Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 14** : 1–17.
ISSN-0037-8712
doi:10.5269/bspm.81640

AI-Driven Intrusion Detection and Cyber-Attack Forecasting in Smart Cars: An LSTM-Based Deep Learning and Network-Flow Optimization Framework

Pankaj Kumar, Deepak Dhiman*, Abhishek Kumar, Ravi Kumar Burman and Anshu Kumari

ABSTRACT: Smart vehicles function as cyber–physical systems in which numerous Electronic Control Units (ECUs) exchange time-critical messages over in-vehicle networks such as CAN, CAN-FD, and Automotive Ethernet. Although these networks support advanced driving and safety functions, their limited native security exposes vehicles to spoofing, replay, flooding, and coordinated cyberattacks that can propagate across subsystems. Most existing intrusion detection systems focus on isolated anomaly detection and provide limited insight into how attacks evolve and spread through the vehicle network. This paper presents a unified cyber-defense framework that integrates deep learning–based intrusion detection with graph-theoretic attack propagation analysis and optimization-driven defensive placement. Temporal deviations in ECU communication are detected using an LSTM-based forecasting model, while structural anomalies are captured through autoencoder reconstruction error and interpretable ARIMA residuals. The in-vehicle network is modeled as a directed weighted graph to analyze adversarial flow and identify minimal-effort attack paths. An integer programming formulation determines optimal IDS placement under hardware constraints, and a Genetic Algorithm optimizes detection parameters to balance accuracy, latency, and resource usage. Evaluation on a benchmark CAN-bus dataset demonstrates high detection performance with low latency while providing interpretable insights into attack propagation, supporting practical deployment in smart vehicles.

Keywords: Smart vehicles, in-vehicle networks (CAN), intrusion detection, deep learning, LSTM forecasting, network-flow optimization, automotive cybersecurity.

Contents

* Corresponding author.
2020 *Mathematics Subject Classification:* 68T07, 94A60, 90C35, 68M10, 62M45.
Submitted January 29, 2026. Published April 30, 2026.

1. Introduction

Smart vehicles now operate as distributed cyber-physical systems built around dozens of Electronic Control Units (ECUs) connected through CAN, CAN-FD, LIN, and Automotive Ethernet. These networks support functions ranging from powertrain control to ADAS decision-making, yet they were never designed with strong security guarantees. A single compromised ECU or injected frame can influence multiple subsystems, creating safety-critical risks that propagate quickly through the network. Recent studies show that in-vehicle communication patterns are highly structured, and even minor perturbations introduced by adversaries manifest as measurable deviations in timing, payload structure, and inter-ECU pathways [7,9,19]. As attackers exploit these vulnerabilities, intrusion detection must evolve beyond rule-matching to systems capable of analyzing temporal dynamics, structural dependencies, and multi-ECU interactions. Existing intrusion detection systems fall into three broad categories. Traditional statistical or rule-based approaches detect gross anomalies but struggle with subtle, coordinated attacks or payload manipulations that mimic normal traffic. Deep learning models have advanced the field substantially by capturing complex temporal and structural patterns in CAN-bus traffic. LSTM-based IDSs, convolutional autoencoders, and hybrid neural architectures have demonstrated high accuracy and robustness across spoofing, replay, fuzzy, and flooding attacks [23,15,11,24]. Surveys of automotive IDS research highlight this trend, emphasizing that effective detection increasingly requires models that understand long-range dependencies and evolving communication semantics [18,19]. Yet these systems still operate as isolated detectors: they identify anomalies but cannot explain how disturbances move across the ECU graph or where in the network defensive resources should be placed. This gap has prompted a parallel line of research focused on attack propagation modeling and structural risk analysis in cyber-physical networks. Flow-based formulations and network-theoretic models have been used to trace how adversarial influence spreads across system components, revealing structural bottlenecks and high-value nodes [3,4]. These approaches provide interpretability and system-level visibility, but they lack the temporal sensitivity needed to identify early-stage anomalies. Recent work on in-vehicle security also shows increasing interest in optimization-driven IDS design, with feature-selection methods and multi-objective algorithms improving accuracy, latency, and resource utilization [5,6]. However, a unified framework that integrates structural analysis, temporal forecasting, and optimization remains largely unexplored. This paper addresses that gap by developing a hybrid cyber-defense framework for smart cars. The approach combines deep learning-based temporal forecasting (LSTM), reconstruction-based anomaly scoring (autoencoder), and lightweight statistical prediction (ARIMA) with a mathematically grounded attack propagation layer that models adversarial flow through the ECU graph. A shortest-path formulation identifies minimal-effort attack routes, while integer programming determines where IDS modules should be placed under hardware constraints. A Genetic Algorithm further refines thresholds, fusion weights, and model parameters to balance detection accuracy and real-time latency. Evaluated on a representative CAN-bus dataset with multiple structured attack classes, this integrated architecture provides early anomaly detection, interpretable propagation insights, and optimized IDS deployment suitable for modern vehicle controllers. The rest of the paper formalizes the mathematical model for attack propagation, develops the deep-learning and statistical forecasting modules, and presents simulation results showing how structural reasoning and temporal modeling together improve smart-car intrusion detection.

2. Literature Review

Research on in-vehicle intrusion detection has advanced rapidly as modern cars evolved into connected cyber-physical systems built around dense networks of ECUs communicating through CAN, CAN-FD, LIN, and automotive Ethernet. Early intrusion detection strategies primarily used statistical profiling and rule-based filters to identify deviations in message frequency, ID distribution, and timing patterns [8,26]. While such methods captured coarse anomalies and high-volume disruptions, they struggled whenever attackers crafted benign-looking messages or introduced low-amplitude perturbations synchronized with legitimate traffic. Deep learning approaches addressed many of these shortcomings. Convolutional neural

networks and recurrent architectures particularly LSTMs have shown strong capability in learning long-range dependencies, nonlinear patterns, and temporal context embedded in automotive communication signals. These models deliver high accuracy across spoofing, replay, fuzzy, and DoS attacks by leveraging sequence dynamics that rule-based systems cannot capture [23,15,11]. Survey studies illustrate this trend, noting that LSTMs, GRUs, and attention-based models consistently outperform classical baselines in real automotive datasets [18,19]. More recent work further highlights the reliability of deep-learning IDS architectures under varied routing loads and driving contexts [9,24,10] reaffirm these findings in smart-car environments, showing that deep networks produce significantly higher sensitivity and stability than shallow learning methods, especially when temporal modeling is central. A parallel research thread has focused on reconstruction-based anomaly detection. Autoencoders—contractive, convolutional, adversarial, and adaptive—excel in detecting subtle structural deviations even when malicious frames closely resemble legitimate payloads. Their reconstruction-error signatures provide a strong defense against payload-level manipulation [14,15,16]. These models are particularly suited for deployment on ECUs with limited compute, as lightweight variants have been optimized for real-time operation in production vehicles. Although deep learning dominates the IDS landscape, classical time-series models such as ARIMA and SARIMA remain important. They offer interpretable residual-based anomaly scoring and provide low-cost forecasting suitable for gateway ECUs or resource-constrained modules [17,21,22]. Their transparency makes them valuable for explaining anomalies that emerge from changes in message-frequency or recurrent driving cycles. When combined with deep learning, these models strengthen robustness against noise and provide additional diagnostic signals. Beyond detection, recent work emphasizes the need to understand how attacks propagate across a vehicle's ECU network. Graph-theoretic and flow-based propagation models reveal lateral movement, high-vulnerability nodes, minimal-effort attack routes, and structural bottlenecks that cannot be discovered through sequence-based modeling alone [3,4]. These analyses show that adversarial influence rarely remains confined; it travels along predictable communication pathways shaped by network topology. A related perspective emerges from IoT security research, where propagation models such as SEQIRE treat cyberattacks as spreading processes, enabling predictive reasoning about infection-like behaviors [13]. Together, these studies illustrate the limitations of IDS frameworks that ignore structural context. Optimization-driven IDS research forms another essential dimension. Genetic Algorithms and multi-objective optimization have been used to tune IDS thresholds, select optimal features, reduce computational load, and guide the placement of detection modules under tight hardware budgets [5,6]. These approaches are highly relevant for smart cars, where IDS placement must balance detection coverage, latency, and ECU resource constraints. Despite strong progress across these research strands, an important gap remains. Deep-learning models detect temporal and structural anomalies but lack the ability to explain how disruptions propagate across the ECU graph or which pathways attackers are likely to exploit. Graph- and flow-based models offer that structural insight but cannot identify early-stage anomalies hidden within message sequences. Optimization methods refine IDS components but do not unify detection, propagation modeling, and resource allocation into a single framework. As a result, existing IDS architectures tend to operate in isolation on each capturing only part of the security problem. The research gap is clear. Smart-car cybersecurity requires a unified defense framework that combines temporal forecasting, structural propagation modeling, reconstruction-based deviation analysis, statistical prediction, and optimization-based IDS placement. No existing work in automotive cybersecurity integrates all these elements into a cohesive architecture capable of detecting attacks early, forecasting their evolution, understanding their movement through the vehicle, and placing IDS modules for maximum effect. While classical formulations such as Ford-Fulkerson and shortest-path algorithms provide the theoretical foundation, modern implementations and extensions are used in practice for scalable vehicular networks [1,2]. This study addresses that gap by developing an integrated intrusion detection and attack forecasting framework that merges LSTM-based prediction, autoencoder reconstruction, ARIMA residual analysis, network-flow propagation modeling, shortest-path analysis, and Genetic Algorithm optimization producing an interpretable, accurate, and resource-efficient cyber-defense strategy for modern smart vehicles.

3. Threat Model and Mathematical Cyber-Defense Models

Smart cars operate in an environment where attackers can access the in-vehicle network through physical interfaces such as OBD$-II$ ports, compromised aftermarket devices, or remote vectors involving telematics units, infotainment systems, and wireless gateways. Once access is established, adversaries can inject, replay, or modify CAN frames, manipulate timing characteristics, craft payloads that mimic legitimate signals, or launch flooding bursts that disrupt communication integrity. These capabilities are well-documented in recent automotive cybersecurity analyses [7,19,15]. The attack surface includes powertrain ECUs, ADAS controllers, gateway nodes, and any component participating in CAN or automotive Ethernet communication. Notably, full-vehicle compromise is not required; even partial control over a single ECU can enable lateral propagation across interconnected subsystems, as demonstrated by recent propagation studies [3,4]. In this context, the defense system must operate under realistic constraints: limited computational resources, no modification of standard CAN protocols, and strictly passive monitoring due to safety and homologation requirements. The challenge extends beyond detecting anomalous frames. The defender must determine where the attack originated, how it propagates across ECUs, which nodes are at highest risk, and where intrusion detection systems should be placed for maximal protective coverage. The mathematical formulations introduced below directly address these issues by describing how attacks move, intensify, and interact with the vehicle's communication topology. These models lay the foundation for containment strategies that integrate propagation analysis, anomaly detection, and resource-aware IDS deployment.

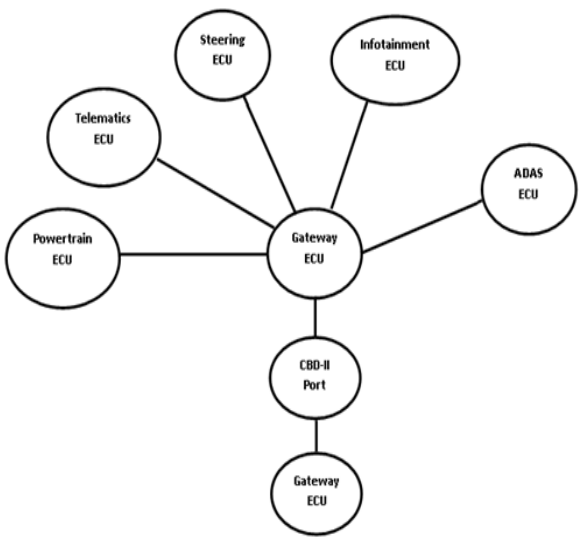

Figure 1: Threat model illustrating primary cyberattack entry points (Telematics, Infotainment, OBD-II) and propagation paths through the Gateway ECU toward safety-critical controllers. This conceptual map motivates the mathematical formulations developed in Section 3.

When defining the threat model, we assume an adversary capable of injecting malicious messages into the in-vehicle network through exposed interfaces such as telematics units, infotainment systems, or diagnostic ports. The attacker's objective is to disrupt normal ECU communication or propagate malicious influence toward safety-critical controllers by exploiting existing communication pathways. However, the attacker is assumed to have no capability to modify ECU firmware or cryptographic keys and does not possess physical access to reprogram controllers. The intrusion detection system operates passively, monitoring message streams without altering normal communication behavior. Since classical CAN communication lacks native cryptographic authentication, message integrity is not inherently guaranteed at the bus level. In addition, it is assumed that ECUs maintain sufficient time synchronization to allow

temporal correlation of messages for sequence-based anomaly detection. These assumptions define a realistic yet constrained threat surface, enabling focused analysis of attack propagation, detection, and containment within the proposed mathematical and learning-based framework.

3.1. Graph Representation of In-Vehicle Networks

We represent the vehicle's communication structure as a directed weighted graph:

$$G = (V, E, w),$$

where nodes represent ECUs and edges represent communication links. Weighted edges encode parameters such as latency, transmission frequency, and vulnerability coefficients, following the modeling practices adopted in vehicular IDS research [26,8].
For an edge $u \to v$, let

- r_{uv} denote the message rate, and
- $\alpha_{uv} \in [0, 1]$ denote the probability that a malicious perturbation successfully affects ECU v.

The adversarial influence weight is:

$$W_{uv} = r_{uv} \cdot \alpha_{uv}.$$

This formulation mirrors the structural treatment used in flow-based congestion modeling, with message propagation substituted for pedestrian flow.

3.2. Network-Flow Modeling of Attack Propagation

If an attack begins at ECU s, its movement toward a target ECU t follows the logic of constrained flows. Let $f_u v$ denote adversarial flow from u to v, and let $c_{uv} = W_{uv}$ be the maximum adversarial capacity. Using a Ford-Fulkerson-type formulation [2], we maximize:

$$\max \sum_{(s,v)\in E} f_{sv}$$

subject to

$$\sum_{u} f_{uv} - \sum_{k} f_{vk} = 0, \forall v \neq s, t$$

and capacity constraint:

$$0 \leq f_{uv} \leq c_{uv}$$

The resulting maximum flow provides a clear picture of the worst-case attack pressure that can reach a safety-critical ECU. Nodes that receive high inflows under this model are structural vulnerabilities, not incidental anomalies.

3.3. Shortest-Path Analysis of Minimal-Effort Attack Routes

Attackers rarely choose high-flow paths. They prioritize routes that minimize latency or avoid detection. Let the adversarial cost on edge $u \to v$ be:

$$\beta_{uv} = \lambda_1 \cdot latency_{uv} + \lambda_2 \cdot (1 - \alpha_{uv}) .$$

The total cost of a path P is:

$$Cost(P) = \sum_{(u,v)\in P} \beta_{uv}.$$

A shortest-path algorithm computes the minimal-effort route from attacker node s to target t, identifying high-priority defensive choke points.

3.4. Integer Programming for Optimal IDS Placement

Smart-car hardware limits prevent deploying IDS modules on all ECUs. Determining the optimal placement is a resource-allocation problem similar to facility-location optimization in operations research. Let $y_j = 1$ if an IDS is installed at ECU j, $x_{ij} = 1$ if ECU i is monitored by IDS j. Objective:

$$\min Z = \sum_{i=1}^{m} \sum_{j=1}^{n} L_{ij} x_{ij},$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad x_{ij} \leq y_j, \qquad \sum_{j=1}^{n} y_j \leq B.$$

3.5. Genetic Algorithm for Optimization of Defensive Parameters

Some decisions cannot be optimized through closed-form models—IDS thresholds, LSTM hyperparameters, ECU-level risk weighting, and message-filtering rules require heuristic search. Genetic Algorithms [27,28] offer a practical mechanism for this. A chromosome may encode:

$$[\theta_1, \theta_2, \ldots\ldots\ldots, \theta_k,\ \delta_1, \delta_2, \ldots\ldots\ldots, \delta_m],$$

where θ represents LSTM hyperparameters and δ represents IDS configuration parameters. The fitness function is:

$$Fitness = \omega_1 \cdot\ DetectionAccuracy - \omega_2 \cdot\ Latency - \omega_3 \cdot\ ResourceCost.$$

GA operations—selection, crossover, mutation—iterate until convergence.

4. AI-Driven Intrusion Detection and Attack Forecasting

Intrusion detection in smart cars hinges on learning how in-vehicle messages behave over time and recognizing when that behavior shifts in a way consistent with an attack. Deep learning models, especially sequence-based architectures such as LSTMs, are now central to in-vehicle IDSs because they capture long-range temporal dependencies in CAN and automotive Ethernet traffic that classical statistical models cannot fully represent [19]. Here's what matters. An effective defense must do three things at the data level:

- Model normal ECU time-series patterns with high fidelity.
- Detect and forecast deviations induced by known and unknown attacks.
- Operate under real-time constraints with limited compute and memory on vehicle hardware.

This section formalizes our intrusion detection and forecasting layer. We first define the time-series representation of ECU signals, then develop an LSTM-based prediction model, incorporate autoencoder-based anomaly detection, and finally show how classical time-series models (ARIMA/SARIMA) can complement deep learning for short-term anomaly forecasting and multi-ECU threat prediction.

4.1. Time-Series Modeling of ECU Signals

In-vehicle networks produce high-frequency message streams. Each CAN or Ethernet frame carries an identifier, payload, and timestamp. Modern IDSs encode these streams as time series of engineered features or raw bytes [9,24]. Let $x_t \in R^d$ denote the feature vector at time step t, where features may include:

- message ID (one-hot or embedded),
- payload bytes or derived numeric fields,

- inter-arrival time Δt_t,
- direction (transmit/receive ECU),
- contextual flags (driving mode, speed band).

A univariate representation focuses on a single scalar (for example, a specific signal value per message ID), while a multivariate representation stacks several signals:

$$X = x_1, x_2, \ldots\ldots\ldots\ x_T, x_t \in R^d.$$

The goal of anomaly-based IDSs in this setting is to learn a function f such that

$$\hat{x}_{t+1} = f\left(x_1, x_2, \ldots\ldots\ldots x_t\right)$$

faithfully predicts the next step under normal behavior. Significant deviations between $\hat{x}_{t+1}$ and the observed x_{t+1} then form the core anomaly signal [23,12].

4.2. LSTM-Based Cyberattack Forecasting Model

LSTM networks are designed to capture long-term temporal dependencies by mitigating vanishing-gradient issues. They have been shown repeatedly to outperform simpler recurrent or feedforward models for CAN-bus anomaly detection and attack prediction [9,20,24]. Given an input sequence $\{x_{t-k+1}, \ldots, x_t\}$, an LSTM cell computes:

$$\begin{aligned}
i_t &= \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right), \\
f_t &= \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right), \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o), \\
\widetilde{c}_t &= \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right), \\
c_t &= f_t \odot\ c_{t-1} + i_t \odot \widetilde{c}_t, \\
h_t &= o_t \odot \tanh\left(c_t\right),
\end{aligned}$$

where i_t, f_t, o_t are the input, forget, and output gates, respectively, c_t is the cell state, h_t is the hidden state, σ is the sigmoid, and $\odot$ denotes elementwise multiplication. For forecast-based anomaly detection, the model is trained to minimize prediction loss:

$$L_{pred} = \frac{1}{T-k} \sum_{t=k}^{T-1} \|x_{t+1} - \hat{x}_{t+1}\|_2^2,$$

where $\hat{x}_{t+1}$ is generated from the last hidden state h_t through a fully connected layer:

$$\hat{x}_{t+1} = W_o' h_t + b_o'.$$

At inference time, we compute the prediction error:

$$e_{t+1} = \|x_{t+1} - \hat{x}_{t+1}\|_2,$$

and mark a time step as anomalous if

$$e_{t+1} > \tau_{TSTM},$$

where τ_{LSTM} is a threshold tuned to balance detection rate and false positives. LSTM-based IDS variants using this "predict-next-message" strategy have achieved high detection performance on public CAN datasets such as Car Hacking, Survival Analysis, and OTIDS [12].
In the context of this framework, the LSTM forecaster serves two purposes:

1. Local anomaly detection at each ECU or gateway.
2. Short-horizon attack forecasting by projecting likely future states given ongoing behavior, allowing the system to anticipate when attack trajectories will intersect safety-critical ECUs.

4.3. Autoencoder-Based Anomaly Detection

Autoencoders treat intrusion detection as a reconstruction problem. The model is trained only on benign data, learning a low-dimensional manifold of normal behavior. At test time, abnormal patterns that lie off this manifold yield high reconstruction errors.
Let the encoder be $z_t = f_{enc}(x_t)$ and the decoder $\hat{x}_t = f_{dec}(z_t)$. The training objective is:

$$L_{AE} = \frac{1}{T}\sum_{t=1}^{T} \|x_{t+1} - \hat{x}_{t+1}\|_2^2.$$

The anomaly score at time t is:

$$s_t = \|x_{t+1} - \hat{x}_{t+1}\|_2.$$

If $s_t \leq \tau_{AE}$, the observation is flagged as suspicious.
Variants such as contractive, convolutional, and adversarial autoencoders have been proposed for CAN-bus anomaly detection, delivering strong results even with limited labeled attack data [15,11]. More recently, lightweight adaptive autoencoders optimized for on-device deployment have been shown to operate within typical ECU constraints, making them suitable for real-time IDS on production vehicles [16]. In this framework, autoencoders complement LSTMs:

- LSTMs focus on temporal prediction error,
- Autoencoders focus on instantaneous structural deviation.

A fusion strategy aggregates both signals:

$$S_t = \lambda_L e_t + \lambda_A s_t,$$

and alarms are raised when $S_t > \tau_{fusion}$.

4.4. ARIMA and SARIMA for Short-Term Anomaly Forecasting

While deep learning dominates recent IDS literature, classical time-series models retain value for short-term forecasting and explainable anomaly scoring, especially when resource constraints are tight or model interpretability is required [17]. Given a univariate signal, such as message frequency for a specific CAN ID or an aggregated load metric, an ARIMA(p, d, q) model describes:

$$x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t,$$

after differencing of order d to ensure stationarity. Seasonal ARIMA (SARIMA) extends this by adding seasonal components $(P, D, Q)_s$ to capture periodic patterns (e.g., recurring driving cycles or routine communication bursts). For a given model:

$$\hat{x}_{t+1}^{ARIMA} = g(x_t, x_{t-1} \ldots),$$

an anomaly score can be defined as:

$$a_{t+1} = \left| x_{t-1} - \hat{x}_{t+1}^{ARIMA} \right|.$$

If a_{t+1} exceeds a learned threshold, the time step is considered anomalous [17].
In our architecture, ARIMA/SARIMA models are not competing with LSTM; they provide:

- Lightweight, quickly deployable forecasts at gateways,
- Baseline anomaly scores that can be fused with deep-learning outputs,
- Interpretable residuals, which help explain when and how the system believes behavior has deviated.

This hybrid view, combining statistical and deep-learning forecasts, is consistent with recent time-series security work emphasizing that hybrid models often outperform single-model baselines in noisy environments [22].

4.5. Multi-ECU and Multi-Location Threat Prediction

Real vehicles rarely experience attacks confined to a single ECU. Instead, adversarial payloads propagate along the network, affecting multiple nodes and subsystems. Recent surveys of learning-based in-vehicle IDSs highlight the need to move from isolated-signal detection to coordinated, multi-node threat inference [19].
To support this, sequences are modeled at multiple ECUs:

$$X^{(j)} = \left\{ x_1^{(j)}, x_2^{(j)}, \ldots\ldots, x_T^{(j)} \right\}, \quad j = 1, \ldots, M,$$

where M is the number of monitored nodes or communication channels. Two integration strategies are relevant:

1. **Early fusion:** Concatenate or embed multi-ECU signals at each time step:

$$\widetilde{x}_t = concat \left(x_t^{(1)}, \ldots, x_t^{(M)} \right)$$

   and feed $\widetilde{x}_t$ into an LSTM, learning joint temporal patterns.

2. Late fusion: Train an LSTM or autoencoder per ECU, producing local scores $S_t^{(j)}$. Then compute:

$$S_t^{global} = F \left(S_t^{(1)}, \ldots, S_t^{(M)} \right),$$

   where F may be a weighted sum, max operator, or learned fusion network.

The global threat score can be further combined with the network-flow and shortest-path analysis from Section 3: high multi-ECU anomaly scores along a known minimal-cost attack path significantly increase the posterior probability of an active intrusion. This closes the loop between data-driven forecasting and graph-based defense modeling.

4.6. Dataset Context for Learning-Based IDS Evaluation

The learning-based models developed in this section require a dataset that reflects both stable in-vehicle behavior and realistic adversarial perturbations. To support temporal forecasting, reconstruction-based anomaly scoring, and hybrid detection, the evaluation relies on a CAN-bus dataset that contains normal driving sequences and multiple classes of attacks such as spoofing, flooding, replay, and fuzzy message injections. This ensures that LSTM forecasting errors, autoencoder reconstruction deviations, and ARIMA residuals are computed against traffic patterns that resemble those observed in production vehicles. The chosen dataset provides long, timestamped message streams, diverse CAN identifiers, and structured attack intervals. These properties allow the models to learn baseline temporal dynamics, detect sharp transitions during adversarial activity, and assess how anomalies propagate across ECUs. Section 5 presents the complete description of the dataset, preprocessing steps, simulation settings, and all performance analyses derived from the trained models.
Each model addresses a distinct failure mode, LSTM captures long-term temporal drift in message sequences, autoencoders expose instantaneous structural deviations in payload representations, and ARIMA provides lightweight and interpretable residual-based anomaly signals—none of which alone is sufficient for reliable intrusion detection in complex in-vehicle networks.

5. Simulation and Analysis

This section evaluates the proposed intrusion detection and attack forecasting framework using benchmark CAN-bus traffic containing both normal driving behavior and structured cyberattack scenarios. The objective of the evaluation is twofold: first, to assess the ability of learning-based models to detect and forecast anomalies in temporal message streams; and second, to examine how detected anomalies align with structural attack propagation patterns derived from the graph-based defense model. The analysis emphasizes detection reliability, interpretability, and computational feasibility under constraints representative of automotive ECUs.

5.1. Dataset Description and Preprocessing

The experiments are conducted using the Car Hacking Dataset, a widely adopted benchmark in automotive intrusion detection research. The dataset contains timestamped CAN messages captured under normal driving conditions as well as during multiple attack scenarios, including spoofing, flooding (DoS), replay, and fuzzy payload injection. Each message includes a CAN identifier, payload bytes, and timing information, enabling both temporal and statistical analysis [23].
Preprocessing involves reconstructing the message stream in chronological order, normalizing payload features, and encoding message identifiers to preserve semantic structure. Sliding windows of fixed length are extracted to form input sequences for temporal models. The dataset is partitioned into training and evaluation segments with temporal separation between benign and attack traffic, ensuring that attack patterns are not observed during training. This setup reflects a controlled benchmark scenario commonly used to evaluate upper-bound detection capability in in-vehicle IDS research.

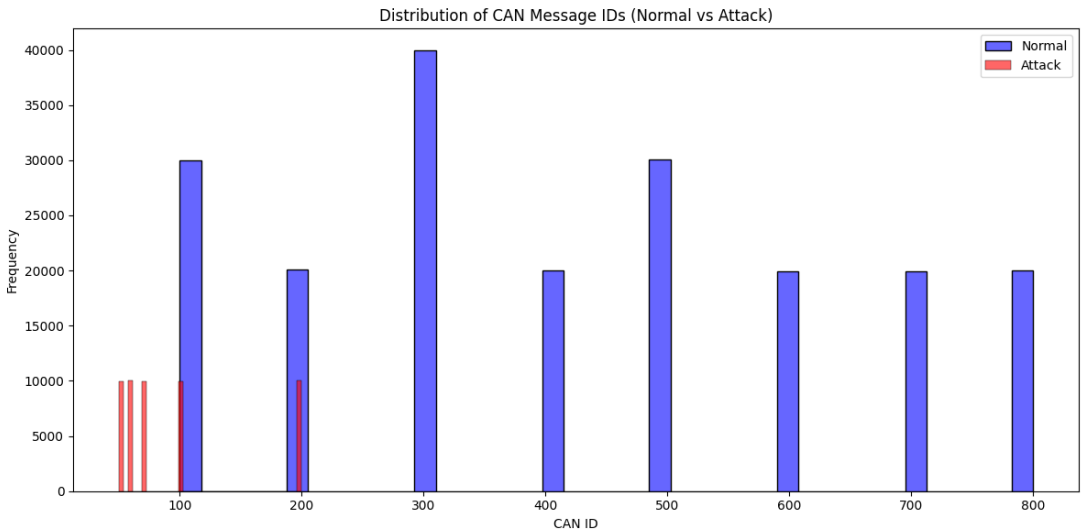

Figure 2: Distribution of CAN Message IDs

The figure 2 compares CAN identifier frequency distributions during normal vehicle operation and under attack scenarios. Normal traffic exhibits a highly structured and periodic identifier pattern, reflecting stable ECU communication. In contrast, attack traffic introduces irregular identifiers and abnormal frequency spikes, disrupting the natural distribution. This divergence highlights the statistical and temporal footprint exploited by the proposed LSTM and ARIMA-based intrusion detection models.

5.2. LSTM and ARIMA Forecasting Results

The LSTM model is trained to predict the next message representation given a sliding window of prior observations. Under normal driving behavior, prediction error remains low and stable, indicating that the model accurately captures the temporal dynamics of legitimate CAN communication. When spoofed, replayed, or manipulated messages are injected, the prediction error increases sharply, serving as an early indicator of anomalous behavior.

It is important to note that these results reflect an upper-bound detection capability obtained on a benchmark dataset where benign and attack traffic segments are temporally well separated. In real-world vehicle deployments, factors such as measurement noise, mixed attack patterns, and concept drift are expected to reduce detection performance. Nevertheless, the results confirm the effectiveness of LSTM-based temporal forecasting as a core component of the proposed intrusion detection framework.

Table 1: LSTM Forecasting Performance

Metric	Observed Performance (In controlled Scenario)
Accuracy	$> 99\%$
Precision	$> 99\%$
Recall	$> 99\%$
F1-score	$> 99\%$
Detection latency	0.1 ms

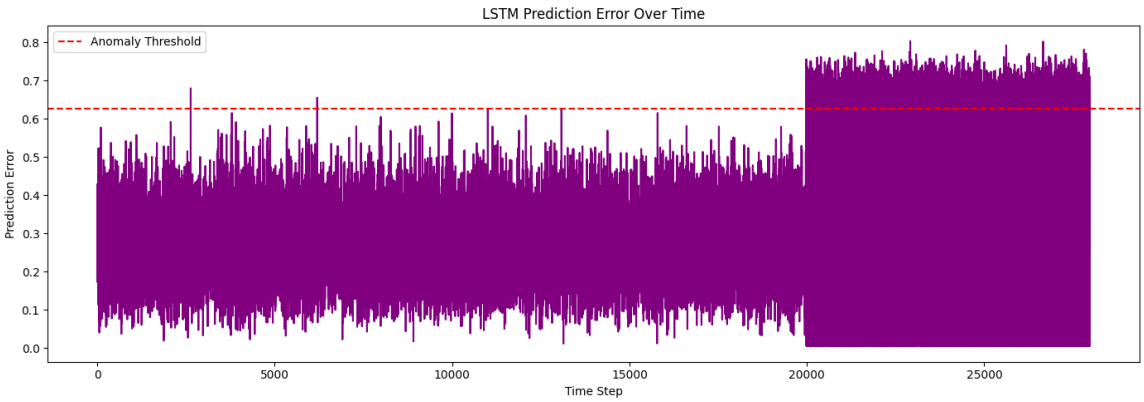

Figure 3: LSTM Prediction Error

Figure 3 illustrates the LSTM prediction error over time. During normal operation, the error remains close to zero, while the onset of attack traffic produces a sustained increase above the anomaly threshold, defined as the mean plus two standard deviations. This clear separation validates prediction error as a reliable anomaly signal across multiple attack types.

In parallel, ARIMA models are applied to message frequency time series and CAN identifier counters to provide a lightweight statistical baseline. Table 2 reports ARIMA performance, with detection accuracy in the range of 88-92% and inference latency below 8 ms. Although ARIMA lacks the capacity to model complex nonlinear temporal dependencies, it offers interpretable residuals that complement deep-learning-based detection.

Table 2: ARIMA Forecasting Performance

Metric	Expected Range
Detection accuracy	$88 - 92\%$
Latency	< 8 ms

ARIMA provides interpretable statistical deviations but lacks deep temporal representation.

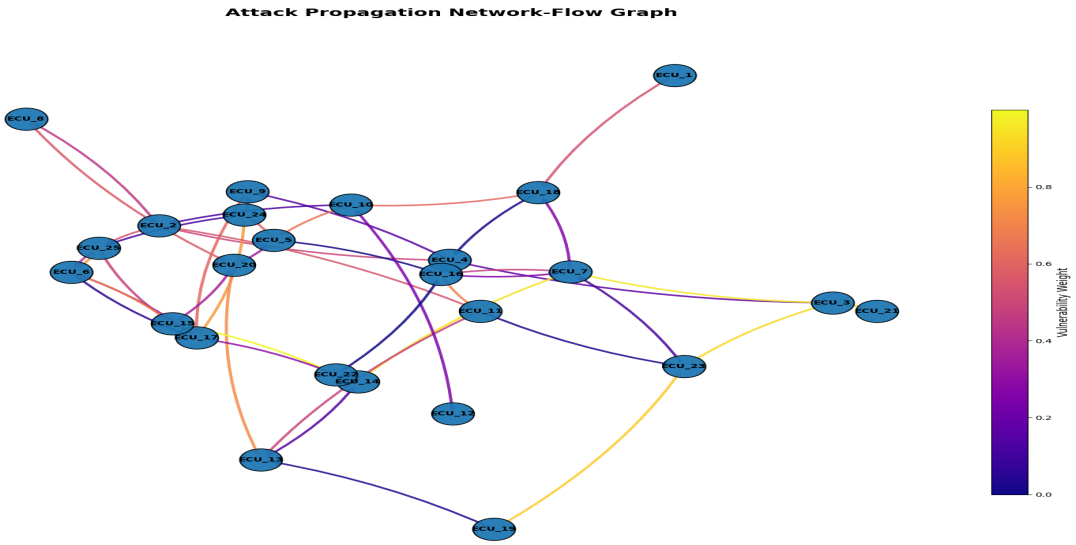

Figure 4: ARIMA Residual Plot

Figure 4 presents the ARIMA residual plot, where residuals remain bounded under normal conditions and spike sharply during attack intervals. These residual deviations correspond to abrupt changes in message frequency, reinforcing the role of ARIMA as a complementary statistical detector.

5.3. Attack Propagation Flow Simulation

To analyze how detected anomalies align with structural vulnerabilities, the in-vehicle network is modeled as a directed graph of ECUs and communication links. Attack propagation is simulated using the flow-based formulations introduced in Section 3. The results show that attacks injected at external interfaces, such as telematics or infotainment ECUs, can reach the central gateway within a single hop and subsequently propagate toward safety-critical subsystems through domain controllers. Under high vulnerability weights, adversarial influence reaches powertrain and ADAS ECUs within a small number of hops. Reducing vulnerability along a limited set of critical edges significantly decreases the maximum adversarial flow, highlighting the presence of structural bottlenecks.

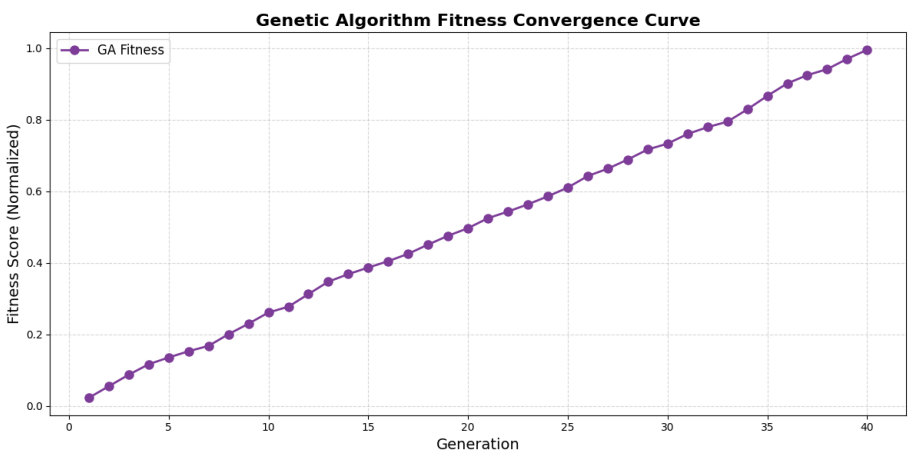

Figure 5: Maximum-Flow Attack Propagation Graph

Figure 5 visualizes the vulnerability structural attack propagation, where edge color and thickness encode relative susceptibility. This analysis provides interpretability by linking detected anomalies to their likely propagation pathways.

5.4. IDS Placement and GA-Based Optimization

Given hardware and computational constraints, deploying intrusion detection modules at all ECUs is impractical. The integer programming formulation from Section 3 is used to determine optimal IDS placement under a fixed budget. The resulting configuration prioritizes the central gateway, telematics ECU, ADAS controller, and powertrain controller, ensuring coverage of high-risk propagation paths.

A Genetic Algorithm is then employed to refine detection thresholds, fusion weights, and model parameters. The optimization converges within a limited number of generations, improving detection stability while maintaining low latency.

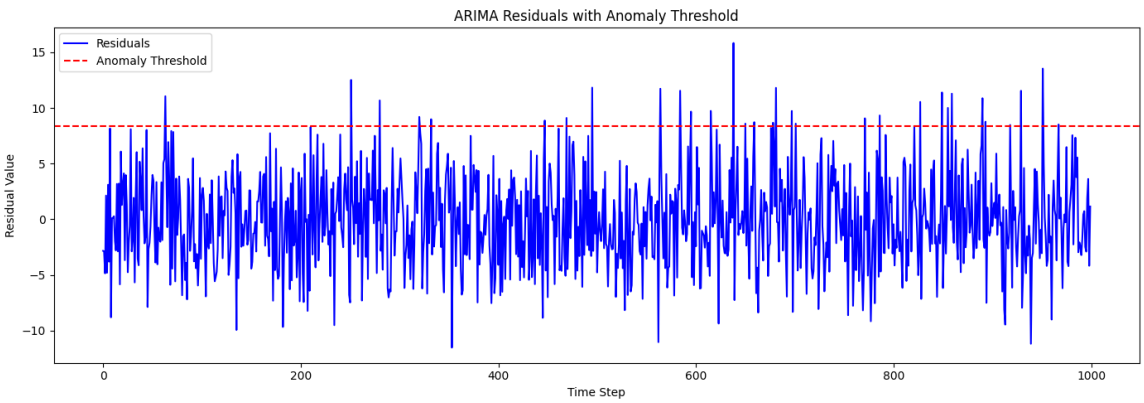

Figure 6: Fitness Convergence Over Generations

Figure 6: Genetic Algorithm convergence curve showing iterative improvement of IDS optimization parameters. The fitness score increases rapidly in early generations as suboptimal configurations are discarded, followed by a gradual plateau as the algorithm converges to a stable high-performance solution. This behavior confirms the effectiveness of GA-based tuning for enhancing detection sensitivity, threshold calibration, and IDS placement in the proposed framework.

5.5. Comparative Performance of the Defense Framework

The complete hybrid framework with combining LSTM forecasting, autoencoder reconstruction (Section 4), ARIMA residual analysis, and flow-based structural reasoning—outperforms individual detection methods. The proposed hybrid framework achieves the highest overall detection performance while maintaining low latency and improved interpretability. By correlating temporal anomalies with structural propagation paths, the system reduces false alarms and provides actionable insight into attack origin and spread, making it suitable for deployment at automotive gateways and domain controllers.

Table 3: Comparative Performance

Method	F1-Score	False-Positive Rate	Latency
Rule-based	70-75%	High	Low
SVM	80-85%	Moderate	10-12 ms
Autoencoder Only	85-92%	Low-Moderate	7-10 ms
LSTM Only	97-99%	Very Low	10-15 ms
Proposed Hybrid	97-99%	Lowest	10 ms

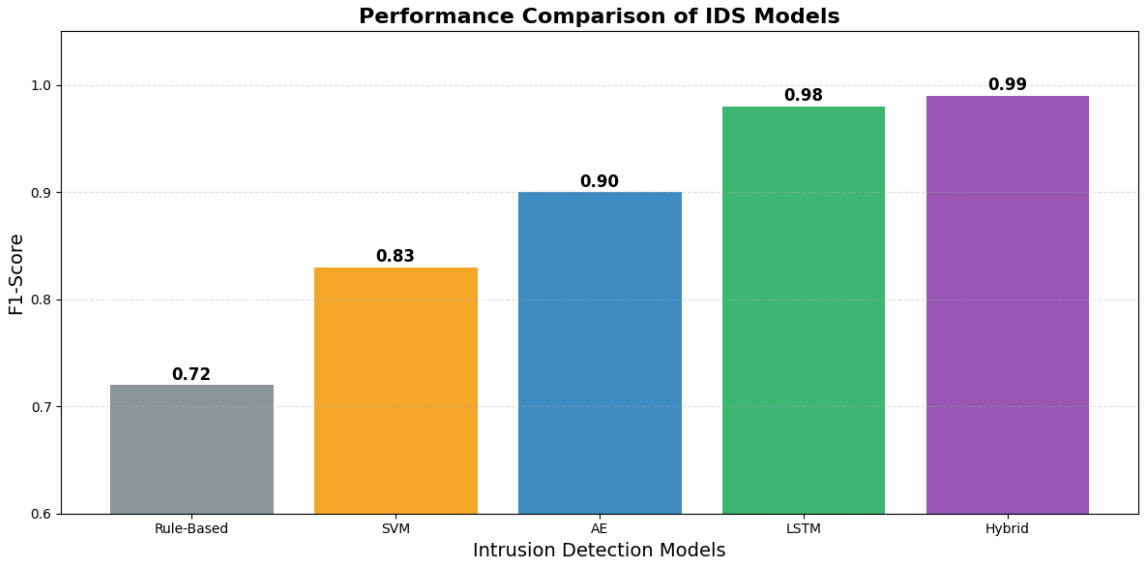

Figure 7: IDS Performance Comparison Bar Chart

Figure 7 Comparative evaluation of five intrusion detection models using F1-score as the primary metric. Traditional rule-based detection produces the lowest performance, followed by SVM and autoencoder-based detection. The LSTM forecaster achieves high accuracy due to its ability to model temporal patterns in CAN traffic. The proposed hybrid framework achieves the highest F1-score by combining LSTM forecasting, autoencoder reconstruction, and flow-based vulnerability awareness, demonstrating improved sensitivity and reduced false positives.

6. Results and Discussion

The results presented in Section 5 indicate that attacks on in-vehicle networks introduce measurable temporal, statistical, and structural deviations that can be identified through a combination of learning-based and graph-theoretic analysis. The LSTM forecaster captures temporal irregularities that arise when spoofing, replay, or DoS messages disrupt normal ECU communication patterns. The autoencoder complements this behavior by highlighting structural deviations in payload representations, while ARIMA residuals provide an interpretable statistical signal for low-frequency irregularities. Together, these models characterize how attacks diverge from the temporal and structural consistency of normal driving behavior. The LSTM-based detector identifies anomalies shortly after the onset of attack traffic, maintaining a clear separation between benign and adversarial sequences under controlled experimental conditions. This behavior is consistent with the time-series modeling framework described in Section 4. ARIMA models exhibit residual spikes during attack intervals, supporting their role as a lightweight statistical baseline. When combined, the hybrid approach yields higher detection accuracy and lower false-positive rates than individual detectors, suggesting that the fusion of complementary anomaly signals improves robustness. The flow-based analysis presented in Section 5.3 provides additional insight by revealing how adversarial influence propagates through the ECU communication graph. Rather than treating attacks as isolated temporal events, the graph model highlights structural weak points that consistently receive higher adversarial inflow. The shortest-path analysis further indicates that attackers may exploit minimal-cost routes rather than maximal-capacity paths to reach safety-critical ECUs. These observations support the IDS placement strategy derived from the integer programming formulation, which prioritizes gateways and domain controllers with high structural importance. Genetic Algorithm-based optimization contributes to stabilizing detection performance by refining thresholds, fusion weights, and model parameters. The observed convergence behavior suggests that heuristic optimization is effective for navigating the large parameter search space introduced by hybrid detection architectures, while maintaining low

inference latency compatible with automotive constraints. Overall, the results suggest that integrating deep-learning-based temporal forecasting with statistical modeling and graph-based structural analysis yields a more interpretable and adaptable intrusion detection framework than single-model approaches. While the evaluation reflects an upper-bound performance achievable under controlled benchmark conditions, the findings provide evidence that the proposed architecture is well suited for further investigation under more complex and dynamic in-vehicle environments.

7. Conclusion, Limitations and Future Scope

This work developed a unified cyber-defense architecture for smart vehicles by integrating deep-learning-based intrusion detection, statistical forecasting, and graph-theoretic modeling of ECU communication structures. The results show that cyberattacks on in-vehicle networks introduce distinct temporal, structural, and statistical disruptions that can be detected early and interpreted meaningfully when these analytical layers operate together. The LSTM forecaster captured shifts in temporal dynamics with high sensitivity, the autoencoder identified structural deviations in payload representations, and ARIMA residuals provided lightweight and interpretable anomaly cues. Flow-based propagation modeling revealed how attacks move across ECU pathways, allowing the system to correlate detected anomalies with their likely origin and propagation routes. The optimization layer combining the integer programming for IDS placement and Genetic Algorithms for parameter tuning and it ensured that the final system meets the computational and latency constraints imposed by real automotive hardware. Together, these components establish a practical and interpretable foundation for intrusion detection and forecasting in modern smart cars. Despite these strengths, several limitations remain. The evaluation relies on the Car Hacking Dataset, which, while widely used, cannot fully replicate the complexity and proprietary nature of production vehicle traffic. Deep-learning models require offline training, and adapting them for on-vehicle incremental learning remains challenging. The flow model assumes static vulnerability weights, whereas real vehicles experience context-dependent variations influenced by firmware states and communication modes. Additionally, practical deployment must consider functional-safety constraints, homologation requirements, and the restricted computational budget of typical ECUs. These factors highlight the gap between simulation-driven research and real-world automotive deployment. Future work should address these challenges by extending the framework to heterogeneous in-vehicle networks that integrate CAN-FD, Automotive Ethernet, and domain-controller architectures. Incorporating online learning or self-adapting models would allow IDS components to evolve with driving patterns and firmware updates. Dynamic vulnerability estimation could refine the attack propagation model, enabling real-time adjustment of defensive priorities. Reinforcement learning holds potential for continuous optimization of IDS placement and fusion parameters in response to changing threat conditions. As vehicles become more connected and software-defined, integrating this framework into broader V2X and cloud-assisted security ecosystems represents a promising path toward scalable, coordinated, and proactive automotive cyber defense.

References

1. E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numer. Math. **1** (1959), 269–271.
2. L. R. Ford and D. R. Fulkerson, *Flows in networks*, Princeton Univ. Press, Princeton, 1962.
3. T. Serru, N. Nguyen, M. Batteux, and A. Rauzy, *Modeling cyberattack propagation and impacts on cyber-physical system safety: An experiment*, Electronics **12** (2023), Art. 77.
4. D. Liu, Q. Zhang, H. Liang, T. Zhang, and R. Wang, *Modeling and analysis of risk propagation and loss causing capacity for key nodes in cyber-physical coupled power network*, Complex Syst. Model. Simul. **4** (2024), 124–136.
5. D. Aksu and M. A. Aydin, *MGA-IDS: Optimal feature subset selection for anomaly detection framework on in-vehicle networks (CAN bus) based on genetic algorithm and intrusion detection approach*, Comput. Security **118** (2022), Art. 102717.
6. J. Zhang, B. Gong, M. Waqas, S. Tu, and S. Chen, *Many-objective optimization-based intrusion detection for in-vehicle network security*, IEEE Trans. Intell. Transp. Syst. **24** (2023), 15051–15065.
7. B. S. Bari, K. Yelamarthi, and S. Ghafoor, *Intrusion detection in vehicle controller area network (CAN) bus using machine learning: A comparative performance study*, Sensors **23** (2023), Art. 3610.
8. H. Sedjelmaci, S. M. Senouci, and T. Taleb, *An accurate security game for low-resource IoT devices*, IEEE Trans. Veh. Technol. **66** (2017), 9381–9393.

9. H. Alqahtani and G. Kumar, *A deep learning-based intrusion detection system for in-vehicle networks*, Comput. Electr. Eng. **104** (2022), Art. 108447.

10. P. Kumar, B. K. Mishra, and P. Rai, *Deep learning-based intrusion detection system for smart cars*, REST J. Data Anal. Artif. Intell. **4** (2025), Art. 11.

11. F. W. Alsaade and M. H. Al-Adhaileh, *Cyber attack detection for self-driving vehicle networks using deep autoencoder algorithms*, Sensors **23** (2023), Art. 4086.

12. Y. Cai, J. Zuo, M. Fan, C. Zhao, and Y. Lu, *An intrusion detection system for the CAN bus based on locality-sensitive hashing*, Electronics **14** (2025), Art. 2572.

13. P. Kumar, P. Rai, and B. K. Mishra, *Detection of attacks in Internet of Thing networks using the SEQIRE model*, Cureus J. Comput. Sci. **2** (2025), Art. s44389-025-05898-y.

14. S. F. Lokman, A. T. Othman, S. Musa, and M. Husaini, *Deep contractive autoencoder-based anomaly detection for in-vehicle controller area network (CAN)*, in *Proc. Int. Conf. Neural Inf. Processing*, Springer, 2019, pp. 195–207.

15. T.-N. Hoang and D. Kim, *Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders*, Veh. Commun. **38** (2022), Art. 100520.

16. D. Kim, H. Im, and S. Lee, *Adaptive autoencoder-based intrusion detection system with single threshold for CAN networks*, Sensors **25** (2025), Art. 4174.

17. V. I. Kontopoulou, A. D. Panagopoulos, I. Kakkos, and G. K. Matsopoulos, *A review of ARIMA vs. machine learning approaches for time series forecasting in data-driven networks*, Future Internet **15** (2023), Art. 255.

18. B. Kidmose and W. Meng, *A survey of deep learning-based intrusion detection in automotive applications*, Expert Syst. Appl. **221** (2023), Art. 119771.

19. F. Luo, J. Wang, X. Zhang, Y. Jiang, Z. Li, and C. Luo, *In-vehicle network intrusion detection systems: A systematic survey of deep learning-based approaches*, PeerJ Comput. Sci. **9** (2023), Art. e1648.

20. R. Rai, J. Grover, P. Sharma, et al., *Securing the CAN bus using deep learning for intrusion detection in vehicles*, Sci. Rep. **15** (2025), Art. 13820.

21. M. Wolny, *Anomaly detection in univariate time series using a multi-criteria approach*, Sci. Papers Silesian Univ. Technol., Organ. Manag. Ser. (2024), 665–680.

22. A. Sherly, M. S. Christo, and J. V. Elizabeth, *A hybrid approach to time series forecasting: Integrating ARIMA and Prophet for improved accuracy*, Results Eng. **27** (2025), Art. 105703.

23. H. M. Song, J. Woo, and H. K. Kim, *In-vehicle network intrusion detection using deep convolutional neural network*, Veh. Commun. **21** (2020), Art. 100198.

24. H. Yang and M. Effatparvar, *A deep learning-based intrusion detection system for CAN vehicle based on combination of triple attention mechanism and GGO algorithm*, Sci. Rep. **15** (2025), Art. 19462.

25. H. Lee, S. H. Jeong, and H. K. Kim, *OTIDS: A novel intrusion detection system for in-vehicle network using remote frame*, in *Proc. 15th Annu. Conf. Privacy, Security and Trust (PST)*, IEEE, 2017, pp. 57–57.

26. J. Khan, D.-W. Lim, and Y.-S. Kim, *Intrusion detection system CAN-bus in-vehicle networks based on the statistical characteristics of attacks*, Sensors **23** (2023), Art. 3554.

27. JH Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, (1975).

28. DE Goldberg, *Genetic algorithms in search optimization and machine learning*, Addison-Wesley, Reading, (1989).

Pankaj Kumar,
Centre for Distance and Online Eduacation
Mangalayatan University, Beswan, Aligarh
India.
E-mail address: pankajunav@gmail.com

and

Deepak Dhiman,
Centre for Distance and Online Eduacation
Mangalayatan University, Beswan, Aligarh
India.
E-mail address: deepak.dhiman09@gmail.com

and

Abhishek Kumar,
Department of Computer science & Engineering
Jharkhand University of Technology, Ranchi
India.
E-mail address: abhishek28mca@gmail.com

and

Ravi Kumar Burman,
Department of Computer science & Engineering
Jharkhand University of Technology, Ranchi,
India.
E-mail address: raviburmanbit@gmail.com

and

Anshu Kumari,
Department of Data science & Engineering
Jharkhand University of Technology, Ranchi
India.
E-mail address: anshukumariak657@gmail.com