Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.077582

ARTICLE

# Lightweight and Explainable Anomaly Detection in CAN Bus Traffic via Non-Negative Matrix Factorization

**Anandkumar Balasubramaniam and Seung Yeob Nam***

Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, Republic of Korea

*Corresponding Author: Seung Yeob Nam. Email: synam@ynu.ac.kr

Received: 12 December 2025; Accepted: 02 February 2026; Published: 09 April 2026

**ABSTRACT:** The increasing connectivity of modern vehicles exposes the in-vehicle controller area network (CAN) bus to various cyberattacks, including denial-of-service, fuzzy injection, and spoofing attacks. Existing machine learning and deep learning intrusion detection systems (IDS) often rely on labeled data, struggle with class imbalance, lack interpretability, and fail to generalize well across different datasets. This paper proposes a lightweight and interpretable IDS framework based on non-negative matrix factorization (NMF) to address these limitations. Our contributions include: (i) evaluating NMF as both a standalone unsupervised detector and an interpretable feature extractor (NMF-W) for classical, unsupervised, and deep sequence models; (ii) providing comprehensive benchmarking on the car-hacking dataset (CHD), demonstrating improved robustness in mixed-attack and cross-attack scenarios, with class imbalance addressed through oversampling and class weighting; (iii) offering a component-level interpretability analysis that links NMF factors to meaningful CAN traffic patterns; and (iv) validating cross-dataset transferability on the offset-ratio and time interval-based intrusion detection system (OTIDS) dataset. Additional ablation and efficiency studies confirm the practical feasibility of deploying NMF-based IDS on embedded automotive controllers. Overall, this work presents a balanced IDS solution that combines detection accuracy, computational efficiency, and explainability, thereby advancing the security of in-vehicle networks.

**KEYWORDS:** Anomaly detection; CAN bus; explainable AI; intrusion detection system (IDS); non-negative matrix factorization (NMF)

## 1 Introduction

The rapid integration of connectivity features in modern vehicles, including the vehicle-to-everything (V2X) communication, infotainment systems, and over-the-air (OTA) updates, has significantly expanded the attack surface of in-vehicle networks [1]. The controller area network (CAN) bus, which is the primary communication protocol for electronic control units (ECUs), was originally designed for efficiency and reliability rather than security [2,3]. In particular, CAN lacks fundamental mechanisms such as authentication, encryption, and intrusion tolerance, rendering it highly vulnerable to adversarial manipulation [4]. Attacks such as denial-of-service (DoS), fuzzy injection, Gear spoofing, and RPM spoofing have demonstrated the ability to disrupt critical vehicle functions such as braking, steering, and engine control [5,6].

CAN bus communication relies on the broadcast of short, fixed-length messages called frames. As illustrated in Fig. 1, each CAN frame contains an identifier (CAN ID) that encodes priority, a data length code (DLC) specifying the payload size, and up to eight bytes of data. While this lightweight design ensures real-time performance, it offers no message origin authentication or payload encryption and requires that every

node on the bus receive all frames. These properties make CAN efficient for embedded automotive control but simultaneously expose it to spoofing, injection, and DoS attacks when adversaries gain bus access [7].

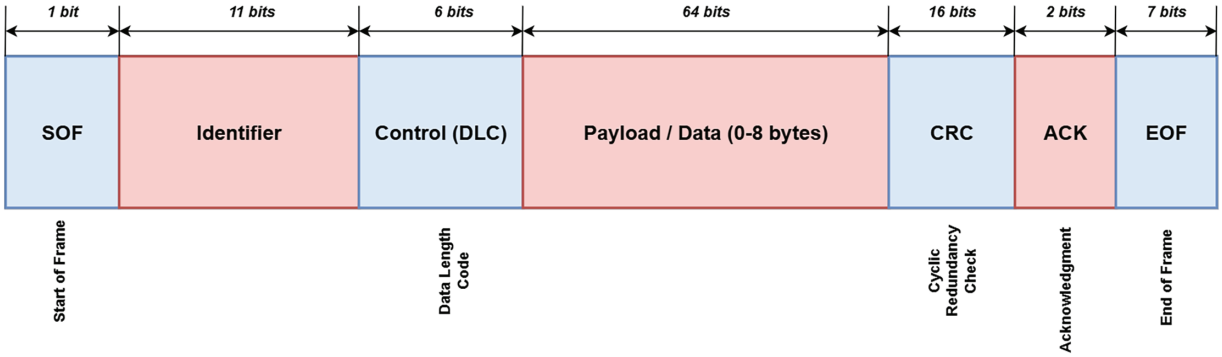



**Figure 1:** Structure of a standard CAN data frame.

Despite extensive research into intrusion detection systems (IDS) for CAN traffic, several challenges remain unresolved [8–11]. The CAN protocol lacks native authentication or encryption, and the lightweight nature of in-vehicle ECUs limits the deployment of computationally intensive solutions. In real-world settings, attack messages form only a tiny fraction of total traffic, creating a severe class imbalance that degrades the performance of conventional machine learning classifiers [12–14]. Many IDS approaches rely on complex black-box machine learning (ML) and deep learning (DL) models that, while accurate, provide little insight into why anomalies are detected [15,16]. Furthermore, raw feature-based models often exploit artificial signatures in injected datasets, limiting generalization to unseen attack types [12,16]. Finally, IDS solutions must operate within the limited computational and memory resources of ECUs, necessitating lightweight yet practical approaches.

In practical deployments, these limitations directly affect operational reliability and safety. First, because malicious injections are rare compared to normal traffic, even a low false-positive rate can produce frequent alarms, leading to alert fatigue or unnecessary mitigation actions that may disrupt vehicle functionality. Second, models that overfit dataset-specific artifacts (e.g., injected message signatures or logging biases) may show near-perfect accuracy offline but fail under real driving conditions, where traffic distributions vary across vehicle models, software versions, and driving contexts. Third, limited interpretability makes it difficult for security engineers to validate detections, diagnose root causes, and perform post-incident analysis or OTA regression checks. Finally, computationally intensive IDS pipelines may be impractical on resource-constrained ECUs and gateways that must meet strict real-time deadlines. These real-world considerations motivate the need for lightweight, generalizable, and explainable IDS designs for CAN bus monitoring.

This paper addresses these challenges by leveraging non-negative matrix factorization (NMF) for anomaly detection in CAN bus traffic [17,18]. NMF is explored both as a standalone detector and as a feature extractor for supervised, unsupervised, and sequence-based models. We perform a comparative evaluation of NMF-derived feature representations ($\mathbf{W}$) against raw features in classical classifiers such as random forest (RF), logistic regression (LR), support vector machines (SVM), and extreme gradient boosting (XGB), as well as the unsupervised isolation forest (IF), and DL models such as convolutional neural networks (CNN), long short-term memory (LSTM) networks, and attention-based architectures. We also conduct attack-specific and cross-attack generalization experiments, highlighting NMF's ability to uncover latent overlaps and transferable anomaly patterns. To further strengthen robustness, we investigate

class-imbalance handling strategies, including the synthetic minority over-sampling technique (SMOTE), adaptive synthetic sampling (ADASYN), and class-weighted training, and analyze their impact on Raw vs. NMF features. Mixed-attack training and cross-dataset validation are used to assess transferability, while interpretability, efficiency, and ablation studies demonstrate that NMF provides a lightweight, explainable pathway to deployable IDS solutions.

Unlike earlier NMF-based intrusion detection studies, which mainly used NMF for feature selection or dimensionality reduction in IT or IoT network data, this study provides a systematic investigation of NMF for automotive CAN intrusion detection. We extend prior efforts by leveraging NMF both as a standalone unsupervised anomaly detector and as a unified, interpretable feature representation for classical, unsupervised, and sequence-based models. Our approach advances the field through comprehensive benchmarking—including attack-wise, cross-attack, and cross-dataset evaluations—detailed component-level interpretability analysis linking latent factors to meaningful CAN traffic structures, and discussion of ECU deployment feasibility. These contributions position NMF as a transparent and lightweight alternative to conventional DL–centric IDS approaches for in-vehicle networks.

The remainder of this paper is organized as follows. Section 2 reviews recent studies on CAN intrusion detection, including transformer-based IDS, interpretable DL models, federated learning frameworks, and NMF-based approaches, as well as sequence modeling and imbalance handling. Section 3 presents the proposed NMF-based framework and methodology, followed by Section 4, which describes the experimental setup and datasets. Section 5 details the experimental analysis, covering standalone NMF detection, baseline and rule-based comparisons, attack-wise evaluation, attack localization, cross-attack and cross-dataset generalization, imbalance handling, interpretability, ablation studies, and efficiency evaluation, together with limitations and threats to validity. Finally, Section 6 concludes the paper and outlines directions for future research.

## 2 Related Work

Intrusion detection in in-vehicle networks has attracted significant research attention due to the lack of built-in authentication and encryption in the CAN protocol. Several surveys provide a comprehensive overview of ML- and DL-based IDS approaches, covering methods from statistical timing analysis to deep neural architectures, and emphasize that intrusion detection remains the primary line of defense against attack injection [19,20]. These works also highlight widely adopted datasets, such as the car-hacking dataset (CHD) and the offset-ratio and time-interval-based intrusion detection system dataset (OTIDS), and emphasize that although DL methods achieve high accuracy, issues of generalization, imbalance, and interpretability remain unresolved [10,21]. Recent advances in interpretable DL for intrusion detection have incorporated explainable artificial intelligence (XAI) techniques such as shapley additive explanations (SHAP), local interpretable model-agnostic explanations (LIME), and RuleFit to provide feature-level explanations for deep neural network (DNN) predictions, enabling greater transparency without sacrificing accuracy [22–24]. However, these methods show that ensuring consistent explanations and deriving operationally actionable insights—especially for ECU-level diagnostics on automotive CAN traffic—remains an open challenge.

Deep models have been a dominant research direction for CAN intrusion detection. Recurrent autoencoder frameworks, such as INDRA, learn the temporal dynamics of normal CAN signals and detect deviations via reconstruction errors, achieving high detection rates but often at a high computational cost [25,26]. LSTM- and gated recurrent unit (GRU)-based autoencoders have been applied and shown to achieve near-perfect performance on benchmark datasets [27], while convolutional and attention-based backbones have also been explored for sequential anomaly detection in CAN traffic [28]. More recently, transformer-based architectures have been deployed for CAN intrusion detection, leveraging self-attention

to model long-range dependencies in CAN streams and improve multiclass anomaly classification, though they often incur notable resource demands [29–31]. Recent work has also explored ensemble-based and hybrid intrusion detection models for vehicular and Internet of Vehicular Things (IoVT) environments. For example, ensemble frameworks that combine multiple machine learning classifiers have been proposed to improve robustness against diverse attack patterns and class imbalance, reporting strong detection performance on widely used network intrusion datasets in IoV contexts [32]. While these approaches benefit from model diversity and high accuracy, they typically rely on complex decision fusion and operate as black-box systems, offering limited interpretability at the traffic-pattern level. In contrast, our work emphasizes a single, lightweight factorization-based representation tailored to CAN bus traffic, explicitly exposing latent communication patterns that support both effective detection and component-level interpretability. Beyond accuracy, some studies stress the importance of distributed IDS deployment across ECUs to prevent single points of failure and minimize communication overhead, further motivating lightweight designs. In parallel, federated learning for IDS has attracted attention for its ability to enable privacy-preserving, distributed anomaly detection across vehicles or ECUs. Recent studies demonstrate that federated models deliver robust collaborative intrusion detection while maintaining local data privacy, even across heterogeneous automotive or IoT settings [33–35]. These findings align with our experimental setup, which contrasts raw traffic features with compact latent representations and evaluates both classical and sequence-aware classifiers. In parallel, transfer learning has been explored to improve cross-dataset adaptability of CAN IDS [36], achieving strong generalization across datasets but without addressing the interpretability limitations of deep architectures.

Hardware-efficient IDS has also received growing attention, with field-programmable gate array (FPGA)-based ECUs recently proposed to accelerate inference for onboard intrusion detection. By consolidating lightweight ML models (e.g., quantised multi-layer perceptron) into FPGA architectures, these systems achieve more than twice the power efficiency and significantly lower latency compared to GPU baselines [37]. Such results underscore the necessity of compact representations and efficient algorithms for real-time deployment, an aspect directly addressed in our work by adopting NMF as a low-overhead factorization method.

Classical ML and statistical approaches, including SVMs, RFs, cumulative sum (CUSUM) tests, and entropy-based detectors, have been widely applied to CAN traffic [38]. While effective against simple flooding or replay scenarios, these methods often overfit to dataset-specific artifacts or fail under subtle payload manipulation attacks. Furthermore, their reliance on raw feature distributions makes them sensitive to class imbalance, a fundamental challenge since attack frames typically constitute only a tiny fraction of real traffic [39]. Data augmentation techniques such as SMOTE and ADASYN have been introduced in IDS to alleviate imbalance and improve minority-class recall, though their effectiveness for CAN bus anomaly detection remains underexplored. Our experiments extend this line by systematically benchmarking oversampling on both raw and NMF-transformed feature spaces.

While NMF itself has seen limited application in CAN intrusion detection, it has been successfully leveraged in other intrusion detection contexts. Early works demonstrated its utility for profiling system or user behaviors [40,41], where NMF decomposed audit trails into additive patterns that distinguished normal from anomalous activity. More recent studies extended NMF for feature selection and hybrid detection, applying it to malware identification and IoT intrusion detection [42,43]. NMF has also been adapted into enhanced variants, such as optimal brain surgeon NMF (OBS-NMF) for improved clustering quality in anomaly detection [44], and neighborhood-structure-assisted NMF (NS-NMF) for unsupervised point-wise anomaly detection [45]. Beyond IDS, NMF has been applied in other domains as a tool for interpretable anomaly detection. Physics-enhanced NMF has been shown to improve anomaly identification

in mechanical systems by decomposing complex signals into additive, human-understandable components [46]. Similarly, spatio-temporal NMF has been employed in traffic pattern mining, where it captures latent structure and reduces dimensionality without losing interpretability [47]. These successes motivate our use of NMF for CAN traffic, positioning it as both a standalone detector and a feature extractor that enhances the performance of downstream ML/DL models while preserving interpretability.

While prior studies have applied NMF to general network intrusion detection, these efforts differ fundamentally from our work. Early works focused on profiling or fast detection in IT networks, whereas we target the distinctive domain of in-vehicle CAN traffic. Other efforts proposed algorithmic enhancements to NMF (e.g., OBS-NMF, NS-NMF), but we show that even standard NMF, when coupled with supervised and sequential IDS models, achieves high performance while remaining lightweight and explainable. More recent IDS applications have used NMF for feature extraction or hybrid selection, but rarely evaluated its role as both a standalone anomaly detector and as an integrative feature space across multiple backbones. Our work therefore provides a comprehensive benchmark of NMF in CAN intrusion detection, spanning attack-wise, cross-attack, cross-dataset, imbalance, and efficiency evaluations. Compared with previous NMF-based IDS work that primarily focused on feature extraction or hybrid detection in IT or IoT contexts, our study conducts a broader, more integrated evaluation of automotive CAN intrusion detection, emphasizing interpretability, cross-dataset transferability, and deployment feasibility. In summary, prior research underscores three significant gaps: (i) most CAN IDS rely on black-box DL models that achieve accuracy but offer limited transparency, (ii) class imbalance is a persistent obstacle, and (iii) efficiency is rarely considered in realistic ECU-constrained environments. Against this backdrop, our work introduces NMF as a lightweight, interpretable, and transferable representation, benchmarked across both tabular and sequential classifiers, and rigorously evaluated under attack-wise, cross-attack, imbalance-handling, and mixed-attack scenarios.

## 3 Methodology

To investigate the effectiveness of NMF for anomaly detection in CAN bus traffic, we design a comprehensive framework that integrates preprocessing, feature extraction, representation learning, and broad evaluation across multiple models (classical ML, unsupervised, and temporal sequence models) to assess NMF's utility. Fig. 2 provides an overview of the proposed framework. The process begins with CAN traffic data collected from CHD [10], which contains both normal in-vehicle communication and multiple categories of injected attacks (DoS, Fuzzy, Gear spoofing, RPM spoofing). This dataset serves as the primary benchmark for evaluating the proposed framework's detection capabilities. The second dataset (OTIDS) [48] is reserved for cross-dataset validation and discussed in Section 5.10.

In the preprocessing stage, raw CAN frames are aggregated into fixed-length windows using an overlapping scheme with a 0.1-s overlap and a 0.05-s stride, thereby capturing short-lived attacks in the feature space. From each window, statistical features are extracted to represent message dynamics, including CAN ID frequencies, payload entropy, and mean payload values. In labeled experiments, the attack-flag ratio is additionally included as a diagnostic feature. These features jointly capture both structural (e.g., ID distributions) and content-level (e.g., payload statistics) variations in the traffic. The resulting feature matrix $\mathbf{X}$ is then factorized using NMF into two low-rank nonnegative matrices, with the coefficient matrix $\mathbf{W}$ serving as an interpretable and compact representation for subsequent analysis.

Downstream detection is performed using multiple approaches to assess NMF's utility. First, the NMF reconstruction error in Eq. (6) is used directly as an unsupervised anomaly score. Second, the latent representation $\mathbf{W}$ is used as input to classical classifiers (RF, LR, SVM, and XGB) and to the unsupervised IF, allowing comparison with raw-feature baselines. Third, temporal DL models, including CNNs, LSTMs,

and Attention-based architectures, are employed to capture sequential dependencies in CAN traffic. Finally, class imbalance is addressed through oversampling methods such as SMOTE and ADASYN, as well as class-weighted training strategies. All approaches are evaluated using the metrics formally defined in Section 5.1. This evaluation protocol enables a rigorous analysis of NMF as both a standalone anomaly detector and a feature extractor that improves the performance of supervised and sequence-based models.

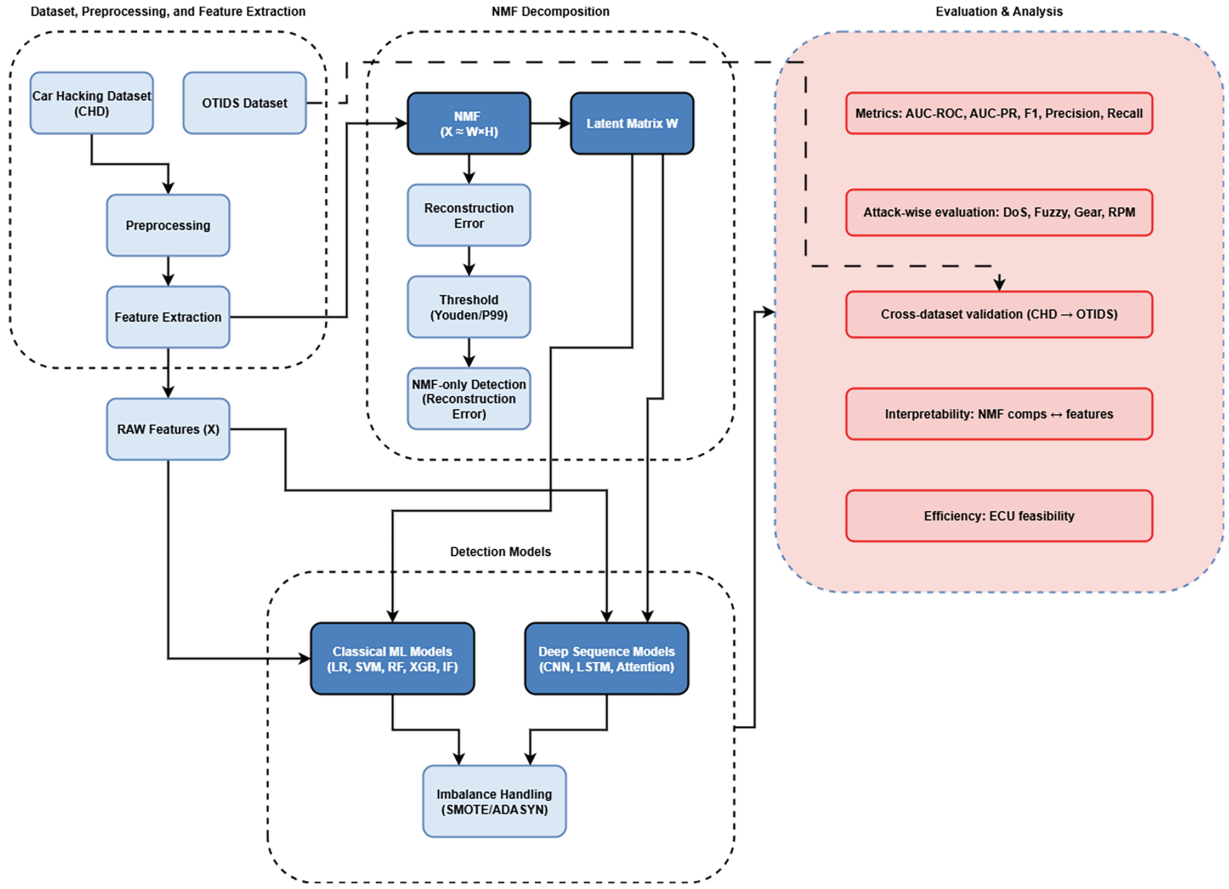



**Figure 2:** Proposed framework for CAN bus anomaly detection using NMF as both a standalone detector and a feature extractor.

### *3.1 NMF for Feature Representation*

NMF is a dimensionality reduction technique that decomposes a nonnegative matrix into additive parts. We employ NMF as both a dimensionality-reduction method and an interpretable representation-learning framework for CAN bus traffic. NMF has been extensively studied in the literature for its algorithms and diverse applications, particularly for uncovering parts-based, interpretable latent representations [18,49]. Fig. 3 provides a schematic illustration.

Given an input feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where $n$ is the number of windows and $d$ is the feature dimension, NMF approximates $\mathbf{X}$ as the product of two nonnegative matrices:

$$\mathbf{X} \approx \mathbf{WH}, \quad \mathbf{W} \in \mathbb{R}_{+}^{n \times k}, \ \mathbf{H} \in \mathbb{R}_{+}^{k \times d}, \tag{1}$$

where $k \ll d$ is the latent rank chosen based on the elbow method and stability analysis of the reconstruction error (see Section 5.2.2). Following the ML convention, we refer to $\mathbf{W}$ as the *coefficient (activation) matrix*,

since each column encodes how strongly a latent component is activated across windows, and **H** as the *basis matrix*, since its rows capture interpretable feature patterns.

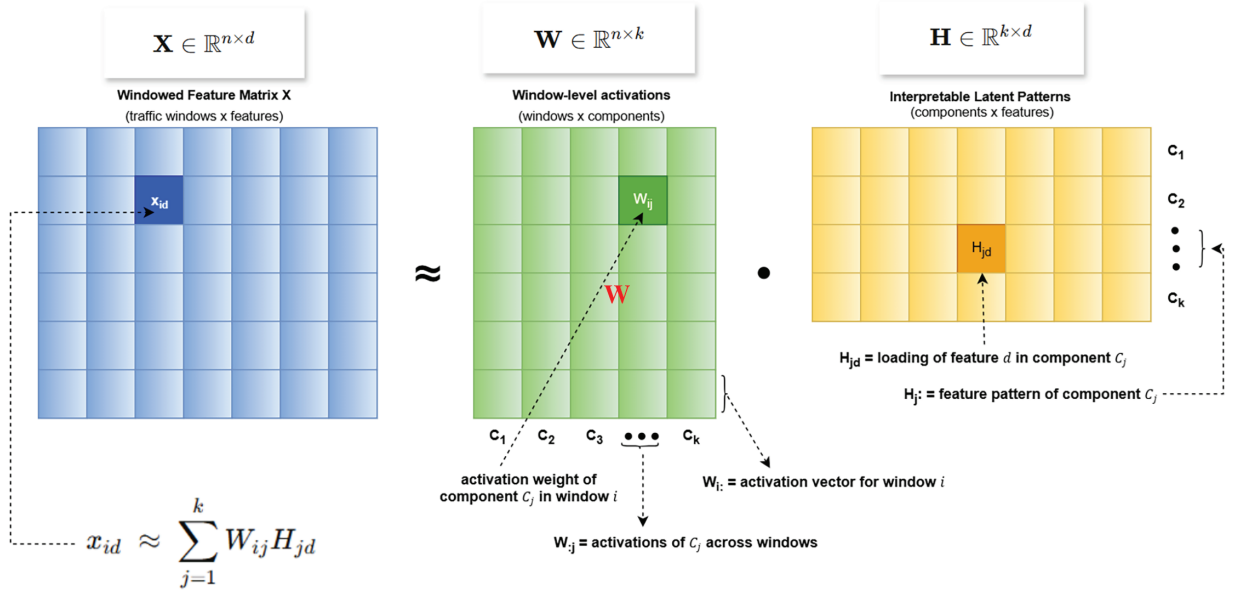



**Figure 3:** Illustration of NMF decomposition: **X** is factorized into **W** (window-level activations) and **H** (latent feature patterns).

Each latent component $C_j (j = 1, \ldots, k)$ is defined as:

$$C_j := \left(W_{:j},\, H_{j:}\right),$$

where: $W_{:j}$ (the $j$-th column of **W**) captures how strongly $C_j$ is activated across the $n$ time windows, and $H_{j:}$ (the $j$-th row of **H**) encodes the feature loading pattern that characterizes $C_j$.

Thus, a single window vector $\mathbf{x}_{i\cdot}$ is reconstructed as a nonnegative linear combination of latent components:

$$\mathbf{x}_{i\cdot} \approx \sum_{j=1}^{k} W_{ij}\, H_{j\cdot}. \tag{2}$$

In this way, **W** provides a window-level activation profile, while **H** captures interpretable basis patterns across features. The optimization objective is:

$$\min_{\mathbf{W},\mathbf{H} \geq 0} \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2, \tag{3}$$

which is solved iteratively using coordinate descent [50].

The choice of NMF is motivated by both the structure of CAN traffic and the practical requirements of in-vehicle intrusion detection. First, NMF's non-negativity constraint naturally aligns with CAN-derived features such as message counts, ID frequencies, entropy values, and payload statistics, allowing the model to operate directly on semantically meaningful inputs without additional transformations. Second, NMF provides a parts-based, additive decomposition that matches the compositional nature of CAN traffic, where each time window is formed by the superposition of messages from multiple ECUs and CAN IDs.

In this formulation, traffic windows are expressed as nonnegative combinations of latent components, each capturing a coherent traffic pattern (e.g., dominant IDs, DLC profiles, or payload entropy behavior), enabling transparent interpretation of anomalous activity. Finally, normal CAN traffic exhibits highly regular and repetitive structure, leading to a stable low-rank representation under NMF. Attacks disrupt this structure, resulting in abnormal component activations or increased reconstruction error. This makes NMF particularly suitable for unsupervised anomaly detection under severe class imbalance, while maintaining lightweight computation and interpretability essential for ECU-constrained environments.

In this study, we adopt standard NMF as a stable and interpretable baseline. While variants such as sparse NMF or temporal NMF can impose additional structure, they introduce extra regularization parameters and temporal dependencies that complicate controlled comparison across classifiers and datasets. Our objective is not to optimize NMF variants, but to systematically evaluate whether a simple, lightweight factorization already provides meaningful gains in interpretability, robustness, and efficiency for CAN intrusion detection. A quantitative comparison with other unsupervised detectors is provided in Section 5.11.

#### *3.1.1 Data Splitting and Preprocessing*

Since NMF requires nonnegative inputs, we preprocess the features using MinMax scaling on the training set, ensuring that the validation and test data are transformed consistently. Any negative values or NaNs are clipped to zero:

$$\tilde{\mathbf{X}} = \max\left(0, \text{MinMax}(\mathbf{X})\right). \tag{4}$$

To ensure a fair comparison, the same train-fitted scaler is applied consistently across all baselines: the MinMax scaler is fitted on the training split and reused to transform validation/test data without refitting. Raw-feature baselines are evaluated under the same normalization pipeline, so differences between Raw and NMF–W results reflect representation/model effects rather than inconsistent scaling.

To preserve temporal consistency, we adopt a per-attack time-based split: 70% of each attack subset is allocated to training, 10% to validation, and 20% to testing. The splits are then concatenated across all attack types, ensuring class balance and chronological order.

#### *3.1.2 Anomaly Scoring via Reconstruction Error*

For standalone anomaly detection, NMF is first trained on *normal* traffic. Given a new window $\tilde{x}_i$, its latent representation $\mathbf{w}_i$ is obtained by projecting it onto the learned basis $\mathbf{H}$ (i.e., solving for the row in $\mathbf{W}$ that best reconstructs $\tilde{x}_i$). The reconstruction is then computed as

$$\hat{x}_i = \mathbf{w}_i\mathbf{H}, \quad \mathbf{w}_i = \text{NMF.transform}(\tilde{x}_i). \tag{5}$$

The anomaly score is defined as the residual norm:

$$s(x_i) = \left\|\tilde{x}_i - \hat{x}_i\right\|_2, \tag{6}$$

where $\tilde{x}_i - \hat{x}_i$ is the reconstruction error. Intuitively, anomalous windows cannot be well represented by parts learned from normal data and thus yield higher residuals.

#### *3.1.3 Threshold Selection*

We evaluate two thresholding strategies to convert continuous scores into binary predictions:

- **Youden's J Statistic:** Based on the validation *receiver operating characteristic* (ROC) curve, we select the threshold $t^\star = \arg\max_t \{\mathrm{TPR}(t) - \mathrm{FPR}(t)\}$.
- **99th Percentile ($\mathbf{P_{99}}$):** We compute $t_{99}$ as the 99th percentile of reconstruction errors on training-normal windows, representing a one-sided anomaly quantile.

Youden's $J$ is derived from validation ROC curves, while $P_{99}$ corresponds to the 99th percentile of training-normal residuals. The use of both ROC-optimized and distribution-based thresholds ensures robustness. Table 1 summarizes the chosen thresholds for each attack scenario.

**Table 1:** Selected thresholds for anomaly scoring using NMF reconstruction errors.

| Attack Type | Threshold (Youden) | Threshold ($P_{99}$) |
|---|---|---|
| DoS | 1.0021 | 0.4516 |
| Fuzzy | 0.3158 | 0.4226 |
| Gear | 0.2196 | 0.4549 |
| RPM | 0.2380 | 0.4529 |

Across all attack scenarios, both Youden-optimized and percentile-based thresholds yield consistent trends in detection performance, with only minor quantitative differences. Youden's $J$ typically provides slightly higher balanced accuracy when validation labels are available, while the percentile-based threshold offers a stable, label-free alternative suitable for deployment. In practice, percentile thresholds (e.g., $P_{99}$) are preferable in operational settings where attack labels are unavailable, whereas Youden's $J$ is useful for controlled evaluation and benchmarking.

### *3.1.4 Interpretability via NMF Components*

Beyond anomaly detection, the factorization provides interpretable insights into CAN traffic behavior. The basis matrix $\mathbf{H}$ contains $k$ components (with $k = 16$ in our experiments based on Section 5.2.2), each expressed as a sparse, nonnegative weighting over features. For component $j$, we extract the top-$m$ defining features:

$$\mathcal{F}_j = \text{Top-}m \ \arg\max_f H_{j,f}, \tag{7}$$

where $H_{j,f}$ denotes the weight of feature $f$ in component $j$. These feature sets highlight traffic patterns such as dominant CAN IDs, abnormal DLC distributions, or payload entropy bursts. Similarly, the activation matrix $\mathbf{W}$ provides temporal dynamics, where $\mathbf{W}_i$ indicates the contribution of each component to window $i$. By analyzing $\mathbf{W}$ across time, one can observe bursts of activation corresponding to an attack. The detailed interpretability results, including component-feature heatmaps and temporal activation plots, are presented in Section 5.

**Practical interpretation.** From an analysis perspective, these component–feature associations allow an analyst to reason about the *type* of abnormal behavior driving a detected anomaly. For example, components dominated by a small set of CAN IDs with high activation may indicate flooding-like behavior, whereas components weighted toward payload entropy or DLC statistics may reflect spoofing or manipulation patterns. We emphasize that this interpretability is demonstrated through offline analysis in this study; validating its effectiveness in real-world analyst workflows is left for future work.

**Dimensionality reduction vs. semantic structure.** Although NMF–W reduces the original feature dimension, the observed robustness gains cannot be attributed to dimensionality reduction alone. If

reduced dimensionality were the sole factor, similar improvements would be expected from generic compression techniques. Instead, our results show that NMF–W consistently improves cross-attack and cross-dataset generalization while preserving interpretable structure, whereas raw features—even at higher dimensionality—often exhibit brittle transfer. This suggests that the semantic, parts-based organization of NMF components, which align with meaningful CAN traffic patterns (e.g., ID dominance, entropy bursts, timing structure), plays a central role. Dimensionality reduction contributes to noise suppression, but the structured latent representation is the primary driver of robustness and interpretability in hybrid models.

With the NMF formulation and its interpretability clarified, we now present the benchmark datasets that serve as the basis for evaluating our framework.

### *3.2 Dataset Description*

To evaluate the proposed framework, two publicly available benchmark datasets were utilized. Both datasets provide real CAN traffic traces recorded from production vehicles under normal and attack conditions, enabling reproducible experimentation in vehicular intrusion detection.

#### *3.2.1 CHD*

In-vehicle networks for modern vehicles are based on the CAN protocol, which lacks robust security measures, making it vulnerable to intrusion. To support cybersecurity research, the hacking and countermeasure research lab (HCRL) released CHD, comprising real CAN traffic captured via an OBD-II port during message-injection attacks on an actual vehicle. It includes four different attack types: DoS, fuzzing, Gear spoofing, and RPM gauge spoofing, as well as normal (attack-free) traffic. These four attack patterns are widely studied in CAN IDS literature [5,7] as they capture three essential categories of adversarial behavior: flooding (DoS), random fuzzing, and spoofing of critical control signals (Gear spoofing and RPM spoofing).

CHD dataset includes **300 intrusion attempts**, where each attempt corresponds to a **3–5 s burst of continuous message injection** embedded within an otherwise normal driving sequence. Within these bursts, attack frames are transmitted at high frequency (e.g., Fuzzy injects a frame every 0.5 ms and DoS injects every 0.3 ms), resulting in tens of thousands of attack frames per intrusion interval. Thus, "300 intrusion attempts" refers to *temporal attack bursts*, while the millisecond-scale injection rate describes the *per-frame timing* inside each burst. Each CHD subset contains approximately 3–4 million CAN messages, including 0.5–0.6 million injected attack frames depending on the attack type. Every record includes a timestamp, CAN ID, DLC, payload bytes, and a flag indicating whether the message is normal or injected. Tables 2 and 3 summarizes the message statistics and describes the characteristic behavior of each attack category in CHD, respectively.

**Table 2:** CHD statistics.

| Attack Type | Messages | Normal | Attack |
|---|---|---|---|
| DoS Attack | 3,665,771 | 3,078,250 | 587,521 |
| Fuzzy Attack | 3,838,860 | 3,347,013 | 491,847 |
| Spoofing the drive gear | 4,443,142 | 3,845,890 | 597,252 |
| Spoofing the RPM gauze | 4,621,702 | 3,966,805 | 654,897 |
| Normal | 988,987 | 988,872 | – |

**Table 3:** Description of attack types in CHD.

| Attack Type | Description |
|---|---|
| DoS | Floods the CAN bus with high-priority 0 × 0000 messages at 0.3 ms intervals, blocking legitimate communication. |
| Fuzzy | Injects random CAN IDs and payloads at 0.5 ms intervals, increasing entropy and creating unpredictable traffic patterns. |
| Gear Spoofing | Manipulates transmission control by injecting crafted CAN frames to alter gear-shift signals. |
| RPM Spoofing | Sends forged engine RPM messages, leading to false instrument cluster readings. |

*3.2.2 OTIDS Dataset*

The OTIDS dataset was collected from a Kia Soul vehicle during real-world driving and contains both normal CAN traffic and three attack categories: *DoS*, *Fuzzy*, and *Impersonation*. In contrast to the temporally segmented design of CHD, OTIDS presents a more realistic distribution in which long spans of normal driving are intermittently interspersed with injected frames. The attack patterns occur throughout the driving sequence rather than as isolated bursts, reflecting deployment-like conditions where intrusions may be sparse and embedded within extended periods of routine operation.

OTIDS includes timestamped CAN frames with ID, DLC, and payload fields. Although DoS and Fuzzy attacks share similarities with those in CHD, OTIDS also includes Impersonation attacks, in which an adversary injects forged frames that mimic legitimate IDs with manipulated payload values. A detailed description of all OTIDS attack types is provided in Table 4. Because OTIDS captures genuine driving dynamics and naturally imbalanced traffic, it is particularly suited for testing the cross-dataset generalization of IDS models. In this work, CHD is used as the primary benchmarking dataset, while OTIDS serves to evaluate out-of-distribution performance (Section 5.10). A high-level comparison of CHD and OTIDS is provided in Table 5, highlighting their complementary characteristics for IDS evaluation.

**Table 4:** Description of attack types in OTIDS.

| Attack Type | Description |
|---|---|
| DoS | Injects continuous high-priority 0 × 0000 CAN messages in a short cycle, overwhelming the bus and preventing legitimate communication. |
| Fuzzy | Injects messages with spoofed random CAN IDs and random payload values, increasing entropy and creating unpredictable traffic patterns. |
| Impersonation | Injects forged messages that mimic a legitimate node's arbitration ID (e.g., ID = 0 × 164) with manipulated payload values, causing the receiver to accept attacker-generated data as authentic. |

**Table 5:** Comparison of CHD and OTIDS.

| Aspect | CHD | OTIDS |
|---|---|---|
| Collection Environment | Production vehicle, OBD-II port | Kia Soul vehicle, real driving environment |

(Continued)

**Table 5 (continued)**

| Aspect | CHD | OTIDS |
|---|---|---|
| Attack Types | DoS, Fuzzy, Gear Spoofing, RPM Spoofing | DoS, Fuzzy, Impersonation |
| Traffic Distribution | Balanced design: each attack subset has millions of normal and ~0.5M injected frames | Realistic skew: long spans of normal traffic with relatively sparse attacks |
| Message Format | Timestamp, CAN ID, DLC, DATA [0–7], Flag (R/T) | Timestamp, CAN ID, DLC, DATA [0–7] |
| Typical Use in This Work | Primary benchmark for model training and evaluation | Cross-dataset validation for generalization |
| Approx. Duration | 30–40 min per attack type | Variable, recorded during actual driving |

### *3.3 Feature Extraction*

To transform raw CAN bus traffic into a structured representation suitable for anomaly detection, a time-sliding-window feature-extraction pipeline was developed. This process converts sequences of individual CAN frames into fixed-length feature vectors that capture statistical, structural, and entropy-based properties of traffic windows. Raw CAN traffic streams were segmented into overlapping windows of 0.1 s with a stride of 0.05 s. Each window was labeled as an attack if at least one injected (flag "T") frame was present; otherwise, it was labeled normal. This ensures that even short-lived injections are captured during training and evaluation. Since raw CAN frames consist only of timestamps, IDs, and payload bytes, applying NMF directly is infeasible. Instead, statistical and distributional features are first extracted to provide a meaningful numeric representation that NMF can decompose into interpretable latent factors.

Feature extraction was performed separately for each dataset subset (Normal, DoS, Fuzzy, Gear Spoofing, and RPM Spoofing), and attack-wise feature files were saved for per-attack evaluation. A combined dataset (Normal + all attacks) was also constructed for holistic benchmarking.

The feature extraction pipeline is illustrated in Fig. 4. Raw CAN frames, consisting of a timestamp, CAN ID, DLC, and data bytes, are segmented into overlapping windows. Within each window, multiple groups of features (mentioned in Table 6) are computed. This modular pipeline ensures that each traffic window is transformed into a structured feature vector, enabling subsequent dimensionality reduction via NMF and classification by baseline or sequence-based models.

**Handcrafted Feature Complexity.** Although the feature set contains more than one hundred window-level statistics (ID frequencies, DLC histograms, payload entropy, per-ID dynamics, etc.), these quantities are derived from simple linear-time aggregations (counts, histograms, means, and variances) over raw CAN frames. Importantly, these features are *generic* and not tuned to any specific attack signature; they describe fundamental structural properties of CAN traffic that remain meaningful across datasets.

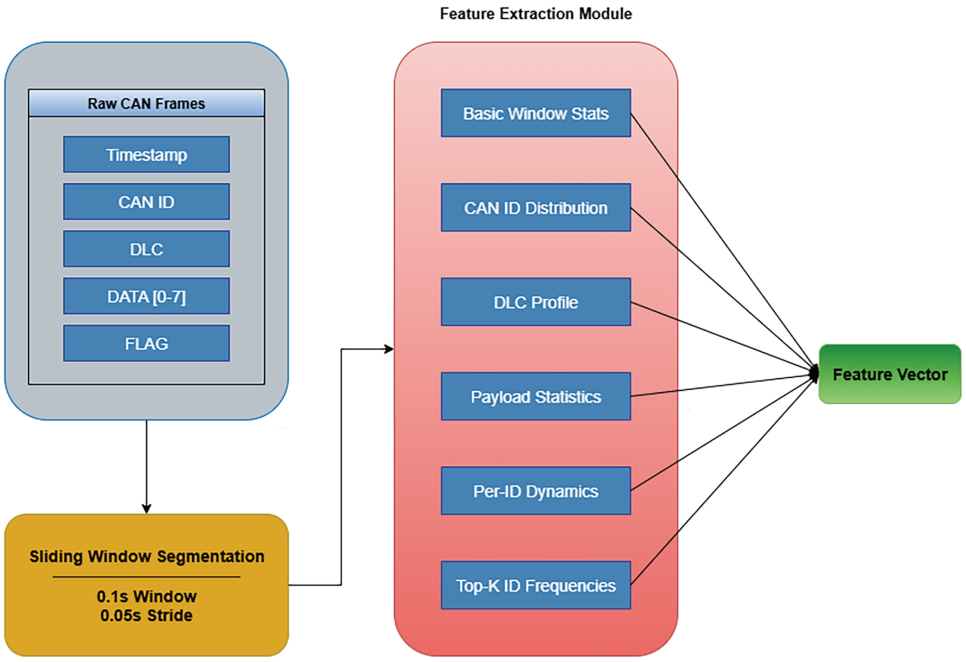



**Figure 4:** Feature extraction pipeline.

**Table 6:** Summary of extracted features for CAN bus windows.

| Feature Group | Feature Name(s) | Description |
|---|---|---|
| Basic Window Statistics | `msg_count`, `msg_rate` | Total number of CAN frames in the window and frames per second. |
| | `interarrival_mean`, `interarrival_std` | Mean and standard deviation of inter-arrival times between consecutive messages. |
| CAN ID Distribution | `unique_id_count` | Number of distinct CAN IDs present in the window. |
| | `id_entropy` | Shannon entropy of CAN ID distribution. |
| | `topID_dominance` | Proportion of the most frequent ID relative to total messages. |
| | `new_id_count`, `new_id_ratio` | Number and ratio of CAN IDs not seen in normal traffic vocabulary. |
| DLC Profile | `dlc_mean`, `dlc_std` | Mean and standard deviation of DLC values. |
| | `dlc_hist_0` – `dlc_hist_8` | Normalized histogram of DLC values from 0 to 8. |
| Payload Statistics | `payload_mean`, `payload_std` | Mean and standard deviation of payload byte values. |
| | `payload_entropy` | Shannon entropy of pooled payload bytes (256-bin histogram). |
| | `nonzero_byte_ratio` | Ratio of non-zero payload bytes to total payload size. |

(Continued)

**Table 6 (continued)**

| Feature Group | Feature Name(s) | Description |
|---|---|---|
| Per-ID Dynamics | `per_id_payload_var_mean`, `per_id_payload_var_max` | Mean and maximum variance of payload values across messages of the same CAN ID. |
| | `per_id_repeat_ratio_mean` | Average fraction of repeated payload frames per CAN ID. |
| Top-K ID Frequencies | `idfreq_X` (for top-100 IDs in normal traffic) | Frequency counts of the most common 100 CAN IDs observed in normal traffic, providing interpretable ID-specific features. |

Furthermore, the downstream models do not operate directly on the full high-dimensional feature vector (with $d \approx 150$–$250$ depending on dataset alignment and histogram resolution). Instead, NMF compresses this space into a compact latent representation of only $k = 16$ components (Section 5.2.2), substantially reducing dependence on individual handcrafted attributes while preserving the dominant traffic patterns. This is supported empirically: in the cross-dataset evaluation (Section 5.10), detectors trained on NMF-W features transfer successfully from CHD to OTIDS, whereas raw-feature models degrade significantly. These results suggest that the learned NMF components capture dataset-agnostic temporal and structural regularities, mitigating the concern that handcrafted features may be overly specialized or brittle.

### *3.4 Model Architectures*

We evaluate both classical ML baselines and sequence models, using either the raw statistical features or the NMF coefficient space **W**. This yields two families: *Raw* baselines and *NMF–W* hybrids.

#### *3.4.1 Classical Baselines*

**LR:** A linear decision function with $\ell_2$ regularization, serving as a calibrated baseline. **SVM:** Margin-based classifier with RBF kernel, considered as a baseline margin classifier, though its scalability in high-dimensional CAN traffic can be limited. **RF:** Bagged decision trees, robust to feature heterogeneity, offering out-of-bag stability. **XGB:** Gradient-boosted trees providing strong supervised performance and calibrated probabilities. **IF:** Unsupervised anomaly detector isolating anomalies via random partitioning; evaluated on both Raw and NMF–W feature spaces without requiring attack labels. For *Raw* models, features are standardized (zero-mean, unit-variance). For *NMF–W*, MinMax scaling is applied before NMF (fit on training-normal), then **W** is standardized before classification. Hyperparameters follow standard defaults with validation-based selection where applicable.

#### *3.4.2 Sequence Models*

We further consider temporal models applied to fixed-length sequences of windows. Let $\mathbf{Z} \in \mathbb{R}^{T \times d}$ denote a sequence of windowed features, where $T$ is the temporal context (number of consecutive windows) and $d$ is the dimensionality of the feature space (either raw handcrafted features or NMF activations **W**). Unless stated otherwise, sequence labels are assigned using the "any" rule, i.e., if any window within the sequence is labeled anomalous, the entire sequence is considered anomalous. **CNN (1D):** Two temporal convolutional layers (rectified linear unit (ReLU)) followed by global average pooling and a dense sigmoid output; captures local temporal structures with low latency. **LSTM:** A stacked configuration (64, then

32 units) modeling long-range temporal dependencies, followed by a dense sigmoid output. **Scaled Dot-Product Attention:** Query/key/value projections over time with an Attention layer and global average pooling, enabling adaptive temporal weighting.

### *3.4.3 Hybrid Use of NMF*

In the hybrid setting, classifiers use the low-dimensional, interpretable activations **W** rather than the full raw feature vector. This reduces dimensionality and yields models whose predictions can be traced back to human-interpretable NMF components (via **H**), enhancing explainability while also improving robustness to class imbalance and dataset shift.

## 4 Experiment Setup

We evaluate the proposed framework using the CHD, which contains normal traffic and four attack types (DoS, Fuzzy, Gear spoofing, RPM spoofing). Each subset is sorted chronologically and partitioned into 70% for training, 10% for validation, and 20% for testing, ensuring temporal consistency and preventing leakage. Normal windows from the training set are used to fit the NMF model, while validation and test sets contain both normal and attack windows. Attack-wise feature tables and a merged dataset were prepared using the feature extractor described in Section 3.3. Preprocessing involves MinMax scaling for NMF inputs, with values clipped to enforce nonnegativity; the latent rank $k$ is selected via elbow and stability analysis. For hybrid models, the NMF coefficient matrix **W** is standardized before classification. Classical baselines, along with the unsupervised IF and temporal sequence models (CNN, LSTM, and Attention), are trained on both raw and NMF–W features. The sequence models are trained using Adam ($10^{-3}$), batch size 256, and early stopping. We report *area under the receiver operating characteristic curve* (AUC–ROC), *area under the precision–recall curve* (AUC–PR), *f1-score* (F1), precision, recall, and accuracy; for NMF-only detection, thresholds are set using Youden's $J$ statistic or the 99th percentile of training-normal residuals. All experiments were run in Python on a Linux workstation with an Intel Core i7-12700 CPU, 32 GB of RAM, and 1 TB of storage.

## 5 Experiment Results and Performance Evaluation

This section provides a comprehensive evaluation of the proposed NMF-based intrusion detection framework. Section 5.1 first introduces the evaluation metrics used throughout the study. Section 5.2 reports the standalone NMF detection performance, followed by Section 5.3, which compares baseline classifiers with their NMF–W counterparts on merged CAN traffic. Rule-based lookup table baselines are presented in Section 5.4 for reference to conventional in-vehicle IDS strategies. Sections 5.5 and 5.6 analyze attack-wise behavior and localization performance, while Section 5.7 examines cross-attack generalization to assess robustness against unseen attack types. Section 5.8 explores imbalance handling and robustness improvements through oversampling and class weighting. Section 5.9 provides a detailed interpretability analysis of NMF components and model decisions. Cross-dataset validation between CHD and OTIDS is presented in Section 5.10, and Section 5.12 situates our approach in comparison with representative state-of-the-art (SOTA) CAN IDS architectures. Section 5.13 presents ablation studies to quantify the contribution of feature groups and design choices. Section 5.14 discusses computational efficiency and ECU deployment feasibility, followed by Section 5.15, which outlines key limitations and threats to validity. Together, these experiments comprehensively evaluate detection performance, generalization capability, interpretability, and deployment readiness of the proposed NMF-based IDS.

### 5.1 Evaluation Metrics

Detection performance is evaluated using standard metrics for binary classification and anomaly detection. We report the area under the receiver operating characteristic curve (AUC-ROC), the area under the precision–recall curve (AUC-PR), and the F1-score. AUC-ROC measures overall class separability, while AUC-PR is particularly informative under severe class imbalance, where attack samples are much rarer than normal traffic. The F1-score summarizes the trade-off between Precision and Recall.

Let $TP$, $FP$, $TN$, and $FN$ denote the numbers of true positives, false positives, true negatives, and false negatives, respectively. The metrics are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \tag{8}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad \text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{9}$$

The ROC curve plots Recall (TPR) against the false-positive rate (FPR)

$$\text{FPR} = \frac{FP}{FP + TN}, \tag{10}$$

and its area is reported as AUC-ROC. Similarly, the precision–recall (PR) curve plots Precision as a function of Recall, and its area is reported as AUC-PR.

Because CAN intrusion detection is highly imbalanced, our analysis focuses primarily on AUC-ROC, AUC-PR, and F1. Accuracy and Precision/Recall are reported for completeness in Sections 5.2 and 5.3, while all comparative evaluations consistently rely on AUC-ROC, AUC-PR, and F1.

### 5.2 Standalone NMF Detection Results

We first evaluate NMF as a standalone unsupervised anomaly detector following the methodology described in Section 3. The model is trained exclusively on normal windows and evaluated on validation and test sets using reconstruction residuals.

#### 5.2.1 Data Splits

Each attack subset (Normal, DoS, Fuzzy, Gear spoofing, RPM spoofing) is chronologically split into 70% training, 10% validation, and 20% testing. Table 7 summarizes the resulting distribution of normal and attack windows.

**Table 7:** Distribution of samples across train, validation, and test splits.

| Split | Normal | Attack |
|---|---|---|
| Train | 67,442 | 88,333 |
| Val | 9722 | 12,533 |
| Test | 42,885 | 1625 |

The time-based split preserves temporal ordering to prevent leakage between adjacent windows and reflects realistic deployment conditions, where IDS training relies on historical attack traces while future traffic is predominantly normal. The resulting class imbalance is an inherent property of the CHD dataset rather than an artifact of the proposed pipeline. Robustness under imbalance is examined explicitly in later sections, where relative performance trends remain consistent.

### *5.2.2 Latent Rank Selection*

To select the number of NMF components $k$, we consider reconstruction error, stability across random initializations, and interpretability. Fig. 5 summarizes the elbow and stability curves, and Table 8 reports the corresponding values.

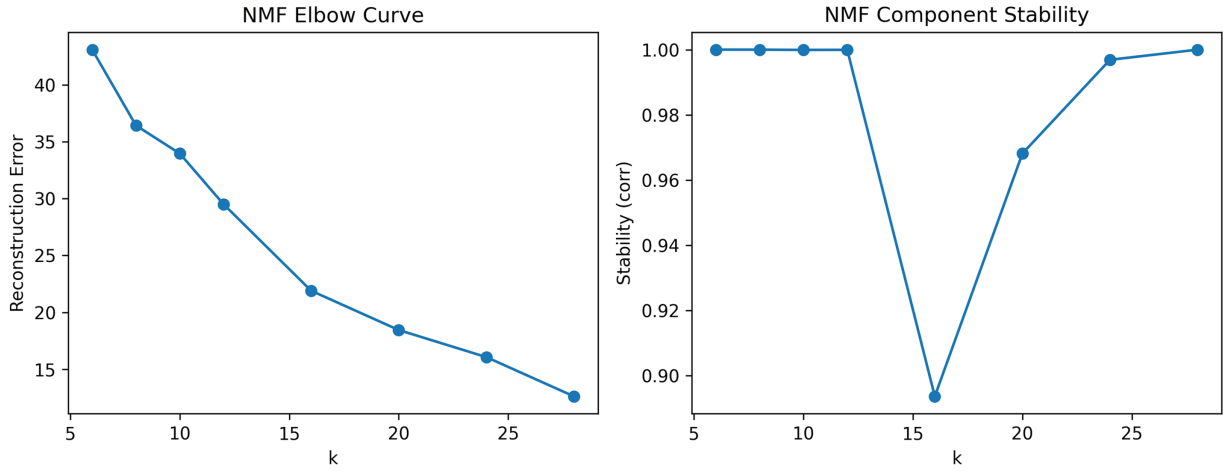



**Figure 5:** Model selection for NMF. (**a**) Reconstruction error vs. $k$ (elbow curve). (**b**) Stability analysis using average Pearson correlation across 5 random initializations.

**Table 8:** Quantitative summary of reconstruction error and stability for different NMF component counts $k$.

| $k$ | Reconstruction Error | Stability (Corr) |
|---|---|---|
| 8 | 41.2 | 1.00 |
| 12 | 26.4 | 0.99 |
| 16 | 17.0 | 0.90 |
| 20 | 15.5 | 0.98 |
| 24 | 14.7 | 0.99 |

Reconstruction error decreases monotonically with increasing $k$ and exhibits a clear elbow around $k = 16$, where error drops by 58.7% relative to $k = 8$. Stability remains high across all tested values (>0.90 Pearson correlation). Beyond $k = 16$, additional components provide marginal error reduction while fragmenting interpretable structure. Results remain qualitatively stable for $k \in \{12, 16, 20\}$, including cross-dataset trends (CHD → OTIDS). Based on this trade-off, we select $k = 16$ for all experiments.

### 5.2.3 Interpretability of Components

The learned basis matrix **H** exhibits semantically meaningful structure. As shown in Fig. 6, components specialize in subsets of traffic attributes such as message rate, entropy, DLC profiles, and CAN-ID frequencies. Table 9 lists the top-10 defining features per component, confirming this specialization.

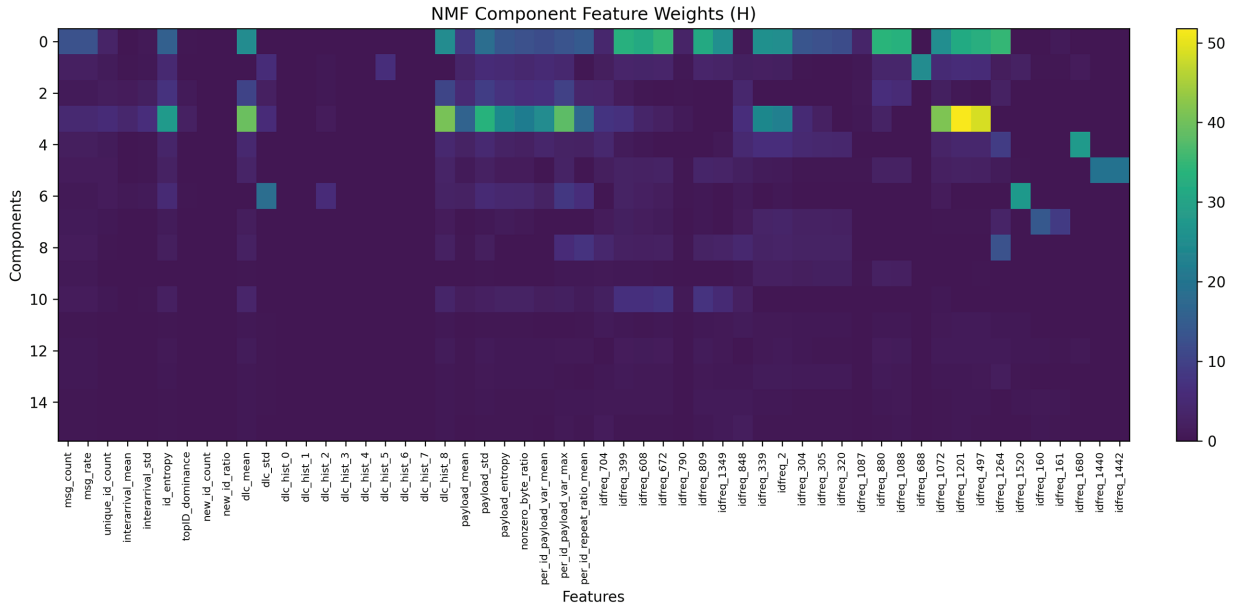



**Figure 6:** Heatmap of NMF basis matrix **H** ($k = 16$). Distinct components emphasize interpretable subsets of features such as message rate, payload entropy, and CAN-ID frequencies.

**Table 9:** Top-10 weighted features per NMF component.

| Comp | Top Features (weight) |
|---|---|
| 0 | idfreq_1264 (35.03), idfreq_672 (34.65), idfreq_880 (33.89), idfreq_1088 (33.14), idfreq_399 (32.82), idfreq_497 (32.44), idfreq_1201 (31.56), idfreq_809 (31.55), idfreq_608 (31.54), idfreq_339 (25.87) |
| 1 | idfreq_688 (25.00), dlc_hist_5 (6.38), idfreq_1201 (6.16), dlc_std (6.08), idfreq_497 (5.75), payload_std (5.73), idfreq_1072 (5.31), per_id_payload_var_mean (5.15), payload_entropy (5.08), id_entropy (4.76) |
| 2 | dlc_hist_8 (10.62), dlc_mean (10.32), per_id_payload_var_max (9.76), payload_std (9.36), id_entropy (7.27), payload_entropy (7.15), per_id_payload_var_mean (6.85), nonzero_byte_ratio (6.59), idfreq_880 (6.49), idfreq_1088 (5.59) |
| 3 | idfreq_1201 (51.79), idfreq_497 (48.56), idfreq_1072 (41.44), dlc_hist_8 (40.77), dlc_mean (39.50), per_id_payload_var_max (38.15), payload_std (33.41), id_entropy (28.05), per_id_payload_var_mean (24.75), payload_entropy (24.12) |
| 4 | idfreq_1680 (28.01), idfreq_1264 (9.15), idfreq_2 (6.66), idfreq_339 (6.49), payload_std (4.55), idfreq_848 (4.46), idfreq_304 (4.46), dlc_mean (4.37), dlc_hist_8 (4.36), idfreq_305 (4.15) |
| 5 | idfreq_1442 (19.47), idfreq_1440 (19.47), idfreq_809 (3.24), dlc_hist_8 (3.17), dlc_mean (3.14), idfreq_1349 (3.04), idfreq_672 (3.01), per_id_payload_var_max (2.85), idfreq_399 (2.68), idfreq_1088 (2.57) |
| 6 | idfreq_1520 (27.67), dlc_std (18.43), per_id_payload_var_max (8.42), per_id_repeat_ratio_mean (6.32), dlc_hist_2 (5.91), payload_std (5.25), id_entropy (5.20), payload_entropy (4.41), nonzero_byte_ratio (4.06), dlc_hist_8 (2.90) |
| 7 | idfreq_160 (14.00), idfreq_161 (8.86), idfreq_2 (3.42), idfreq_1264 (3.11), idfreq_339 (3.06), idfreq_304 (2.56), idfreq_305 (2.50), idfreq_320 (2.27), idfreq_399 (1.96), idfreq_608 (1.57) |

(Continued)

**Table 9 (continued)**

| Comp | Top Features (weight) |
|---|---|
| 8 | idfreq_1264 (12.82), per_id_repeat_ratio_mean (7.34), per_id_payload_var_max (5.98), idfreq_848 (4.38), idfreq_704 (3.90), idfreq_2 (3.10), idfreq_1349 (3.06), idfreq_339 (3.02), idfreq_305 (2.78), idfreq_320 (2.73) |
| 9 | idfreq_880 (2.41), idfreq_2 (2.22), idfreq_1088 (2.18), idfreq_339 (2.10), idfreq_320 (1.91), idfreq_305 (1.91), idfreq_304 (1.82), idfreq_704 (0.74), msg_count (0.65), msg_rate (0.65) |
| 10 | idfreq_672 (7.60), idfreq_809 (7.46), idfreq_399 (6.84), idfreq_608 (6.80), idfreq_1349 (5.34), dlc_mean (3.45), dlc_hist_8 (3.44), payload_entropy (2.81), nonzero_byte_ratio (2.63), idfreq_848 (2.19) |
| 11 | idfreq_848 (1.12), idfreq_704 (0.94), idfreq_1201 (0.92), idfreq_497 (0.89), idfreq_1349 (0.77), idfreq_339 (0.72), idfreq_1072 (0.71), idfreq_2 (0.67), idfreq_809 (0.57), idfreq_1088 (0.54) |
| 12 | idfreq_399 (1.22), idfreq_672 (1.22), idfreq_880 (1.22), idfreq_1088 (1.19), idfreq_608 (1.15), dlc_hist_8 (1.09), idfreq_809 (1.08), dlc_mean (1.04), idfreq_1264 (0.95), idfreq_497 (0.95) |
| 13 | idfreq_1264 (1.49), idfreq_2 (1.20), idfreq_339 (1.16), idfreq_305 (0.86), idfreq_880 (0.85), idfreq_304 (0.85), idfreq_320 (0.84), idfreq_399 (0.82), idfreq_1088 (0.78), idfreq_672 (0.76) |
| 14 | idfreq_848 (0.88), idfreq_1349 (0.88), idfreq_809 (0.82), idfreq_704 (0.77), idfreq_160 (0.75), idfreq_672 (0.73), idfreq_399 (0.65), idfreq_1088 (0.65), idfreq_608 (0.63), idfreq_880 (0.59) |
| 15 | idfreq_848 (1.51), idfreq_704 (1.25), idfreq_1349 (0.85), per_id_payload_var_max (0.69), idfreq_809 (0.58), idfreq_160 (0.53), dlc_mean (0.44), idfreq_1088 (0.44), dlc_hist_8 (0.43), idfreq_1201 (0.42) |

The coefficient matrix **W** captures temporal activation dynamics. As shown in Fig. 7, components remain stable during normal traffic and exhibit sharp activation bursts aligned with attack intervals, indicating that NMF captures discriminative temporal patterns.

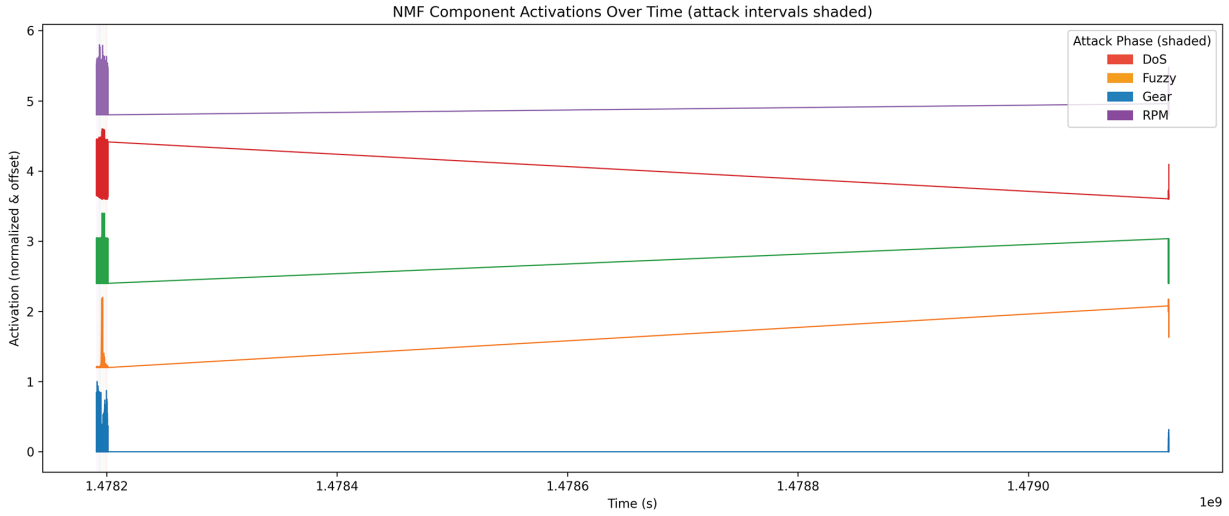



**Figure 7:** Temporal activations of NMF components (**W**). Shaded intervals denote attacks, showing strong activation bursts aligned with anomalies.

### *5.2.4 Component Purity*

Component purity analysis (Table 10) shows that several components (e.g., C2, C4, C7, C8, C10, C14) are strongly associated with specific attack types, while others remain dominated by normal traffic. This separation indicates that NMF learns selective latent factors that activate for particular intrusion patterns, supporting interpretable attribution of anomalies.

**Table 10:** Component purity at top-1% activations.

| Comp | Attack Ratio | Normal | DoS | Fuzzy | Gear | RPM |
|---|---|---|---|---|---|---|
| 0 (C0) | 0.606 | 877 | 116 | 1 | 717 | 515 |
| 1 (C1) | 0.152 | 1888 | 0 | 338 | 0 | 0 |
| 2 (C2) | 1.000 | 0 | 0 | 2213 | 6 | 7 |
| 3 (C3) | 0.876 | 277 | 281 | 232 | 543 | 893 |
| 4 (C4) | 0.862 | 307 | 43 | 1152 | 275 | 449 |
| 5 (C5) | 0.856 | 320 | 155 | 372 | 859 | 520 |
| 6 (C6) | 0.922 | 173 | 270 | 1239 | 228 | 316 |
| 7 (C7) | 0.952 | 107 | 87 | 1585 | 202 | 245 |
| 8 (C8) | 0.896 | 231 | 1445 | 7 | 336 | 207 |
| 9 (C9) | 0.774 | 503 | 188 | 39 | 707 | 789 |
| 10 (C10) | 0.894 | 237 | 42 | 227 | 704 | 1016 |
| 11 (C11) | 0.755 | 545 | 207 | 19 | 886 | 569 |
| 12 (C12) | 0.310 | 1537 | 475 | 60 | 148 | 6 |
| 13 (C13) | 0.334 | 1483 | 0 | 728 | 0 | 15 |
| 14 (C14) | 0.900 | 222 | 0 | 0 | 93 | 1911 |
| 15 (C15) | 0.043 | 2130 | 8 | 47 | 41 | 0 |

#### *5.2.5 Detection Performance*

Detection performance is evaluated using two thresholding strategies defined in Section 3: Youden's $J$ statistic and the 99th percentile ($P_{99}$) of reconstruction errors on training-normal windows. Table 11 summarizes test-set results. Both strategies achieve near-perfect AUC-ROC (0.9999) and AUC-PR (0.9989). Youden's $J$ yields balanced sensitivity and specificity, while $P_{99}$ provides a slightly more conservative detector suitable for safety-critical deployment.

**Table 11:** Standalone NMF detection results under Youden's $J$ and $P_{99}$ thresholds.

| Threshold | AUC-ROC | AUC-PR | Acc | Prec | Rec | F1 |
|---|---|---|---|---|---|---|
| Youden's $J$ | 0.9999 | 0.9989 | 0.9994 | 0.9920 | 0.9920 | 0.9920 |
| $P_{99}$ | 0.9999 | 0.9989 | 0.9992 | 0.9836 | 0.9938 | 0.9887 |

Overall, standalone NMF demonstrates that reconstruction residuals alone can reliably distinguish normal from anomalous CAN traffic while yielding interpretable latent components aligned with attack phases. The near-saturated scores are partly driven by the strong separability of CHD flooding attacks, motivating the more challenging cross-attack and cross-dataset evaluations presented in later sections.

### *5.3 Baseline Classifiers vs. NMF-W Features (Merged Data)*

While Section 5.2 demonstrated that standalone NMF can detect anomalies using reconstruction errors, it is also important to assess whether the NMF-transformed representation ($\mathbf{W}$) preserves discriminative information for conventional classifiers. We therefore compare baseline models trained on the original handcrafted feature set (Raw) against the same models trained on NMF-W features, evaluating NMF as an interpretable dimensionality-reduction stage.

All models are trained on the merged dataset spanning normal traffic and all attack types. Performance is reported using AUC-ROC, AUC-PR, Accuracy, Precision, and F1. Experiments are repeated across multiple random seeds and averaged.

Table 12 shows that Raw features provide slightly higher absolute performance across most metrics, consistent with the higher-dimensional representation retaining maximal discriminative information. However, NMF-W remains highly competitive: for example, RF achieves AUC-ROC 0.9972 and AUC-PR 0.9964, and XGB achieves AUC-ROC 0.9981 and AUC-PR 0.9977. Fig. 8 visualizes the Raw vs. NMF-W comparison for AUC-PR, AUC-ROC, and F1. Overall, NMF-W yields a compressed and interpretable feature space while maintaining strong detection performance.

**Table 12:** Performance of baseline classifiers on raw features vs. NMF-W features (merged dataset).

| Model | Feat. | AUC-ROC | AUC-PR | Acc | Prec | F1 |
|---|---|---|---|---|---|---|
| RF | Raw | 1.0000 | 1.0000 | 0.9998 | 0.9990 | 0.9994 |
| RF | NMF-W | 0.9972 | 0.9964 | 0.9948 | 0.9620 | 0.9757 |
| LR | Raw | 1.0000 | 1.0000 | 0.9990 | 0.9981 | 0.9985 |
| LR | NMF-W | 0.9835 | 0.9818 | 0.9540 | 0.9200 | 0.9329 |
| SVM | Raw | 1.0000 | 1.0000 | 0.9991 | 0.9989 | 0.9995 |
| SVM | NMF-W | 0.9829 | 0.9805 | 0.9571 | 0.9220 | 0.9372 |
| XGB | Raw | 1.0000 | 1.0000 | 0.9993 | 0.9990 | 0.9995 |
| XGB | NMF-W | 0.9981 | 0.9977 | 0.9750 | 0.9640 | 0.9816 |
| IF | Raw | 0.9881 | 0.9852 | 0.9510 | 0.9312 | 0.9449 |
| IF | NMF-W | 0.9699 | 0.9633 | 0.9330 | 0.9011 | 0.9124 |

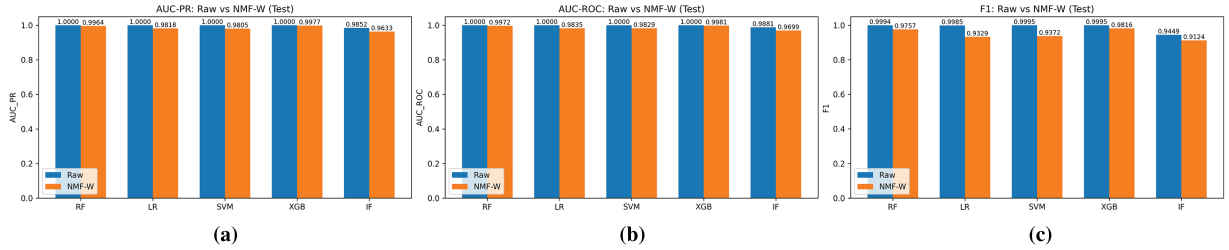



**Figure 8:** Comparison of classifier performance using raw vs. NMF-W features: (**a**) AUC-PR; (**b**) AUC-ROC; (**c**) F1-score.

**Classifier-specific behavior.** Performance differences reflect how models exploit the NMF-W representation. XGB performs best on NMF-W because boosting can capture nonlinear interactions among latent components while remaining robust in low dimension. LR and SVM also benefit from the compact, less noisy latent space, which can make separation more linearly accessible. In contrast, RF and IF typically benefit from richer feature heterogeneity in higher-dimensional spaces; compressing the feature space into $k$ components reduces split diversity and can lead to modest degradation.

**IF behavior under NMF transformation.** IF occasionally degrades on NMF-W because it isolates anomalies via random partitioning, a mechanism that is often more effective when many heterogeneous features are available. NMF-W compresses the data into a low-rank, semantically structured space, which can reduce the effectiveness of random splits. The magnitude of this effect is dataset dependent and is more visible

on CHD, where Raw features already separate flooding attacks strongly. This does not affect the suitability of NMF-W for supervised or hybrid models, where we observe consistent robustness and interpretability gains.

In summary, classifiers trained on Raw features achieve marginally higher in-domain performance, while NMF-W offers a compact, interpretable representation that remains close to Raw (typically within 1%–3%). The near-ceiling results in both spaces are influenced by CHD flooding attacks, whose extreme message rates and ID patterns create large distribution shifts from normal traffic. This trade-off supports NMF as an efficient and explainable preprocessing stage for real-time IDS pipelines in resource-constrained vehicular settings, while later cross-dataset experiments (Section 5.10) provide a more stringent evaluation of generalization.

### *5.4 Lookup Table and Rule-Based Baselines*

Automotive ECUs commonly employ lightweight rule-based checks, such as ID whitelisting and message-rate monitoring, to detect malformed CAN traffic. To assess whether such rules can replace learned detectors, we implement two representative baselines using the same 70/10/20 chronological split and windowed features as the ML models.

**ID Whitelist.** A window is flagged as an attack if it contains at least one CAN identifier not observed in the normal-training subset (`new_id_count` $> 0$), simulating a lookup table of valid IDs.

**DoS-rate Threshold.** A window is flagged as DoS if its message rate exceeds $\mu + 3\sigma$ computed from normal-training windows, corresponding to a simple frequency-based flooding detector.

Table 13 reports AUC-PR, AUC-ROC, and F1-scores for these rule baselines, alongside Random Forest (RF) models trained on Raw and NMF–W features.

**Table 13:** Rule baselines vs. ML models (AUC-PR/AUC-ROC/F1).

| Test Attack | Whitelist(ID) | DoS-Rate | RF-Raw | RF-NMF–W |
|---|---|---|---|---|
| DoS | 1.000/1.000/1.000 | 0.049/0.511/0.043 | 1.000/1.000/0.888 | 0.990/0.9998/0.850 |
| Fuzzy | 1.000/1.000/1.000 | 0.145/0.530/0.112 | 1.000/1.000/0.955 | 0.998/0.9998/0.952 |
| Gear | –/–/0.000 | –/–/0.000 | –/–/0.000 | –/–/0.000 |
| RPM | –/–/0.000 | –/–/0.000 | –/–/0.000 | –/–/0.000 |

ID whitelisting performs well for DoS and Fuzzy attacks in this dataset, as both introduce identifiers absent during normal operation. In contrast, both rule-based methods fail entirely for Gear and RPM spoofing, which preserve valid CAN IDs and do not induce abnormal message rates. These results highlight the inherent limitations of static rules and motivate the use of statistical and representation-learning approaches, such as NMF–W combined with RF, which capture subtler distributional deviations beyond identifier or rate checks.

### *5.5 Attack-Wise Evaluation*

We conduct attack-wise evaluations to analyze how NMF-transformed features ($\mathbf{W}$) behave under different intrusion types: DoS, Fuzzy, Gear spoofing, and RPM spoofing. For each attack, baseline classifiers trained on raw features are compared against those trained on NMF–W using AUC-ROC, AUC-PR, and F1-score on the test set.

### 5.5.1 DoS and Fuzzy Attacks

Figs. 9 and 10 report results for DoS and Fuzzy injections. Both Raw and NMF–W representations achieve near-saturated performance, with AUC-ROC and AUC-PR values close to 1.0 and F1-scores exceeding 0.99 across supervised classifiers. This confirms that flooding-style attacks are highly separable in the CHD dataset. Importantly, NMF–W preserves this strong detection performance while reducing feature dimensionality. Minor performance drops are observed for IF under NMF–W, consistent with its sensitivity to low-dimensional transformations.

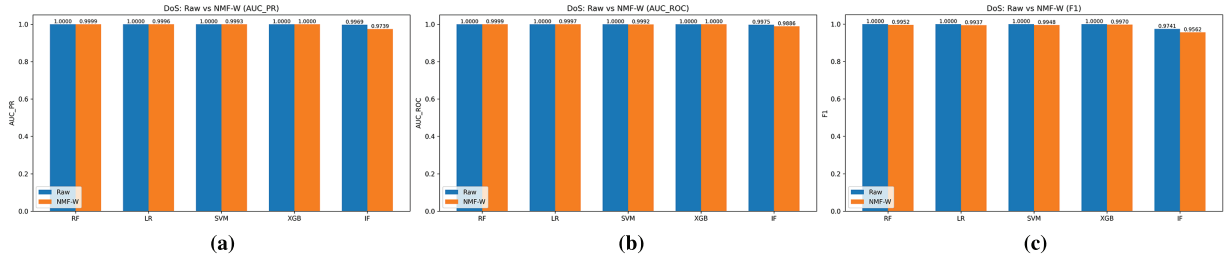



**Figure 9:** DoS attack: raw vs. NMF–W features across (**a**) AUC-PR, (**b**) AUC-ROC, and (**c**) F1-score.

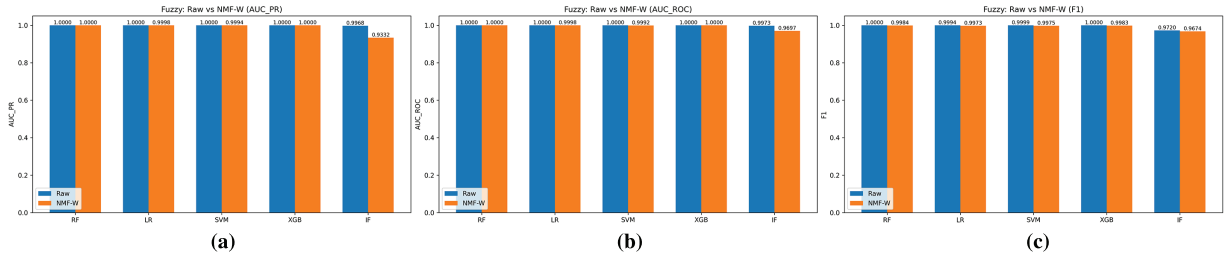



**Figure 10:** Fuzzy attack: raw vs. NMF–W features across (**a**) AUC-PR, (**b**) AUC-ROC, and (**c**) F1-score.

### 5.5.2 Gear and RPM Spoofing Attacks

Figs. 11 and 12 summarize results for Gear and RPM spoofing, which represent subtler manipulations that preserve valid CAN identifiers and message rates. While raw features again yield near-perfect metrics for supervised classifiers, NMF–W maintains strong performance for RF, LR, and SVM (AUC $\geq$ 0.99). In contrast, IF shows noticeable degradation under NMF–W (e.g., F1 $\approx$ 0.84–0.89), indicating that unsupervised detectors relying on random partitioning benefit less from compressed, structured representations. These results highlight the increased difficulty of detecting spoofing attacks and the limitations of purely unsupervised methods.

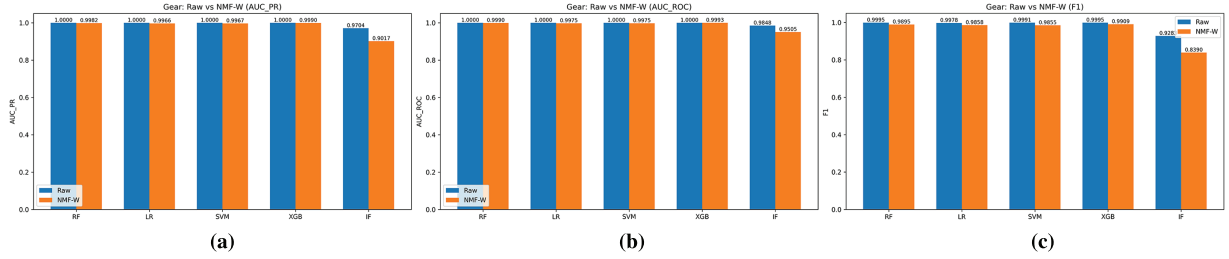



**Figure 11:** Gear spoofing attack: raw vs. NMF–W features across (**a**) AUC-PR, (**b**) AUC-ROC, and (**c**) F1-score.

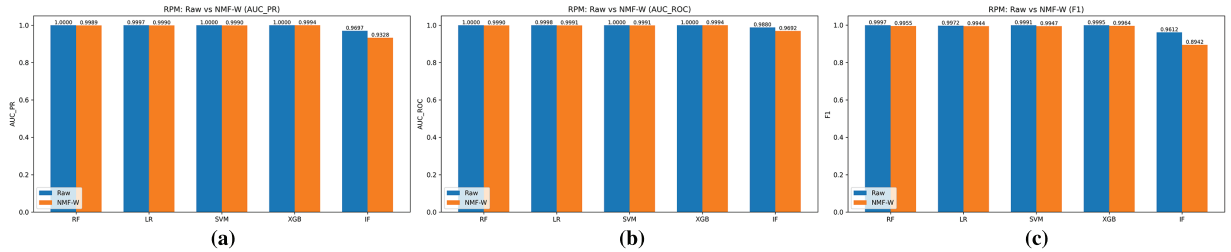



**Figure 12:** RPM spoofing attack: raw vs. NMF–W features across (**a**) AUC-PR, (**b**) AUC-ROC, and (**c**) F1-score.

Table 14 aggregates the averaged performance across classifiers. While raw features achieve near-perfect scores due to strong dataset separability, these results may overestimate real-world robustness. NMF–W remains competitive across all attack types, particularly for DoS and Fuzzy attacks, while offering reduced dimensionality and improved interpretability. For subtler spoofing attacks, performance degradation is primarily observed in unsupervised models, reinforcing the importance of combining NMF representations with supervised or hybrid detectors for deployment.

**Table 14:** Attack-wise performance comparison of raw vs. NMF–W features (averaged across classifiers).

| Attack | AUC-ROC | AUC-PR | F1 |
|---|---|---|---|
| DoS | 0.9999 | 0.9998 | 0.996 |
| Fuzzy | 0.9998 | 0.9997 | 0.995 |
| Gear | 0.9890 | 0.9810 | 0.940 |
| RPM | 0.9905 | 0.9830 | 0.945 |

### *5.6 Attack Localization Analysis*

Window-level labeling assigns an attack label to a 0.1 s window whenever it contains at least one injected frame. Although this convention is standard in CAN intrusion detection, it raises a concern: if a window contains only a few injected frames alongside many normal frames, does the entire interval meaningfully reflect anomalous behavior? To examine this, we perform a fine-grained localization analysis to assess whether attack-induced disturbances remain detectable at shorter timescales.

We use the following terminology:

- **Attack frame:** a CAN frame flagged as injected by the dataset (CHD flag = “T”).
- **Normal frame:** a frame not labeled as injected.
- **Attack window:** a 0.1 s window containing at least one attack frame.
- **Micro-window:** a non-overlapping 10 ms segment obtained by subdividing an attack window.

A statistical baseline is first constructed from normal traffic at the 10 ms resolution. For each normal micro-window, we compute the message rate (`msg_rate`) and the Shannon entropy of the CAN ID distribution (`id_entropy`). From these, we estimate the 99th percentile of `msg_rate` and the 1st–99th percentile range of `id_entropy`, which define typical normal behavior.

For each attack type (DoS, Fuzzy, Gear spoofing, RPM spoofing), all attack windows are subdivided into 10 ms micro-windows, yielding a total of $N_{\text{mw}}$ micro-windows. Each micro-window is evaluated using three indicators:

1. **Has Attack:** contains at least one injected frame.
2. **Rate Anomaly:** `msg_rate` exceeds the normal 99th percentile.
3. **Entropy Anomaly:** `id_entropy` lies outside the normal 1st–99th percentile range.

A micro-window is considered *localized* if it satisfies at least one indicator.

Table 15 reports localization statistics normalized by $N_{\text{mw}}$. Across attack types, approximately 41%–43% of micro-windows contain injected frames. However, a larger fraction—between 45% and 56%—exhibits at least one anomaly indicator relative to normal behavior. For example, in the DoS subset, 42.91% of micro-windows contain attack frames, while 52.92% are localized by rate or entropy deviations. Similar trends are observed for Gear and RPM spoofing, where roughly 45% of micro-windows fall outside normal statistical bounds despite only 41%–42% containing injected frames.

**Table 15:** Micro-window localization statistics (10 ms resolution) within attack windows.

| Attack Type | Total Micro-windows | Has Attack (%) | Rate Anom. (%) | Entropy Anom. (%) | Localized (%) |
|---|---|---|---|---|---|
| DoS | 559,311 | 42.91 | 3.73 | 52.83 | 52.92 |
| Fuzzy | 587,403 | 43.36 | 6.29 | 50.30 | 55.69 |
| Gear | 485,762 | 41.20 | 17.97 | 44.59 | 45.22 |
| RPM | 486,350 | 41.97 | 20.42 | 44.88 | 45.42 |

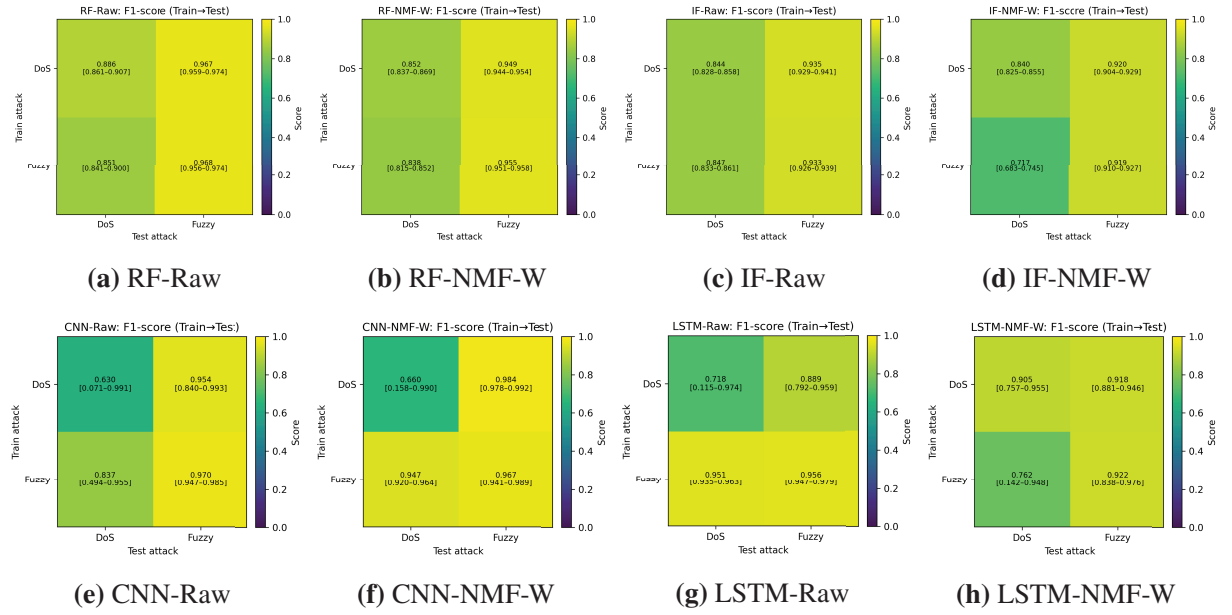
Note: "Localized" denotes micro-windows exhibiting at least one anomaly indicator.

These results demonstrate that window-level labeling does not indiscriminately mark long stretches of normal traffic as anomalous. Instead, injected frames create localized temporal distortions in traffic statistics that remain detectable at the 10 ms scale. Operationally, the proposed system functions as an IDS: a 0.1 s attack window serves as a coarse-grained alert that can trigger lightweight, automated follow-up checks at finer resolution (e.g., micro-window analysis) within the ECU or gateway. The localization results are therefore intended to support operator awareness and hierarchical IDS workflows rather than frame-level intrusion prevention. Limitations of window-based detection for highly stealthy or adaptive attacks are discussed further in Section 5.15.

### *5.7 Cross-Attack Generalization*

To evaluate robustness under unseen attack types, we conducted a cross-attack generalization study using the two attack classes with sufficient temporal coverage in the CHD dataset: *DoS* and *Fuzzy*. Both attacks span long durations, yielding adequate attack-labeled windows after the 70/10/20 chronological split. In contrast, Gear and RPM spoofing occur only in short bursts, producing sparse or absent attack windows in some train–test combinations; their cross-attack metrics are therefore statistically unreliable. Accordingly, our controlled analysis focuses on the DoS–Fuzzy pair, while implications for short-duration attacks are discussed in Section 5.15. All reported cross-attack results include 95% bootstrap confidence intervals for AUC-ROC, AUC-PR, and F1.

**Tabular models (RF and IF).** Fig. 13 summarizes cross-attack F1-scores for RF, IF, CNN, and LSTM using Raw and NMF-W features (full heatmaps and Δ-analyses are provided in the supplementary materials, Figs. S1–S7). Same-attack performance is consistently strong, with AUC-ROC values near 1.0 and F1-scores above 0.85. This reflects the fact that CHD's DoS and Fuzzy attacks induce pronounced distributional shifts (high-rate, payload-saturated bursts), making them easier to separate from normal traffic than more subtle real-world attacks.




**(a)** RF-Raw **(b)** RF-NMF-W **(c)** IF-Raw **(d)** IF-NMF-W

**(e)** CNN-Raw **(f)** CNN-NMF-W **(g)** LSTM-Raw **(h)** LSTM-NMF-W

**Figure 13:** Cross-attack F1-score heatmaps with 95% confidence intervals for RF, IF, CNN, and LSTM using raw and NMF-W features.

Across train–test combinations, two consistent patterns emerge. First, training on Fuzzy generalizes well to DoS, with RF and IF achieving stable AUC-ROC values (0.97–0.99) and F1-scores typically above 0.83 for both Raw and NMF-W representations. Second, the reverse direction (DoS → Fuzzy) yields lower but still positive transfer, with F1-scores ranging from 0.84 to 0.96. This asymmetry suggests that Fuzzy attacks introduce more diverse structural perturbations that support downstream generalization, whereas DoS attacks—dominated by payload saturation at fixed CAN IDs—provide less transferable structure.

The effect of NMF-W is attack dependent. For RF, NMF-W slightly reduces F1-scores in some settings while preserving high AUC-ROC values. For IF, NMF-W stabilizes detection when trained on DoS but reduces F1 when trained on Fuzzy. Supplementary Δ-heatmaps (Figs. S6 and S7) show that NMF-W introduces structured shifts rather than uniform gains, reflecting the underlying separability of each attack type.

**Sequence models (CNN, LSTM, Attention).** We further evaluated CNN, LSTM, and Attention-based architectures under the same DoS ↔ Fuzzy protocol (full metric heatmaps in Figs. S3–S5). All sequence models achieve strong same-attack detection, with AUC-ROC values above 0.98 and F1-scores in the 0.90–0.99 range, confirming their ability to capture attack-specific temporal signatures.

Under cross-attack evaluation, the same asymmetry persists. Models trained on Fuzzy generalize more reliably to DoS (AUC-ROC 0.94–0.99; F1 0.83–0.97) than those trained on DoS and tested on Fuzzy, which exhibit greater F1 variability despite high AUC-ROC values. CNN and Attention show moderate degradation under DoS → Fuzzy, while LSTM exhibits wider confidence intervals, reflecting sensitivity to the simpler temporal structure of DoS attacks.

Overall, the benefit of NMF-W for sequence models is mixed: it improves performance in some cases (e.g., CNN under Fuzzy → DoS) but introduces additional variance in others. These findings indicate that deep sequence models already learn strong latent representations from Raw features, and that cross-attack generalization is primarily governed by intrinsic differences in attack structure rather than model choice. The consistent asymmetry across tabular and sequence models reinforces the dataset-driven nature of cross-attack challenges in CAN intrusion detection.

### *5.8 Class Imbalance and Robustness Enhancements*

We study three complementary strategies to address skewed labels and improve robustness: (i) tabular resampling with SMOTE/ADASYN, (ii) sequence models with class-weighted training and fixed-recall reporting, and (iii) mixed-attack training (two attacks) evaluated on a held-out third attack. Across all settings, we compare Raw features against NMF representations (**W**).

**Rationale for imbalance handling.** Severe class imbalance is intrinsic to CAN intrusion detection, where attack windows represent a small fraction of overall traffic. We adopt SMOTE and ADASYN as representative data-level techniques because they increase minority-class support without discarding normal samples. SMOTE interpolates between existing minority instances, while ADASYN emphasizes hard-to-learn regions, making both suitable for tabular CAN features and for analyzing how oversampling interacts with Raw vs. NMF-W representations. Alternative strategies such as undersampling or cost-sensitive losses were not emphasized here, as undersampling removes informative normal traffic and loss reweighting is model-specific and not applicable to unsupervised or hybrid settings (e.g., standalone NMF). Our goal is not to propose a new imbalance method, but to systematically assess how standard techniques interact with NMF-based representations under realistic CAN traffic skew.

**Tabular resampling (SMOTE/ADASYN).** Using the merged tabular dataset, we trained {RF, LR, SVM, XGB} under three settings: *pure* (no resampling), *SMOTE*, and *ADASYN*, for both Raw and NMF-W. Fig. 14 summarizes test AUC-PR, AUC-ROC, and F1. Raw-feature models already operate near ceiling, and resampling yields only marginal changes. In contrast, NMF-W shows small but consistent gains in F1 and AUC-PR under SMOTE/ADASYN (e.g., XGB with ADASYN achieves $F1 \approx 0.98$ and AUC-ROC $\approx 0.998$), indicating that synthetic minority generation more effectively populates low-density regions of the compact latent space. Overall, resampling is most beneficial when operating in the NMF representation.

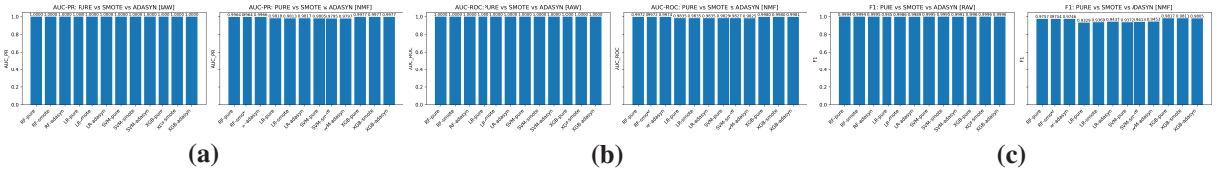
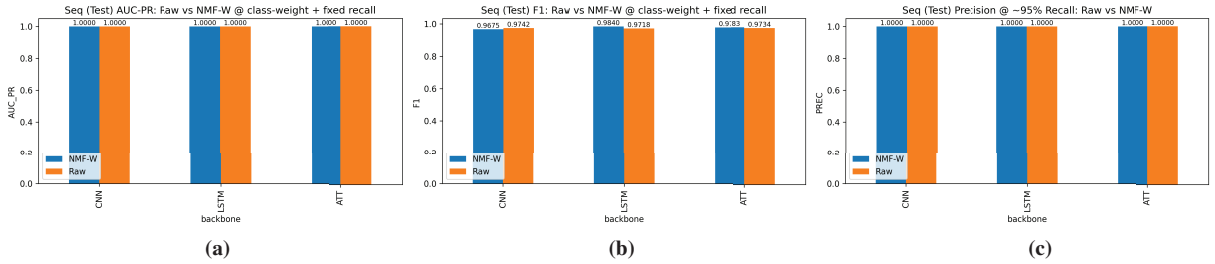



**Figure 14:** Tabular resampling on the merged dataset (SMOTE/ADASYN): (**a**) Test AUC-PR; (**b**) Test AUC-ROC; (**c**) Test F1-score for Raw and NMF-W across {RF, LR, SVM, XGB}.

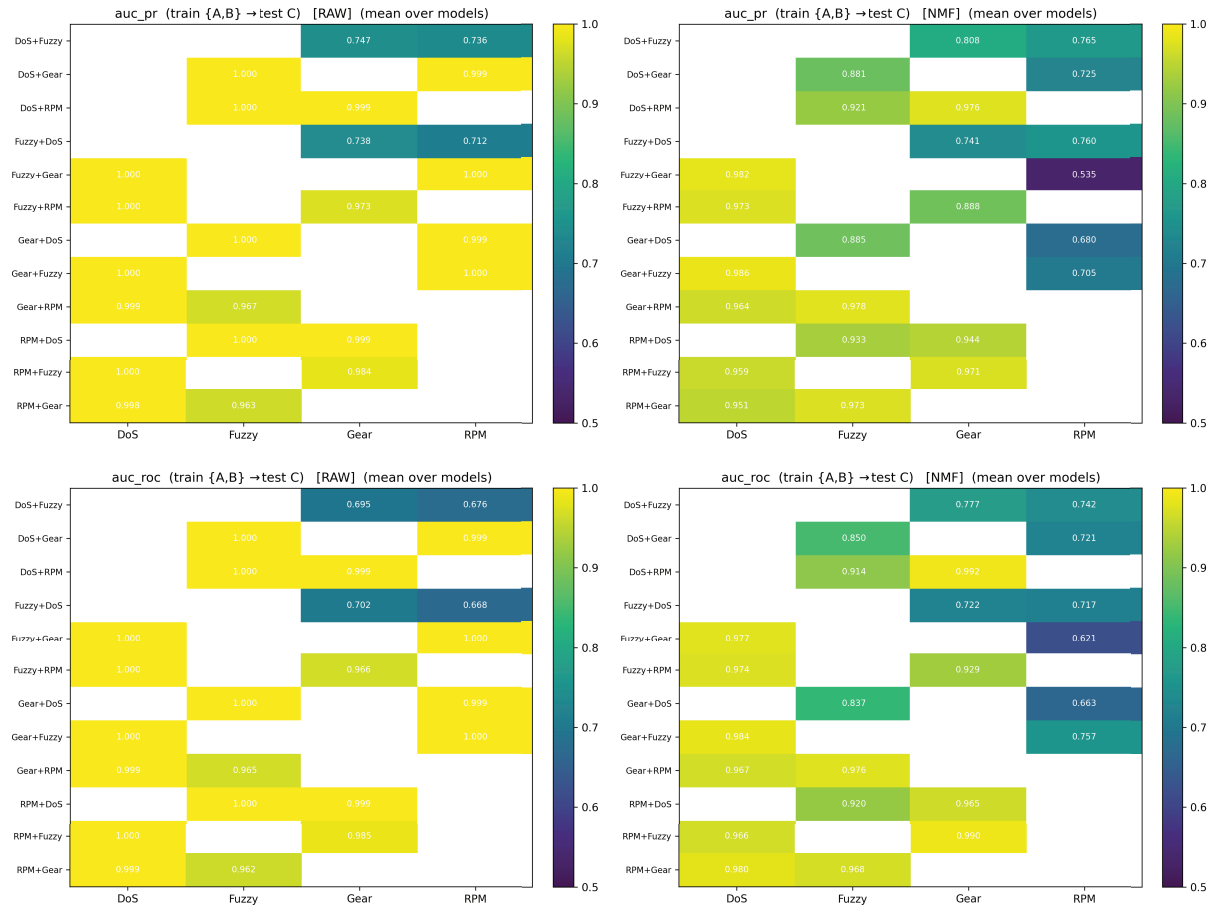
**Sequence models with class-weighting.** On the merged sequence dataset, we trained {CNN, LSTM, Attention} using class-weighted loss for both Raw and NMF-W features. Fig. 15 reports test AUC-PR, F1, and precision at a fixed high recall. All architectures achieve strong performance (AUC-PR $\gtrsim 0.94$, accuracy $\approx 0.998$). Class-weighting improves minority sensitivity, while maintaining high precision at strict recall levels. In several cases, NMF-W provides slightly improved recall–F1 trade-offs, indicating that compact representations can complement temporal modeling under imbalance.

**Mixed-attack training (2 → 1).** To simulate low-label robustness, we trained models on mixtures of two attacks and evaluated them on a held-out third attack. Fig. 16 shows cross-condition heatmaps for Raw and NMF-W. When the held-out target is subtle (Gear/RPM), Raw representations often exhibit brittle generalization (e.g., $F1 \approx 0$), whereas NMF-W retains non-trivial transfer (e.g., DoS+Fuzzy → Gear with XGB on NMF-W yields $F1 \approx 0.60$ and AUC-PR $\approx 0.98$). When the target is DoS or Fuzzy, both feature spaces generalize strongly. These results indicate that compact, part-based NMF features provide a more transferable basis under attack-mixture training, particularly for subtle targets.




**(a)** **(b)** **(c)**

**Figure 15:** Sequence models with class-weighting on merged data: (**a**) Test AUC-PR; (**b**) Test F1; (**c**) Precision at fixed recall.




**(a)** Raw features **(b)** NMF-W features

**Figure 16:** Mixed-attack training (2 → 1) on tabular data: AUC-PR (top) and AUC-ROC (bottom). NMF-W is more stable for subtle held-out targets (Gear/RPM), while both spaces perform near ceiling for DoS/Fuzzy.

**Generalization to unseen attacks.** Several experiments in this section explicitly assess robustness to previously unseen attack behaviors. The mixed-attack training setup approximates realistic deployment scenarios in which new attack patterns are encountered without explicit labels. Results show that NMF-W representations retain meaningful detection capability for unseen and subtle targets, while raw-feature models may collapse under the same conditions. In addition, standalone NMF operates without attack labels and thus inherently supports anomaly detection beyond known signatures. While no IDS can guarantee

detection of all unknown attacks, these findings indicate that the proposed framework exhibits meaningful generalization to novel deviations from learned normal traffic patterns.

### *5.9 Interpretability Analysis*

#### *5.9.1 NMF Interpretability*

A key motivation for adopting NMF is its ability to produce interpretable, parts-based representations of CAN traffic. Here, we focus on *internal interpretability*: we assess whether NMF components are consistent and semantically meaningful within our datasets and models, without claiming that their usefulness for human operators has been validated in real-world workflows. NMF yields a basis matrix **H** that highlights subsets of traffic features and a coefficient matrix **W** that provides per-window component activations. We analyze interpretability in three ways: phase prototypes, distinctive component attribution, and inter-phase similarity.

**Phase prototypes.** We aggregate mean **W** activations for each traffic phase (Normal, DoS, Fuzzy, Gear spoofing, RPM spoofing). As shown in Fig. 17, distinct activation profiles emerge: specific components activate strongly for DoS and Fuzzy injections, while others remain near normal baselines, indicating that NMF captures latent factors aligned with attack semantics.

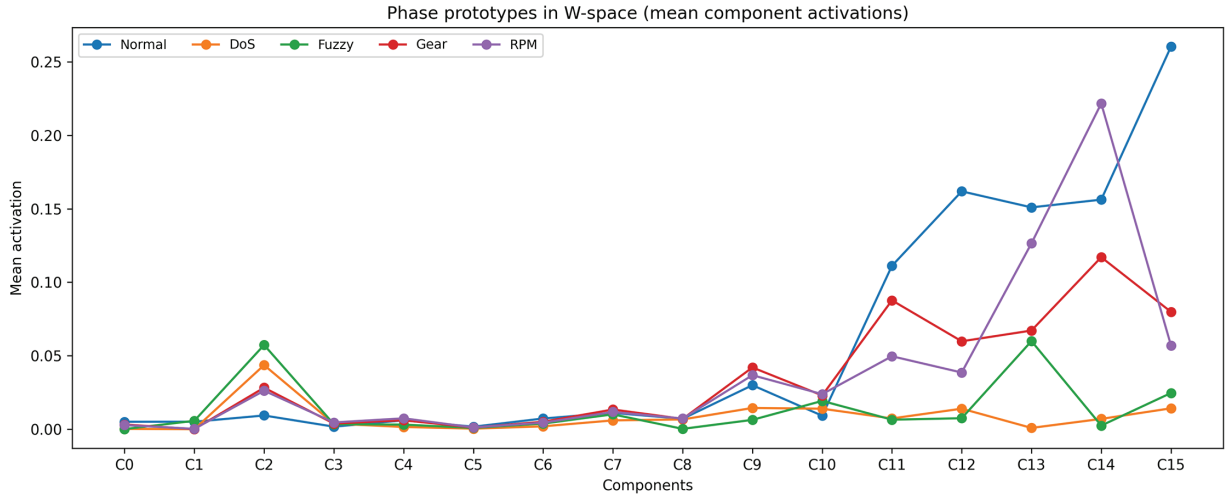



**Figure 17:** Average NMF component activations per traffic phase. Distinct peaks highlight parts-based structure aligned with DoS, Fuzzy, Gear spoofing, and RPM spoofing attacks.

**Distinctive component attribution.** We identify the top-5 components per attack type by computing $z$-scores of mean activations relative to Normal. Table 16 shows recurring components (e.g., C2 across all attacks, and C3/C10 across multiple phases) with strong discriminative activations, providing stable attribution handles for interpreting anomalies in the latent space.

**Cross-attack component activation.** Several components are consistently activated across multiple attack types (e.g., C2 across all attacks; C3 and C10 across multiple phases), indicating that NMF captures both *shared* anomaly structure (generic deviations such as entropy distortions or sustained rate changes) and *phase-specific* behavior. This mixture of shared and specialized components provides an interpretable mechanism for robustness under cross-attack and cross-dataset evaluation.

**Table 16:** Top-5 distinctive NMF components per attack phase (z-score vs. Normal).

| Phase | Rank | Component | Z_vs._Normal | MeanAct_Phase | MeanAct_Normal |
|---|---|---|---|---|---|
| DoS | 1 | C2 | 7.1948 | 0.0436 | 0.0093 |
| DoS | 2 | C3 | 1.8064 | 0.0038 | 0.0017 |
| DoS | 3 | C10 | 0.7106 | 0.0139 | 0.0092 |
| DoS | 4 | C8 | −0.1155 | 0.0066 | 0.0071 |
| DoS | 5 | C5 | −0.2623 | 0.0004 | 0.0017 |
| Fuzzy | 1 | C2 | 10.0727 | 0.0573 | 0.0093 |
| Fuzzy | 2 | C10 | 1.5016 | 0.0192 | 0.0092 |
| Fuzzy | 3 | C3 | 1.4732 | 0.0034 | 0.0017 |
| Fuzzy | 4 | C1 | 0.0506 | 0.0057 | 0.0051 |
| Fuzzy | 5 | C5 | −0.1635 | 0.0009 | 0.0017 |
| Gear | 1 | C2 | 3.9696 | 0.0282 | 0.0093 |
| Gear | 2 | C10 | 2.0608 | 0.0230 | 0.0092 |
| Gear | 3 | C3 | 1.8774 | 0.0039 | 0.0017 |
| Gear | 4 | C9 | 0.6551 | 0.0420 | 0.0299 |
| Gear | 5 | C7 | 0.4476 | 0.0134 | 0.0111 |
| RPM | 1 | C2 | 3.5383 | 0.0262 | 0.0093 |
| RPM | 2 | C3 | 2.5212 | 0.0046 | 0.0017 |
| RPM | 3 | C10 | 2.1864 | 0.0238 | 0.0092 |
| RPM | 4 | C14 | 0.7582 | 0.2217 | 0.1562 |
| RPM | 5 | C4 | 0.6781 | 0.0074 | 0.0061 |

**Component-level summary and phase similarity.** Table 17 summarizes each component via purity at top-1% activations and mean activations per phase. To compare phases globally, we compute a **W**-space prototype for each phase by averaging per-window activations and measure cosine similarity between prototypes (Fig. 18). Normal traffic is most similar to Gear spoofing, while DoS and Fuzzy are more distant, consistent with the separability observed in earlier experiments.

**Table 17:** Component-level summary: purity at top 99th percentile (W) and mean activations per phase.

| Comp. | Attack_ Ratio | Normal | DoS | Fuzzy | Gear | RPM | $\bar{W}_{Normal}$ | $\bar{W}_{DoS}$ | $\bar{W}_{Fuzzy}$ | $\bar{W}_{Gear}$ | $\bar{W}_{RPM}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C0 | 0.6060 | 877 | 116 | 1 | 717 | 515 | 0.0051 | 0.0002 | 0.0001 | 0.0032 | 0.0028 |
| C1 | 0.1518 | 1888 | 0 | 338 | 0 | 0 | 0.0051 | 0.0000 | 0.0057 | 0.0000 | 0.0002 |
| C2 | 1.0000 | 0 | 0 | 2213 | 6 | 7 | 0.0093 | 0.0436 | 0.0573 | 0.0282 | 0.0262 |
| C3 | 0.8756 | 277 | 281 | 232 | 543 | 893 | 0.0017 | 0.0038 | 0.0034 | 0.0039 | 0.0046 |
| C4 | 0.8621 | 307 | 43 | 1152 | 275 | 449 | 0.0061 | 0.0015 | 0.0031 | 0.0061 | 0.0074 |
| C5 | 0.8562 | 320 | 155 | 372 | 859 | 520 | 0.0017 | 0.0004 | 0.0009 | 0.0011 | 0.0012 |
| C6 | 0.9223 | 173 | 270 | 1239 | 228 | 316 | 0.0073 | 0.0019 | 0.0038 | 0.0051 | 0.0044 |
| C7 | 0.9519 | 107 | 87 | 1585 | 202 | 245 | 0.0111 | 0.0060 | 0.0101 | 0.0134 | 0.0117 |
| C8 | 0.8962 | 231 | 1445 | 7 | 336 | 207 | 0.0071 | 0.0066 | 0.0002 | 0.0070 | 0.0073 |
| C9 | 0.7740 | 503 | 188 | 39 | 707 | 789 | 0.0299 | 0.0144 | 0.0064 | 0.0420 | 0.0367 |
| C10 | 0.8935 | 237 | 42 | 227 | 704 | 1016 | 0.0092 | 0.0139 | 0.0192 | 0.0230 | 0.0238 |

(Continued)

**Table 17 (continued)**

| Comp. | Attack_ Ratio | Normal | DoS | Fuzzy | Gear | RPM | $\bar{W}_{Normal}$ | $\bar{W}_{DoS}$ | $\bar{W}_{Fuzzy}$ | $\bar{W}_{Gear}$ | $\bar{W}_{RPM}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C11 | 0.7552 | 545 | 207 | 19 | 886 | 569 | 0.1112 | 0.0073 | 0.0065 | 0.0877 | 0.0496 |
| C12 | 0.3095 | 1537 | 475 | 60 | 148 | 6 | 0.1619 | 0.0140 | 0.0075 | 0.0598 | 0.0386 |
| C13 | 0.3338 | 1483 | 0 | 728 | 0 | 15 | 0.1510 | 0.0009 | 0.0599 | 0.0671 | 0.1265 |
| C14 | 0.9003 | 222 | 0 | 0 | 93 | 1911 | 0.1562 | 0.0070 | 0.0025 | 0.1171 | 0.2217 |
| C15 | 0.0431 | 2130 | 8 | 47 | 41 | 0 | 0.2604 | 0.0142 | 0.0246 | 0.0798 | 0.0570 |

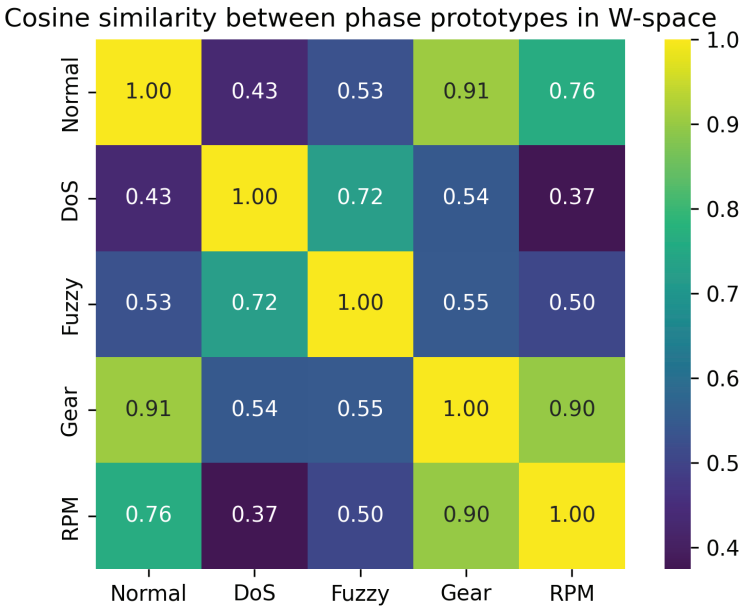



**Figure 18:** Cosine similarity between phase prototypes in **W**-space. Lower similarity implies better separability between phases.

Overall, these analyses indicate that NMF yields semantically meaningful latent components that are internally consistent across multiple views. Assessing how these interpretations assist security analysts or automotive engineers in real workflows is left for future work. Additional violin plots of per-component activation distributions are provided in the supplementary materials (Figs. S8–S10).

### *5.9.2 Model Interpretability*

To complement NMF component interpretability, we examine how supervised classifiers prioritize signals when trained on raw handcrafted features vs. NMF-derived latent components. We report feature importances for RF and XGB and absolute coefficients for LR in both spaces.

**Raw feature space.** Table 18 shows that raw-feature models rely heavily on a small set of identifier-specific features (e.g., `idfreq_*`) and related count-based indicators. While these features yield near-perfect performance on CHD, they can reflect dataset-specific artifacts (e.g., injected IDs and burst patterns), motivating the use of structured representations for improved robustness.

**Table 18:** Top features selected by RF, XGB, and LR in the raw feature space.

| Rank | RF (Gini) | XGB (Gain) | LR (\|coef\|) |
|---|---|---|---|
| 1 | dlc_hist_2 | idfreq_1349 | new_id_count |
| 2 | idfreq_1349 | msg_count | idfreq_399 |
| 3 | idfreq_1088 | idfreq_1088 | idfreq_880 |
| 4 | idfreq_2 | new_id_ratio | idfreq_1088 |
| 5 | idfreq_809 | new_id_count | idfreq_1072 |

**NMF-W feature space.** In the NMF-W space, models distribute attention across multiple latent components (Table 19). Notably, components such as C2 and C3—also highlighted as distinctive in Section 5.9—recur as influential signals across classifiers, providing a consistent bridge between unsupervised component semantics and supervised decision-making. Since each component maps back to a sparse subset of original features via **H**, these rankings offer a compact and traceable explanation of model behavior. Full ranked lists are provided in the supplementary materials (Tables S1 and S2).

**Table 19:** Top components selected by RF, XGB, and LR in the NMF-W feature space.

| Rank | RF (Gini) | XGB (Gain) | LR (\|coef\|) |
|---|---|---|---|
| 1 | C2 | C11 | C8 |
| 2 | C3 | C1 | C6 |
| 3 | C10 | C8 | C2 |
| 4 | C7 | C14 | C10 |
| 5 | C4 | C12 | C1 |

### *5.10 Cross-Dataset Validation (CHD → OTIDS)*

OTIDS represents an independent dataset collected from a different vehicle platform and includes attack realizations not observed during CHD training, providing a stringent test of robustness under dataset shift. To evaluate cross-domain transfer, all models were trained on CHD and evaluated on both the CHD-test split (in-domain) and OTIDS (out-of-domain), which contains CAN traffic from a real KIA Soul vehicle under DoS, Fuzzy, and Impersonation attacks. Table 20 summarizes supervised results, and Table 21 reports standalone NMF reconstruction-error performance.

**Table 20:** Cross-dataset results (training on CHD, testing on CHD-test and OTIDS). Metrics include AUC-ROC, AUC-PR, and F1-score at a validation-calibrated recall of approximately 0.95.

| Model/Space | CHD-test (in-domain) | | | OTIDS (out-of-domain) | | |
|---|---|---|---|---|---|---|
| | AUC-ROC | AUC-PR | F1@R ≈ 0.95 | AUC-ROC | AUC-PR | F1@R ≈ 0.95 |
| RF (Raw) | 0.99999 | 0.99998 | 0.978 | 0.9780 | 0.9839 | 0.000 |
| XGB (Raw) | 0.99999 | 0.99999 | 0.976 | 0.9393 | 0.9546 | 0.197 |
| LR (Raw) | 0.99998 | 0.99998 | 0.974 | 0.3672 | 0.4563 | 0.685 |
| RF (NMF-W) | 0.99740 | 0.99667 | 0.968 | 0.9962 | 0.9963 | 0.443 |
| XGB (NMF-W) | 0.99828 | 0.99794 | 0.971 | 0.9740 | 0.9830 | 0.560 |
| LR (NMF-W) | 0.98276 | 0.98147 | 0.949 | 0.9298 | 0.9647 | 0.495 |

**Table 21:** Standalone NMF (trained on CHD-normal) evaluated on OTIDS via reconstruction error.

| Setting | AUC-ROC | AUC-PR |
|---|---|---|
| NMF (CHD-normal) → OTIDS (reconstruction error) | 0.286 | 0.562 |

**Supervised models.** All classifiers achieved near-saturated performance on CHD-test (AUC-ROC/AUC-PR ≈ 1.0). Under transfer to OTIDS, Raw-feature models degrade sharply in thresholded performance, despite sometimes retaining high AUC values. For example, RF (Raw) attains AUC-ROC = 0.978 but collapses to $F1 = 0.0$ at the calibrated recall point, indicating that its decision boundary does not translate reliably under feature-level shift. LR (Raw) also degrades substantially on OTIDS (AUC-ROC = 0.367, AUC-PR = 0.456), reflecting strong sensitivity to dataset-specific feature distributions. In contrast, NMF-W provides consistently stronger transfer: RF (NMF-W) achieves AUC-ROC = 0.996, AUC-PR = 0.996, and $F1 = 0.443$, while XGB (NMF-W) reaches AUC-ROC = 0.974, AUC-PR = 0.983, and $F1 = 0.560$. These results suggest that NMF-W yields a compact representation that reduces spurious reliance on dataset-specific raw-feature artifacts and improves robustness under cross-dataset shift.

**Standalone NMF.** Direct reconstruction-error scoring generalizes poorly across datasets (AUC-ROC = 0.286), although AUC-PR remains moderate (0.562). This indicates that reconstruction residual magnitudes are not inherently domain-invariant, as they can be affected by shifts in CAN ID distributions, payload statistics, and logging conditions. Importantly, the latent representation ($\mathbf{W}$) still retains transferable structure that downstream classifiers can exploit more effectively than Raw features, yielding substantially improved OTIDS results.

**Limitations of cross-dataset transfer.** This evaluation reflects transfer between two specific public datasets rather than universal generalization. CHD and OTIDS differ in vehicle platform, CAN ID distributions, logging configurations, and attack execution styles, introducing substantial domain shift. Therefore, the observed performance—especially for standalone reconstruction-error scoring—should be interpreted as evidence of *relative* robustness of NMF-W rather than a guarantee of deployment-level generalization. Broader validation across additional vehicles, attack types, and data-collection settings is required, as discussed in Section 5.15.

### *5.11 Comparison with Other Unsupervised Detectors*

Table 22 shows that NMF achieves the highest F1-score and precision at a fixed high-recall operating point, substantially outperforming IF and slightly exceeding Autoencoder performance. While Autoencoders provide competitive accuracy, they rely on opaque latent representations, whereas NMF yields additive, parts-based components that are directly interpretable in terms of CAN traffic structure. These results confirm that NMF provides a favorable balance between detection performance, interpretability, and robustness for unsupervised CAN bus intrusion detection.

**Table 22:** Comparison of unsupervised anomaly detectors trained on normal traffic only (CHD), evaluated at a fixed high-recall operating point (R ≈ 0.95).

| Model | AUC-ROC | AUC-PR | Precision | Recall | F1 |
|---|---|---|---|---|---|
| NMF (k = 16) | 0.9999 | 0.9989 | 0.9981 | 0.9877 | 0.9929 |
| IF | 0.9992 | 0.9802 | 0.7536 | 0.9883 | 0.8552 |
| Autoencoder | 0.9992 | 0.9935 | 0.9816 | 0.9852 | 0.9834 |

### *5.12 Comparison with State-of-the-Art CAN IDS*

To contextualize the proposed framework within existing CAN IDS research, we qualitatively compare it with representative learning-based systems: GIDS [10], CANet [12], and INDRA [25]. These methods span GAN-based detection, LSTM autoencoders, and GRU autoencoders. Table 23 summarizes their architectures, datasets, interpretability, and deployment footprint.

**Table 23:** Qualitative comparison with representative SOTA CAN IDS approaches.

| Method | Architecture | Dataset | Interpretability | Deploy. Footprint |
|---|---|---|---|---|
| GIDS [10] (2018) | GAN (DNN, DeconvNet) | CHD | ✗ | N/S |
| CANet [12] (2020) | LSTM Autoencoder | Real, SynCAN (ETAS/Bosch) | ✗ | ~8.7 MB |
| INDRA [25] (2020) | GRU Autoencoder | SynCAN (ETAS/Bosch) | ✗ | ~443 kB |
| **Proposed NMF-based IDS** | NMF + Classifier | CHD, OTIDS | ✓ | **10.6 kB (NMF-H) + <1 kB (LR)** |

Note: "Deploy. Footprint" reflects model-size statistics reported in original or subsequent benchmark studies.

**DL baselines.** CANet employs an LSTM autoencoder and reports a footprint of approximately 8.7 MB (as benchmarked in [25]), while INDRA reduces this to roughly 443 kB using a GRU autoencoder. GIDS, based on GAN and DNN components, does not report explicit model-size statistics, but GAN-based IDS typically require multi-megabyte models. Although these approaches demonstrate strong detection performance on their respective datasets, they rely on opaque latent representations and incur substantial computational and memory costs, limiting transparency and deployment on resource-constrained ECUs.

**Proposed NMF-based IDS.** In contrast, the proposed framework yields a compact and interpretable latent representation. Each NMF component corresponds to semantically meaningful traffic patterns, such as CAN ID frequency structure, payload entropy, or DLC behavior. With $k = 16$, the learned basis matrix $\mathbf{H}$ and associated parameters require only **10.6 kB**, enabling deployment on highly constrained automotive ECUs. The resulting latent features can be paired with extremely lightweight classifiers (e.g., LR, <1 kB) or more expressive models when resources permit.

Overall, while SOTA DL-based IDS architectures achieve high accuracy, they typically trade interpretability and deployment efficiency for performance. The proposed NMF-based IDS offers a complementary alternative that combines competitive detection capability with explicit interpretability and a markedly smaller deployment footprint, making it well suited for real-time, in-vehicle intrusion detection.

### *5.13 Ablation Studies*

To quantify the contribution of individual design choices, we conduct ablation experiments along three axes: (i) removal of raw feature groups, (ii) variation of the NMF latent rank, and (iii) comparison between raw features and NMF-derived representations. Feature ablations are performed by removing entire semantically coherent groups (Table 24) while keeping all remaining descriptors unchanged. The groups capture complementary aspects of CAN traffic: burstiness and timing (**RATES_TIMING**), payload length usage (**DLC_STATS**), byte-level variability (**PAYLOAD_STATS**), per-ID variability (**PER_ID_STATS**), ID frequency structure (**ID_FREQ**), and novelty with respect to normal traffic (**NOVELTY**).

**Table 24:** Ablation groups and member features.

| Group | Features included (window-level) |
|---|---|
| **RATES_TIMING** | `msg_count`, `msg_rate`, `interarrival_mean`, `interarrival_std`. |
| **DLC_STATS** | `dlc_mean`, `dlc_std`, `dlc_hist_0`..`dlc_hist_8`. |
| **PAYLOAD_STATS** | `payload_mean`, `payload_std`, `payload_entropy`, `nonzero_byte_ratio`. |
| **PER_ID_STATS** | `per_id_payload_var_mean`, `per_id_payload_var_max`, `per_id_repeat_ratio_mean`. |
| **ID_FREQ** | Top-$K$ per-ID frequency counters `idfreq_{cid}`. |
| **NOVELTY** | `new_id_count`, `new_id_ratio`, `id_entropy`, `unique_id_count`, `topID_dominance`. |

**Raw feature group removal.** Table 25 reports test-set F1 scores after removing individual raw feature groups. Across RF, XGB, and LR, performance remains largely stable, with only minor degradation when removing **RATES_TIMING** or **NOVELTY**. This indicates substantial redundancy among handcrafted descriptors and suggests that strong performance in the raw space does not depend on any single feature group, reinforcing the motivation for a compact representation.

**Table 25:** Test-set F1 scores for raw feature-group ablations.

| Ablated group | RF | XGB | LR |
|---|---|---|---|
| ID_FREQ | 0.987 | 0.975 | 0.975 |
| DLC_STATS | 0.982 | 0.976 | 0.975 |
| PAYLOAD_STATS | 0.977 | 0.976 | 0.974 |
| PER_ID_STATS | 0.981 | 0.976 | 0.974 |
| RATES_TIMING | 0.979 | 0.974 | 0.974 |
| NOVELTY | 0.977 | 0.975 | 0.973 |

**NMF component sweep.** We evaluate the effect of the latent rank by sweeping $k \in \{8, 12, 16, 20\}$. As shown in Fig. 19 and Table 26, performance improves markedly from $k = 8$ to $k = 16$ but saturates thereafter. Gains beyond $k = 16$ are marginal relative to the increased model size and reduced interpretability. Accordingly, $k = 16$ is adopted as a balanced operating point.

**Raw vs. NMF–W comparison.** Table 27 directly compares raw features with NMF–W ($k = 16$). Raw features achieve slightly higher in-distribution F1 scores, reflecting the richness of the original feature space. However, NMF–W reduces dimensionality by more than an order of magnitude while retaining competitive performance. This compact representation proves advantageous in cross-attack and cross-dataset evaluations, particularly for LR, which benefits from the structured latent space.

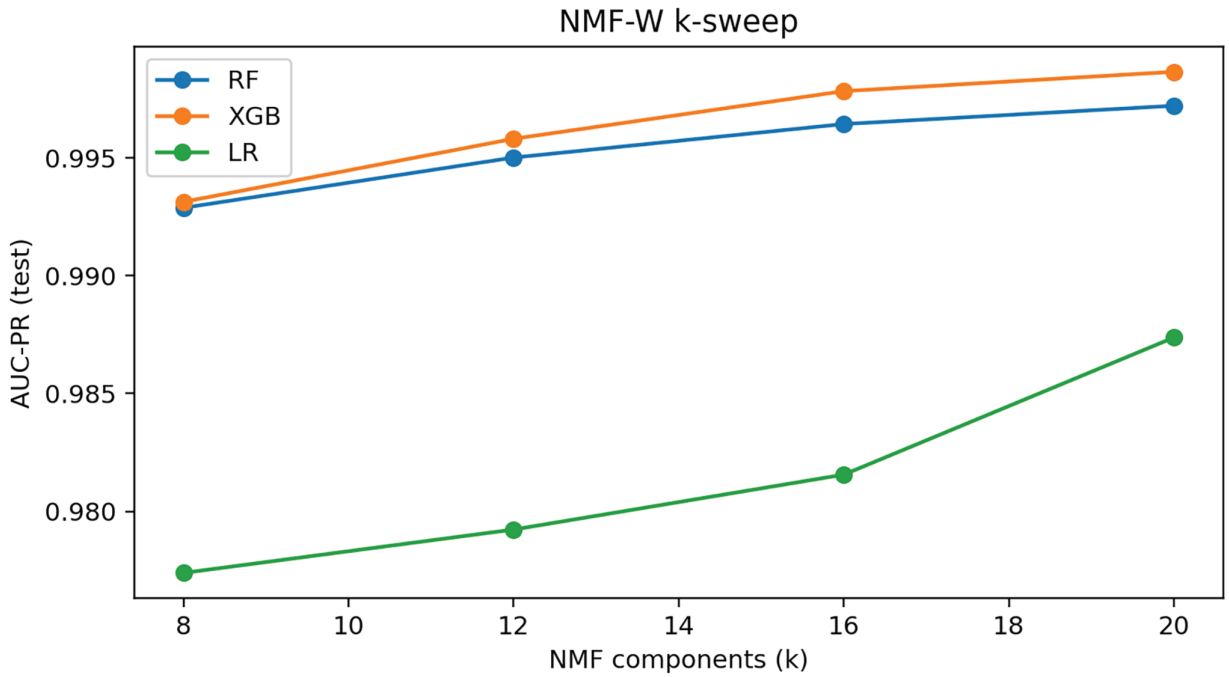



**Figure 19:** AUC-PR on the test set for NMF-W component sweep ($k = 8, 12, 16, 20$).

**Table 26:** Performance of NMF-W at varying numbers of components.

| *k* | RF (AUC-PR) | XGB (AUC-PR) | LR (AUC-PR) |
|---|---|---|---|
| 8 | 0.993 | 0.993 | 0.977 |
| 12 | 0.995 | 0.996 | 0.979 |
| 16 | 0.996 | 0.998 | 0.982 |
| 20 | 0.997 | 0.999 | 0.988 |

**Table 27:** Comparison of RAW vs. NMF–W ($k = 16$) features on the test split.

| Model | RAW | | | NMF–W | | |
|---|---|---|---|---|---|---|
| | ROC | PR | F1 | ROC | PR | F1 |
| LR | 0.99998 | 0.99998 | 0.974 | 0.9829 | 0.9816 | 0.948 |
| RF | 0.99999 | 0.99998 | 0.978 | 0.9973 | 0.9964 | 0.968 |
| XGB | 0.99999 | 0.99999 | 0.976 | 0.9982 | 0.9978 | 0.971 |

Overall, the ablation results show that raw feature performance is driven by redundant descriptors, while NMF–W provides a compact and robust alternative that preserves accuracy and improves downstream generalization and interpretability.

### *5.14 Efficiency Evaluation and ECU Deployment Feasibility*

We evaluate computational efficiency in terms of training time, inference latency, and serialized model size. Table 28 summarizes tabular models (RF, XGB, LR) on Raw and NMF-W representations, while Table 29 reports forward-pass timing for sequence models (CNN, LSTM, Attention).

**Table 28:** Training and inference efficiency for tabular models on Raw vs. NMF-W features. Model size is reported after serialization.

| Space | Model | Train (s) | Inference (ms/1k) | Size (MB) |
|---|---|---|---|---|
| RAW | RF | 9.91 | 0.094 | 18.5 |
| RAW | XGB | 1.35 | 0.0009 | 0.70 |
| RAW | LR | 1.51 | 0.00018 | < 0.01 |
| NMF-W | RF | 21.11 | 0.085 | 219.6 |
| NMF-W | XGB | 1.36 | 0.00073 | 2.00 |
| NMF-W | LR | 0.31 | 0.00005 | < 0.01 |

**Table 29:** Forward-pass efficiency of sequence models for synthetic 20-step windows.

| Model | Params | Forward (s) | ms/1k |
|---|---|---|---|
| CNN | 33.9k | 0.023 | 5.6 |
| LSTM | 45.1k | 0.126 | 30.7 |
| ATT | 14.8k | 0.024 | 5.9 |

**Tabular models.** On Raw features, RF trains in 9.9 s and infers in 0.094 ms per 1k windows, while XGB and LR train within ~1.5 s with sub-millisecond inference. With NMF-W, training times remain low for XGB/LR, and inference latency stays well below real-time requirements across all models.

**NMF overhead (offline vs. runtime).** NMF training is performed offline and yields compact artifacts: for $k = 16$, the basis **H** and scaling parameters require only **10.6 kB**. Fitting NMF on normal windows takes 6.12 s (Table 30). At runtime, the NMF-W step reduces to lightweight matrix–vector operations.

**Table 30:** Timing analysis of standalone NMF detection and amortized per-frame cost.

| Operation | Time | Unit |
|---|---|---|
| NMF fit (train-normal, offline) | 6.12 | s |
| Validation scoring (per window) | 0.062 | ms |
| Test scoring (per window) | 0.060 | ms |
| Full-stream scoring (per window) | 0.086 | ms |
| Average frames per window | 158 | Frames |
| Amortized test cost per frame | 0.38 | µs |
| Amortized full-stream cost per frame | 0.54 | µs |

**Model size and deployability.** Deployability is dominated by the downstream classifier rather than NMF. The RF model on NMF-W is large (219.6 MB) due to the 400-tree ensemble and deep split structures, and thus exceeds typical ECU flash budgets. In contrast, XGB (2.0 MB) and LR (< 1 kB) remain compact, indicating that NMF itself does not impose a storage burden. Reducing the number of trees proportionally reduces RF size (e.g., ~110 MB at 200 trees) with negligible accuracy change in preliminary tests (F1 variation < 0.3%).

**Sequence models.** CNN, LSTM, and Attention contain 14k–45k parameters and require 5–31 ms per 1k windows (Table 29). Although slower than tabular models, their forward-pass latency remains feasible on higher-end automotive processors.

**Feasibility for ECU deployment.** From an embedded perspective, NMF (10.6 kB for k = 16) paired with lightweight classifiers such as LR or XGB fits within common ECU flash memory ranges (2–16 MB) and supports real-time inference (<1 ms per 1k windows). Therefore, the NMF representation is ECU-friendly, and practical feasibility depends primarily on the downstream classifier.

**Runtime efficiency under ECU constraints.** Beyond model size, inference latency is a critical factor for embedded deployment. Our timing analysis shows that standalone NMF scoring requires less than 0.1 ms per window, which translates to an amortized cost below 1 μs per CAN frame under realistic traffic densities. This latency is well within real-time constraints of automotive CAN buses, where frame intervals are typically on the order of hundreds of microseconds. Since all computationally intensive steps—feature extraction from logs, NMF fitting, and classifier training—are performed offline, the on-vehicle workload is limited to lightweight matrix–vector operations and threshold checks, making continuous deployment feasible on resource-constrained ECUs.

**Window-based vs. packet-based IDS.** A key design choice in this work is the use of a *window-based* IDS rather than a purely packet- (frame-) based detector. Packet-based IDS [38,39,51] operate at the granularity of individual CAN frames and prioritize immediate reaction with minimal latency. In practice, such systems commonly rely on lightweight rule-based mechanisms, including CAN ID whitelisting and instantaneous message-rate thresholding, due to their low computational cost and suitability for deployment on resource-constrained ECUs.

In contrast, the proposed window-based NMF IDS aggregates CAN frames over short temporal windows (0.1 s in this study) and analyzes their collective statistical structure. This aggregation enables the detection of more subtle anomalies that preserve valid CAN IDs and per-frame constraints, such as spoofing attacks with realistic timing and payload patterns. While window-based detection introduces a bounded alert delay determined by the window size, it provides richer temporal context and improved robustness without exceeding real-time ECU constraints.

As shown in Table 30, NMF scoring requires approximately 0.06–0.09 ms per window, corresponding to an amortized per-frame cost of 0.38–0.54 μs given an average of 158 frames per window. Executing the same matrix-based analysis independently for every CAN frame would incur substantially higher overhead. Window-based processing therefore enables near-real-time detection while maintaining low computational cost.

**Quantitative comparison with packet-based IDS.** To contextualize this design choice, we quantitatively compare the proposed window-based NMF IDS against a conventional packet-based IDS implemented using per-frame CAN ID whitelisting and instantaneous rate thresholds [38,39,51]. From Table 31, it is clear that Packet-level detection achieves very low per-frame cost (0.33–0.35 μs) and near-zero alert latency, but its effectiveness is strongly attack-dependent. While DoS and Fuzzy attacks are detected reliably (F1 ≈ 0.99), detection performance collapses for subtle spoofing attacks such as Gear and RPM (F1 ≈ 0, AUC-PR < 0.35).

In contrast, the window-based NMF IDS maintains a comparable amortized per-frame cost (0.38–0.54 μs) while introducing a bounded alert delay determined by the window size (median ≈ 50 ms, 95th percentile ≈ 95 ms). This modest delay enables robust detection across both high-rate and subtle attack types, illustrating a trade-off between immediacy and robustness that reflects fundamental architectural choices rather than implementation limitations.

**Table 31:** Quantitative comparison between packet-based IDS and the proposed window-based NMF IDS.

| Metric | Packet-based IDS (Rule-based) | Window-based NMF IDS |
|---|---|---|
| Detection granularity | Per CAN frame | 0.1 s window (158 frames avg.) |
| Per-frame cost | 0.33–0.35 µs | 0.38–0.54 µs (amortized) |
| Alert latency | < 1 ms | ≤100 ms (window-bounded) |
| DoS detection (F1) | ≈1.00 | ≈1.00 |
| Fuzzy detection (F1) | ≈0.99 | ≈0.99 |
| Gear spoofing (F1) | ≈0.00 | > 0.90 |
| RPM spoofing (F1) | ≈0.00 | > 0.90 |
| Model type | Rule-based thresholds | Interpretable latent model |

Overall, packet-based IDS offer immediate response with minimal computation but are inherently limited to detecting simple rule violations only for known attack patterns. The proposed window-based NMF IDS operates as a standalone detector with comparable computational cost and substantially improved robustness to subtle attacks, making it a practical and complementary alternative for in-vehicle intrusion detection.

*Practical Deployment and Update Strategy*

Vehicle CAN traffic characteristics may evolve over time due to ECU software updates, hardware changes, or driving-pattern shifts. Therefore, practical deployment of an IDS requires not only a small memory footprint, but also the ability to update or retrain models efficiently as vehicle behavior changes.

The proposed NMF-based IDS separates offline training from on-vehicle inference. All computationally expensive steps—feature extraction from raw logs, NMF factorization, and classifier training—are performed on a backend analysis server. The ECU stores only the final artifacts (the NMF basis **H** and a lightweight classifier) and performs simple matrix–vector multiplications and threshold checks at runtime.

**Updating models when new attacks emerge.** As new attack traces are collected in the field (e.g., via a central gateway ECU or a fleet-level logging infrastructure), they can be aggregated offline and used to retrain or fine-tune the NMF model and downstream classifiers. Because the deployed artifacts are lightweight (Section 5.14), updates can be delivered OTA by replacing only the NMF basis and classifier parameters.

**Handling changes in normal CAN traffic over time.** Normal CAN traffic may drift due to ECU software updates or hardware changes. Two mechanisms address this: (i) decision thresholds (e.g., the $P_{99}$ reconstruction-error or classifier score used to define "attack windows") can be recalibrated using recent normal windows without modifying the model; and (ii) when distribution shifts exceed a safe margin, the NMF and classifier can be retrained offline on updated normal data and redeployed via an OTA patch.

**Integration with existing automotive security frameworks.** The NMF-based IDS complements rule-based defences such as ID whitelisting and rate limiting. In a typical in-vehicle security architecture, the module resides on a gateway ECU or security processor, receives time-windowed features from a monitoring component, and outputs anomaly scores or alerts. Rule-based checks handle obvious violations (e.g., unknown IDs or extreme DoS rates), while the NMF-based detector identifies subtle or previously unseen patterns. Alerts can be forwarded to a vehicle security manager or backend analysis system for correlation with other logs. This modular design ensures compatibility with existing automotive security workflows while adding a lightweight, interpretable anomaly-detection layer.

### *5.15 Limitations & Threats to Validity*

The findings of this study are influenced by several constraints related to datasets and evaluations. First, the CHD and OTIDS datasets include only a limited set of attack behaviors. Flooding attacks (DoS, Fuzzy) appear as long continuous intervals, whereas Gear spoofing and RPM spoofing occur only in short bursts. Under the chronological 70–10–20 split used to avoid temporal leakage, these short-lived spoofing attacks yield too few attack-labeled windows for stable cross-attack analysis, restricting quantitative generalization to the DoS–Fuzzy pair. Moreover, the flooding attacks in CHD induce extreme message-rate and payload patterns that make them inherently easier to distinguish from normal traffic than more subtle real-world intrusions; thus, the near-saturated AUC values reported on CHD should be interpreted as optimistic upper bounds rather than typical field performance. This limitation reflects constraints of current public datasets rather than shortcomings of the proposed NMF-based framework. Second, the observed asymmetry in cross-attack generalization (Fuzzy → DoS stronger than DoS → Fuzzy) arises from fundamental differences in attack structure: Fuzzy introduces high-entropy variability, whereas DoS produces uniform traffic saturation. This pattern is consistent across Raw features, NMF–W, and multiple backbones (RF, IF, CNN, LSTM, Attention), indicating that it is a dataset property rather than a modeling artifact. Third, the chronological split naturally leads to an imbalanced train-test distribution because attacks in CHD occur early in the logs. Although this setup more closely mimics real deployment scenarios, it reduces comparability with studies that use random stratified sampling, potentially leading to conservative performance estimates. Fourth, in cross-dataset evaluation (CHD → OTIDS), standalone NMF reconstruction error is sensitive to domain shift, whereas NMF–W paired with supervised models transfers more reliably. Stronger generalization claims will require additional datasets, more diverse attack types, and controlled multi-domain benchmarks. Fifth, the attack-localization analysis demonstrates that within 0.1 s attack windows, approximately 45%–56% of the 10 ms micro-windows exhibit statistically abnormal behavior in message rate or CAN ID entropy when compared with normal baselines. This supports the use of window-level alerts as coarse indicators of temporally concentrated disruptions. However, the proposed system remains a window-based IDS, not a frame-based intrusion prevention system (IPS). While the proposed framework demonstrates robustness to unseen attack behaviors across datasets and attack mixtures, the hierarchical strategy (window-level alert followed by finer-grained ECU checks) is primarily validated for the attack patterns in CHD and OTIDS. Highly stealthy or adaptive attacks that closely mimic normal traffic may require finer-resolution analysis or complementary IPS mechanisms. Finally, efficiency and interpretability analyses were conducted offline. Real-ECU deployment, expert validation, and integration with OTA update pipelines were not tested in this study and represent promising directions for future work.

## 6 Conclusion

This paper presented NMF as a lightweight and explainable framework for intrusion detection in automotive CAN bus traffic. In contrast to conventional raw feature-based IDS approaches that tend to overfit dataset artifacts and provide limited interpretability, NMF offers a principled decomposition into additive, non-negative latent components that capture meaningful traffic patterns and remain consistent across models and attack scenarios. Our study systematically evaluated NMF in two roles: as a standalone unsupervised detector based on reconstruction error, and as a feature extractor (NMF–W) for classical classifiers, unsupervised detectors, and deep sequence models. Results demonstrated that NMF-based representations provide competitive detection performance while enhancing robustness and interpretability, positioning NMF as a viable alternative to black-box DL approaches.

**Future Work.** Guided by the limitations identified in Section 5.15, future work will focus on three concrete directions. First, to better capture short-lived and subtle spoofing behaviors, we plan to investigate

NMF variants with limited temporal awareness, such as temporally regularized or sequence-aware NMF formulations that preserve interpretability while incorporating short-range dynamics within windows. Second, to enhance robustness under cross-dataset and cross-vehicle distribution shifts, we will explore lightweight domain-alignment strategies, including component-level normalization, adaptive threshold calibration, and incremental updating of the NMF basis using newly observed normal traffic, without retraining complex downstream models. Third, to bridge the gap between algorithmic interpretability and operational usability, we aim to conduct small-scale expert-in-the-loop evaluations, where automotive security analysts assess whether NMF component activations and basis patterns meaningfully support root-cause analysis, attack triage, and ECU-level diagnostics. Together, these directions provide a concrete and feasible roadmap for strengthening the robustness, practicality, and real-world relevance of NMF-based intrusion detection in automotive networks.

**Acknowledgement:** Not applicable.

**Funding Statement:** This work was supported in part by the Basic Science Research Program through the NRF funded by the Ministry of Education under Grant 2021R1A6A1A03039493, and in part by the Regional Innovation System & Education (RISE) program through the Gyeongbuk RISE CENTER, funded by the Ministry of Education (MOE) and the Gyeongsangbuk-do, Republic of Korea (2025-RISE-15-115).

**Author Contributions:** The authors confirm contribution to the paper as follows: conceptualization, Anandkumar Balasubramaniam and Seung Yeob Nam; methodology, Anandkumar Balasubramaniam and Seung Yeob Nam; software, Anandkumar Balasubramaniam; validation, Anandkumar Balasubramaniam and Seung Yeob Nam; formal analysis, Anandkumar Balasubramaniam; investigation, Anandkumar Balasubramaniam and Seung Yeob Nam; resources, Anandkumar Balasubramaniam; data curation, Anandkumar Balasubramaniam; writing—original draft preparation, Anandkumar Balasubramaniam; writing—review and editing, Anandkumar Balasubramaniam and Seung Yeob Nam; visualization, Anandkumar Balasubramaniam; supervision, Seung Yeob Nam; project administration, Seung Yeob Nam; funding acquisition, Seung Yeob Nam. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The datasets used in this study are publicly available and can be accessed from the HCRL website at https://ocslab.hksecurity.net/Datasets/car-hacking-dataset and https://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset.

**Ethics Approval:** Not applicable.

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

**Supplementary Materials:** The supplementary material is available online at https://www.techscience.com/doi/10.32604/cmc.2026.077582/s1.

## References

1. Petit J, Shladover SE. Potential cyberattacks on automated vehicles. IEEE Trans Intell Transport Syst. 2014;16(2):546–56. doi:10.1109/tits.2014.2342271.
2. Hoppe T, Kiltz S, Dittmann J. Security threats to automotive CAN networks—practical examples and selected short-term countermeasures. Reliab Eng Syst Saf. 2011;96(1):11–25. doi:10.1016/j.ress.2010.06.026.
3. Studnia I, Nicomette V, Alata E, Deswarte Y, Kaâniche M, Laarouchi Y. Survey on security threats and protection mechanisms in embedded automotive networks. In: Proceedings of the 2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W); 2013 Jun 24–27; Budapest, Hungary. p. 1–12. doi:10.1109/DSNW.2013.6615528.

4. Müter M, Asaj N. Entropy-based anomaly detection for in-vehicle networks. In: Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV); 2011 Jun 5–9; Baden-Baden, Germany. p. 1110–5. doi:10.1109/IVS.2011.5940552.
5. Miller C, Valasek C. A survey of remote automotive attack surfaces. Las Vegas, NV, USA: Black Hat; 2014. p. 1–94.
6. Bécsi T, Aradi S, Gáspár P. Security issues and vulnerabilities in connected car systems. In: Proceedings of the 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS); 2015 Jun 3–5; Budapest, Hungary. p. 477–82.
7. Koscher K, Czeskis A, Roesner F, Patel S, Kohno T, Checkoway S, et al. Experimental security analysis of a modern automobile. In: Proceedings of the 2010 IEEE Symposium on Security and Privacy; 2010 May 16–19; Oakland, CA, USA. p. 447–62. doi:10.1109/SP.2010.34.
8. Taylor A, Japkowicz N, Leblanc S. Frequency-based anomaly detection for the automotive CAN bus. In: Proceedings of the 2015 World Congress on Industrial Control Systems Security (WCICSS); 2015 Dec 14–16; London, UK. p. 45–9.
9. Song HM, Woo J, Kim HK. In-vehicle network intrusion detection using deep convolutional neural network. Veh Commun. 2020;21(1):100198. doi:10.1016/j.vehcom.2019.100198.
10. Seo E, Song HM, Kim HK. GIDS: GAN based intrusion detection system for in-vehicle network. In: Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust (PST); 2018 Aug 28–30; Belfast, Ireland. p. 1–6. doi:10.1109/PST.2018.8514157.
11. Marchetti M, Stabili D. READ: reverse engineering of automotive data frames. IEEE Trans Inform Forensic Secur. 2019;14(4):1083–97. doi:10.1109/tifs.2018.2870826.
12. Hanselmann M, Strauss T, Dormann K, Ulmer H. CANet: an unsupervised intrusion detection system for high dimensional CAN bus data. IEEE Access. 2020;8:58194–205. doi:10.1109/ACCESS.2020.2982544.
13. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Jair. 2002;16:321–57. doi:10.1613/jair.953.
14. He H, Bai Y, Garcia EA, Li S. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence); 2008 Jun 1–8; Hong Kong, China. p. 1322–28. doi:10.1109/ijcnn.2008.4633969.
15. Neupane S, Ables J, Anderson W, Mittal S, Rahimi S, Banicescu I, et al. Explainable intrusion detection systems (X-IDS): a survey of current methods, challenges, and opportunities. IEEE Access. 2022;10(7):112392–415. doi:10.1109/access.2022.3216617.
16. Moustafa N, Koroniotis N, Keshk M, Zomaya AY, Tari Z. Explainable intrusion detection for cyber defences in the Internet of Things: opportunities and solutions. IEEE Commun Surv Tutor. 2023;25(3):1775–807. doi:10.1109/COMST.2023.3280465.
17. Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Nature. 1999;401(6755):788–91. doi:10.1038/44565.
18. Lee DD, Seung HS. Algorithms for non-negative matrix factorization. Adv Neural Inf Process Syst. 2001;13:556–62 doi: 10.32614/cran.package.nmfn.
19. 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); 2017 Jun 11–14; Los Angeles, CA, USA. p. 1577–83. doi:10.1109/IVS.2017.7995934.
20. Luo F, Wang J, Zhang X, Jiang Y, Li Z, Luo C. In-vehicle network intrusion detection systems: a systematic survey of deep learning-based approaches. PeerJ Comput Sci. 2023;9:e1648. doi:10.7717/peerj-cs.1648.
21. Kang MJ, Kang JW. Intrusion detection system using deep neural network for in-vehicle network security. PLoS One. 2016;11(6):e0155781. doi:10.1371/journal.pone.0155781.
22. Nair R. Unraveling the decision-making process interpretable deep learning IDS for transportation network security. J Cybersecur Inf Manag. 2023;12(2):69–82. doi:10.54216/jcim.120205.
23. Subasi O, Cree J, Manzano J, Peterson E. A critical assessment of interpretable and explainable machine learning for intrusion detection. arXiv:2407.04009. 2024.

24. Abou El Houda Z, Brik B, Khoukhi L. "Why should I trust your IDS?": an explainable deep learning framework for intrusion detection systems in Internet of Things networks. IEEE Open J Commun Soc. 2022;3:1164–76. doi:10.1109/OJCOMS.2022.3188750.
25. Kukkala VK, Thiruloga SV, Pasricha S. INDRA: intrusion detection using recurrent autoencoders in automotive embedded systems. IEEE Trans Comput Aided Des Integr Circuits Syst. 2020;39(11):3698–710. doi:10.1109/TCAD.2020.3012749.
26. Malar Dhas JP, Isravel DP. Anomaly detection in IoV can bus traffic using variational autoencoder-LSTM with attention mechanism. In: Proceedings of the 2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS); 2024 Dec 17–18; Bengaluru, India. p. 368–73. doi:10.1109/ICICNIS64247.2024.10823311.
27. Kim T, Kim J, You I. An anomaly detection method based on multiple LSTM-autoencoder models for in-vehicle network. Electronics. 2023;12(17):3543. doi:10.3390/electronics12173543.
28. Sun H, Chen M, Weng J, Liu Z, Geng G. Anomaly detection for in-vehicle network using CNN-LSTM with attention mechanism. IEEE Trans Veh Technol. 2021;70(10):10880–93. doi:10.1109/TVT.2021.3106940.
29. Nguyen TP, Nam H, Kim D. Transformer-based attention network for in-vehicle intrusion detection. IEEE Access. 2023;11:55389–403. doi:10.1109/ACCESS.2023.3282110.
30. Chen A, He Z, Zhang D. An anomaly detection model for CAN networks based on CNN and transformer. In: Proceedings of the IECON 2024–50th Annual Conference of the IEEE Industrial Electronics Society; 2024 Nov 3–6; Chicago, IL, USA. p. 1–6. doi:10.1109/IECON55916.2024.10905930.
31. Jo H, Kim DH. Intrusion detection using transformer in controller area network. IEEE Access. 2024;12:121932–46. doi:10.1109/ACCESS.2024.3452634.
32. Ullah I, Khalil I, Bai X, Garg S, Kaddoum G, Shamim Hossain M. An ensemble-based hybrid model for the detection of attacks in the Internet of vehicular things. IEEE Trans Intell Transp Syst. 2025;26(10):17914–27. doi:10.1109/TITS.2025.3547999.
33. Hernandez-Ramos JL, Karopoulos G, Chatzoglou E, Kouliaridis V, Marmol E, Gonzalez-Vidal A, et al. Intrusion detection based on federated learning: a systematic review. ACM Comput Surv. 2025;57(12):1–65. doi:10.1145/3731596.
34. de Oliveira JA, Gonçalves VP, Meneguette RI, de Sousa RT, Guidoni DL, Oliveira JCM, et al. F-NIDS—a network intrusion detection system based on federated learning. Comput Netw. 2023;236(2):110010. doi:10.1016/j.comnet.2023.110010.
35. Idrissi MJ, Alami H, El Mahdaouy A, El Mekki A, Oualil S, Yartaoui Z, et al. Fed-ANIDS: federated learning for anomaly-based network intrusion detection systems. Expert Syst Appl. 2023;234(1):121000. doi:10.1016/j.eswa.2023.121000.
36. Khatri N, Lee S, Nam SY. Transfer learning-based intrusion detection system for a controller area network. IEEE Access. 2023;11:120963–82. doi:10.1109/ACCESS.2023.3328182.
37. Khandelwal S, Shreejith S. A lightweight FPGA-based IDS-ECU architecture for automotive CAN. In: Proceedings of the 2022 International Conference on Field-Programmable Technology (ICFPT); 2022 Dec 5–9; Hong Kong, China. p. 1–9. doi:10.1109/ICFPT56656.2022.9974508.
38. Müter M, Groll A, Freiling FC. A structured approach to anomaly detection for in-vehicle networks. In: Proceedings of the 2010 Sixth International Conference on Information Assurance and Security; 2010 Aug 23–25; Atlanta, GA, USA. p. 92–8. doi:10.1109/ISIAS.2010.5604050.
39. Taylor A, Leblanc S, Japkowicz N. Anomaly detection in automobile control network data with long short-term memory networks. In: Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA); 2016 Oct 17–19; Montreal, QC, Canada. p. 130–9. doi:10.1109/DSAA.2016.20.
40. Wang W, Guan X, Zhang X. Profiling program and user behaviors for anomaly intrusion detection based on non-negative matrix factorization. In: Proceedings of the 2004 43rd IEEE Conference on Decision and Control (CDC); 2004 Dec 14–17; Nassau, Bahamas. p. 99–104. doi:10.1109/CDC.2004.1428613.
41. Guan X, Wang W, Zhang X. Fast intrusion detection based on a non-negative matrix factorization model. J Netw Comput Appl. 2009;32(1):31–44. doi:10.1016/j.jnca.2008.04.006.

42. Lefoane M, Ghafir I, Kabir S, Awan IU, El Hindi K, Mahendran A. Non-negative matrix factorization and latent semantic analysis for hybrid feature selection: a proposed machine learning system for the detection of malicious executable files. IEEE Access. 2025;13(4):138867–82. doi:10.1109/ACCESS.2025.3596483.
43. Sulistianingsih N, Martono GH. Feature selection versus feature extraction models for IoT intrusion detection system using convolutional neural network. In: Proceedings of the 2024 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT); 2024 Nov 28–30; Mataram, Indonesia. p. 35–42. doi:10.1109/COMNETSAT63286.2024.10862350.
44. Ma W, Wu Y, Wang S, Gong M. Improved OBS-NMF algorithm for intrusion detection. In: Bio-inspired computing: theories and applications. Singapore: Springer; 2017. p. 596–613. doi:10.1007/978-981-10-7179-9_47.
45. Ahmed I, Hu XB, Acharya MP, Ding Y. Neighborhood structure assisted non-negative matrix factorization and its application in unsupervised point-wise anomaly detection. J Mach Learn Res. 2021;22(34):1–32.
46. Yan B, Ma X, Sun Q, Shen L. Physics-enhanced NMF toward anomaly detection in rotating mechanical systems. IEEE Trans Rel. 2025;74(3):3911–25. doi:10.1109/tr.2024.3417262.
47. Wang Y, Zhang Y, Wang L, Hu Y, Yin B. Urban traffic pattern analysis and applications based on spatio-temporal non-negative matrix factorization. IEEE Trans Intell Transp Syst. 2022;23(8):12752–65. doi:10.1109/TITS.2021.3117130.
48. Lee H, Jeong SH, Kim HK. 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); 2017 Aug 28–30; Calgary, AB, Canada. doi:10.1109/PST.2017.00017.
49. Berry MW, Browne M, Langville AN, Pauca VP, Plemmons RJ. Algorithms and applications for approximate nonnegative matrix factorization. Comput Stat Data Anal. 2007;52(1):155–73. doi:10.1016/j.csda.2006.11.006.
50. Hsieh CJ, Dhillon IS. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, CA, USA: ACM; 2011. p. 1064–72. doi:10.1145/2020408.2020577.
51. Young C, Zambreno J, Olufowobi H, Bloom G. Survey of automotive controller area network intrusion detection systems. IEEE Des Test. 2019;36(6):48–55. doi:10.1109/MDAT.2019.2899062.