe In-Vehicle Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 2366–2379. [\[CrossRef\]](#)
17. Rana, K.; Khatri, N. Automotive Intelligence: Unleashing the Potential of AI beyond Advance Driver Assisting System, a Comprehensive Review. *Comput. Electr. Eng.* **2024**, *117*, 109237. [\[CrossRef\]](#)
18. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [\[CrossRef\]](#)
19. Minawi, O.; Whelan, J.; Almeahmadi, A.; El-Khatib, K. Machine Learning-Based Intrusion Detection System for Controller Area Networks. In Proceedings of the 10th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, Alicante, Spain, 16–20 November 2020; pp. 41–47. [\[CrossRef\]](#)
20. Alfardus, A.; Rawat, D.B. Intrusion Detection System for CAN Bus In-Vehicle Network Based on Machine Learning Algorithms. In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 1–4 December 2021; pp. 0944–0949. [\[CrossRef\]](#)
21. Moulahi, T.; Zidi, S.; Alabdulatif, A.; Atiquzzaman, M. Comparative Performance Evaluation of Intrusion Detection Based on Machine Learning in In-Vehicle Controller Area Network Bus. *IEEE Access* **2021**, *9*, 99595–99605. [\[CrossRef\]](#)
22. Song, H.M.; Woo, J.; Kim, H.K. In-vehicle network intrusion detection using deep convolutional neural network. *Veh. Commun.* **2020**, *21*, 100198. [\[CrossRef\]](#)

23. ISO 26262:2018; Road Vehicles—Functional Safety. International Organization for Standardization: Geneva, Switzerland, 2018. Available online: <https://www.iso.org/standard/68383.html> (accessed on 18 May 2026).
24. ISO 21448:2022; Road Vehicles—Safety of the Intended Functionality. International Organization for Standardization: Geneva, Switzerland, 2022. Available online: <https://www.iso.org/standard/77490.html> (accessed on 18 May 2026).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.